UNIVERSIDADE FEDERAL FLUMINENSE

PATRICK BLACKMAN SPHAIER

USER INTENT CLASSIFICATION IN NOISY TEXTS: AN INVESTIGATION ON NEURAL LANGUAGE MODELS

NITERÓI

2021

UNIVERSIDADE FEDERAL FLUMINENSE

PATRICK BLACKMAN SPHAIER

USER INTENT CLASSIFICATION IN NOISY TEXTS: AN INVESTIGATION ON NEURAL LANGUAGE MODELS

Dissertation presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfilment of the requirements for the degree of Master of Science. Research area: Computer Science.

Advisor:
ALINE MARINS PAES CARVALHO

NITERÓI

2021

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

S753u Sphaier, Patrick Blackman
USER INTENT CLASSIFICATION IN NOISY TEXTS: AN INVESTIGATION ON NEURAL LANGUAGE MODELS / Patrick Blackman Sphaier ; Aline Marins Paes Carvalho, orientadora. Niterói, 2021.
177 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2021.

DOI: http://dx.doi.org/10.22409/PGC.2021.m.03676377745

1. Aprendizado de maquina. 2. Inteligência Artificial. 3. Produção intelectual. I. Marins Paes Carvalho, Aline, orientadora. II. Universidade Federal Fluminense. Instituto de Computação. III. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

PATRICK BLACKMAN SPHAIER

USER INTENT CLASSIFICATION IN NOISY TEXTS: AN INVESTIGATION ON NEURAL LANGUAGE MODELS

Dissertation presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Master of Science. Topic area: Computer Science.

Approved in July 2021.

APPROVED BY

Prof. Aline Marins Paes Carvalho - Advisor, UFF

Prof. Alexandre Plastino de Carvalho, UFF

Prof. Vládia Célia Monteiro Pinheiro, UNIFOR

Prof. Ronaldo Ribeiro Goldschmidt, IME

Niterói

2021

Acknowledgements

I want to send a special thank you to my advisor, Aline Paes, for her dedication, support, and valuable insights, making this an inspiring experience for me. I also would like to thank the examination committee for accepting evaluate this project. I also want to thank my colleagues and friends who supported me one way or another and helped me accomplish this challenge. Finally, to my family, in special my sons, Theo and Noah, who provided me with the necessary strength and motivation besides the many hours we couldn't enjoy together.

Resumo

Esta dissertação investiga os benefícios do uso de embeddings pré-treinados e ajustados na classificação de intenção do usuário em cenários multi-classe com ruído e sentenças curtas. Conteúdos gerados por usuários são uma fonte fundamental de informações que auxiliam na tomada de decisões em várias tarefas, como marketing online, atendimento a solicitações de clientes e no acompanhamento à resposta da intenção. No entanto, por serem gerados por usuários sem supervisão ou correção, também apresentam vários desafios, como falha em identificar a classe correta devido ao texto limitado, palavras com grafia incorreta e a falta de gramática devido principalmente à forma como a informação é coletada, e as vezes em um estilo linguístico específico. Por outro lado, esta tarefa é naturalmente modelada como um problema de classificação que tem sido amplamente abordado nos últimos anos pela extração de atributos baseados em vetores de embeddings pré-treinados seguido pelo treinamento de um classificador. No entanto, devido à natureza ruidosa das frases coletadas, esse pipeline que usa diretamente embeddings pré-treinados a partir de corpus genéricos pode não funcionar bem. Nesta dissertação, investigamos se tal percepção se mostra empiricamente verdadeira em três conjuntos de dados do mundo real. Além disso, avaliamos o fine-tuning de embeddings pré-treinados com diferentes estratégias para avaliar a mais promissora. No total, avaliamos o desempenho de onze modelos de linguagem, incluindo embeddings gerais pré-treinados, embeddings pré-treinados baseados em tweets, aprendizagem de embeddings do zero e fine-tuning de embeddings pré-treinados. Para verificar se é possível aproveitar uma representação simples para resolver a tarefa de classificação de intenção do usuário, também avaliamos o desempenho de classificadores de vetores esparsos usando uma abordagem de bag-of-words (BOW). Mostramos que o ajuste da linguagem dos embeddings ao vocabulário do conjunto de dados alvo e uma classificação adicional a partir de um modelo BERT - uma tarefa conhecida como Task Adaptive Pre-training (TAPT) - obtém os melhores resultados gerais. No entanto, empregar diretamente a classificação sobre o BOW também pode ser a escolha certa em alguns casos, graças à simplicidade e a baixa utilização de recursos de hardware dessa opção. Também mostramos que que comparar os resultados utilizando um método de interpretabilidade pode ajudar a compreender as predições e também auxiliar na identificação de classes incorretamente rotuladas em um conjunto de dados.

Palavras-chave: embeddings, fine-tuning, datasets de intenção do usuário, multiclasse, interpretabilidade.

Abstract

This dissertation investigates the benefits of using pretrained and fine-tuned embeddings to address user intent classification in noisy, short-text, and multiclass scenarios. We claim that such user-generated content is a fundamental source of information to aid the decision-making in several tasks, such as online marketing, answering requests from customers, and follow-up intent response. However, they also present several challenges, as the misguiding of the class due to the limited text, many misspelled words and lack of proper grammar due mainly to how they can be collected, and, sometimes, a specific linguistic style. On the other hand, the task is naturally modelled as a classification problem that has been widely tackled in the last years by extracting vector-based features from pretrained embeddings followed by the induction of a classifier. However, because of the noisy nature of the collected sentences, this pipeline that directly uses pretrained embeddings from general corpora may not work well. In this dissertation, we investigate if such a perception empirically proves true in three real-world datasets. Furthermore, we evaluate fine-tuning pretrained embeddings with different strategies to observe the most promising one. In total, we evaluate the performance of eleven language-based models, including pretrained general embeddings, tweets-based pretrained embeddings, learning embeddings from scratch, and fine-tuning embeddings. To verify if one can leverage a simple representation to solve the user-intent classification task, we also evaluate the performance of sparse-vector classifiers using a bag-of-words (BOW) approach. We show that adjusting the language of the embeddings to the target dataset vocabulary and an additional classification of a BERT model – a task that is known as Task Adaptive Pretraining (TAPT) – achieves the best overall results. However, directly employing classification over BOW could also be the right choice in some cases, empowered by the simplicity and low-hardware resource requirements of this choice. We also show that analysing the resuls with an interpretability method helps on understanding the predictions and may also help to identify classes incorrectly labelled in a dataset.

Keywords: embeddings, fine-tuning, user-intent datasets, multiclass, interpretability.

List of Figures

List of Tables

List of Abbreviations and Acronyms

ANN	:	Artificial Neural Network;
ASGD	:	Asynchronous Stochastic Gradient Descent;
ASR	:	Automated Speech Recognition;
AWD-LSTM	:	ASGD Weight-Dropped LSTM;
BOW	:	Bag of Words;
BPTT	:	Backpropagation Through Time;
CBOW	:	Continuous Bag of Words;
CNN	:	Convolutional Neural Network;
CV	:	Computer Vision;
FNN	:	Feedforward Neural Network;
GPU	:	Graphic Processing Unit;
LM	:	Language Model;
LSTM	:	Long Short-Term Memory Network;
MLFNN	:	Multilayer Feedforward Neural Network;
MLM	:	Masked Language Model;
MSE	:	Mean Square Error;
NLI	:	Natural Language Inference;
NLP	:	Natural Language Processing;
NLU	:	Natural Language Understanding;
NSP	:	Next Sentence Prediction;
PDA	:	Portable Digital Assistant;
PSTN	:	Public Switched Telephone Network;
QA	:	Question Answering;
ReLU	:	Rectified Linear Unit;
RNN	:	Recurrent Neural Network;
TAPT	:	Task-Adaptive Pretraining;
STLR	:	Slanted Triangular Learning Rate;
TF-IDF	:	Term Frequency–Inverse Document Frequency;
VSM	:	Vector Space Model;

Contents

Chapter 1

Introduction

With the advent of pervasive conversational agents, online marketing, and services based on social networks, it has become crucial to understand what the user of those services intends automatically. Although a complete understanding of whatever the user wants requires aspects that computer systems cannot represent yet, one usually addresses this problem by classifying the utterances. By automatically providing a class to the utterance, the process may benefit from faster decision-making and filtering between simple and complex situations such that humans may only focus on situations that require more sensitive decisions. For instance, identifying a user's intent during a call to a support service may help decide the best human operator the call should be diverted to and serve as crucial business management information. However, even in this simplified framing, the task of *intent classification* faces several challenges such as short utterances, limited and informal vocabulary with specific expressions, lack of grammar correctness, capturing from noisy environments, a large set of intent classes, among other issues, also seen in other tasks such as classification of social network user data [29, 31, 64]. Even though such user-generated content presents those problematic issues, in several situations, they are the only source of information to find out the intention of the user and hence to aid the decision-making process [40].

Consider, for example, a conversational voice-based agent responsible for discovering a customers' intention to redirect him/her to the appropriate service. Usually, the first step in this situation is to acquire what the user verbally says from an automatic speech recognition service and convert it into a textual statement. However, this step may introduce noise into the conversation. No matter how good the automatic speech recognition engine performs, it may be influenced by external sounds, by the user's accent, and even by grammatical errors committed by him/her. For example, the sentence *eu sou meu* *controle remoto que parou do nada estou tentando marcar uma uma gema uma visita de um técnico aqui em casa e nao tô conseguindo*, captured from a call to a cable TV support service is an example of a noisy translation output from an ASR engine. A similar yet less problematic situation happens with content posted by users in social networks and online marketing, as idiomatic expressions and grammar mistakes are often present. In addition, to frame user-intent discovery as a classification task, one may have to elicit several possible classes to contemplate a large set of outcome possibilities. For example, in a large cable TV support service, calls may be classified into 121 classes representing intents such as ask for a remote control replacement, complain about a channel that cannot be accessed or schedule a technical visit. Similarly, data collected from a large online marketing service includes more than 1,000 classes, each one associated to a specific product category, like bicycle wheels, car wheels, leggings or gardening tools.

Recent years have witnessed an explosion of machine learning methods based on numerical vectors of words, sentences, or documents, known as embeddings, to handle natural language-based tasks (see Chapter 2), such as summarization, text classification, question answering, text generation, among others [2]. It has also become a common practice to use embeddings pretrained from large corpora and then inducing a model to the specific task [36]. Moreover, the last couple of years brought attention to another practice with the emergence of deep learning-based methods to generate embeddings, including ELMo [43], BERT [15], ULMFit [22]: to start from a pretrained model and then *fine-tuning* them, *i.e.*, refining the numerical values that represent the words according to the task one needs to solve. With text-based inputs converted into a numerical format, one may follow two general approaches to address the target task. Either one can extract such numeric features and make them the input of a machine learning-based classifier or put together the induction of the numerical representations and the classifier. As most approaches rely on neural networks to induce numerical representations, neural networks-based methods are usually the standard choice to induce the classifiers.

However, most of the time, the corpora used to pretrain such embeddings and the tasks used to evaluate those methods target formal texts, in the sense that the problematic features previously pointed out do not primarily define them. Thus, the question that arises is whether user intent-based systems should also use such methods to address their classification component, even regarding that the utterances consist of short texts generated by regular users – and not experts on the subject at hand – possibly with noisy information. Previous work has focused on creating utterance embeddings and on classifying intent with neural networks-based classifier (see section 2.3), but not with fine-tuning approaches focused on adjusting embeddings to the target domain vocabulary, or comparing the classification performances amongst different fine-tuning methods, such as BERT and ULMFit, to the best of our knowledge. Also, these works focus mainly on English datasets.

1.1 Objectives

Our main objective is to create automated models capable of identifying user intent on datasets that naturally contain noisy sentences distributed amongst a large number of classes. We direct our investigation to neural network-based models, considering the recent advances in this area and their broad use to generate numerical representations from texts. In this context, we aimed to answer the following questions:

- Broadly speaking, considering language model approaches that use static or dense vector representation of features that are either extracted or fine-tuned on downstream tasks, which of these approaches is best suited for intent classification tasks of such noisy texts?
- What is the impact of using such language model approaches generally focused on English corpora when applied in the pre-training and fine-tuning of language models on languages for which there is less availability of research data, such as Brazilian Portuguese?
- What is the impact of using a fine-tuning approach that allows adjusting a generic and publicly available language model to the more specific corpus of a noisy target dataset?

We conducted an extensive experimental evaluation to induce classification models from methods that range from Bag of words, passing through Convolutional Neural Networks and BiLSTMs using features extracted from embeddings, and arriving at recent fine-tuning-based approaches. In addition to the challenging characteristics posed before related to noisy-user generated content, the datasets investigated here also requires us to deal with two other issues. First, they have 64 to 1048 classes, different from the most used datasets representing binary tasks. Second, a subset is written in Brazilian Portuguese to observe if the most successful approaches also benefit tasks in a language other than English. We thoroughly investigate three datasets with those attributes (see Section 3.1) and compare the best strategy for training a neural network intent classifier. We also demonstrated that an interpretability method based on visualisation of the positive or negative contribution of sentence tokens to a classifier output could help understand the outcome of a prediction and highlight the reasons for misclassification.

1.2 Contributions

This dissertation contributes with methodologies and quantitative and qualitative experimental investigations of intent classification of short sentences. Regarding the methodological aspects, we focus on the two main components of modern text classification: generating numerical representations and building a classifier. The first component focus on how to induce numerical representations from texts. Here we investigate sparse representations with BOW, feature extraction with publicly available resources, generating embeddings from scratch from a language either closer to the domain or from the domain itself and adjusting the language model with fine-tuning strategies. The fine-tuning strategies include a task-adaptive pretraining of Portuguese and English BERT models and the strategy designed on ULMFit. The second component concerns how to aggregate word embedding to induce numerical representations for the set of short sentences constituting an example. In this case, we experiment with Bidirectional LSTMs and Convolutional Neural Networks for approaches that induce embeddings for words or characters. Approaches such as BERT already have a mechanism for computing sentence embeddings. The studies conducted here focus on both Portuguese and English languages. Pretrained embeddings and adjusted language models will be made publicly available so that future studies can benefit from them as a starting point. Finally, one of the methods designed and trained here, ULMFit, has helped improve a traditional approach that uses Virtual Operator data to decide an issue reported by a customer.

Regarding the experimental evaluation, the performance results from all these classifiers were compared, so we could better understand which one is best suited for the characteristics of the selected datasets. Considering the qualitative investigation, we visually demonstrated the importance of stop-words on intent classification and the impact of their removal from a dataset. We also included a visual representation of token importance to understand the impact of Task Adaptive Pretraining of BERT models used on intent classification. Lastly, we offered an alternative metric to evaluate the quality of a classifier prediction considering the averaged token importances of a sentence.

1.3 Organization of this Dissertation

This dissertation is organised as follows: Chapter 2 introduces the fundamental concepts concerning neural networks, how features can be represented, language models and architectures employed in this research. Related works are also presented in this chapter. Chapter 3 explains the methodology, the selected datasets and the neural network architectures employed in this work. Chapter 4.1 contains the results from this research. We present the final remarks of this work in Chapter 5.

Chapter 2

Background

In this chapter, we introduce the Deep Learning concepts that are key to understanding its application in Machine Learning and, more specifically, in the field of Natural Language Processing (NLP). The concepts addressed here are the ones employed in the development of this dissertation.

We start by presenting the Artificial Neural Network (ANN), its basic processing unit, the artificial neuron and how neurons are activated by using activation functions. We also show that multiple layers of neurons can be stacked together to form a Multilayer Feedforward Neural Network (MLFNN). The concepts of supervised and unsupervised training are also approached here.

Concerning the training of neural network-based models, We furthermore present the methodology followed in this dissertation on why datasets are split into training, validation and test sets and some of the techniques available to avoid overfitting during neural network training.

Next, we offer a brief description of some of the central neural network architectures applied to NLP used throughout the experiments in this dissertation. Lastly, we cover some of the techniques used to represent documents in NLP as sparse or dense vectors, including different approaches to learning these dense vectors.

2.1 Artificial Neural Networks

Artificial Neuron Networks' history dates back to 1943, with initial attempts to understand the biological brain and its interconnected neurons functioning. In [32], the authors present the idea of an artificial switch accepting input from other connected neurons using electric circuits. Later studies also expose the concept that frequently used connections between neurons become reinforced [55]. The concept of a *perceptron*, an artificial neuron that can be mathematically modeled, is introduced in [48]. The author develops a neurocomputer capable of recognizing characters, which despite its success it is limited to solving linear classification problems. This limitation is exposed in [37], a study that some authors refer to as being responsible for a period of decreasing interest in ANNs also known as *The Quiet Years* [5]. Among the achievements that help renew the ANNs interest is the resurfacing of the backpropagation algorithm in [51], which is initially proposed in [65]. In addition, contributions like *Convolutional Neural Networks (CNN)*, used to recognize handwritten digits [27] help to revive the interest in ANNs. The following years witnessed an increase in computer power, with faster CPUs but also with *Graphic Processing Units (GPU)* becoming generally accessible. Besides, with the popularization of the Internet, cell phones with embedded digital cameras, and other technologies supporting Big Data, an increasing amount of data becomes available to train more robust neural networks. Public datasets like ImageNet [14], a vast collection of annotated images which quickly turned into an annual competition in the search for the most accurate image classification algorithm, are some of the contributions to the massive evolution in the field of Deep Learning. Today we are surrounded by systems built atop neural networks, from smartphone cameras with facial recognition to automated speech-enabled customer support systems, smart assistants, and language translators, to name a few.

The Artificial Neuron

An artificial neuron can be seen as a special switch connected and accepting input from other similar switches. Each connection between neurons has an associated weight, which is then multiplied by the input signal. The weight defines the relevance of that connection and is the computational equivalent of the strength of a synapse - a biological connection between neural cells. The sum of its weighted inputs passes through an activation function that decides if the neuron output is activated or not [26]. *Softmax*, *Sigmoid*, *Rectified Linear Unit* (ReLU) and *Hyperbolic Tangent* are some of the most commonly used activation functions. Generally, the activation function needs to be nonlinear, allowing the neural network to learn nonlinearities in the data. It also needs to be differentiable so that the neural network weights can be optimized during training through backpropagation. The logical representation of a neuron is shown in Figure 2.1.

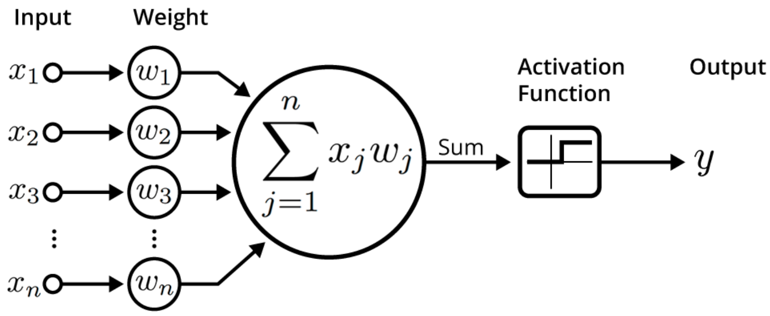


Figure 2.1: An artificial neuron (Figure from [53])

2.1.1 Feedforward Models

Feedforward neural networks (FNN) have their neurons arranged so that there are no feedback loops between layers, meaning that data flows through the neurons in a one-way fashion, as shown in figure 2.2. A *Multilayer Feedforward (MLFNN)* neural network is a type of FNN in which artificial neurons are arranged in layers. A layer may have one or more neurons. Each neuron in one layer serves as input to neurons in the following layer. An MLFNN comprises at least an input layer that receives data to be processed by the network and an output layer that provides the computation results. Besides those, it usually contains hidden layers, which are not externally accessible but have an essential role in transforming and yielding features [18]. An MLFNN is generally trained through *backpropagation*, an algorithm consisting of a *forward* and a *backward* phase. In the forward phase, data enters the input layer and propagates through the network. The output result is then compared with the expected result, and the error, computed according to a loss function, between both results is calculated. In the backward phase, the error calculated during the forward phase is propagated backward through the network, causing an adjustment in its weights to minimize the error computed in the forward phase.

An MLFNN can be trained in either *supervised* or *unsupervised* mode. Supervised mode needs a tagged dataset that will be used during the training phase. This dataset contains not only the inputs which will be used during training but also the expected output associated with that input. The loss computed during training evaluates how far the network output is from the expected result. Loss functions generally used include *Mean Square Error (MSE)* and *Cross Entropy Loss*, among others. Unsupervised mode, on the contrary, uses an unlabelled dataset to train a neural network.

Usually, a small portion of the data is separated from the training set to evaluate the progress of the training phase of a neural network. This *validation set* is used to

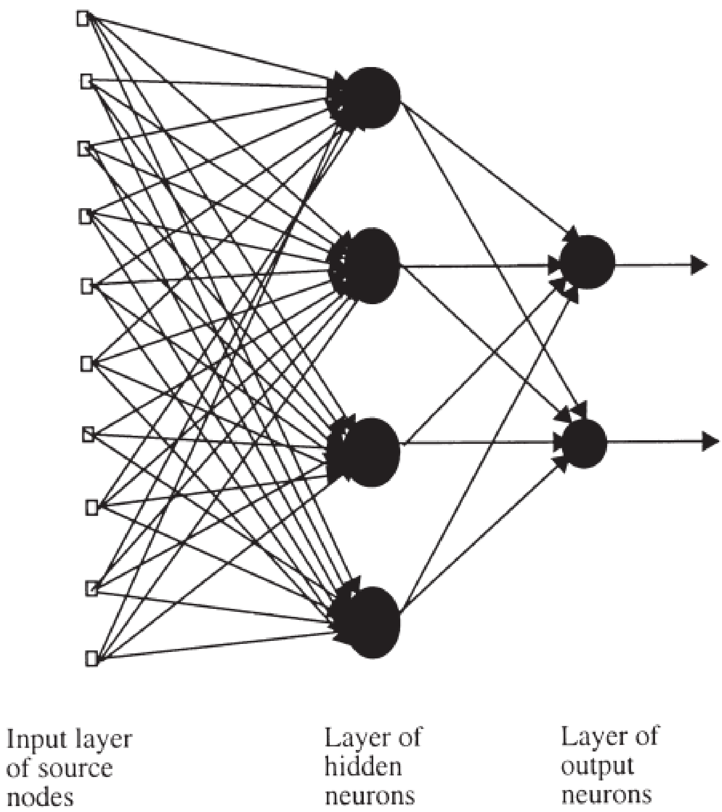


Figure 2.2: A feedforward Neural Network (Figure from [18])

test the neural network model after each training step and provides us with a means of checking how well a model performs with unseen data during training. If the validation loss starts to diverge from the training loss, that can indicate that the model is *overfitting*. Overfitting occurs when the model learns so well about the training set that it loses its ability to generalize and handle unseen data.

Overfitting can also be reduced by applying a technique called *dropout*, in which a portion of the network neurons is randomly deactivated during training. For example, a dropout with p=0.5 means a neuron has a 50% chance of being deactivated during a training step. This partial deactivation of the neural network forces different neurons to learn the same concepts, improving generalization [57]. Regularization is another technique that reduces overfitting by computation of a term added to the training loss that penalizes for high weights [38]. These are just some of the many available approaches to reduce overfitting.

MLFNNs are powerful *Universal Function approximators*, meaning that for any continuous function $f(x)$, there is a neural network $g(x)$ that will approximate it with an acceptable error [21]. However, they also have significant limitations. When applied in image classification, the number of weights that must be trained becomes overwhelmingly high when the image size increases. For instance, a neural network with a hidden layer containing eight neurons, accepting a color image of size 300x300 pixels as input, has 2,160,000 weights to be learned. Also, the spatial relationship between image features is not learned by MLFNNs, which cannot learn sequential information. In NLP, a feedforward neural network trained on document features loses information about the order of the sentences or words in those documents.

These are just some of the problems that led to the search for new architectures, such as Convolutional and Recurrent Neural Networks.

Convolution Neural Networks (CNN) were first applied in image recognition tasks [27]. They are built over the concept of a convolution operation. In a convolution, a filter (also referred to as a kernel) slides through the input matrix, and a scalar product is calculated between the subset of the input matrix covered by the filter (the receptive field) and the filter itself (Figure 2.3). CNNs can scan a large structure to identify local features, which can be combined in a second structure represented by a fixed-size vector. Convolution layers can be hierarchically combined so that more distant and non-contiguous features which are still related can still be detected. When employed over text, it is common to have CNNs with sequential (1D) convolutions [16] that scan k word-vectors at a time, where k is the size of the convolution filter, as shown in figure 2.4.

Pooling layers are also used to reduce the dimensionality of the convolution layer output by calculating the maximum or the average value on each of the sliding windows, thus highlighting relevant features irrespective of their location [16].

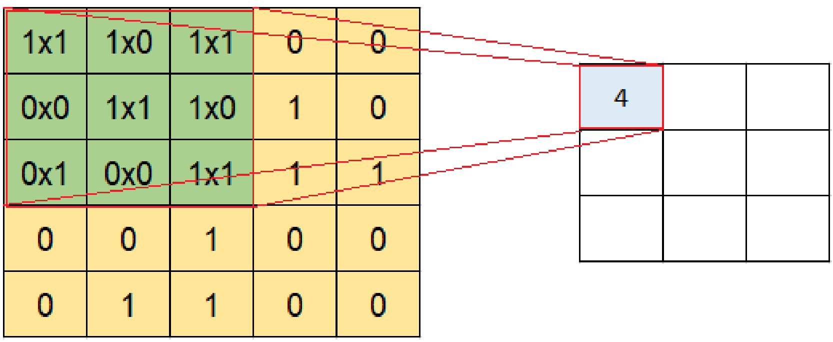


Figure 2.3: Convolution Operation

Although CNNs confer some ability to understand word order, this capability is restricted to identifying local patterns and does not consider patterns on more distant lo-

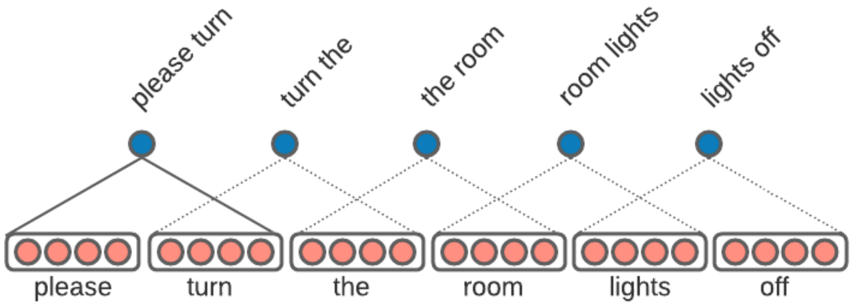


Figure 2.4: Illustrative example of a text convolution with kernel size k=2 (Figure adapted from [16])

cations in a sequence [16]. This and other drawbacks led to the adoption of architectures such as Recurrent Neural Networks.

2.1.2 Recurrent Models

Recurrent Neural Networks (RNN) were developed to be applied on time series or sequential data [50]. They introduce the concept of memory to neural networks and work by taking the hidden state of a feedforward neural network and using it as an additional input at each time step, thus keeping information from the previous states to discover dependency amongst the sequence elements (Figure 2.5). Considering an input sequence $x = (x_1, ..., x_{T\infty})$, the forward pass of an RNN can be described by the set of equations 2.1. Vectors $\mathbf{h}^{(t)}$ and $\mathbf{y}^{(t)}$ represent the hidden state and the output, respectively, at time step t. Vectors u, v, and w are the weights relative to the input, output, and hidden state connections, respectively, and b and c are bias vectors. The activation functions are represented by f and g.

Because of this state-keeping, the output gradients depend on all time steps, and not only the last one. Consequently, after the error is computed during training, it is backpropagated for every time step in the network. This algorithm is referred to as *Backpropagation Through Time (BPTT)* [66].

$$
\begin{aligned}
\mathbf{h}^{(t)} &= f(\mathbf{b} + \mathbf{w}\mathbf{h}^{(t-1)} + \mathbf{u}\mathbf{x}^{(t)}) \\
\mathbf{y}^{(t)} &= g(\mathbf{c} + \mathbf{v}\mathbf{h}^{(t)})
\end{aligned}
\tag{2.1}
$$

When applied in NLP, models that use RNNs can use entire sequences for training while still considering the words' order. They are generally not used alone but combined with other models. For example, an RNN can feed a feedforward neural network for classification tasks, working as an input-transformer for that network. RNNs break the Markov assumption's dependence, allowing a network to learn word dependencies based on all words that precede it [16]. However, because BPTT involves backpropagating the error function through the neurons behind the final output and through all time steps, gradients can get progressively so smaller to the point that the network does not train well. This situation, known as the *vanishing gradient problem* makes RNNs unfit to learn long dependencies [41].

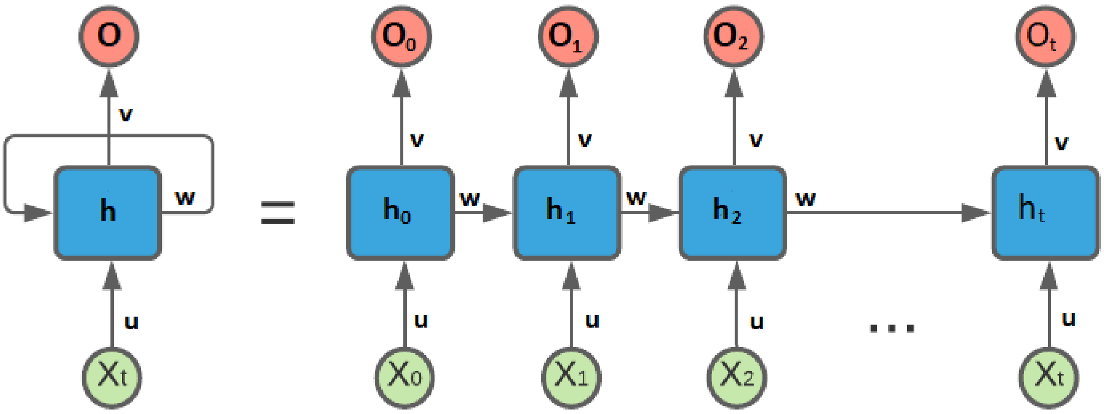


Figure 2.5: A Recurrent Neural Network

Long Short-Term Memory Networks (LSTM) [20] were created to solve the vanishing gradient problem of RNN's. An LSTM is based on a gating architecture in which access to the hidden state vector is controlled by a gate composed by vector $\boldsymbol{g} \in \mathbb{R}^{\ltimes}$ going through a sigmoid function. An LSTM has three of these gates: *input*, *forget* and *output* gates, which decide how and when the hidden state should be updated. Figure 2.6 shows the structure of an LSTM cell.

Bidirectional RNNs or LSTMs (BiRNN or BiLSTM) can be trained by combining the hidden state of a model trained with sequences in one direction with another model trained with sequences in the backward direction. In this way, words occurring both

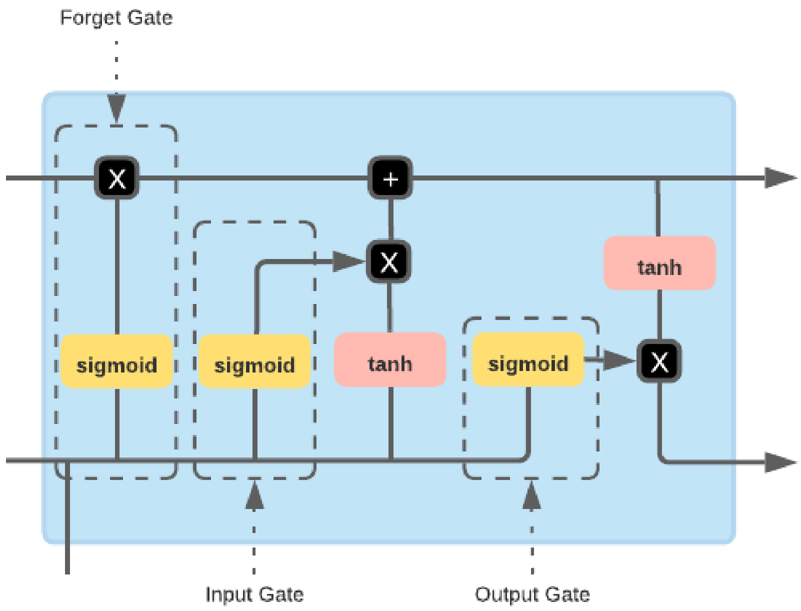


Figure 2.6: An LSTM Neural Network Cell

before and after a specific word can contribute to its representation.

2.1.3 Encoder-Decoder Models

The Encoder-Decoder architecture was first proposed in [11]. It consists of a neural network that *encode* a variable-length input sequence into a fixed-length representation called *context vector* and then *decodes* it into variable-length output sequence (Figure 2.7). This architecture was first applied in machine translation tasks but is also used in other tasks such as speech recognition [44].

In its most common formulation, the *encoder* block consists of an RNN that receives the sequence x as input and reads each word sequentially, updating the hidden state $\mathbf{h}$ according to equation 2.2. The context vector $\mathbf{c}$ is computed from the hidden state after the end of the sequence is reached (signaled by a special end-of-sequence symbol).

$$\begin{aligned} \mathbf{h}_{(t)} &= f(\mathbf{h}_{(t-1)}, \mathbf{x}_{(t)}) \\ \mathbf{c} &= q(\{\mathbf{h}_{(1)}, ..., \mathbf{h}_{(t)}\}) \end{aligned} \tag{2.2}$$

The *decoder* also uses an RNN trained to predict at each time step the next symbol y_t given the context vector $\mathbf{c}$ and the previous hidden state $\mathbf{h}_t$. Differently from a vanilla

RNN though, the hidden state of the decoder t is calculated by

$$\mathbf{h}_{(t)} = g(\mathbf{h}_{(t-1)}, y_{(t-1)}, \mathbf{c}) \tag{2.3}$$

The next symbol's conditional probability is represented by

$$P(y_t|\{y_{t-1}, y_{t-2}, ..., y_1\}, \mathbf{c}) = g(\mathbf{h}_t, y_{t-1}, \mathbf{c}) \tag{2.4}$$

The functions f and q and g are non-linear activation functions.

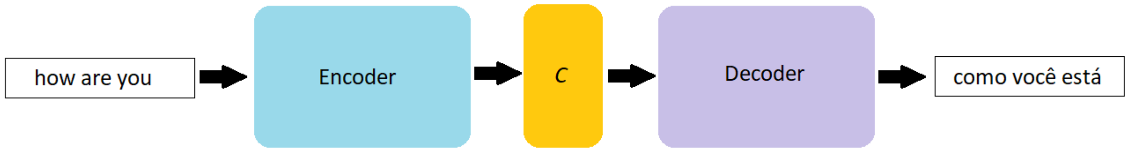


Figure 2.7: Encoder-Decoder architecture. c is the context vector.

Since the context vector has a fixed length, encoding information, especially from long sequences, into a compressed context vector may create an information bottleneck and lead to loss of previously learned representations, mainly those at the beginning of the sequence.

2.1.4 Encoder-Decoder Models with Attention

The attention mechanism was proposed in [4] for neural translation tasks as a means to avoid the "bottleneck" problem inherent to encoder-decoder models when applied to long sequences, as described in 2.1.3. Instead of using a fixed-length context vector, it relies on a body of information composed by the encoder and decoder hidden states and alignment between source and target sequences. The attention mechanism searches for specific positions in the source sentence for each word generated during decoding, looking for relevant information.

In its most basic formulation, the attention mechanism is integrated into an RNN encoder. Differently from the encoder-decoder architecture described in 2.1.3, the conditional probability of the next symbol y_t is conditioned on the input sequence vector x, and is represented by equation 2.5.

$$P(y_t|y_1, ..., y_{t-1}, \mathbf{x}) = g(y_{t-1}, s_t, c_t), \tag{2.5}$$

where s_t, the hidden state at time step t is computed by

$$\mathbf{s}_{(t)} = f(s_{t-1}, y_{t-1}, c_t) \tag{2.6}$$

The encoder, in this case, computes the context vector c_t as a weighted sum of a sequence of *annotations* represented by equation 2.7. These annotations encode information about the input sequence, focusing on each word's surroundings. This encoder uses a biRNN, which computes the hidden states' sequence from both a forward and a backward RNN. The annotation h_j of a word x_j is obtained by the concatenation of $\overrightarrow{h_j}$ and $\overleftarrow{h_j}$, the forward and backward RNNs' hidden states, respectively.

For each anotation h_j, a weight α_{ij} is computed by a softmax function (2.8), where e_{ij} is obtained by a feedforward neural network a that receives the decoder's hidden state s_{i-1} and the annotation h_j of the input sequence (Equation 2.9). This feedforward model is trained together with the remaining components. This description of the attention mechanism is illustrated in figure 2.8.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.7}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{2.8}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.9}$$

2.1.5 Transformer Models

The *Transformer* model architecture is an encoder-decoder that is entirely based on the attention mechanism and uses no RNNs or CNNs to learn global dependencies between input and output [62].

In this model architecture, shown in figure 2.9, the encoder is built by stacking N identical layers, each composed by a multi-head self-attention mechanism followed by

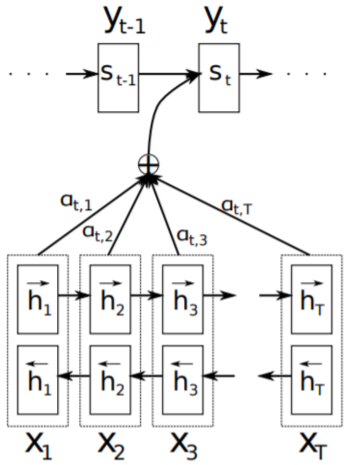


Figure 2.8: An illustration of the Attention mechanism, showing the annotation vectors h_t and their respective attention weights α_t (Figure from [4])

a fully connected feedforward network and layer normalization. Self-attention uses the concept of similarity between Queries and Keys to define an attention filter which is then applied to a Value Vector. In figure 2.10, the diagram in the left side shows how Attention is computed. Two copies of the input embeddings, representing the *Query (Q)* and the *Query (K)*, respectively, have their dot product computed and then scaled before passing through a softmax function. The resulting matrix, the *attention filter* is then multiplied by a third copy of the input embeddings, the *Value (V)* matrix. This multiplication highlights the information to which the network must focus on, or in other words, pay attention to. This set of operations is represented by equation 2.10:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V, \tag{2.10}$$

where d_K is the dimension of Q and K.

The Transformer uses three of these Self-attention functions in parallel, focusing on different representations of the input information. The results of each function are concatenated and fed into a linear layer which outputs the *Multi-head attention*. This concept is depicted on the right side of figure 2.10. This mechanism is used in three different places: encoder-decoder layers and inside both the encoder and decoder layers.

The decoder architecture is similar to the encoder, also consisting of N stacked identical layers. The main difference is an additional third sub-layer performing multi-head attention over the output of the encoder stack.

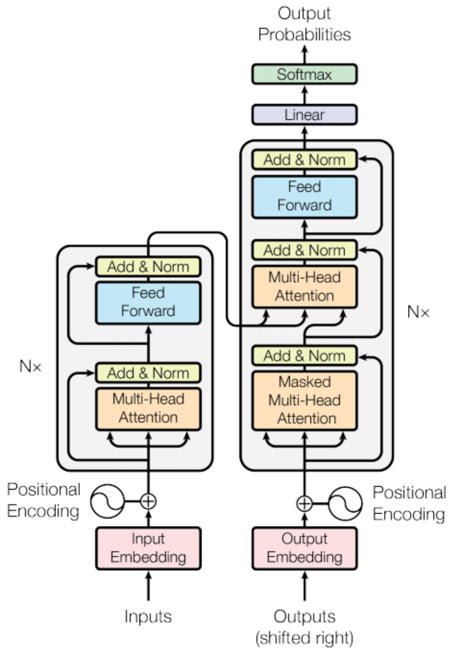


Figure 2.9: The Transformer model architecture (Figure from [62])

According to [62], the Transformer model architecture proved to achieve superior quality and more parallelization, requiring less training time than previous architectures.

2.2 Document Representation in NLP

When putting together Machine Learning and Natural Language Processing, it has become a standard practice to represent the symbolic elements of the language, namely, the words, sentences, or even entire documents, as numeric representations [52, 35]. This section describes the two main approaches used to represent textual content, using sparse vectors or dense vector representations.

2.2.1 Sparse Vector Representation

The concept of *Vector Space Model - VSM* was first proposed in [52] for an information retrieval system, and it is based on the *statistical semantics hypothesis*, which states that meaning can be extracted from statistical patterns of human words usage [61]. The authors proposed the idea that each document in a collection can be represented as a vector in a space vector. The distance between the vectors is proportional to their semantic similarity. Each element of the vector holds the value of some feature associated with a word present in the document's vocabulary. In [52] the authors used a *Term*

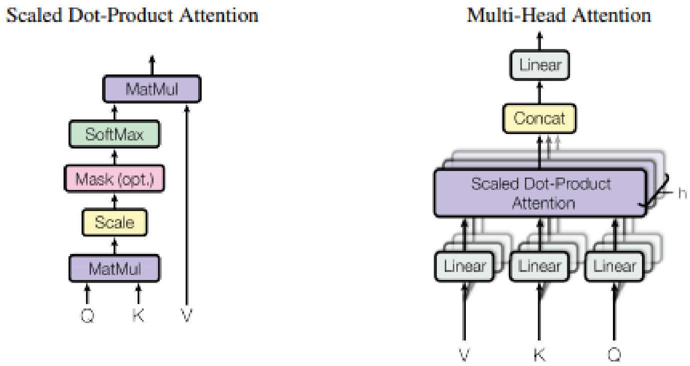


Figure 2.10: Representations of the Scaled Dot-Product Attention (left) and the Multi-Head Attention (right) (Figure from [62])

Frequency–Inverse Document Frequency (TF-IDF) document-matrix but word count or other frequency-based functions can also be used, such as Okapi BM25 [46], or just a binary value indicating the presence or absence of a vocabulary word in the document (a *one-hot* representation). TF-IDF measures the relevance of a word to a document in a set and is calculated by computing the *Term Frequency (TF)* of a word, which in its simplest form is just the count of how many times it appears in a document, and dividing it by the *Inverse Document Frequency (IDF)*. IDF is the logarithm of the ratio between the number of documents and the number of documents containing the word in question. BM25 (BM stands for *Best Match*) is a family of scoring functions commonly used in document ranking, based on query terms appearing in each document.

The use of such frequency-based functions is based on the *Bag of Words hypothesis*, which proposes that the relevance of a document to a query can be indicated by the frequency of words in the document. The term *bag* also refers to the fact that the vector does not carry any information regarding the structure or order in which words appear in the document.

Considering that documents use just a small portion of the vocabulary, the vector representation is *sparse*, meaning that the majority of its elements will have a value of zero.

Consider the example in figure 2.11. The dictionary contains all words present in the dataset, and an index is attributed to each word. Two example sentences are also shown, with their corresponding sparse-vector representations. Each element in the sparse vector informs the presence or absence of that particular word. Since the dictionary size determines vector size, sparse vectors can become quite large. Also, there is no dependency information between words. In the same example, the word *cat* is so unrelated to *dog* as it is to *sat*.

word	index
the	0
cat	1
sat	2
on	3
mat	4
dog	5
saw	6

Example	Vector Representation						
	0	1	2	3	4	5	6
the cat sat on the mat	1	1	1	1	1	0	0
the dog saw the cat	1	1	0	0	0	1	1

Figure 2.11: Sparse-vector encoding example

2.2.2 Dense Vectors Representation

This section introduces the concept of embeddings - dense vectors representing words and the associated syntactic and semantic relationship amongst them - as an alternative approach to a sparse vector representation. We discuss some of the most relevant models proposed for training embeddings and how these models evolved from the evolution from static representations that do not consider different word contexts to contextualized ones. We also explain how transfer learning in NLP evolved from feature extraction to a more clever fine-tuning approach. Figure 2.12 graphically represents the embedding approaches, base models, pretraining, and fine-tuning approaches, which will be discussed here.

2.2.2.1 Static Word Embeddings

The idea of representing words as dense feature vectors, thus uncovering syntactic or semantic relationships amongst them, was built over the concept of distributed representations [19]. Using fixed-length dense vectors to represent words helps to reduce the curse of dimensionality and improves generalization.

In [6], the authors propose the use of a neural network to train a *Language Model (LM)*, a large-scale statistical model of the distribution of word sequences. They also introduce the concept of an *embedding layer* referring to the projection layer where word vectors are input. In [12], the authors build a neural network semi-supervised model with the main purpose of training word embeddings, decoupling it from downstream tasks [3]. The unsupervised pretraining of word-embeddings became popular in 2013, with

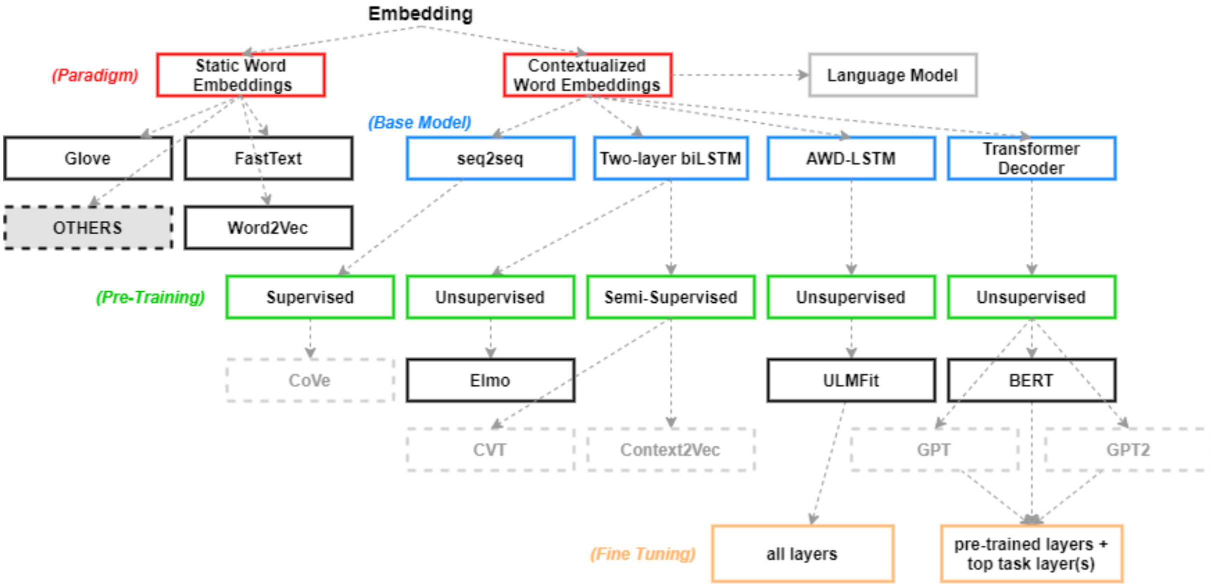


Figure 2.12: Some of the embeddings approaches and models applied in NLP tasks. The boxes with solid lines represent the models investigated in our study.

the development and public availability of LMs pretrained using Word2Vec, a software introduced by [35]. The authors propose two methods to produce a dense vector space containing the distributed relationship between words in vocabulary: (a) *Continuous Bag-of-Words (CBOW)*, which predicts a word based on its surrounding neighbors and (b) *Skip-Gram* which predicts the surrounding context words based on a specific word. The skip-gram algorithm works by creating a vocabulary of words, with each word pointing to its respective word vector. These word vectors are randomly initialized. A window of size m is set, so for each word at position t - the center word - the model tries to maximize the probability of predicting the next and previous m words - the context words - given the center word, as seen in figure 2.13. The CBOW algorithm works in the opposite way.

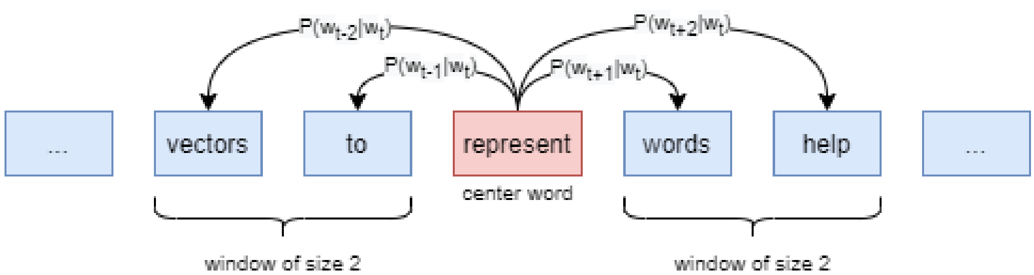


Figure 2.13: Skip-gram model introduced by Word2Vec

GloVe [42] proposes a model using CBOW and Skip-gram for acquiring local context and a method called *Global Matrix Factorization (GMF)* to include global statistics. GMF makes use of a co-occurrence matrix which is built by parsing the corpus vocabulary and calculating the number of words co-occurring in a specific window. For example, using a window of size one, the sentences *I like NLP, I like deep learning* and *I enjoy flying* would generate the co-occurrence matrix represented in Figure 2.14. The words in the vocabulary are listed in the first row and the first column. Next, the algorithm counts how many times a word in the first row appears in the specified window around a word in the first column. In the same example, the word pair *(i, like)* has a count of 2 because it appears twice in the analyzed sentences.

FastText [7] offers a skip-gram model trained with a subword vector representation approach, where words are represented as bags of character n-grams. Each character n-gram is represented by a vector. Thus, words are represented as the sum of such representations.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	0	1	1	0

Figure 2.14: Example of a window-based (word-word) co-occurrence matrix (Figure adapted from [1])

For example, the FastText 3-gram representation of the word *factor* is *<fa, fac, act, cto, tor, or>*. The characters < and > define word boundaries and are used to distinguish the word from an equal n-gram. In this way, using the same example, if the word *factor* is found in the vocabulary, it will be represented as *<factor>*. This strategy not only preserves the meaning of words that otherwise might collide with some subwords n-grams but also helps to capture suffix and prefix meaning [58]. Using this approach also allows the representation of out-of-vocabulary words. For instance, the 3-gram representation of the words *precaution*, *prejudice* and *preview* contain the token *pre*, a prefix whose meaning can be understood by FastText.

The embedding approaches discussed so far generate static embeddings, meaning that a vector representing a word is the same, irrespective of the context in which that word is used. Thus, for instance, the word matter, which has different meanings in the sentences *the dark matter mystery* and *it does not matter* is represented by the same vector. This problem with polysemous words is an essential limitation of static embeddings.

2.2.2.2 Contextualized Word Embeddings

Embeddings from Language Models (ELMo)[43] introduces the concept of *contextualized embeddings*, an approach that can deal with polysemy by considering the surrounding words in a sentence to understand context. ELMo is based on a bidirectional LM architecture (BiLM), which combines a forward model that computes a token's probability given the previous tokens in the sentence and a backward language model running in the opposite direction, which predicts a token based on the token sequence ahead of it, as shown in figure 2.15. ELMo uses two of these BiLMs. Sentence words are input into a character-level CNN and converted into raw word vectors that enter the first biLM, which outputs intermediate word vectors. These vectors are used as input to the second biLM. ELMo vectors are represented by the intermediate and raw word vectors' computed weighted sum. This architecture allows the model to learn different vector representations for the same word, capturing syntax, semantics, and other complex characteristics and variations of such characteristics used in different contexts. The authors trained ELMo embeddings on the 1 Billion Word Language Model Benchmark [9]. Its performance was then tested across six different NLP downstream tasks, achieving new state-of-the-art results on all of them.

2.2.3 Fine-Tuning - Adjusting the Weights of a Model According to a Task

By separating language model pretraining from downstream tasks, the approaches discussed so far demonstrate that the concept of *Transfer Learning*, widely used in *Computer Vision (CV)* tasks, can also be applied to NLP. Instead of being randomly initialized from scratch, a language model can be trained unsupervised on a large source task dataset and then have its weights used on a supervised downstream task trained on the target data. In [49], the authors refer to this mechanism as *adaptation*. Adaptation can occur in either one of two ways: Through *features extraction*, when the pretrained embeddings are used as fixed weights in the downstream task, or through *fine-tuning* when embeddings are adjusted to the target task.

All pretrained language models presented in the previous section are based on the features-extraction method. Letting the embeddings be fine-tuned jointly with the target

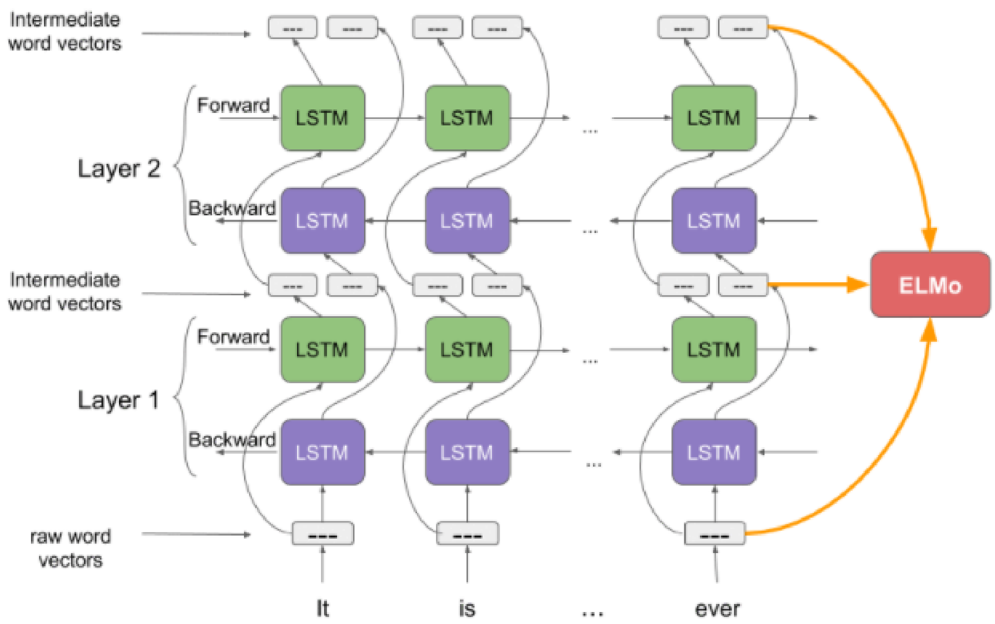


Figure 2.15: ELMo model architecture (Figure from [23])

task may lead to loss of learned embeddings relationships - a phenomenon known as *catastrophic forgetting* [22]. Also, language models trained on small datasets can overfit.

Universal Language Model Fine-tuning (ULMFiT) [22] is a transfer learning method that, according to the authors, addresses both overfitting and catastrophic forgetting issues by introducing a three-step approach for fine-tuning a language model (Figure 2.16):

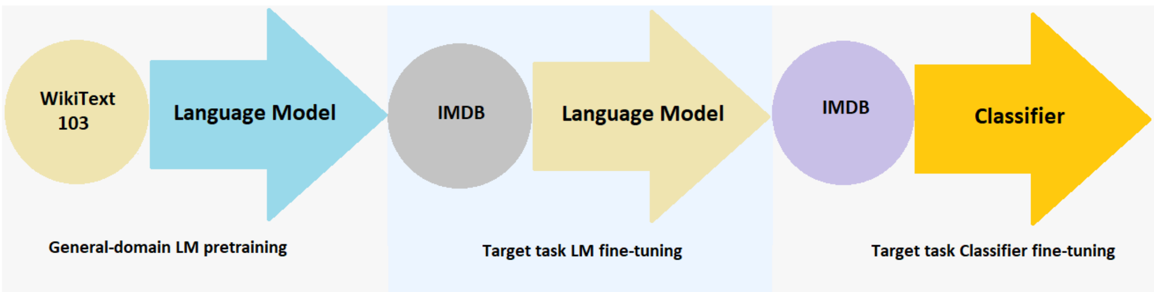


Figure 2.16: ULMFit 3-step approach (Figure adapted from [54])

General-domain LM pretraining: An LM should be trained on a large corpus, capturing the most general aspects of language. Tasks with particularly small datasets benefit from pretraining. The authors pretrained ULMFiT on Wikitext-103 [34], a 103 million words corpus. The LM uses 3 layers of an Asynchronous Stochastic Gradient Descent (ASGD) Weight-Dropped LSTM (AWD-LSTM) architecture [33], a regular LSTM with several dropout hyper-parameters.

Target task LM fine-tuning: In this step, the LM is fine-tuned on the target task data, which generally has a different distribution from the LM source data. This technique, also known as *task-adaptive pretraining (TAPT)*[17] allows training of powerful LMs even for small target datasets. The authors also propose two techniques in this step:

1. *Discriminative Fine-Tuning* allows each model's layer to be trained with different learning rates. This is based on the principle that different layers learn different features, hence requiring different learning rates. The authors empirically concluded that first finding the last layer's learning rate η^L by fine-tuning only the last layer and using $\eta^{l-1} = \eta^l/2.6$ as the learning rate for the lower layers provided the best results.

2. *Slanted Triangular Learning Rates (STLR)* proposes an initial steep linear increase in the learning rate, followed by a slow decay to help the model parameters be fine-tuned. According to the authors, this technique helps quick convergence still at the beginning of training.

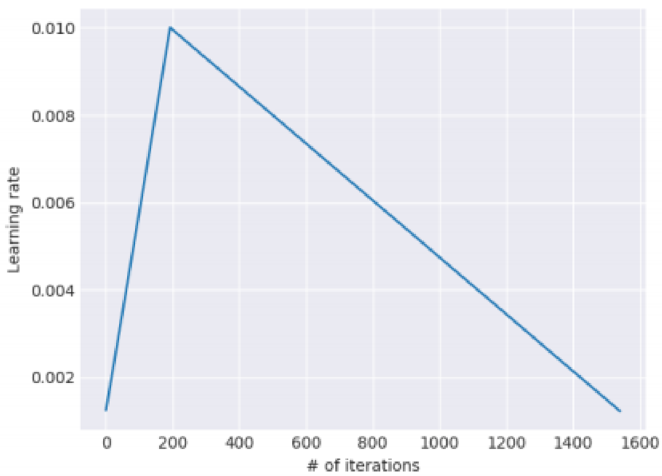


Figure 2.17: STLR in ULMFiT as a function of the number of training iterations (Figure from [22])

Target task classifier fine-tuning: a classifier is built by adding two feedforward layers and a softmax normalization layer to the LM. These layers contain the only

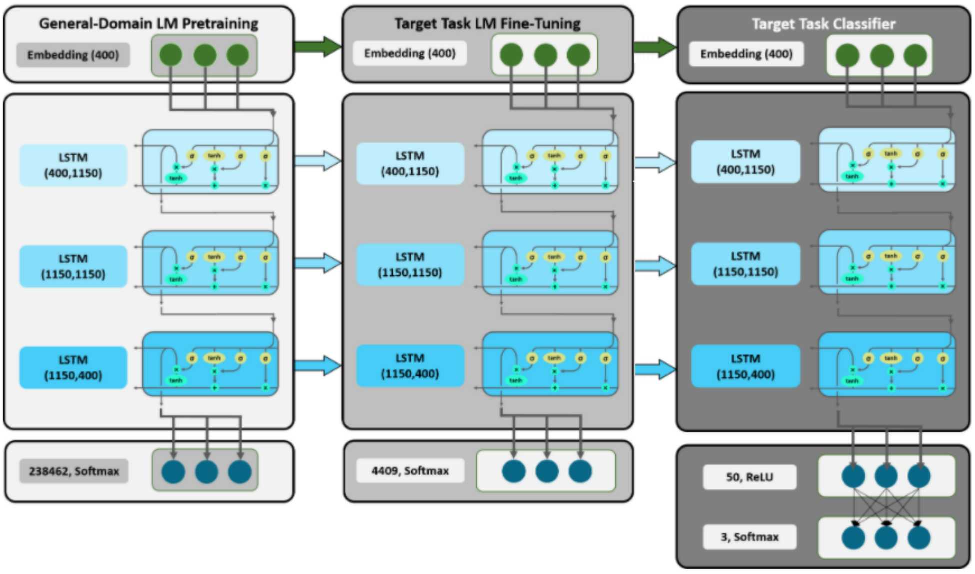


Figure 2.18: STLR in ULMFiT as a function of the number of training iterations (Figure from [22])

weights that will be learned from scratch. Two new techniques are used in this step:

1. *Concat pooling:* The last time step's hidden state vector is concatenated with both the computed max-pooled and mean-pooled vectors, which are calculated over as many time steps as fit in GPU memory. Using only the last time step's hidden state might otherwise lead to loss of relevant information.

2. *Gradual unfreezing:* To avoid catastrophic forgetting, and considering that the last layer contains the most specialized knowledge, the model is gradually unfrozen from the last layer backwards. In the first training epoch, only the last layer is fine-tuned. The next layers are gradually unfrozen and fine-tuned in the subsequent epochs until all layers converge.

ULMFit's general architecture is depicted in Figure 2.18, showing the stack of LSTM layers that compose the model. We can see that the output softmax layer in the diagram in the left contains, as an example, 238,462 dimensions, each one corresponding to a token in the EN Wikipedia vocabulary. In contrast, the middle diagram, representing the LM fine-tuned on the target task, contains only 4,409 dimensions. Token embeddings existing in both source and target vocabulary are kept, but the ones from the original LM that do not exist in the target vocabulary are removed. New tokens present only in the target vocabulary are initialized with the row mean of all source embeddings. In the first epoch, the LSTM layers' weights are frozen, and only the embedding and softmax layers are trained. The weights will be unfrozen for the remaining epochs, allowing the LSTM layers to be fine-tuned (Figure 2.19).

Analogously, gradual unfreezing is applied during the classifier fine-tuning. First, only the classifier's softmax output and the embedding layers are allowed to be updated. Then, only the LSTM weights are kept frozen, and finally, the whole network is fine-tuned in the subsequent epochs (Figure 2.20)

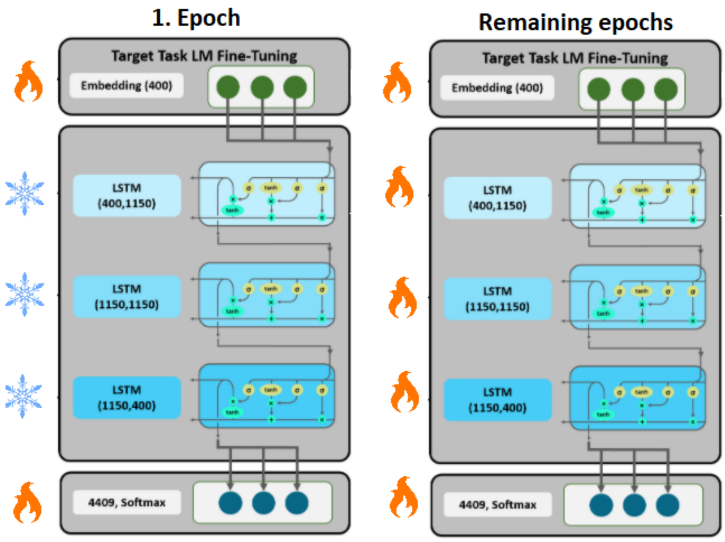


Figure 2.19: Gradual unfreezing applied during LM fine-tuning on the target task. Fire an snow-flake symbols represent non-frozen and frozen layers, respectively.

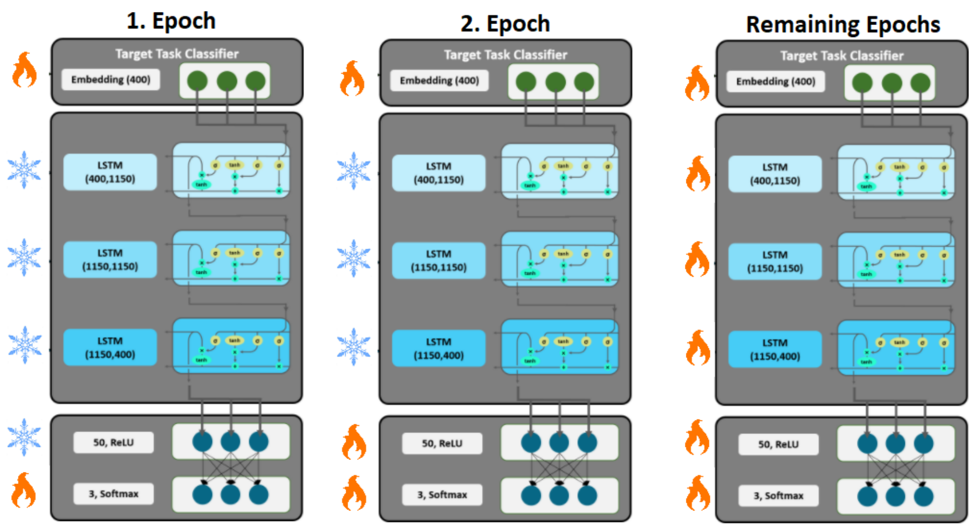


Figure 2.20: Gradual unfreezing applied during classification downstream task

All pretrained language models presented so far are unidirectional, meaning that features are extracted from a left-to-right or a right-to-left LM. ELMo token representation is generated by concatenating a left-to-right and a right-to-left representation, but it still uses two different models.

Bidirectional Encoder Representations from Transformers (BERT)[15] uses the Transformer with an Attention mechanism to learn contextual relations between words or subwords in text. Its architecture consists of an encoder built by stacking several layers of Transformers. Since the objective is to generate contextualized representations, only the encoder part of the transformer is used. Each token representation output by an encoder layer represents features for that token and is used as input to the next encoder layer, as shown in Figure 2.23. BERT is inspired in the Cloze test [60] and uses a *Masked Language Model (MLM)* as a pretraining objective. The LM is trained to predict a token that was previously masked at random in the input sentence. The model also uses *Next Sentence Prediction (NSP)* to pretrain the LM for downstream tasks such as *Question Answering (QA)* and *Natural Language Inference (NLI)*. NSP allows the LM to learn the relationship between two sentences, a knowledge that the LM does not acquire directly.

Before being input into the encoder, BERT sentences are converted to a multi-dimensional vector representation computed by an element-wise sum of token, segment, and positional embedding representations.

Token representations are obtained by first converting sequences into subword tokens using a segmentation algorithm called WordPiece [67], which generates a vocabulary of 30.000 tokens. The vocabulary also contains special tokens used to signal the beginning of a sentence ([CLS]), and to separate sentences packed together as sentence pairs (a sentence pair is referred to as a sequence) used during the NSP task ([SEP]). Examples of inputs using one or two sentences can be seen in figure 2.21. A token embeddings layer converts each token in a sequence into a multi-dimensional vector representation. BERT authors tested the model with representations of 768 and 1024 dimensions.

2 Sentence Input:

```
[CLS] The man went to the store. [SEP] He bought a gallon of milk. [SEP]
```

1 Sentence Input:

```
[CLS] The man went to the store. [SEP]
```

Figure 2.21: BERT examples with one and two sentences and its special tokens

Segment embeddings distinguish tokens from each of the sentences in an input pair. They consist of a 2-vector representation, with index 0 being assigned to tokens from the first and 1 to the second sentences in the pair.

Positional embeddings contain sequential knowledge related to the input sequences, with each position in a sequence containing its embedding vector. BERT allows for up to 512 positional vectors per sequence.

For downstream classification tasks, a classification layer can be added on top of BERT, and its weights will be trained along with the fine-tuning of the entire model. BERT can also be used as a features extraction embedding layer. The [CLS] token acts as a special classification token that can be used to represent a sentence. The authors of the BERT paper suggest that concatenating the last four layers' hidden states provide the best-contextualized embedding representation.

The authors pretrained BERT using the BooksCorpus (800M words) [70] and English Wikipedia (2.500M words). BERT models were initially available to the public on two versions: *BERT Base* has 12 encoding layers, 768 dimensions on its hidden layers, and 12 attention heads, while *BERT Large* has 24 layers, 1024 dimensions on its hidden layers and 16 attention heads. Also, BERT was made available on both *cased* and *uncased versions*, distinguishing between lower and upper case words. BERT pretrained models are available in several languages, amongst these BERTimbau [56], trained on the Brazilian Portuguese corpus BRWAC [63]. BERT authors also offer a multilingual version, trained in 104 languages.

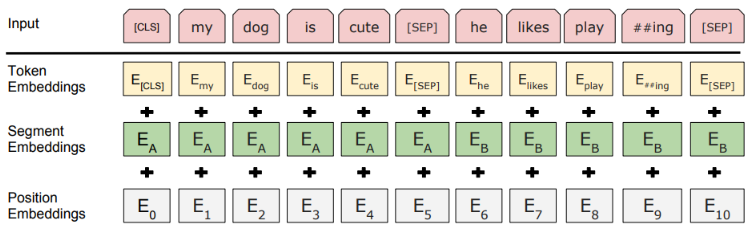


Figure 2.22: BERT input embeddings (Figure from [15])

2.3 Related Work

Previous works have addressed the task of classifying user intent from open-domain dialogue act classification with convolutional and recurrent neural networks, using or not pretrained embeddings. In [24], a Hierarchical Convolutional Neural Network (HCNN) is used to generate word vectors that are fed into a Recurrent Convolutional Neural Network (RCNN) outputting the dialogue act label. A similar approach is used by [28], which per-

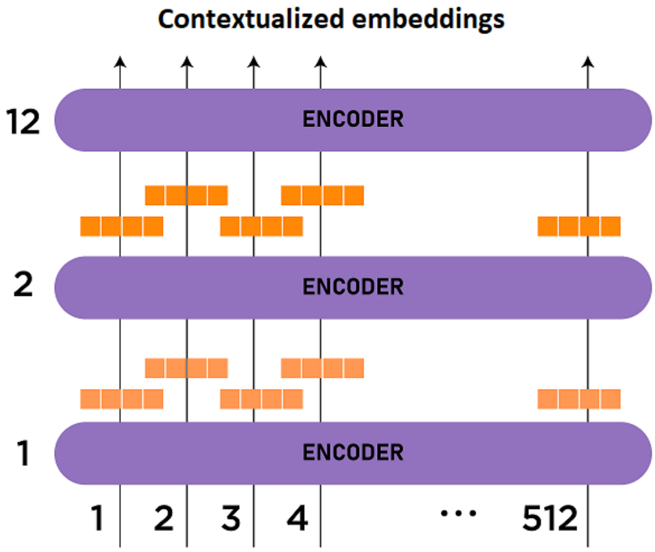


Figure 2.23: A representation of BERT's stack of encoders

forms short-text classification using a model consisting of two parts. The first one uses either an RNN or a CNN to generate a vector representation for each sentence, and the second part uses an LSTM that classifies a sentence based on its vector representation and the representations from preceding sentences. The work of [45] focuses on character-level tokens input into a set of parallel CNNs for dialogue act prediction. A similar approach is presented in [69], using character-level CNNs for multiclass text classification on eight large-scale datasets containing from two to 14 classes.

One of the datasets investigated here, namely, The Virtual Operator dataset, shares attributes with those tasks, but it is part of a system designed to answer customers automatically by phone and redirect them to a more specific problem solver. Thus, it has one additional challenge: the automatically captured talk from the phone is far from perfect. Moreover, besides investigating CNN and LSTM methods from pretrained word and character embeddings, we also include pretraining from tweets and fine-tuning the embeddings via ULMFit. Focusing specifically on user intent classification in conversational agents, in [8] a method is presented for evaluation of commercial Natural Language Understanding (NLU) services. The authors introduce two datasets - ChatBot Corpus, containing 206 questions distributed amongst seven intents from a Telegram chatbot used to answer questions about public transport; and the StackExchange[1] Corpus, which encompasses questions from ask ubuntu and Web Applications, two platforms from StackExchange, which combined, contain 290 questions and 13 intents. In [13], the authors propose a

[1]https://stackexchange.com/

method for the generation of data that can be used to train or evaluate NLU devices. They also made available a dataset consisting of around 16K crowdsourced sentences distributed amongst seven intents. The amount of intents in those datasets is considerably small compared to the Virtual Operator dataset used in this investigation, which contains 121 classes. In [68], three commercial services were compared to a free language model-based tool. The commercial tools perform slightly better, probably due to the much broader set of examples to which they are presented every day. The authors also used a crowdsourced dataset consisting of 25,716 utterances annotated on 64 intents. This dataset is, to our knowledge, the largest publicly available NLU evaluation dataset in terms of classes and was selected for our investigation. In [39], the authors present a methodology for intent classification on a chatbot answering career-related questions, using RNNs connected by a rule-based classifier for category and subcategory classification. To the best of our knowledge, there are no similarly reliable works on intent classification focusing on the Brazilian Portuguese language, which is also the focus of this investigation.

Recently, [30] states that classifying intent from utterance-level in conversational agents is a challenging task due to the size and sparsity of the sentences and the need of representing different languages and domains. To address such challenges, they proposed a method to induce dynamic utterance-level vector representations. This representation uses six metrics - IDF scores of unigrams, character n-grams, word bigrams and trigrams, utterance length and word order - which are used to compute a similarity-based representation for each utterance. This approach achieved a 3% improvement over BOW on supervised classification tasks. In [10], the authors present a model based on BERT for joint intent classification and slot filling that outperforms previous approaches, which modelled intent classification and slot filling separately. This model uses the hidden state of BERT's first special [CLS] token to take intent predictions, and the remaining tokens' hidden states are fed into a softmax layer that outputs the slot filling labels. Here, we investigated the benefits of using fine-tuning and pretrained embeddings. Combining their approach with fine-tuning methods is an exciting venue for future work. In table 2.1, we present a summary of the related works mentioned in this section.

Table 2.1: Summary table of related work.

Reference	Title	Description
[24]	Recurrent Convolutional Neural Networks for Discourse Compositionality	Multiclass Dialogue act classification using RCNN fed with vectors generated by HCNN
[28]	Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks	Multiclass classification of short texts using an LSTM classifier fed with vector representations of the sentence and preceding sentences generated by a RCNN or CNN.
[45]	A Study on Dialog Act Recognition using Character-Level Tokenization	Multiclass dialogue act prediction using character-level tokens input into parallel CNNs
[69]	Character-level Convolutional Networks for Text Classification	Multiclass text classification using character-level CNNs
[8]	Evaluating Natural Language Understanding Services for Conversational Question Answering Systems	A method for evaluation of commercial NLU services. Introduction of two multiclass benchmark datasets.
[13]	Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces	A method for the generation of data that can be used on training or evaluation of NLU devices. Introduction of a new multiclass benchmark dataset.
[68]	Benchmarking Natural Language Understanding Services for building Conversational Agents	Benchmark among three commercial NLU services and a free language model-based tool. Introduction of a new multiclass benchmark dataset with 64 classes.
[39]	Intent Detection and Slots Prompt in a Closed-Domain Chatbot	Multiclass Intent classification on a chatbot using RNNs connected to a rule-based classifier.
[30]	SimVecs: Similarity-Based Vectors for Utterance Representation in Conversational AI Systems	Proposal of a method to induce dynamic utterance-level vector representations using six metrics, based on IDF scores of n-grams, character n-grams, utterance length and word order.
[10]	BERT for Joint Intent Classification and Slot Filling	Joint intent classification and slot filling using BERT.

Chapter 3

Methodology

Our primary goal in this dissertation is to investigate the use of different pretrained embeddings and fine-tuning approaches to solving user intent classification problems in noisy datasets, with a large number of classes and highly imbalanced. We comparatively evaluated different aspects of Language Models pretraining. We evaluated different neural architectures and embeddings approaches, using static or contextualised embeddings, with features extracted or fine-tuned on a downstream task. We investigated if there was any benefit of using a less formal language corpus, such as tweets when pretraining an LM. We also addressed whether the same TAPT approach used in ULMFit can benefit BERT models trained for intent classification, as suggested by [17]. Lastly, we compared the performance of intent classifiers trained on BERT Multilingual and BERT language-specific models. Different pretrained Language Models were trained on a downstream classification task with or without an intermediate task-adaptive fine-tuning step to accomplish this set of investigations. The following sections will provide more details about the different aspects of our study.

User intent data collected from standard platforms such as PDAs or automated customer support services hold one or more of the following attributes: (i.) the examples are short, sparse sentences; (ii.) they are inherently multiclass to uphold for different intents; (iii.) the sentences are usually noisy in the sense that they lack proper grammar; and (iv.) the distribution of sentences per class is skewed. One dataset in English (EN) and two in Brazilian Portuguese (PT-BR) were used in our research.

3.1 The Datasets

3.1.1 Virtual Operator

The *Virtual Operator* dataset contains 669,929 Brazilian Portuguese utterances collected from a customer technical support speech-automated system running on a large telecommunications service provider company. Each of the samples in the dataset corresponds to a customer's answer to the question, "How may I help you?". The sentences are acquired by an Automated Speech Recognition (ASR) engine, which receives audio streams directly from the Public Switched Telephone Network (PSTN) and converts the caller's spoken utterances into text. The quality of the audio stream arriving at the ASR engine is influenced by factors such as the amount of environmental noise, audio level, the quality of the PSTN, the presence of noise-canceling devices, the use of lossy audio codecs, and the presence of more than one talker, amongst others. Such factors, as a consequence, affect the precision of ASR results and contribute to the generation of a noisy dataset.

Each transcribed utterance is fed into a *Deterministic Intent Parser* that uses regular expressions to automatically identify the intent and classify the utterance according to its respective label. Inaccuracies in the set of regular expressions used for each label classification or conflicts between regular expressions - when the utterance matches two or more regular expressions in different sets - can lead to misclassification and add noise to the dataset.

The dataset contains 121 labels, each corresponding to a user's intent when calling the support service. Each sentence is automatically classified using the same deterministic intent parser described in the previous paragraph. So, sentences are also subject to misclassification due to inaccuracies in the parser's regular expressions. The dataset is highly unbalanced - the label with the smallest set has 11 samples, while the most massive set contains 72,762 samples. The complete label distribution is available on appendix A.

Sentence mean token size is 7.6, with a standard deviation of 8.6. The smallest sentence has just one token, whereas the longest one has 72 tokens. Token size distribution is shown in Figure 3.1.

This variability in the length of sentences is partially explained by at least two distinct behaviors amongst the service users. First-time users or users believing that they are talking to a human operator tend to be wordier, while experienced users, or users who are aware they are using an automated system, use concise sentences that contain a single

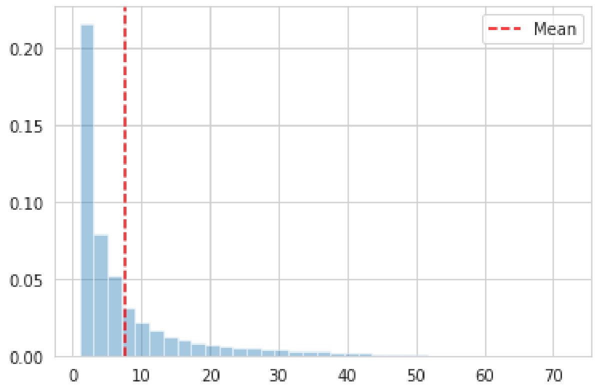


Figure 3.1: Virtual Operator dataset - sentence length distribution (in tokens)

token sequence describing an intent. Some examples of wordy and concise sentences describing the same intents can be seen on table 3.1.

Table 3.1: Examples of wordy and concise sentences describing the same intent

User Profile	Sentence	Label
wordy	é um aparelho que foi acrescentado o quarto e aí nao pega alguns canais a globo 38 nao pega alguns canais nao pegam todos os canais que pegam na sala	Genérico.Canal Globo não pega
concise	nao tenho acesso a globo	
wordy	eu fiz alteraçao no meu plano para ter hd em segundo ponto entao estou aguardando que me traga um modem para o segundo ponto	Qualificado.NãoTéc ponto adicional
concise	pedir ponto adicional	
wordy	á faz uma semana que está dando uma mensagem na tela dizendo que está perdendo o sinal do satélite falta de comunicaçao e voce assiste normal de repente carlos final fica tudo a tela azul e já estou aparelho da [company name] se desliga	Qualificado.Equipamento liga e desliga sozinho
concise	meu aparelho fica desligando	

The most frequent 3-grams and 4-grams, seen in figures 3.2 and 3.3, also show some token sequences, like *motivo da ligação*, *motivo da ligação que* and *da ligação que eu*, which can be associated to wordy sentences.

Figures 3.4 and 3.5 show the distribution of the most frequent tokens in the dataset vocabulary and the most frequent stop words, respectively.

3.1.2 NLU-Evaluation

The *NLU-Evaluation* dataset is built from real user data through crowdsourcing as a benchmark of different NLP tasks[68]. It contains questions and commands representing interactions between a user and his *Portable Digital Assistant (PDA)*, covering the following scenarios: audio, audiobook, calendar, cooking, datetime, email, game, general, IoT, lists, music, news, podcasts, general Q&A, radio, recommendations, social, food takeaway, transport, and weather.

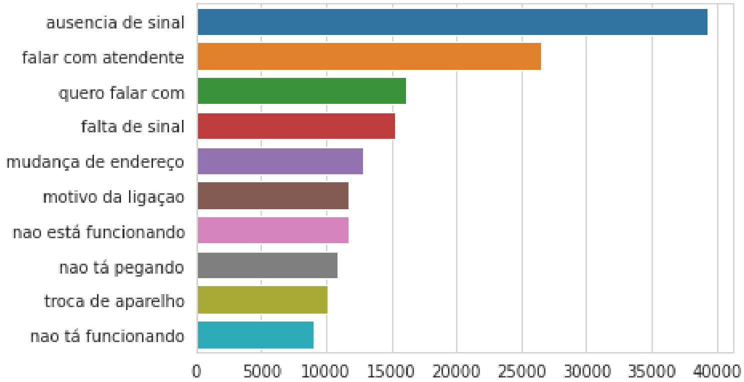


Figure 3.2: Virtual Operator dataset - Most frequent 3-grams

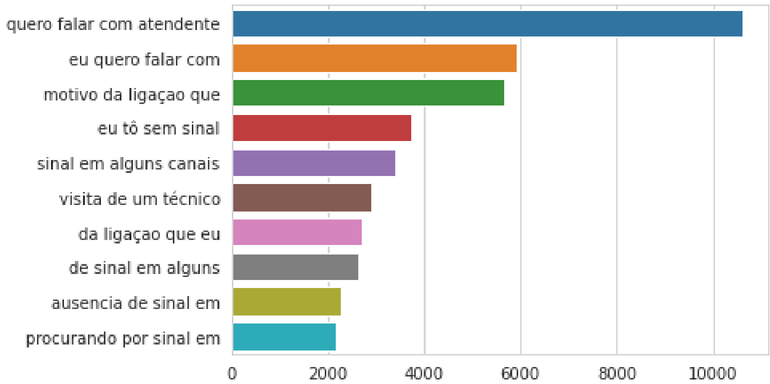


Figure 3.3: Virtual Operator dataset - Most frequent 4-grams

The dataset contains 25,578 user utterances in English, classified in 64 different intents with a mean sentence size of 6.5 and a standard deviation of 3.3. The distribution of sentence lengths shown in Figure 3.6 is less sparse than the one in the Virtual Operator dataset and can be explained by the fact that a user tends to speak to its PDA using concise, short and objective command-like utterances. The label sets range from 171 to 1,218 samples, meaning that this dataset is also highly unbalanced. Label distribution is available on appendix A. A closer look into the data shows some noise, like typos, as in the example *is there a new email in the inbo <unk> from jay*, or occurrences of the same utterance with different labels, such as *agree*, labeled as *general_feedback* in one record, and as *podcasts_play* in another one.

The distribution of most frequent 3-grams and 4-grams (figures 3.7 and 3.8) gives an idea of the command-like or question-based characteristic of the user utterances in this dataset. Likewise, the distribution of the most frequent tokens in the dataset vocabulary and the most frequent stop words can be seen in Figures 3.9 and 3.10, respectively.

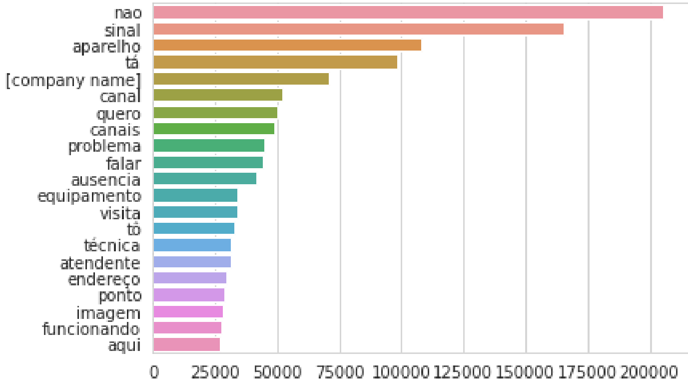


Figure 3.4: Virtual Operator dataset - Most frequent tokens

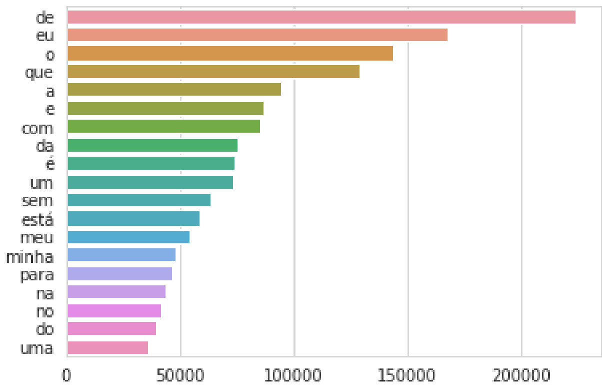


Figure 3.5: Virtual Operator dataset - Most frequent stop words

3.1.3 Mercado Livre - Data Challenge - PT

Mercado Livre - Data Challenge - PT (ML-PT) is a subset of a dataset released by *Mercado Livre* for the *MercadoLibre* Data Challenge 2019[1]. Mercado Livre is an e-commerce website that offers a marketplace to connect buyers and sellers, offering numerous new or used products. Sellers offer their products by providing a short description - limited to 60 characters - and pictures of their selling items. They also need to associate their products to one of the thousands of categories available, and choosing the correct one can be difficult. In this scenario, it is important to have a reliable classification system to help users suggest their products' right category. Although this use case is not precisely intent classification, this dataset shares the same characteristics as the other two. Considering the difficulty of finding public datasets such as this in Brazilian Portuguese, we decided to include it in our investigation.

[1] https://ml-challenge.mercadolibre.com/downloads

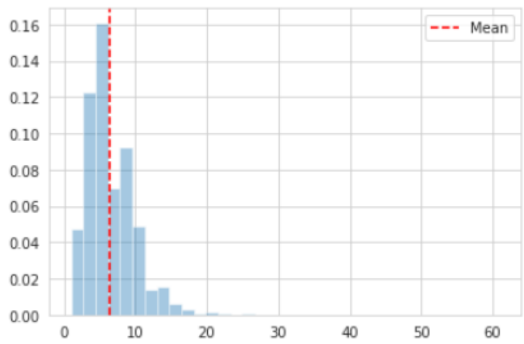


Figure 3.6: NLU-Evaluation dataset - sentence length distribution (in tokens)

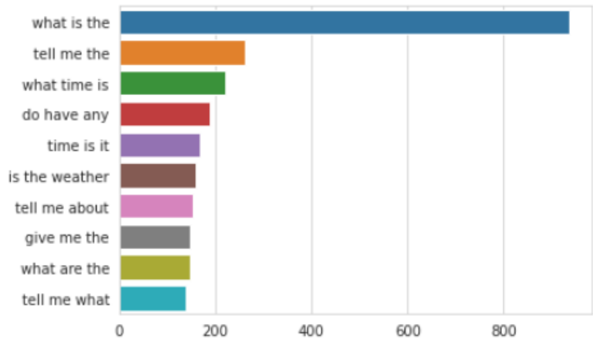


Figure 3.7: NLU-Evaluation dataset - Most frequent 3-grams

The original dataset contains 20 Million product descriptions written by Mercado Livre end-users in Brazilian Portuguese or Spanish. Each sample also has an additional label informing whether the classification is reliable or not. For the scope of this work, we consider only reliable product descriptions written in Portuguese. Also, we discarded labels containing less than ten samples. The filtered dataset contains 692,750 samples divided into 1,048 unbalanced classes with label sets ranging from 10 to 4,711 samples, with a mean sentence length of 8.3 tokens and a standard deviation of 2.2. The sentence-length distribution, shown in figure 3.11, has a different profile from the previously analyzed datasets, which can be explained by the fact that users try to describe their products in as much detail as possible within the 60-character sentence limitation. The sentence-length distribution in characters can be seen in figure 3.12. The sentences are not verified for misspelling or semantic error. Also, users tend to use abbreviations to cope with the 60-character limitation. Some sentences are also truncated by the system, like in the example *tinta acrilica fosco amarelo ouro 3 6l standard suvinil cobr*. These factors contribute to the addition of noise to the dataset.

The distribution of the most frequent tokens (Figure 3.13) lists some special charac-

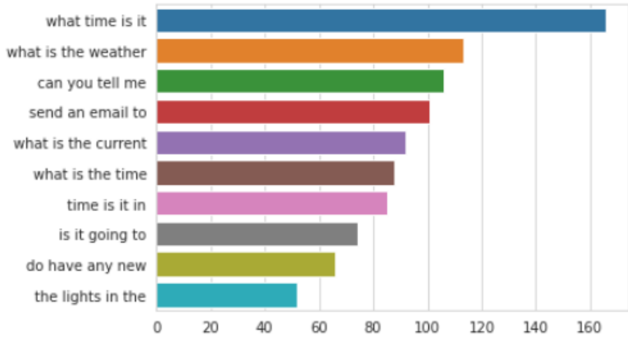


Figure 3.8: NLU-Evaluation dataset - Most frequent 4-grams

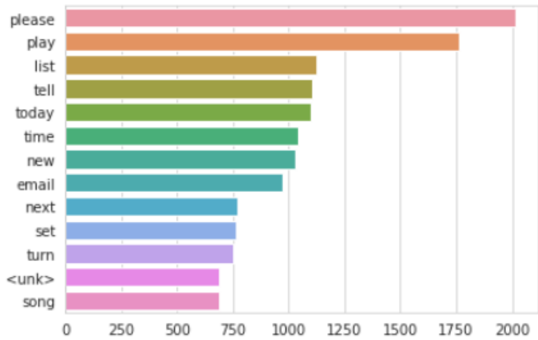


Figure 3.9: NLU-Evaluation dataset - Most frequent tokens

ters, like - and + and numbers, which are generally used in many descriptions as part of a product code or specification, as represented in sentences such as *pilha recarregavel aa com 2 unidades rtu - mo-aa2100c2 - mox* and *papel parede corinthians sc310-01 futebol vinilico lavavel.*

Figures 3.14 and 3.15 show the distribution of the most frequent 3-grams and 4-grams, respectivelly, and figure 3.16, the most frequent stop-words. Table 3.2 summarizes the main features of the three datasets.

Table 3.2: Summary of the investigated datasets main features.

Main Features	**Virtual Operator**	**NLU-Evaluation**	**Mercado Livre**
Language	PT-BR	EN	PT-BR
Sentences	669,929	25,578	692,75
classes	121	64	1,048
Mean sentence size (tokens)	7.6 (s=8.6)	6.5 (s=3.3)	8.3 (s=2.2)

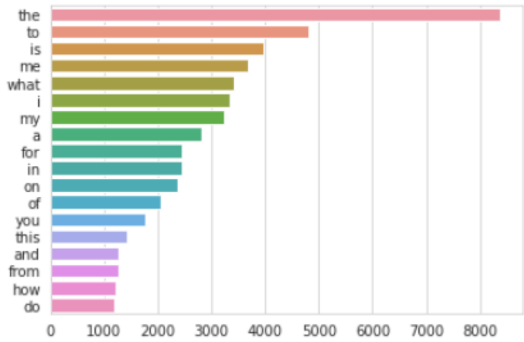


Figure 3.10: NLU-Evaluation dataset - Most frequent stop words

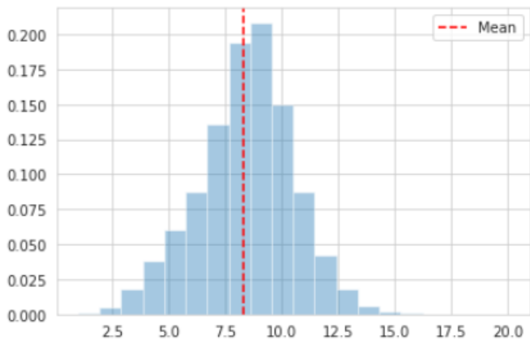


Figure 3.11: ML-PT dataset - sentence length distribution (in tokens)

3.1.4 Training, Validation and Test Sets Creation

To guarantee that the neural network models used in this investigation are evaluated under the same conditions, all three datasets are split into train, validation, and test sets, in *stratified* form - keeping the relative proportion amongst labels of the original dataset. In a first split, 20% of the data are reserved for a test set. Then, the remaining 80% are split into training and validation sets on a 9:1 ratio.

The training and validation sets are used repeatedly during the process of hyper-parameters tuning for each model. Once we are satisfied with the model training hyper-parameters and model performance, a final evaluation of the model using the test set is performed.

3.2 Language Models Investigated

To understand how the different embeddings approaches and neural models affect the ability of a classifier to identify a user's intent accurately, our investigation relied on a set

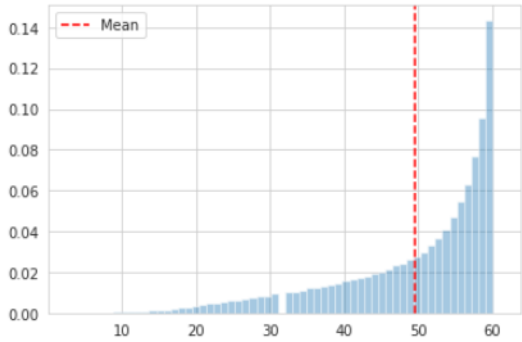


Figure 3.12: ML-PT dataset - sentence length distribution (in characters)

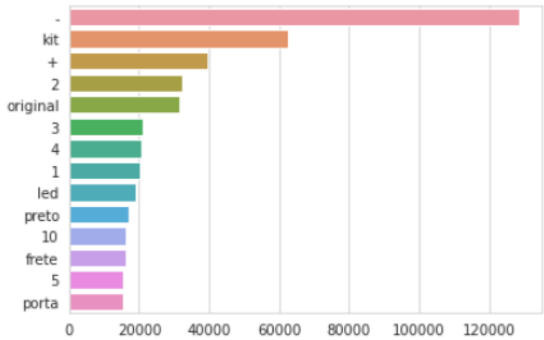


Figure 3.13: ML-PT dataset - Most frequent tokens

of Language Models that were either already pretrained and made available to the public or pretrained for this research. The list of pretrained embeddings included both static and contextualized models. FastText and Word2vec were chosen as static embeddings approaches, whereas ELMo, ULMFit, and BERT models were selected as contextualized models. These models were later fine-tuned on intent classification downstream tasks.

The fastText LMs used in this work were pretrained by the authors on Wikipedia using skip-gram algorithm as described in [7] on both PT-BR and EN, generating vectors with dimension 300.

The Word2vec LMs used in this work were pretrained by us on both target languages using the default CBOW algorithm as per [35]. We pretrained three models using sentences from the training e validation sets of each of the target datasets. We also pretrained two additional LMs using random tweets downloaded from the Internet. Over a period of three weeks, we could download 5,326,164 tweets in PT-BR and 5,084,000 in EN, resulting in corpora of 54,943,878 and 81,840,016 words, respectively.

ELMo pretrained embeddings were available to the public in both target languages.

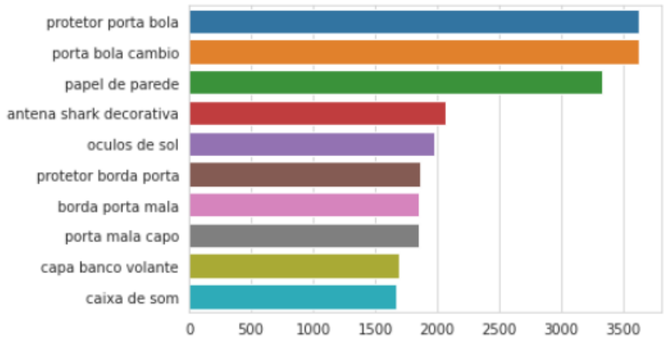


Figure 3.14: ML-PT dataset - Most frequent 3-grams

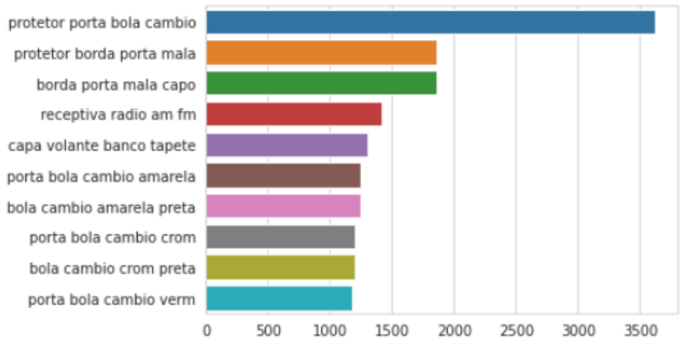


Figure 3.15: ML-PT dataset - Most frequent 4-grams

The EN version was pretrained by the authors of the ELMo paper, and the PT-BR LM was pretrained by researchers from Universidade Federal de Goiás (UFG) [47], using a large corpus from several sources. Both LMs had the same characteristics - LSTM hidden-layer with size 2,048 and output of size 256.

BERT models were also available to the public in the target languages of this research. The authors of the BERT paper provided the EN version, and for the PT-BR LM, BERTimbau was chosen. Since BERTimbau did not offer an uncased version of BERT by the time our experiments were being conducted, we used the cased model on both EN and PT-BR languages. Also, due to computational and time constraints, we opted for the base version. We also used the multilingual version trained by the BERT paper authors to compare the performance results between this model and a language-specific one.

ULMFit models used in this research were pretrained on either Wikipedia or on the same random tweets corpus used to pretrain Word2Vec tweets embeddings. We used the EN Model pretrained on Wikipedia by the authors of the ULMFit paper. Additionally, three more LMs were pretrained by us - one pretrained on the Brazilian Portuguese version of Wikipedia with 100,6 million words, and two more LMs on 5,084,000 randomly collected

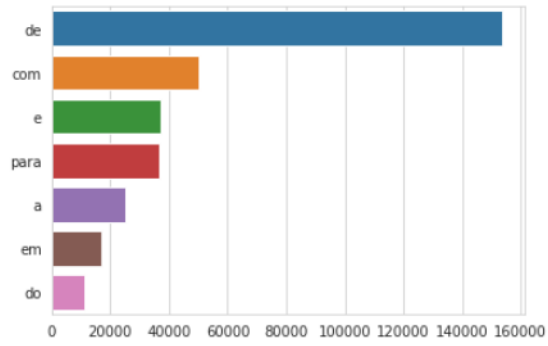


Figure 3.16: ML-PT dataset - Most frequent stop words

tweets in English and 5,326,166 in the Brazilian Portuguese version.

Figure 3.17 shows a graphic summary of the diverse LMs evaluated during this study. The green boxes represent pretrained models that were already available, whereas the orange ones show which models we trained. In total, sixteen Language Models were used in downstream classification tasks, eight of them pretrained by us.

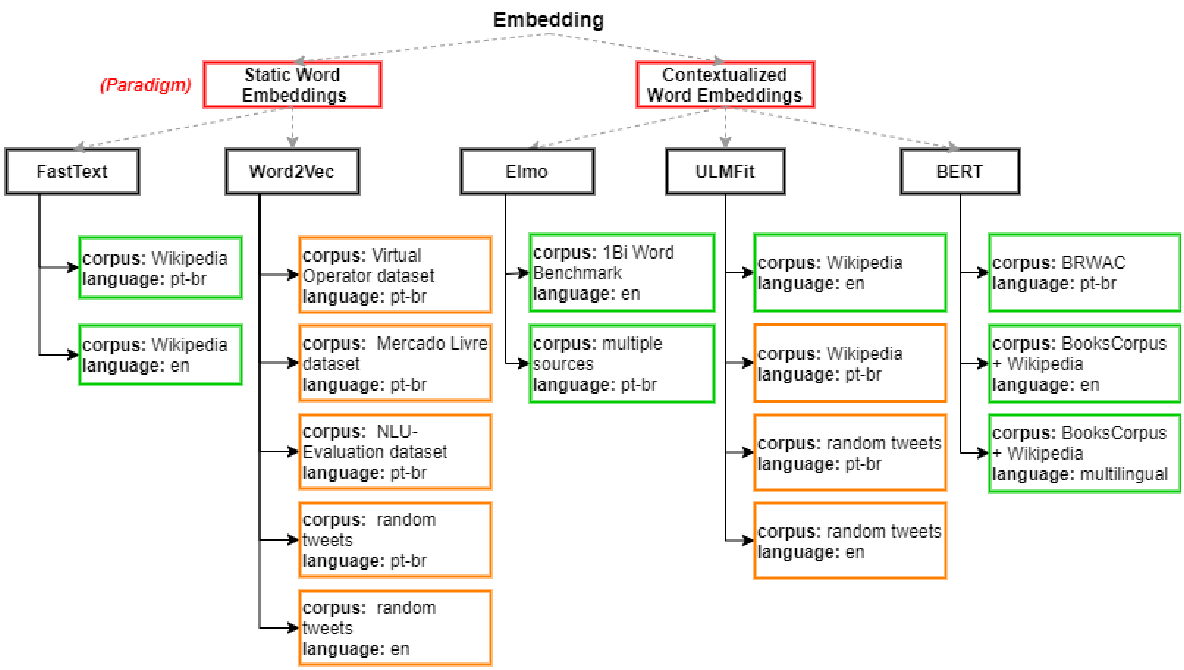


Figure 3.17: pretrained Language Models used in this research - green boxes represent LMs already pretrained and publicly available. Orange boxes show LMs pretrained for this dissertation

3.3 Neural Network Classifiers

This section presents the different vector representation approaches investigated and the neural network classifiers trained on the target datasets to support this investigation. These classifiers architectures used sparse or dense-vector (embeddings) representations. We trained classifiers that used an embedding layer randomly initialized and jointly trained with the remaining neural network layers, loaded from a pretrained model for features extraction, or were the result of fine-tuning an LM on a downstream classification task. All classifiers described in this section were trained on the three datasets described in section 3.1.

Classifier With Sparse-Vector Representation: We built a simple FFNN classifier using one-hot sparse vectors representation, as described in section 2.2.1. Two versions of this classifier were trained - one using the full, unfiltered vocabulary and another with the previous removal of stop-words. The idea here is to understand whether or not stop-words can contain information beneficial to the classification task, depending on the dataset characteristics. We used the list of stop-words provided by the *NLTK*[2] library for both PT-BR and EN.

Classifiers With Pretrained Embeddings For Features Extraction: We evaluated the performance of language models adapted, *i.e.* trained on a downstream classification task using features extraction. In this approach, the weights of the embedding layer are loaded from the LM and are not jointly trained with the classifier (Section 2.2.3). We selected some of the central neural network architectures that are usually applied in NLP tasks - FFNNs, CNNs, LSTMs, and BiLSTMs. We then trained classifiers on embeddings extracted from the Word2Vec, and FastText LMs described in section 3.2. We also trained classifiers using features extracted from an ELMo pretrained embedding layer and fed into an LSTM. Additionally, we trained an FFNN classifier on features extracted from a base BERT model in the target dataset language. We address these classifiers architecture in more details in section 3.4.

Classifiers With Embeddings Jointly Learned from Scratch: We used the same CNNs, LSTMs, and BiLSTMs architectures to train classifiers with an embedding layer that had its weights jointly trained from scratch with the rest of the network.

Classifiers From Fine-Tuned LMs: To evaluate the performance of classifiers trained from LMs fine-tuned on downstream tasks, we selected BERT, and ULMFit LMs

[2]https://www.nltk.org/

introduced in 3.2 and trained classifiers on the target datasets. We used the TAPT approach on ULMFit, as suggested in [22] and introduced in section 2.2.3. Regarding BERT, we fine-tuned LMs on downstream classification tasks with and without an intermediate TAPT step on the target dataset vocabulary, using a language-specific or a multilingual version of the LM.

A total of 17 neural classifiers were trained for each one of the three datasets included in this research. Table 3.3 summarises the combination of diverse vector representations, adaptation modes, LMs, fine-tuning approaches, and neural network architectures we addressed. We provide further details about each classifier in the following sections.

Table 3.3: A summary of the classifiers trained for this research. *N/A* stands for "Not Applicable"

Vector Representation	**Adaptation Mode**	**LM Base Model**	**Pretrained Model**	**TAPT**	**Classifiers Architecture**
Sparse	N/A	N/A	N/A	N/A	FFNN
Dense	N/A	N/A	N/A	N/A	CNN BiLSTM
	Features Extraction	Word2Vec	Target Dataset Corpus	N/A	CNN BiLSTM
			Random Tweets	N/A	CNN BiLSTM
		FastText	Wikipedia on Target Language	N/A	CNN BiLSTM
		ELMo	1 Bi Benchmark Corpus (EN) multiple sources (PT-BR)	N/A	BiLSTM
		BERT	BERT base (language-especific)	N/A	FFNN
	Fine-Tuning	ULMFit	Wikipedia on Target Language	Yes	ULMFit default
			Random Tweets on Target Language	Yes	ULMFit default
		BERT	BERT base (language-especific)	Yes	BERT default
			BERT base (language-especific)	No	BERT default
			BERT base (multilingual)	No	BERT default

3.4 Neural Network Classifier Architectures

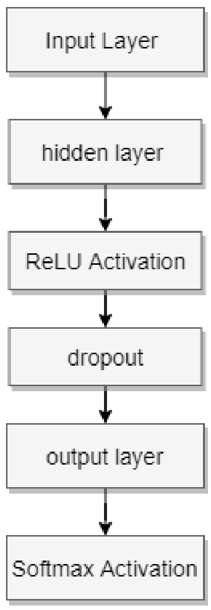
This section describes the architecture models of each of the neural network classifiers used in this research.

Sparse-Vector Classifier: The Sparse-vector classifier model conceptual diagram can be seen in figure 3.4. It consists of a feed-forward neural network with an input layer accepting a one-hot encoded vector with its size corresponding to the vocabulary size of the training and validation sets, combined (see table 3.4). This layer is followed by a hidden layer with 1000 neurons and ReLU activation, a dropout layer, and finally, an output layer with a size equal to the number of labels and Softmax activation. This classifier was implemented using Pytorch [3] Python library.

Table 3.4: Sparse-vector Classifier Architecture

Virtual Operator	**NLU-Evaluation**	**ML-PT**
22417	7370	235867

Table 3.5: one-hot vector sizes for each of the datasets



BiLSTM Classifiers: The BiLSTM classifier archictecture is depicted in figure 3.18. This architecture was used to train classifiers with Word2Vec and FastText embeddings, and also with embeddings jointly learned with the classifier weights. The input layer receives sentence token vectors, which are converted to their dense-vector representations in the embeddings layer. These dense vectors enter the next layer, representing the neural model architecture being tested in the experiment. The next layers follow the same topology as the Sparse-vector classifier - a dropout layer, followed by the output classification layer and softmax activation. These classifiers were implemented using Pytorch Python library.

[3]https://pytorch.org

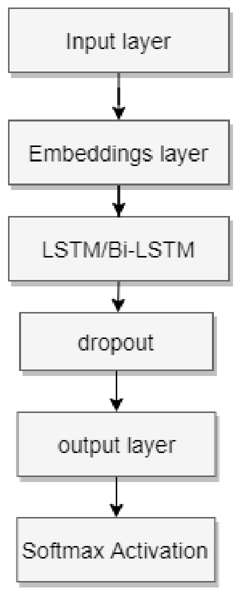


Figure 3.18: LSTM and BiLSTM Classifiers Architecture

CNN Classifiers: The architecture of the CNN classifiers employed in our investigation is quite similar to the LSTM and BiLSTM classifiers, apart from the additional pooling layer after the dropout layer, as per Figure (3.19). The CNN layer contains 256 filters with a kernel size of 4. We also employed this architecture to train classifiers with Word2Vec and FastText embeddings and with embeddings jointly learned with the classifier weights. The CNN classifiers were implemented using Pytorch Python library.

ELMo BiLSTM Classifier: To evaluate classifiers trained on features extracted from ELMo embeddings, we implemented a BiLSTM neural network following the same architecture shown in 3.18, using AllenNLP [4] Python library.

ULMFit Classifier from Fine-tuned LMs: The Fastai[5] library, provided by the ULMFit creators, implements both the Language Model and Classifier described in their work and was used to train both the LM and classifier for these experiments, using the default configuration.

BERT Classifier from Fine-tuned LMs: The BERT classifier, fine-tuned from the LMs previously mentioned, was trained using Hugging Face Transformers [6] library.

[4] https://allennlp.org/
[5] https://www.fast.ai/
[6] https://huggingface.co/transformers/

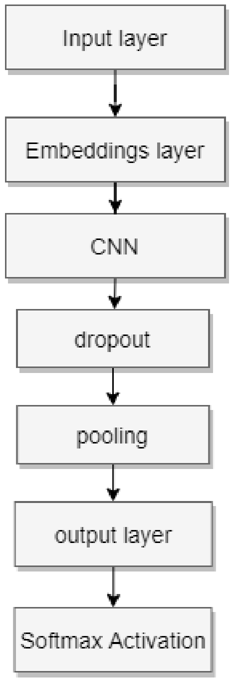


Figure 3.19: CNN Classifiers Architecture

The library implements the classifier as described in [15] - adding a sequence classification layer on top of a BERT LM. To investigate the use of TAPT on BERT models, we also used the same library with the default implementation of the BERT model for pretraining - adding MLM and NSP layers on top of a BERT LM. In our classifiers, only MLM is used. The whole model was fine-tuned during training.

BERT Classifier From LM Extracted Features To evaluate the performance of a classifier trained on features extracted from a BERT Model, we extracted the [CLS] token representation from contextual embeddings of the four last transformers heads. They were concatenated before being fed into an FFNN consisting of input, dropout, and output layers, followed by Softmax activation, as shown in Figure 3.20. The pretrained BERT model was loaded using Hugging Face Transformers Python library, and the FFN was implemented on Pytorch.

The main hyperparameters used during training of the CNN, BiLSTM, FFFN, ELMo, BERT, and BERT for Features Extraction classification models depicted in this section are listed in table 3.6. ULMFit classifiers hyperparameters are listed on table 3.7. All language models and classifiers were trained on a Nvidia Tesla P100 GPU.

Table 3.6: Main hyperparameters used during training of the CNN, BiLSTM, FFNN, ELMo, BERT and BERT for Features Extraction classifiers.

Hyperparameter	**CNN**	**BiLSTM**	**FFNN**	**ELMo**	**BERT**	**BERT for Features Extraction**
Optimizer	Adam	Adam	Adam	Adam	AdamW	AdamW
Scheduler	Reduce learning rate on plateau. Early Stop	Reduce learning rate on plateau. Early Stop.	Reduce learning rate on plateau. Early Stop.	Early Stop	Linear Scheduler with warmup	Linear Scheduler with warmup
max. Epochs	30	30	30	30	100	100
learning rate	1e-3	1e-3	1e-3	3e-2	2e-5	1e-5
gradient clipping	0.25	0.25	-	-	1.0	1.0

Table 3.7: Hyperparameters used on ULMFit Classification models, grouped by Step (target task or Classifier fine-tuning), and freezing status.

Step	**Freezing status**	**Hyper-parameter**	**NLU-Evaluation**		**Virtual Operator**		**Mercado Livre**	
			Tweets LM	**Wiki LM**	**Tweets LM**	**Wiki LM**	**Tweets LM**	**Wiki LM**
Target Task LM Fine-Tuning	**Freeze General Domain LM**	**epochs**	7	5	5	5	5	5
		learning rate	1e-2	3e-2	2e-2	3e-2	5e-2	3e-2
		momentums	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)
	Unfreeze General Domain LM	**epochs**	5	5	5	5	5	5
		learning rate	5e-3	3e-2	8e-3	8e-3	8e-3	8e-3
		momentums	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)
Target Task Classifier Fine-Tuning	**Freeze Target LM**	**epochs**	10	7	10	10	10	10
		learning rate	5e-2	6e-2	5e-2	1.2e-1	1e-1	8e-2
		momentums	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)
	Unfreeze last two layers	**epochs**	2	2	2	2	2	2
		learning rate	slice(1e-2/2.6^4, 1e-2)	slice(7e-3/2.6^4, 7e-3)	slice(1e-1/2.6^4, 1e-1)	slice(1e-1/2.6^4, 1e-1)	slice(7e-2/2.6^4, 7e-2)	slice(8e-2/2.6^4, 8e-2)
		momentums	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)
	Unfreeze next layer	**epochs**	4	4	4	4	4	4
		learning rate	slice(8e-3/2.6^4, 8e-3)	slice(7e-3/2.6^4, 7e-3)	slice(5e-2/2.6^4, 5e-2)	slice(5e-2/2.6^4, 5e-2)	slice(2e-3/2.6^4, 2e-3)	slice(5e-2/2.6^4, 5e-2)
		momentums	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)
	Unfreeze all layers	**epochs**	4	5	4	4	4	4
		learning rate	slice(4e-3/2.6^4, 4e-3)	slice(1e-4/2.6^4, 1e-4)	slice(1e-3/2.6^4, 1e-3)	slice(1e-3/2.6^4, 1e-3)	slice(7e-3/2.6^4, 7e-3)	slice(1e-3/2.6^4, 1e-3)
		momentums	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)	(0.8, 0.7)

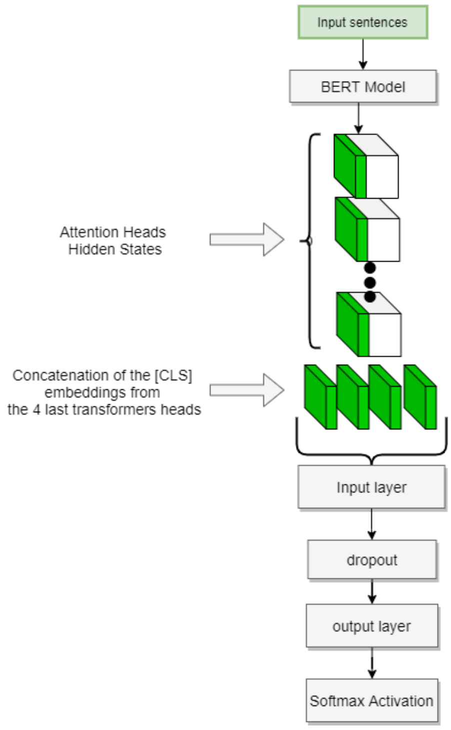


Figure 3.20: How BERT features are extracted and fed into a neural network.

Chapter 4

Results

In this chapter, we present the experimental results obtained with the trained models and strategies described in Chapter 3. The classification results were compared using accuracy, which also corresponds to the micro-averaged F1-score. We used Captum library [25] to visualize which tokens positively or negatively contribute to a sentence classification. Captum implements a series of attribution algorithms to calculate an attribution score for each sentence token. When visualizing token attributions in a sentence, tokens with positive attribution scores are displayed in shades of green, the darker shades representing higher attribution scores. Analogously, tokens with negative attribution scores are surrounded by shades of red.

4.1 General Results

We summarize the classification results for each dataset on Table 4.1, grouped by Vector Representation, Language Model, and Classifier Architecture. Broadly speaking, BERT LMs fine-tuned on a downstream classification task achieved the best overall performance on all three datasets. The TAPT approach had the highest accuracy on both *NLU-Evaluation* and *Virtual Operator*, with 0.790 and 0.966, respectively. There was no improvement when applying TAPT over BERT on the *Mercado Livre* dataset compared to a classifier trained on BERT Base, a result that is further investigated in this chapter. Figure 4.1 shows three scattered plots for these best-performing classifiers, with class support plotted on the x axis and class accuracy on the y axis. We can see that both *Mercado Livre* classifier using BERT Base, and *Virtual Operator* classifier employing TAPT over BERT have similar patterns, with classes with smaller support being associated to lower accuracies. *NLU Evaluation classifer* using TAPT over BERT base, on the other hand, showed a more dispersed pattern. However, still, lower accuracies could be, in general, related to smaller class support.

Table 4.1: Classification accuracies for each of the analyzed datasets, grouped by Vector Representation, Language Model, and Classifier Architecture. FFNN$^+$ represents BOW models trained on sentences without stop-words, whereas a * highlights the best results achieved using sparse or dense vectors features extraction. The best overall values are shown in bold.

Vector Representation		**LM**	**Dataset**			
			Classifier Architecture	**Virtual Operator**	**NLU-Evaluation**	**Mercado Livre PT**
Sparse Vector		N/A	FFNN	0.910	0.768*	0.945*
			FFNN$^+$	0.895	0.735	0.945*
Features Extraction	Embd/Class jointly trained	N/A	BiLSTM	0.942	0.728	0.938
			CNN	0.929	0.743	0.937
	Word2Vec	Random Tweets	BiLSTM	0.922	0.750	0.915
			CNN	0.906	0.732	0.877
		Dataset Vocabulary	BiLSTM	0.935	0.692	0.937
			CNN	0.903	0.658	0.921
	FastText	publicly available in target language	BiLSTM	0.935	0.722	0.927
			CNN	0.928	0.719	0.923
	ELMo	publicly available in target language	BiLSTM	0.916	0.736	0.915
	BERT	BERT Base in target lang	FFNN	0.947*	0.755	0.944
Fine-Tuning	BERT	BERT Base in target lang	BERT Classifier	0.965	0.788	**0.950**
		Multilingual	BERT Classifier	0.943	-	**0.950**
		BERT target lang + TAPT	BERT Classifier	**0.966**	**0.790**	**0.950**
	ULMFit	Wikipedia	ULMFit Classifier	0.965	0.764	0.944
		Random Tweets	ULMFit Classifier	0.965	0.776	0.941

ULMFit had a similar performance to TAPT on BERT, with slightly lower accuracy - 0.965 on *Virtual Operator*, 0.776 on *NLU-Evaluation* and 0.944 on *Mercado Livre*. Pretraining a ULMFit LM on random tweets favored classification on the *NLU-Evaluation* dataset. In contrast, an LM pretrained on Wikipedia showed better performance on the *Mercado Livre* dataset. One hypothesis for this observation is that datasets with smaller sentences could benefit from an LM pretrained on short sentences as tweets, while an LM pretrained on Wikipedia would favor datasets containing longer and more descriptive sentences.

BERT also had the best performance when considering classifiers trained on LMs using a features extraction approach, with an accuracy of 0.947 on *Virtual Operator*, 0.755 on *NLU-Evaluation* and 0.944 on *Mercado Livre*, 1.97%, 4.43% and 0.63%, respectively, below their BERT TAPT approach counterparts.

The strategy of jointly training the embedding layer from scratch with the classifier

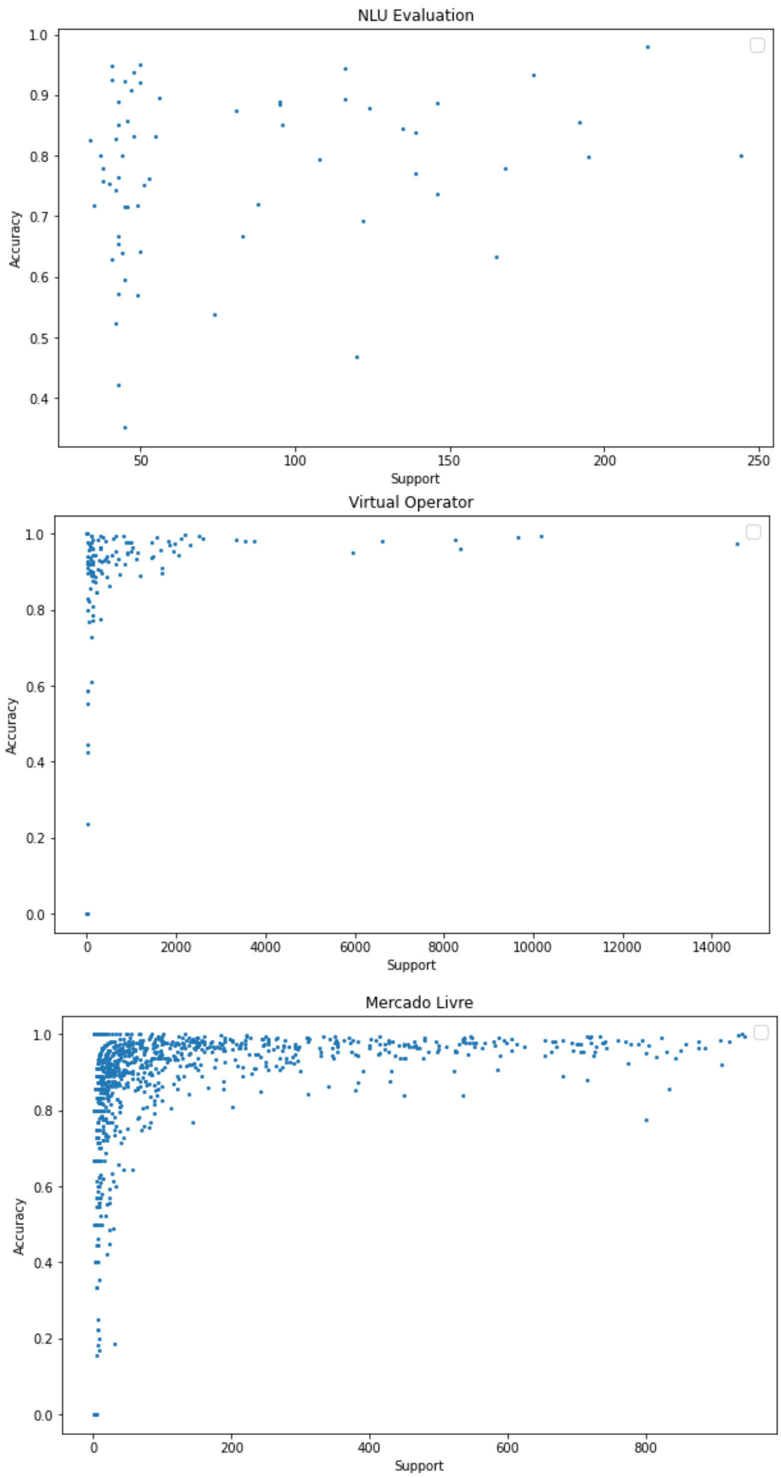


Figure 4.1: Scattered plots showing per-class support versus accuracy for the best overall classifiers on each of the investigated datasets.

model was superior to all other feature extraction approaches, excluding BERT, on two of the datasets. The BiLSTM classifier using this approach achieved 0.942 on *Virtual Operator*, and 0.938 on *Mercado Livre*. *NLU-Evaluation*, on the other hand, presented a superior performance on the BiLSTM classifier trained on the Word2Vec Tweets LM, with an accuracy of 0.750. We believe this result can also be related to the specific characteristics of the *NLU-Evaluation* dataset, containing concise, short, and objective command-like utterances.

Considering the model architectures employed on classifiers using features extracted from embeddings, BiLSTMs had superior overall performance compared to CNNs. Accuracy was on average 2.01% higher when using a BiLSTM, except for the *NLU-Evaluation* classifier with jointly trained embeddings. This classifier had an accuracy of 0.743 when a CNN model was trained against 0.728 on a BiLSTM classifier, representing a 2.06% difference.

Looking at the results of the classifiers that use a sparse-vectors representation on a BOW approach, we can see that FFFN trained on sentences that include stop-words outperform all features extraction approaches and also ULMFit pretrained on Wikipedia on two of the datasets. Accuracy on the *NLU-Evaluation* BOW classifier (0.768) was 1.72% higher than the BERT features extraction classifier (0.755) and 0.52% higher than ULMFit pretrained on Wikipedia (0.764). Performances on *Mercado Livre* BOW (0.945) and BERT features extraction (0.944) classifiers were quite similar, but BOW was 0.11% superior to ULMFit pretrained on Wikipedia. Conversely, BOW performance on *Virtual Operator* was outperformed by almost all classifier approaches, except for CNNs with embeddings pretrained on Word2Vec using random tweets or the dataset's vocabulary.

We also evaluated the role of stop-words in the performance of BOW classifiers by training a different set of models after removing stop-words during the dataset preprocessing. Both *Virtual Operator* and *NLU-Evaluation* classifiers experienced a drop on accuracy after removal of stop-words - -1.65% and -4.30% respectively. *Mercado Livre* classifier was not impacted. These results demonstrate that stop-words may represent features that convey relevant information for classification tasks, depending on the dataset characteristics.

4.2 Comparing Different Feature Representations

This section compares the results obtained on classifications tasks for each dataset from a feature representation strategy. Here, the term *Features Extraction* encompasses all approaches in which features were either extracted directly from sentence tokens or extracted from embeddings that were pretrained on a formal, publicly available vocabulary corpus before being fed into an aggregation layer. FFNNs trained on BOW features and models using features extracted from FastText, ELMo and BERT LMs are included under this group. Next, we grouped all models that used an embedding layer pretrained on a more specific vocabulary, closer to the dataset's domain using Word2Vec, or which embeddings were jointly pretrained with the classifier. We called this group *Embeddings Training*. The last group, *Fine-Tuning*, includes classifiers trained from BERT LMs with our without an intermediate TAPT step or on ULMFit LMs pretrained on either Wikipedia or Random tweets. For each group, we also present the strategy and aggregation layer that achieved the best results.

Table 4.2 shows the results on *NLU-Evaluation*. Using a sparse vector representation to extract BOW features to feed an FFNN was the best Features Extraction approach, achieving an accuracy of 0.768. As an embeddings training approach, the winner was Wor2Vec trained on random tweets feeding a BiLSTM, with an accuracy of 0.750. Finally, the best Fine-tuning strategy and the overall winner was BERT + TAPT using BERT Default classification layer, with an accuracy of 0.790.

Table 4.2: *NLU-Evaluation* classification results, grouped by feature representation approach.

Approach	**Best Strategy**	**Accuracy**	**Aggregator**
Features Extraction	Sparse Vector (BOW)	0.768	FFNN
Embeddings Training	Word2Vec on tweets	0.750	BiLSTM
Fine-Tuning	BERT + TAPT	0.790	BERT Default
Overall	**BERT + TAPT**	**0.790**	**BERT Default**

Results for *Virtual Operator* are shown in table 4.3. In the Static Features Extraction group, BERT sentence features extraction using an FFNN aggregation layer had the best performance, with an accuracy of 0.947. In the Embeddings Training group, jointly trained embeddings on a BiLSTM was the winning strategy, achieving an accuracy of 0.942. Lastly, in the Fine-Tuning group, BERT + TAPT LM trained on BERT's default classification head obtained the best accuracy of 0.966. This was also the best overall strategy for this dataset.

Table 4.4 presents results from *Mercado Livre* classifiers, showing that a sparse vector representation using BOW features to fed an FFNN was the best approach amongst all Features Extraction approaches, reaching an accuracy of 0.945. Considering Embeddings Training, an embedding layer jointly trained with a BiLSTM classification layer achieved the highest accuracy, of 0.938. Finally, in the Fine-Tuning group, BERT Base in the target language, Bert Multilingual, and BERT + TAPT had the same performance, with

Table 4.3: *Virtual Operator* classification results, grouped by feature representation approach.

Approach	Best Strategy	Accuracy	Aggregator
Features Extraction	BERT	0.947	FFNN
Embeddings Training	Jointly Trained	0.942	BiLSTM
Fine-Tuning	BERT + TAPT	0.966	BERT Default
Overall	**BERT + TAPT**	**0.966**	**BERT Default**

an accuracy of 0.950. Provided that BERT Base in the target language required fewer steps than the other approaches, we considered this to be an important consideration which led us to select it as the winning strategy on this dataset.

Table 4.4: *Mercado Livre* classification results, grouped by feature representation approach.

Approach	Best Strategy	Accuracy	Aggregator
Features Extraction	Sparse Vector (BOW)	0.945	FFNN
Embeddings Training	Jointly Trained	0.938	BiLSTM
Fine-Tuning	Bert Base in target Lang	0.950	BERT Default
Overall	**Bert Base in target Lang**	**0.950**	**BERT Default**

In the next section, we analyze the impact of stop-words in closer detail.

4.3 The Role of Stop-words on BOW

In order to understand the reason behind the loss of classification performance associated with the removal of stop-words on *NLU-Evaluation*, we selected the three classes that had the most significant impact on this dataset. Table 4.5 lists the most impacted classes on *NLU-Evaluation*, their respective accuracies on classifiers trained with and without stop-words, and the associated reduction on accuracy, with values ranging from -17.78% to -27.87%. The complete per-class performance comparison for this dataset is available on section B.1.

Table 4.5: List of classes on *NLU-Evaluation* that had the most significant impact on accuracy after removal of stop-words

ID	Class Name	Accuracy		Reduction
		With stop-words	Without stop-words	
10	QA_open_query	0.409	0.295	-27.87%
32	general_mistake	0.500	0.378	-24.40%
0	calendar_notification	0.388	0.319	-17.78%

We selected some example sentences from each of these classes and listed them in Table 4.6. For each example, we present the original sentence with stop-words and also the sentence after stop-words removal. We also plotted the average feature importances for each predicted class to better understand the role of stop-words on the results achieved. The graph in figure 4.2 plots the feature importance of each token that impacted, positively or not, in the classification on class *QA_ Open_ query*. The top chart shows feature importances when stop-words are included, and the bottom chart when they are removed. Considering sentence (1), we can see that the stop-words *you, me* and *my* positively contribute to the correct classification. Also, both *my* and *you* have negative importance on class *datetime_ query*, shown on figure 4.3). However, when stop-words are not considered, the influence of tokens *date* and *time*, both with high importance on class *datetime_ query*, becomes relevant enough to favour classification under this label.

Table 4.6: Examples of sentences extracted from *NLU-Evaluation* which were incorrectly classified when stop-words were removed. We present the sentence with its stop-words and also without them.

ID	Sentence	Predicted Class	Correct
1	could you please tell me which time will be the best time for me to date my lover	QA_open_query	Yes
	could please tell time best time date lover	datetime_query	No
2	can you tell me how to measure my shoe size	QA_open_query	Yes
	tell measure shoe size	QA_factoid	No
3	that is not correct	general_mistake	Yes
	correct	general_feedback	No
4	that was not what i was looking for try it again	general_mistake	Yes
	looking try	general_feedback	No
5	tell me when i have a work meeting coming up	calendar_notification	Yes
	tell work meeting coming	calendar_query_event	No
6	can you remind me tomorrow morning about my dinner plans for the weekend	calendar_notification	Yes
	remind tomorrow morning dinner plans weekend	calendar_set_event	No

Regarding sentence (2), classification under class *QA_ Open_ query* is influenced by tokens *please*, *tell*, *me*, *how* and *my*. Without stop-words, classification under this class is influenced only by token *tell* and therefore, the sentence is classified under the label *QA_factoid*, despite the lack of important features in the sentence favouring classification on this class (Figure 4.4). Sentence (3) is an interesting example of meaning inversion due to removal of stop-words. Although token *this* has a small, but negative impor-

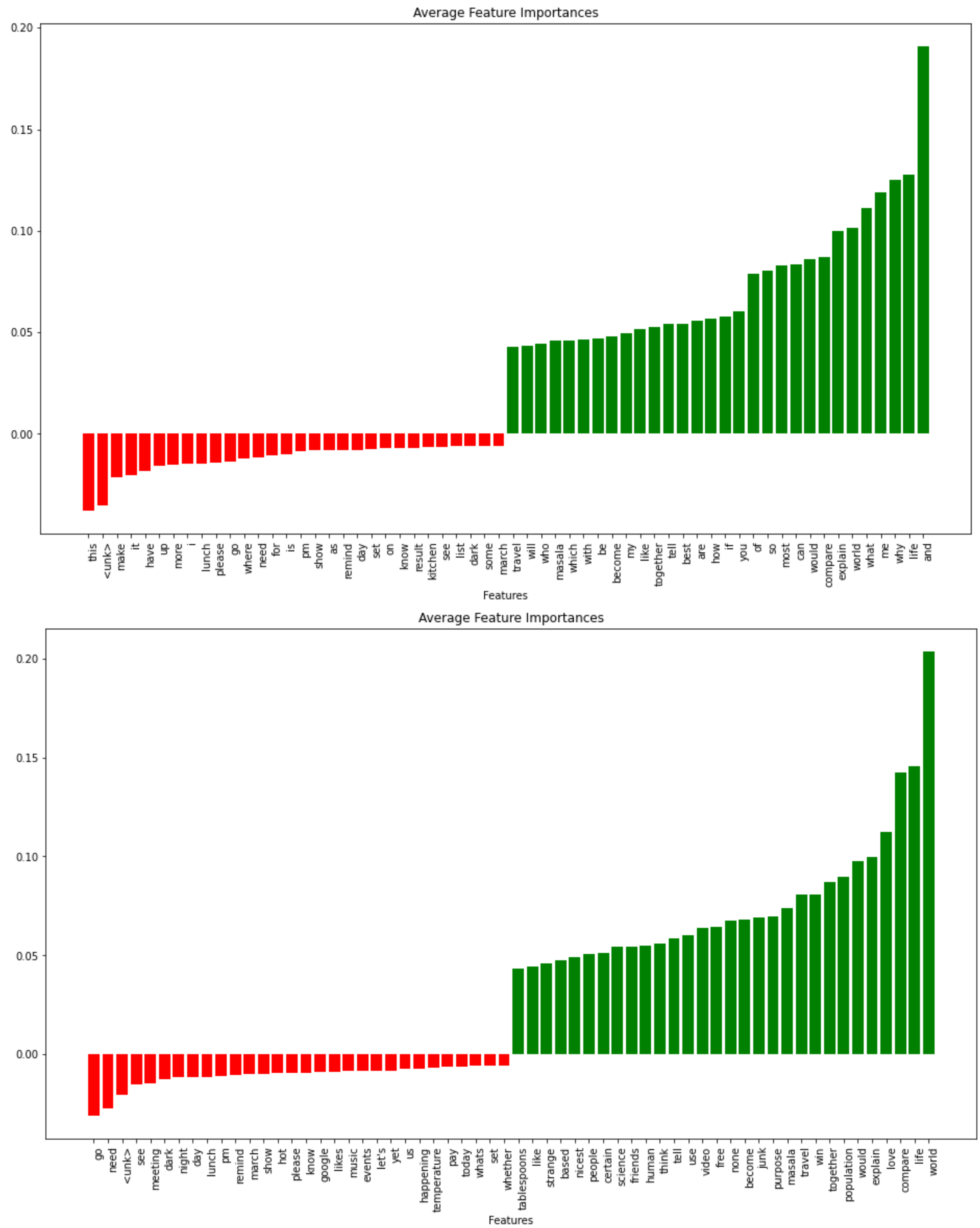


Figure 4.2: Average feature importances on NLU-Evaluation class *QA_ open_ query* when stop-words are considered (top) and removed from the dataset (bottom)

tance, *not* is the second most important token for the *general_ mistake* class, according to figure 4.5. Besides, *correct* has similar average importance on both *general_ feedback* and *general_ mistake*. Figure 4.6 presents the average feature importances for class *gen-*

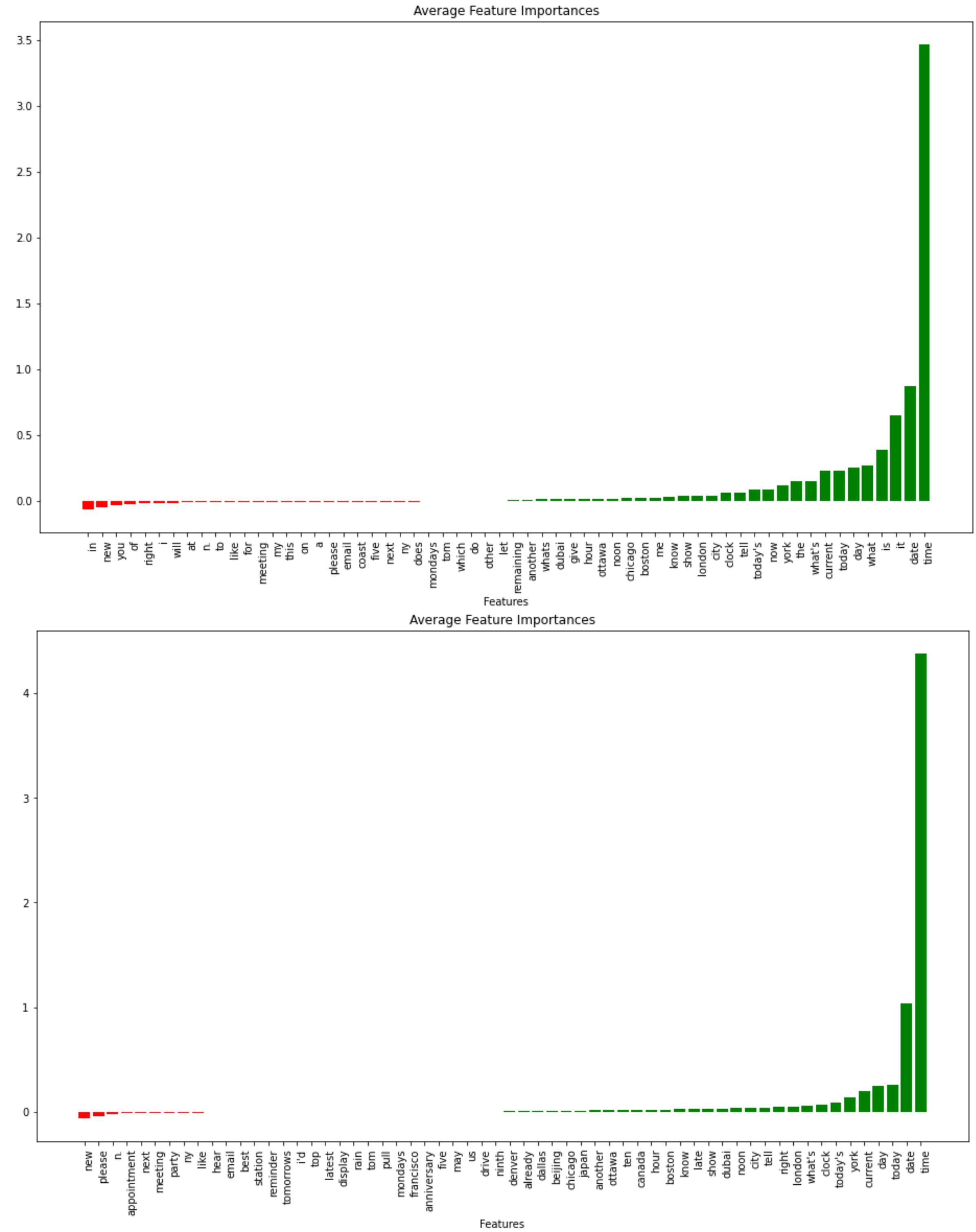


Figure 4.3: Average feature importances on NLU-Evaluation class *datetime_ query* when stop-words are considered (top) and removed from the dataset (bottom)

eral_ feedback.

The removal of stop-words from sentence (4) implies in loss of meaning. From a BOW

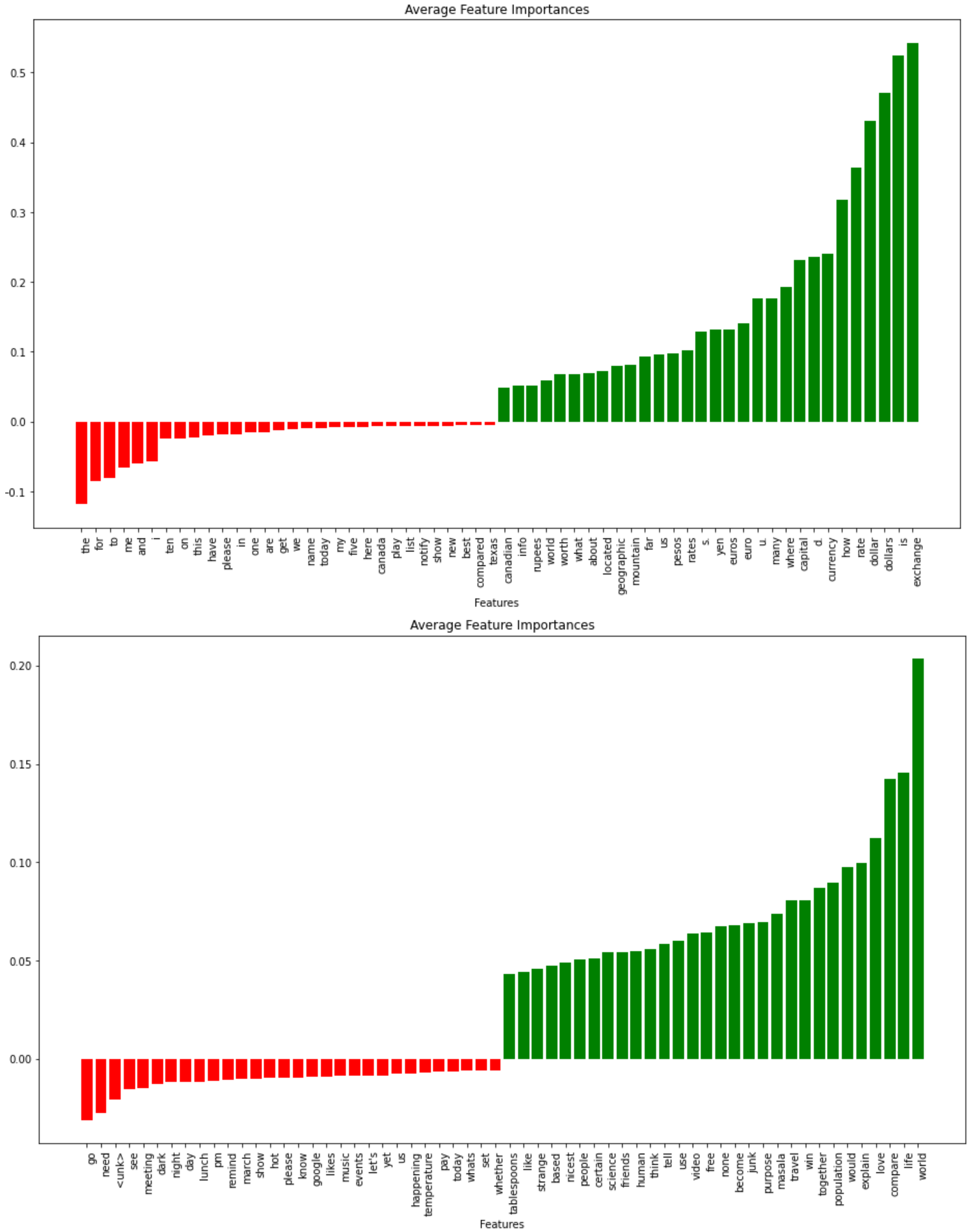


Figure 4.4: Average feature importances on NLU-Evaluation class *QA_factoid* when stop-words are considered (top) and removed from the dataset (bottom)

perspective, classification under class *general_mistake* is highly dependent on tokens *that*, *was*, *not* and *again*. Moreover, tokens *looking* and *try*, figuring as important tokens for

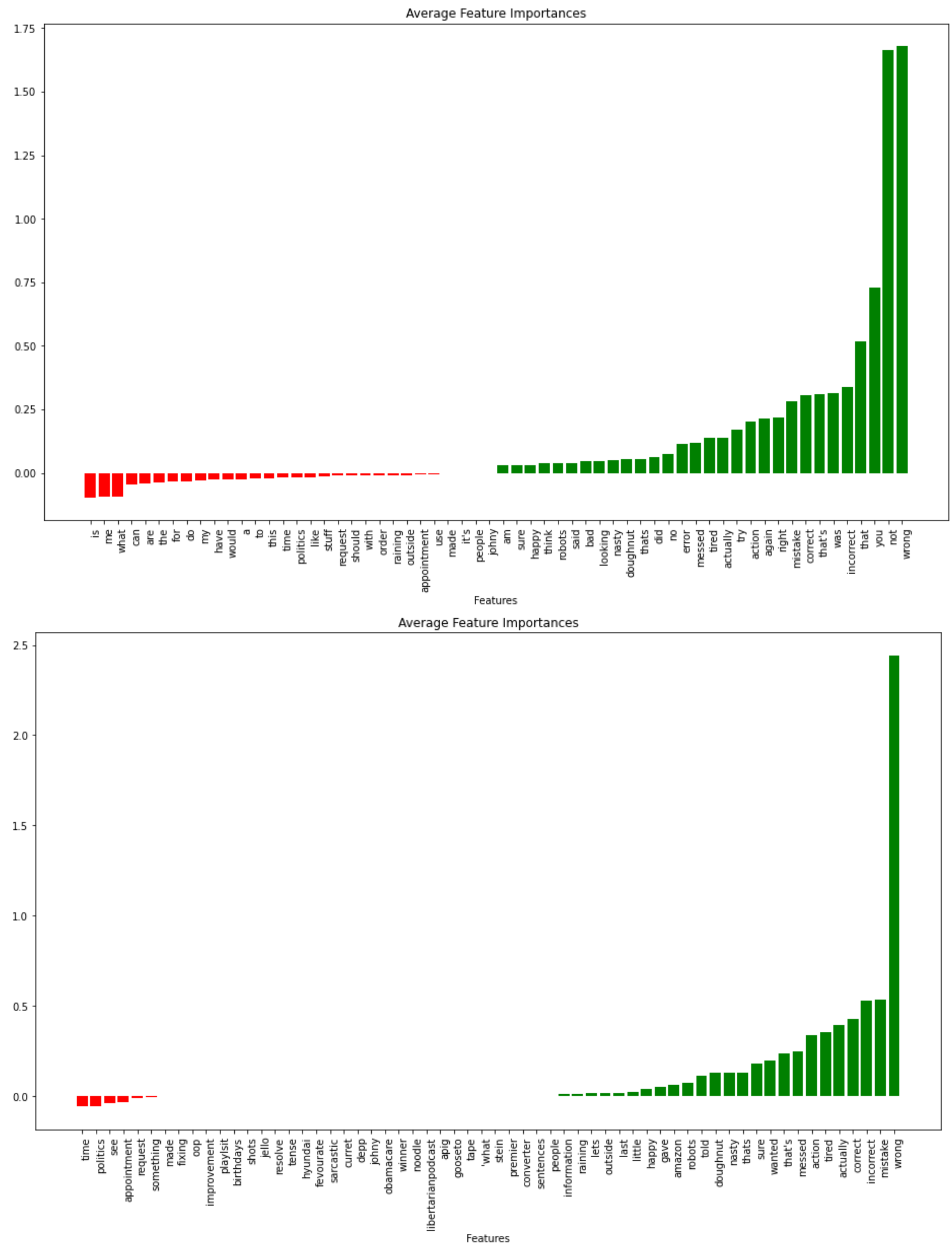


Figure 4.5: Average feature importances on NLU-Evaluation class *general_ mistake* when stop-words are considered (top) and removed from the dataset (bottom)

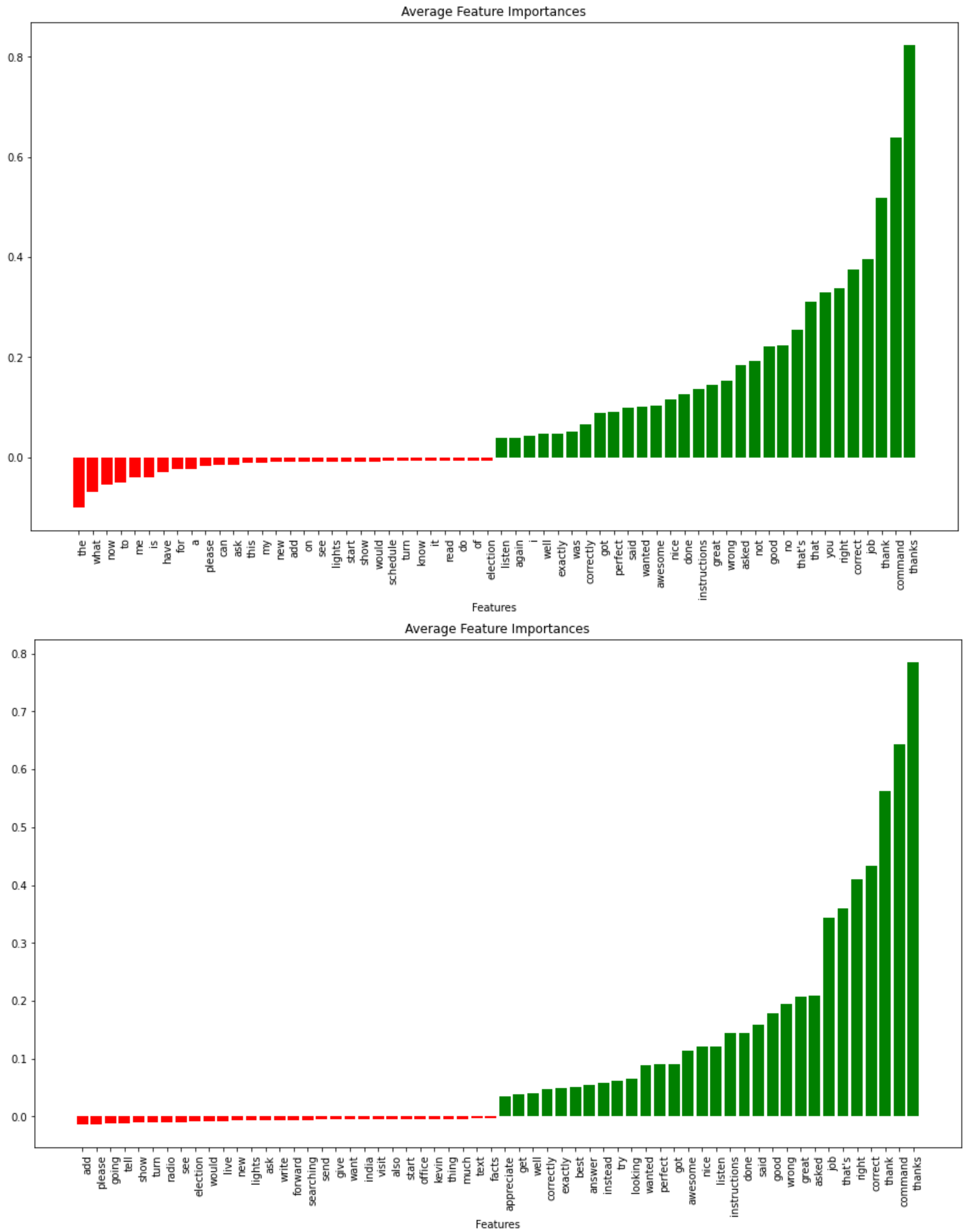


Figure 4.6: Average feature importances on NLU-Evaluation class *general_feedback* when stop-words are considered (top) and removed from the dataset (bottom)

class *general_mistake* when stop-words are present, do not appear as important features for this same class after removal of stop-words, but positively impact classification on class *general_feedback*. Sentences (5) and (6) have similar behavior regarding the presence or absence of stop-words. Their removal leads to loss of relevant information, which would contribute to the classification under the correct class.

For the analysis of the *Virtual Operator* dataset, we also focused on the most negatively impacted classes, as per table 4.7. The complete per-class performance comparison for this dataset can be seen on section B.2. However, we identified four of these classes with relevant mislabelling issues that would impact this investigation during this analysis. For example, class *Qualificado.Cancelar [carrier name]*, with a support of 50[1], had 37 incorrectly labeled sentences that belonged, in fact, to other classes. Class *Sintomas.Qualificado.Travado exceto 200* had 16 mislabelled sentences, from a total of 17. Regarding class *Sintomas.Genérico.Código sim*, sentences in fact belonged to classes *Sintomas.Genérico.Texto ou código na tela* or *Sintomas.Qualificado.Código 56* but were mistakenly labeled in this class. Class *Sintomas.Qualificado.Cliente está longe* also had examples that, in fact, belonged to other classes. Our analysis considered classes *Qualificado.Equipamento travado*, *Qualificado.Áudio atrasado* and *Genérico.Equipamento quebrado G*, which were considered to be more accurately labeled.

Table 4.7: List of classes on *Virtual Operator* that had the most significant impact on accuracy after removal of stop-words

ID	**Class Name**	**Accuracy**		**%**
		With stop-words	**Without Stop-words**	
90	Qualificado.Cancelar [carrier]	0,356	0,098	-72,5%
112	Qualificado.Travado exceto 200	0,250	0,091	-63,6%
64	Genérico.Código sim	0,449	0,213	-52,6%
108	Qualificado.Cliente está longe	0,348	0,244	-29,9%
76	Qualificado.Equipamento travado	0,494	0,353	-28,5%
117	Qualificado.Áudio atrasado	0,667	0,500	-25,0%
11	Genérico.Equipamento quebrado G	0,859	0,734	-14,6%

The classification of sentences under class *Qualificado.Equipamento travado* was impacted by the removal of token *só*, a stop-word. In the distribution of feature importances for this class, displayed in figure 4.7, we can see the high relevance of tokens *só*, *pegando* and *globo*. Removing token *só* led to a higher number sentences being missclassified under class *Genérico.Canal comum não pega (G)*, as shown in the example sentences on table 4.8. An analysis of the most important features for class *Genérico.Canal comum não pega (G)* (Figure 4.8) reveals that the same tokens *pegando* and *globo* were amongst the most important ones. Similarly, class *Genérico.Equipamento quebrado G* had tokens *aparelho*, *com*, *defeito*, *problema* and *quebrado* figuring as the most important features,

[1]Support stands for the number of samples in a specific class of a dataset.

with *com*, a stop-word, being the second one, as shown in figure 4.9. Its removal implied in less differentiation ability from classes with similarly relevant tokens, such as *Genérico.operadora não funciona* and *Genérico.Problema com equipamento* (Figures 4.10 and 4.11, respectively).

Table 4.8: Examples of sentences extracted from *Virtual Operator* which were incorrectly classified when stop-words were removed. We present the sentence with its stop-words and also without them.

ID	Sentence	Predicted Class	Correct
1	o aparelho de tv só tá funcionando a globo nao pega mais nenhum canal	Qualificado.Equipamento travado	Yes
	aparelho tv tá funcionando globo nao pega nenhum canal	Genérico.Canal comum não pega (G)	No
2	a minha tv só tá funcionando a globo	Qualificado.Equipamento travado	Yes
	tv tá funcionando globo	Genérico.Canal comum não pega (G)	No
3	quero ver os canais net nao tá pegando só tá pegando a globo	Qualificado.Equipamento travado	Yes
	quero ver canais net nao tá pegando globo	Genérico.Canal comum não pega (G)	No
4	meu aparelho está com entrada hdmi estragada	Genérico.Equipamento quebrado G	Yes
	aparelho entrada hdmi estragada	Genérico.Problema com equipamento	No
5	é o aparelho está com defeito aparelho slim hd com defeito	Genérico.Equipamento quebrado G	Yes
	aparelho defeito aparelho slim hd defeito	Genérico.Problema com equipamento	No
6	o motivo da ligaçao aparelho que está com hdmi quebrado	Genérico.Equipamento quebrado G	Yes
	motivo ligação aparelho hdmi quebrado	Genérico.operadora não funciona	No
7	motivo da ligaçao porque eu coloco no canal pode ser aqui bebe pode ser qualquer um outro canal ele fica a uns 30 segundos com audio normal de voz e depois some o áudio só fica na imagem e som no áudio aí	Qualificado.Áudio atrasado	Yes
	motivo ligação porque coloco canal pode ser aqui bebe pode ser qualquer outro canal fica uns 30 segundos audio normal voz some áudio fica imagem som áudio ai	Qualificado.Apenas imagem sem áudio	No

Class *Qualificado.Áudio atrasado* had only six examples in the test set, and four of these sentences predicted labels matched their respective set true labels when stop-words were considered. However, one of these sentences (sentence (7) on table 4.8) had a different predicted label when stop-words were discarded. Looking closer to the sentence, we identified that this sentence was incorrectly labeled in the test set and belonged to

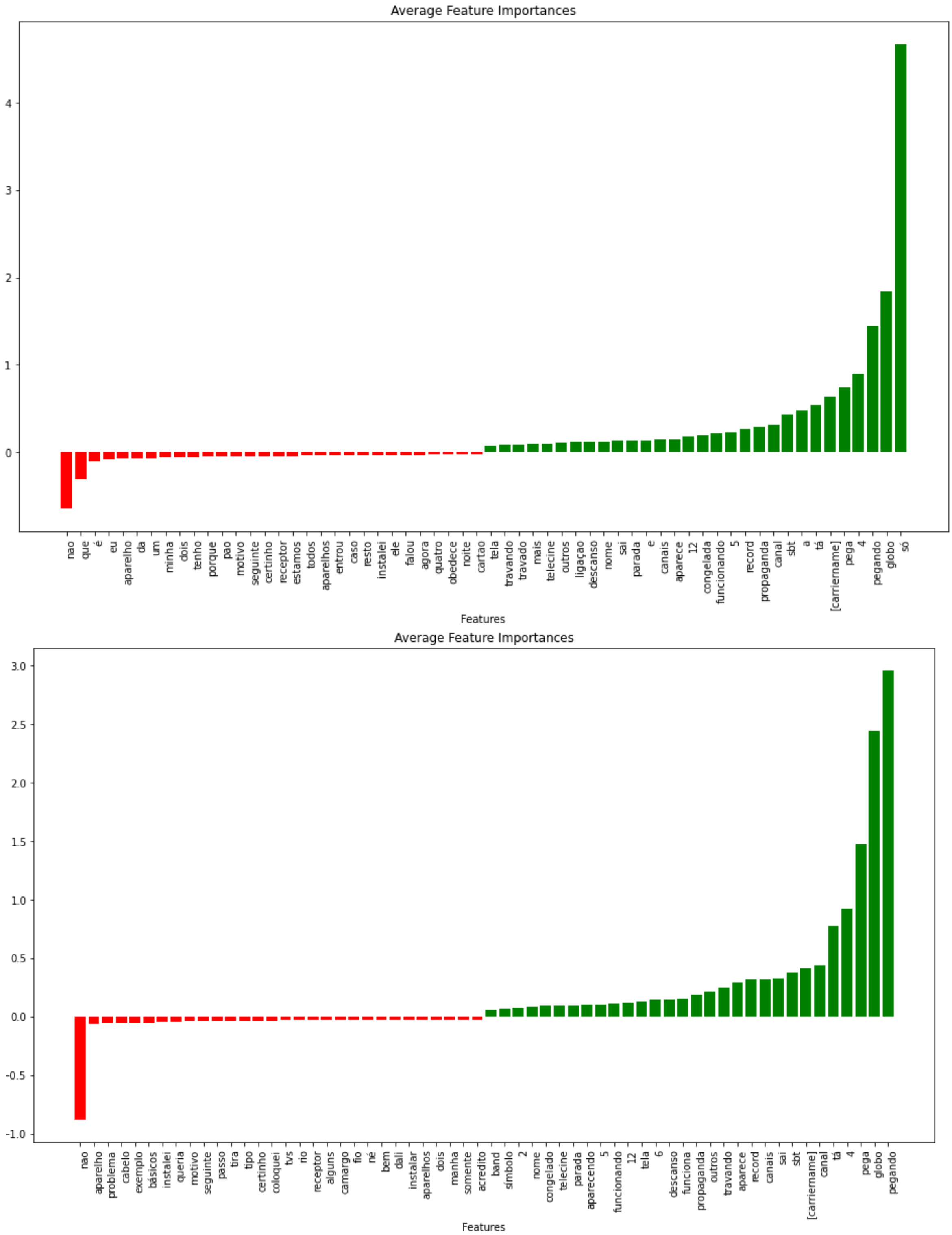


Figure 4.7: Average feature importances on *Virtual Operator* class *Qualificado.Equipamento travado* when stop-words are considered (top) and removed from the dataset (bottom).

class *Qualificado.Apenas imagem sem áudio*, which was correctly predicted by the classifier trained on sentences with no stop-words. The small support of this class allowed us to

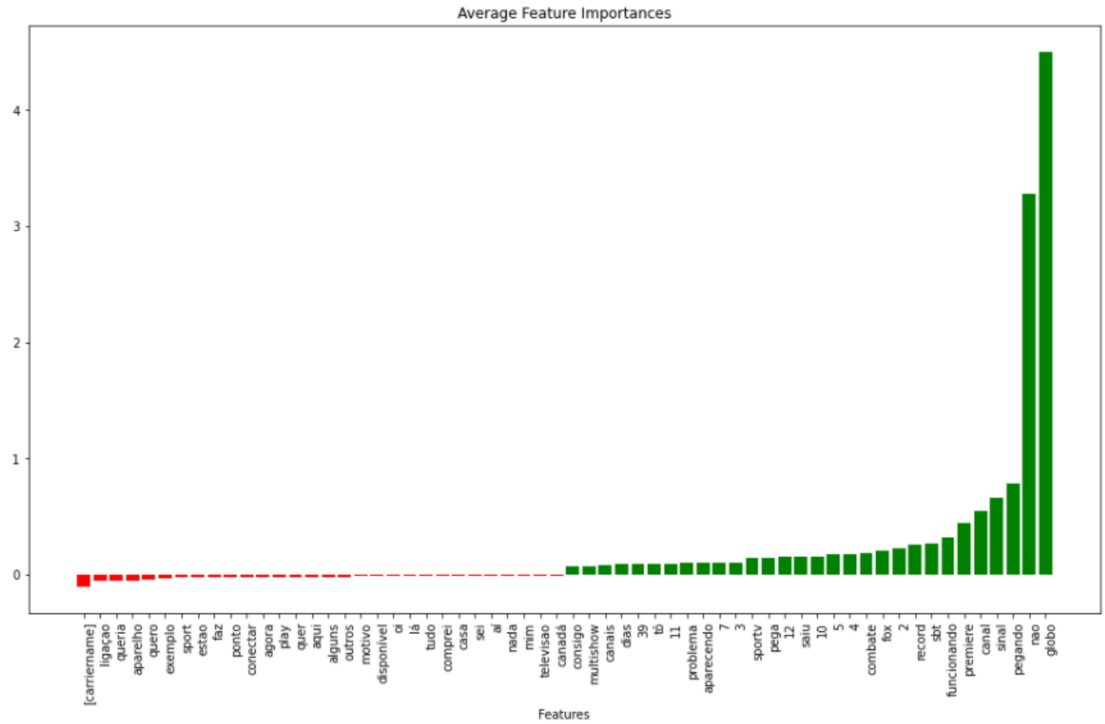


Figure 4.8: Average feature importances on *Virtual Operator* class *Genérico.Canal comum não pega (G)* when stop-words are removed from the dataset

observe that the removal of stop-words also affected the set of features that influenced classification under a specific label. For instance, the distribution of important features for class *Qualificado.Áudio atrasado* contained 38 tokens when stop-words were not removed, but only five tokens after their removal, as shown in Figure 4.12. We conclude that removing stop-words may lead to loss of information which is not only carried by them but also to other tokens which may have some relationship with these stop-words.

4.4 Comparing BERT Base and BERT Base + TAPT Results

We followed the same approach presented in the previous section to investigate the impact of applying TAPT over a BERT Base LM. We compared the performances of BERT fine-tuned on a downstream classification task with BERT fine-tuned on a downstream task with an intermediate TAPT step. We conducted this evaluation for all three datasets, selecting the three most positively and negatively impacted classes, considering their accuracies. We collected some examples from each of these classes and applied Captum to identify features that contributed to the predicted class output on each of these examples.

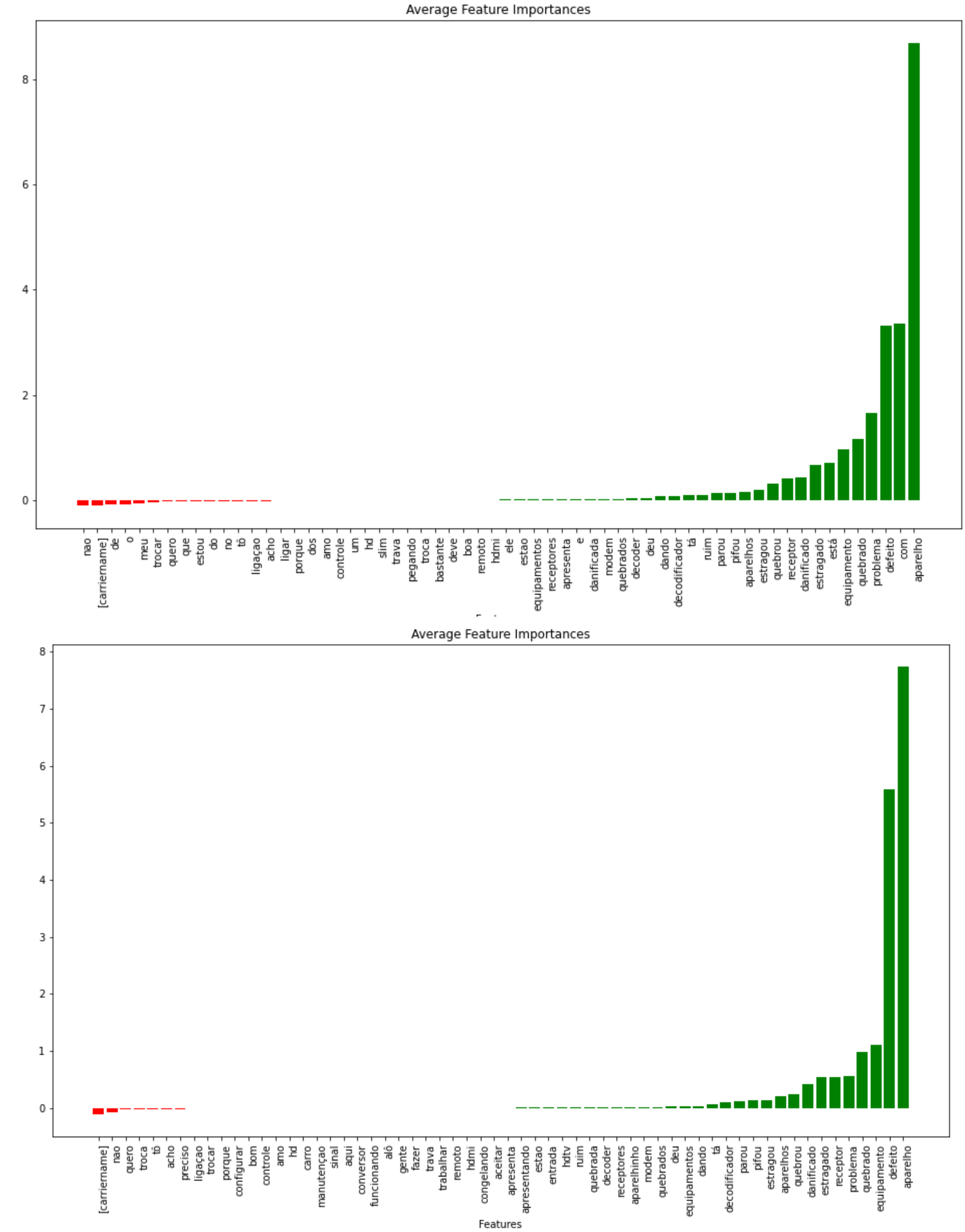


Figure 4.9: Average feature importances on *Virtual Operator* class *Genérico.Equipamento quebrado G* when stop-words are considered (top) and removed from the dataset (bottom)

Table 4.9 lists the classes selected for the analysis of *NLU-Evaluation* dataset BERT Base and BERT Base + TAPT intent classifiers. We used the same criteria in the previous

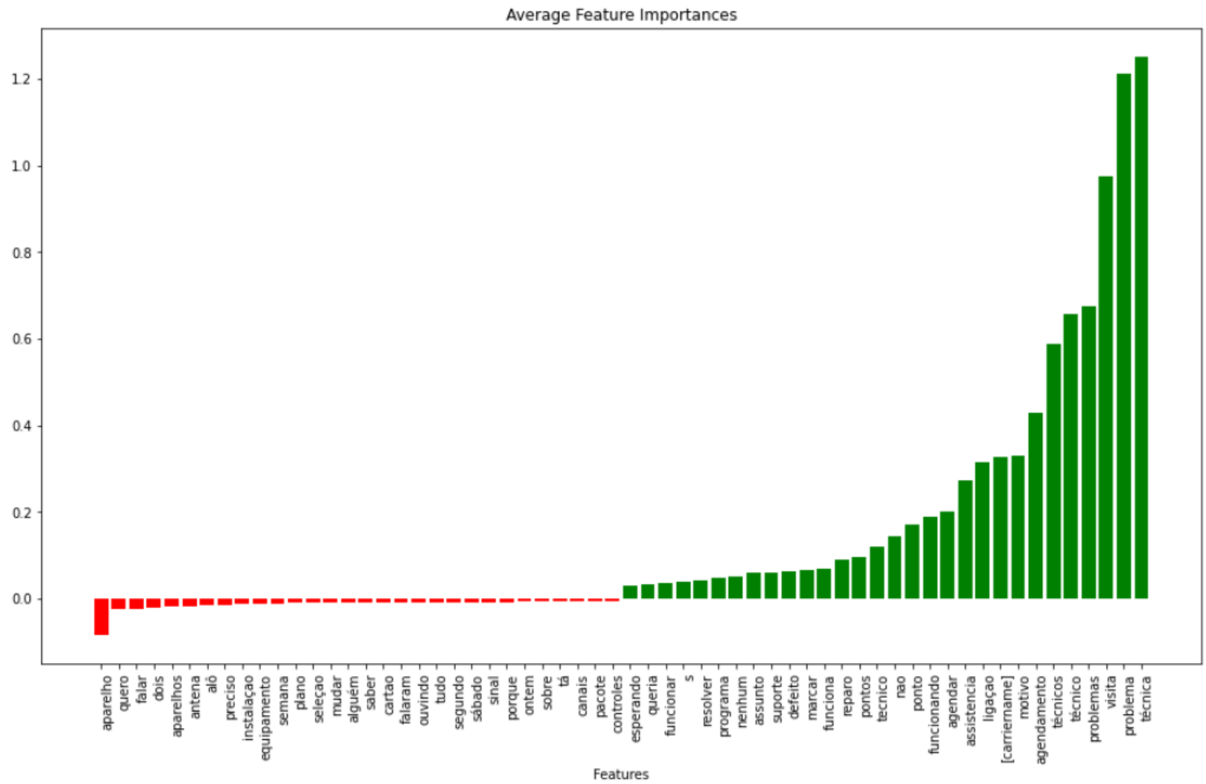


Figure 4.10: Average feature importances on *Virtual Operator* class *Genérico.operadora não funciona* when stop-words are removed from the dataset

analysis - selecting the three classes that most benefited from BERT + TAPT and the three that had the highest reduction in their accuracies. Figure 4.13 shows some examples taken from the three classes that had the highest positive impact. For each example, the true and predicted labels, the attribution score, and the features that most influenced the classifier prediction are displayed. The sentence attribution score is computed as the sum of the individual attribute scores from all sentence tokens. Each sentence is listed twice - the first occurrence shows the result from the BERT Base classifier and the incorrectly predicted label, and the second one, from the BERT + TAPT classifier, showing the predicted output matching its respective true label.

We can see that, for classes 34 (*alarm_ query*) and 1 (*transport_ directions*), sentence attribution scores are higher on the TAPT classifier when compared to the BERT Base one, caused either by enforcement on tokens that positively contribute to the classification or by the reduction on the negative tokens contributions. For instance, in the sentence *do i have any alarms set for tomorrow*, there is a reduction in the negative contribution of token *tomorrow*, but also, token *i* becomes a positive contribution, enforcing the role of n-gram *do i have any alarm* in the classification output. On the other hand, class

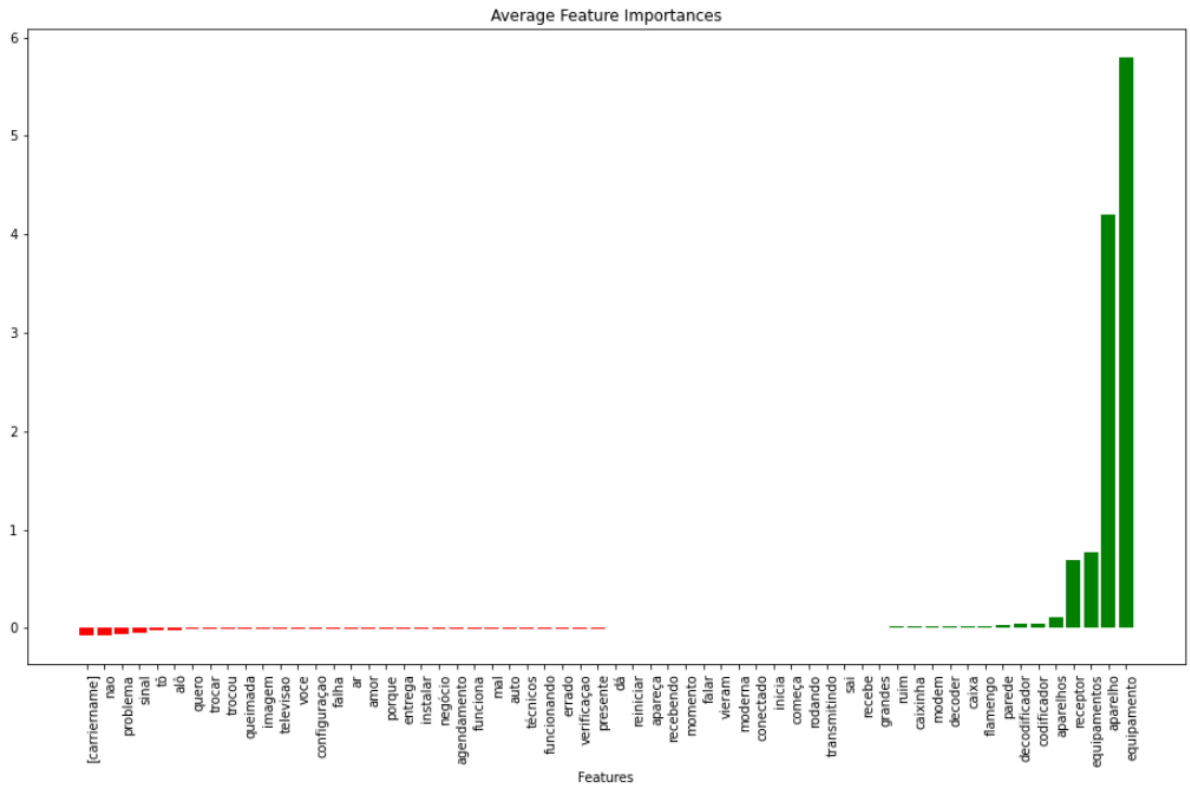
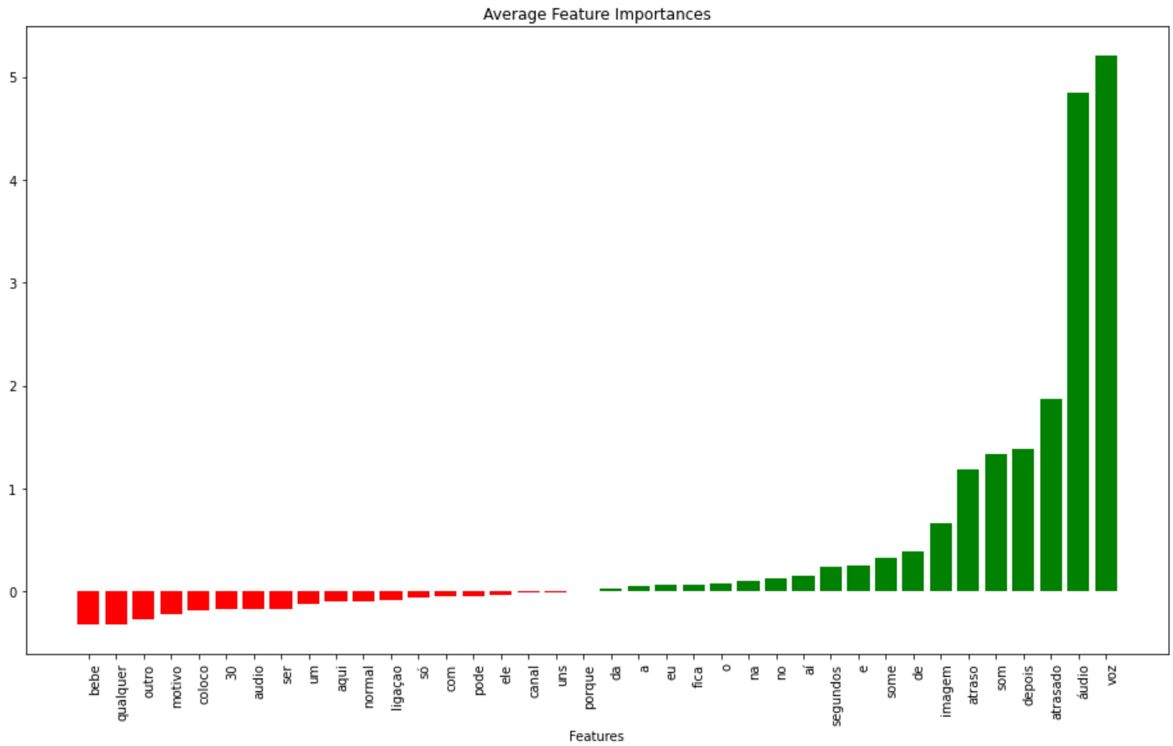


Figure 4.11: Average feature importances on *Virtual Operator* class *Genérico.Problema com equipamento* when stop-words are removed from the dataset

Table 4.9: *NLU-Evaluation* classes selected for investigation. Classes 34, 1 and 18 had the highest improvement on their accuracy when TAPT was used. Classes 32, 13 and 0, conversely, had their accuracy degraded.

ID	Class Name	Accuracy		%
		BERT BASE	BERT BASE + TAPT	
34	alarm_query	0.727	0.800	10.04%
1	transport_directions	0.576	0.629	9.20%
18	recommendation_locations	0.713	0.764	7.15%
32	general_mistake	0.608	0.569	-6.41%
13	music_question	0.711	0.639	-10.13%
0	calendar_notification	0.415	0.352	-15.18%

18 (*recommendation_locations*), despite the improvement in the predictions when TAPT was used, experienced a reduction in its sentences attribution scores. We also computed token importances on those sentences classified by the TAPT classifier on class 18, in respect to class 38 (*takeaway_order*), which was the class that was wrongly predicted by the BERT Base classifier but had a higher mean attribution score. These attribution scores, shown in figure 4.14, were still lower than those computed with respect to class 18, the true label class. Pretraining the LM on the dataset vocabulary led to an overall



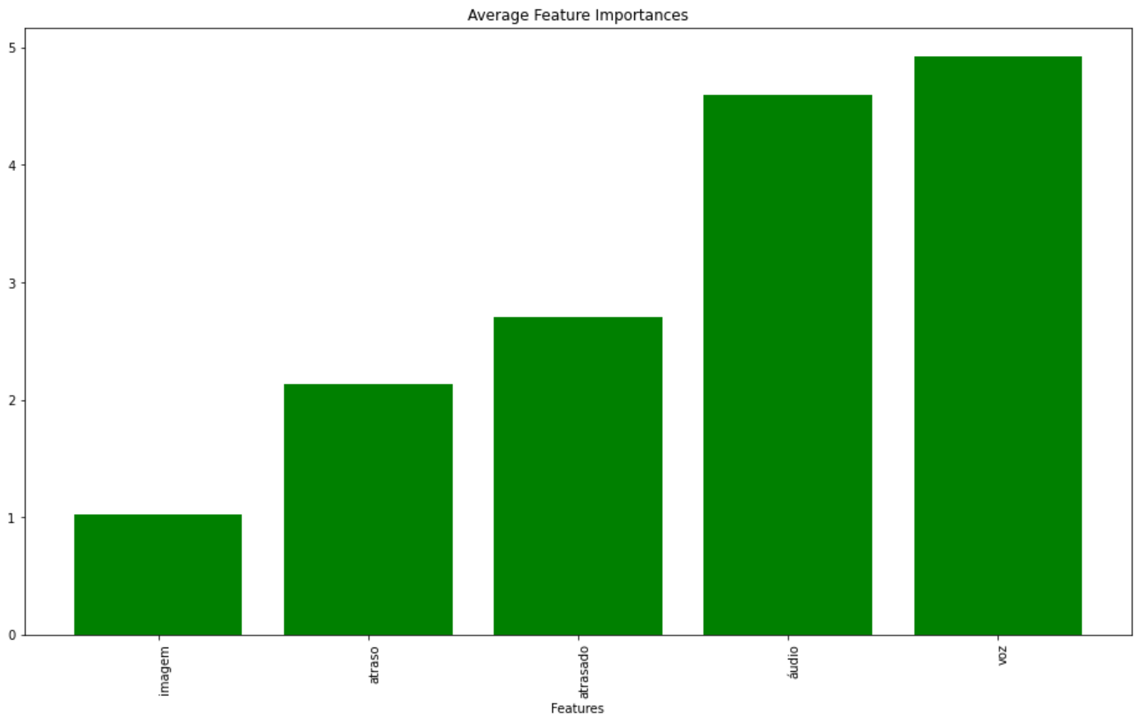


Figure 4.12: Average feature importances on *Virtual Operator* class *Qualificado.Áudio atrasado* when stop-words are considered (top) and removed from the dataset (bottom)

reduction in the mean attribute score for the true label, which was still higher than the attribution score for the class that was wrongly predicted by the BERT Base classifier.

When analyzing the three classes to which BERT + TAPT was detrimental (Fig-

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Score	Word Importance
34	45 (0.50)	1.40	[CLS] do i have any alarm ##s set for tomorrow [SEP]
34	34 (0.61)	1.90	[CLS] do i have any alarm ##s set for tomorrow [SEP]
34	47 (0.58)	0.53	[CLS] is my reminder alarm set for dance class [SEP]
34	34 (0.47)	1.67	[CLS] is my reminder alarm set for dance class [SEP]
34	62 (0.68)	0.28	[CLS] remind me about my alarm ##s today [SEP]
34	34 (0.69)	1.04	[CLS] remind me about my alarm ##s today [SEP]
1	11 (0.57)	1.19	[CLS] how can i go from b ##ost ##on to new yo ##rk by train [SEP]
1	1 (0.65)	1.42	[CLS] how can i go from b ##ost ##on to new yo ##rk by train [SEP]
1	39 (0.23)	0.26	[CLS] how do i get to du ##nk ##in don ##uts in at ##lant ##ic city n ##j [SEP]
1	1 (0.54)	1.25	[CLS] how do i get to du ##nk ##in don ##uts in at ##lant ##ic city n ##j [SEP]
18	38 (0.74)	1.30	[CLS] are there any good pizza places around here [SEP]
18	18 (0.50)	0.34	[CLS] are there any good pizza places around here [SEP]
18	38 (0.54)	1.68	[CLS] i want to find some chin ##ese food what is near me [SEP]
18	18 (0.69)	0.76	[CLS] i want to find some chin ##ese food what is near me [SEP]
18	38 (0.83)	0.80	[CLS] i need some su ##shi what ' s closest [SEP]
18	18 (0.48)	0.58	[CLS] i need some su ##shi what ' s closest [SEP]

Figure 4.13: Example sentences from the 3 classes that most benefited from TAPT on *NLU-Evaluation*, showing, for each example, the true and predicted labels, the attribution score and the features that most influenced the classifier prediction. Each sentence is listed twice - the first occurrence shows the result from the BERT Base classifier, and the second one, from the BERT + TAPT classifier.

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
18	18 (0.50)	38	0.15	[CLS] are there any good pizza places around here [SEP]
18	18 (0.69)	38	0.25	[CLS] i want to find some chin ##ese food what is near me [SEP]
18	18 (0.48)	38	0.65	[CLS] i need some su ##shi what ' s closest [SEP]

Figure 4.14: sentences belonging to class 18 (*recommendation_ locations*) which had their attribution scores and token importances computed for class 38 (*takeaway_ order*). These attribution scores are lower than those computed for the true class label.

ure 4.15), we could identify that, for some classes, the misclassifications favored specific classes. For instance, sentences belonging to class 0 (*calendar_ notification*) were wrongly predicted by BERT + Base under class 62 (*reminder_ set*). We also computed, using the BERT + TAPT classifier, the attribution scores for these sentences with respect to their true labels and compared them with the scores computed for the predictions output by the classifier. This comparison can be seen in figure 4.16. Attribution scores were, in general, higher when computed concerning the predicted class than the true class. One possible explanation for this behavior is that some classes may represent similar intents, making it harder for the classifier to learn differences between them. In fact, classes 0 (*calendar_notification*) and 62 (*reminder_set*), for example, represent very similar ideas, sharing tokens like *remind*, *lunch*, and *about*, which are relevant to classification under both classes.

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Score	Word Importance
0	0 (0.64)	0.84	[CLS] remind upcoming meeting with em ##ine ##m [SEP]
0	19 (0.91)	0.27	[CLS] remind upcoming meeting with em ##ine ##m [SEP]
0	0 (0.51)	1.73	[CLS] can you remind me tomorrow morning about my dinner plans for the weekend [SEP]
0	62 (0.82)	0.65	[CLS] can you remind me tomorrow morning about my dinner plans for the weekend [SEP]
0	0 (0.58)	0.72	[CLS] remind me about the meeting [SEP]
0	62 (0.51)	0.76	[CLS] remind me about the meeting [SEP]
0	0 (0.59)	1.92	[CLS] remind me about my lunch date for mon ##day [SEP]
0	62 (0.75)	0.89	[CLS] remind me about my lunch date for mon ##day [SEP]
13	13 (0.68)	1.98	[CLS] who ' s that song by [SEP]
13	33 (0.57)	0.50	[CLS] who ' s that song by [SEP]
13	13 (0.25)	1.00	[CLS] title [SEP]
13	17 (0.38)	1.00	[CLS] title [SEP]
13	13 (0.55)	0.47	[CLS] who is the singer of hotel ca ##li ##fo ##rn ##ia [SEP]
13	28 (0.47)	0.48	[CLS] who is the singer of hotel ca ##li ##fo ##rn ##ia [SEP]
32	32 (0.29)	1.04	[CLS] next time you should [SEP]
32	44 (0.39)	0.82	[CLS] next time you should [SEP]
32	32 (0.51)	0.69	[CLS] please correct yourself [SEP]
32	42 (0.39)	0.75	[CLS] please correct yourself [SEP]
32	32 (0.63)	1.29	[CLS] start over [SEP]
32	42 (0.66)	1.41	[CLS] start over [SEP]

Figure 4.15: Example sentences from the three classes which accuracy degraded when TAPT was applied on *NLU-Evaluation*, showing, for each example, the true and predicted labels, the attribution score and the features that most influenced the classifier prediction. Each sentence is listed twice - the first occurrence shows the result from the BERT Base classifier, and the second one, from the BERT + TAPT classifier.

For the analysis of *Virtual Operator*, we focused on the classes listed on table 4.10. In figure 4.17 we list some sentences taken from the classes that had an improvement on their accuracies when TAPT was used. The BERT Base classifier failed to correctly predict these sentences' classes, but the classifier using the BERT + TAPT strategy correctly classified them. We can observe in these classes that the average attribution scores were, in general, higher on those sentences classified by the classifier trained with TAPT, as occurred with *NLU-Evaluation*, with some tokens switching from a negative to a positive contribution. For example, on class 107 (*Genérico.Problema com troca de canal*), n-grams *##o troca* and *o## muda*, initially presenting a negative importance on BERT Base and contributing to the wrong prediction on class 72 (*Genérico.Canal travado*), switch to a positive contribution when TAPT is used, enforcing the importance of n-grams *na ##o*

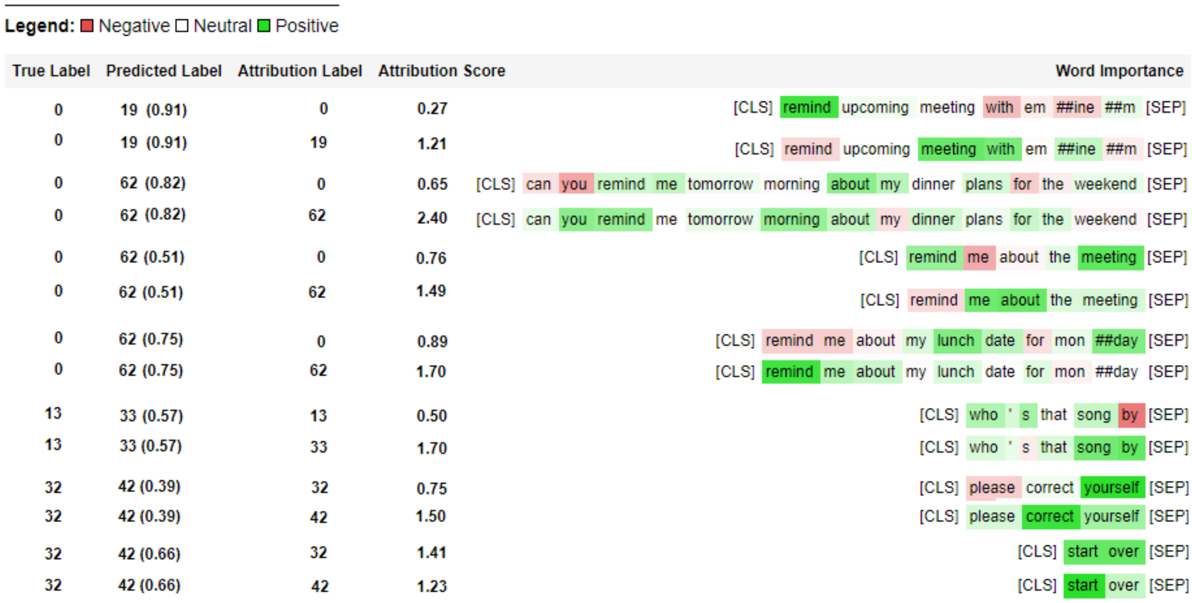


Figure 4.16: List of example sentences from *NLU-Evaluation*, showing token importances and mean attribution scores calculated with respect to the sentence's true class (first sentence) and predicted class (second sentence) for classes where TAPT was detrimental. Attribution scores were in general higher for the predicted class than the scores of the respective true class.

troca and *na ##o muda*, which influence the correct classification on class 72.

Table 4.10: *Virtual Operator* classes selected for investigation. Classes 107, 91 and 105 had the highest improvement on their accuracy when TAPT was used. Classes 15, 84 and 115, conversely, had their accuracy degraded.

ID	Class Name	Accuracy		%
		BERT BASE	BERT BASE + TAPT	
107	Genérico.Problema com troca de canal	0.485	0.588	21.24%
91	Qualificado.Recarga	0.808	0.926	14.60%
105	Qualificado.Habilitar recurso de senha	0.836	0.919	9.93%
15	Qualificado.Técnico não resolveu	0.662	0.611	-7.70%
84	Qualificado.Controle quebrado	0.857	0.786	-8.28%
115	Genérico.Promessa de oferta	0.629	0.552	-11.99%

The analysis on sentences belonging to the classes that experienced lower performance on BERT + TAPT, listed in Figure 4.18 shows a reduction in the mean attribute scores for all analyzed examples. Figure 4.19 shows that, as occurred with *NLU-Evaluation*, mean attribution scores for examples in these classes were higher when computed with respect to the predicted label than when computed concerning the true label. Also, we identified that all analyzed sentences were incorrectly labelled, and in four of them BERT

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Score	Word Importance
107	72 (0.75)	0.26	[CLS] motivo que na ##o na ##o troca os canais é o aparelho que ficou [SEP]
107	107 (1.00)	2.49	[CLS] motivo que na ##o na ##o troca os canais é o aparelho que ficou [SEP]
107	16 (0.76)	1.10	[CLS] problema ao trocar o canal no aparelho s ##ky [SEP]
107	107 (0.64)	1.66	[CLS] problema ao trocar o canal no aparelho s ##ky [SEP]
107	14 (1.00)	0.00	[CLS] é o sinal canal na ##o muda [SEP]
107	107 (1.00)	0.44	[CLS] é o sinal canal na ##o muda [SEP]
91	115 (0.44)	0.80	[CLS] é porque eu fiz uma reca ##r ##ga de r $ 54 e ele liberado no canal 12 na ##o o cara falou eu lig ##ue ##i para libera ##r sexta - feira [SEP]
91	91 (1.00)	1.22	[CLS] é porque eu fiz uma reca ##r ##ga de r $ 54 e ele liberado no canal 12 na ##o o cara falou eu lig ##ue ##i para libera ##r sexta - feira [SEP]
91	17 (1.00)	1.67	[CLS] eu gostaria de fazer uma reca ##r ##ga só isso [SEP]
91	91 (0.96)	1.53	[CLS] eu gostaria de fazer uma reca ##r ##ga só isso [SEP]
91	115 (0.71)	0.11	[CLS] eu fiz a reca ##r ##ga de man ##ha e os canais de filmes na ##o esta ##o entrando [SEP]
91	91 (1.00)	1.39	[CLS] eu fiz a reca ##r ##ga de man ##ha e os canais de filmes na ##o esta ##o entrando [SEP]
91	4 (0.48)	1.58	[CLS] al ##ô eu t ##ô com um problema humana já caiu o sinal da s ##ky por assinatura s ##ky pré - pago reca ##r ##ga hoje na ##o [SEP]
91	91 (1.00)	2.00	[CLS] al ##ô eu t ##ô com um problema humana já caiu o sinal da s ##ky por assinatura s ##ky pré - pago reca ##r ##ga hoje na ##o [SEP]
91	17 (0.94)	0.79	[CLS] eu t ##ô ligando porque eu quero fazer uma reca ##r ##ga para s ##ky pré - pago [SEP]
91	91 (1.00)	2.02	[CLS] eu t ##ô ligando porque eu quero fazer uma reca ##r ##ga para s ##ky pré - pago [SEP]
91	55 (0.54)	0.30	[CLS] eu t ##ô querendo pegar o código do número do cliente do meu aparelho para mim fazer uma reca ##r ##ga [SEP]
91	91 (1.00)	1.51	[CLS] eu t ##ô querendo pegar o código do número do cliente do meu aparelho para mim fazer uma reca ##r ##ga [SEP]
91	(0.45) 0	0.12	[CLS] o que eu fiz uma reca ##r ##ga na minha s ##ky e no dia 2 e ainda na ##o entrou e eu já lig ##ue ##i para aí já está quase a quinta vez que eu lig ##o para aí ninguém resolve meu problema [SEP]
91	91 (1.00)	1.18	[CLS] o que eu fiz uma reca ##r ##ga na minha s ##ky e no dia 2 e ainda na ##o entrou e eu já lig ##ue ##i para aí já está quase a quinta vez que eu lig ##o para aí ninguém resolve meu problema [SEP]
105	95 (1.00)	1.01	[CLS] o meu e - ma ##il do cadas ##tro mudou em ta ##o manda ##ndo confirmar sen ##ha no por e - ma ##il [SEP]
105	105 (0.94)	1.56	[CLS] o meu e - ma ##il do cadas ##tro mudou em ta ##o manda ##ndo confirmar sen ##ha no por e - ma ##il [SEP]

Figure 4.17: Example sentences from the 3 classes that most benefited from TAPT on *Virtual Operator*, showing, for each example, the true and predicted labels, the attribution score and the features that most influenced the classifier prediction. Each sentence is listed twice - the first occurrence shows the result from the BERT Base classifier, and the second one, from the BERT + TAPT classifier.

+ TAPT was able to predict the correct class correctly. It is possible that classes facing a lower accuracy on BERT + TAPT classifiers in fact contain mislabelled samples.

Regarding *Mercado Livre*, the list of analysed classes is shown in Table 4.11, and Figure 4.20 presents examples of sentences for which classification by BERT Base failed, but were correctly classified by BERT + TAPT. As in the previous analysis, the attribution score was also higher on the BERT + TAPT examples. The use of TAPT also enforced

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Score	Word Importance
115	115 (0.75)	2.20	[CLS] eu contra ##te ##i um novo serviço da s ##ky aonde o tele ##cin ##e está liberado e na ##o liber ##ou em todos os meus pontos eu queria saber o que tá acontecendo [SEP]
115	78 (1.00)	-1.34	[CLS] eu contra ##te ##i um novo serviço da s ##ky aonde o tele ##cin ##e está liberado e na ##o liber ##ou em todos os meus pontos eu queria saber o que tá acontecendo [SEP]
15	15 (1.00)	2.50	[CLS] ol ##ha só eu queria que vie ##s ##se técnico aqui porque na ##o tem a televis ##ao na ##o sai do do do do primeiro dos caído ##s h ##tn ##1 . está funcionando [SEP]
15	76 (0.38)	0.39	[CLS] ol ##ha só eu queria que vie ##s ##se técnico aqui porque na ##o tem a televis ##ao na ##o sai do do do do primeiro dos caído ##s h ##tn ##1 . está funcionando [SEP]
15	15 (1.00)	1.84	[CLS] eu quero pedir uma venda de um técnico aqui para poder olhar a televis ##ao para mim porque tá sem sinal total [SEP]
15	13 (1.00)	-1.27	[CLS] eu quero pedir uma venda de um técnico aqui para poder olhar a televis ##ao para mim porque tá sem sinal total [SEP]
15	15 (1.00)	2.56	[CLS] al ##ô bom dia eu queria falar com alguém que me ajud ##e mand ##e um técnico aqui para casa porque minha s ##ky na ##o está funcionando eu estou com um aparelho queima ##do [SEP]
15	(1.00) 9	-0.60	[CLS] al ##ô bom dia eu queria falar com alguém que me ajud ##e mand ##e um técnico aqui para casa porque minha s ##ky na ##o está funcionando eu estou com um aparelho queima ##do [SEP]
84	(1.00) 84	1.62	[CLS] é o meu controle que tá ruim quero trocar a cor de controle [SEP]
84	(1.00) 26	-0.85	[CLS] é o meu controle que tá ruim quero trocar a cor de controle [SEP]

Figure 4.18: Example sentences from the 3 classes which accuracy degraded when TAPT was applied on *Virtual Operator*, showing, for each example, the true and predicted labels, the attribution score and the features that most influenced the classifier prediction. Each sentence is listed twice - the first occurrence shows the result from the BERT Base classifier, and the second one, from the BERT + TAPT classifier.

the occurrence of some n-grams which are more specific to the vocabulary, like *teclado i ##pad pro* and *mi ##di nova ##tion*. Looking into examples taken from the classes that had the highest negative impact on TAPT, in Figure 4.21, the same reduction in the mean attribution scores can be identified on the majority of the sentences. It is also possible to identify that some tokens, like *carne*, used in a sentence labelled under class 765 (*MEAT_GRINDERS*) and tokens *fras* and *##queira*, which appear together on class 996 (*MAKEUP_TRAIN_CASES*) to form the word *frasqueira* become more important when computed concerning their respective predicted classes - 320 (*FOOD_PROCESSORS*) and 774 (*TOILETRY_BAGS*). This can be seen when sentence attribution scores are computed with respect to the predicted classes on BERT + TAPT, as shown in Figure 4.22. We also observed that classes 320 and 774 had a support of 135 and 40, whereas classes 765 and 996 had supports of 3 and 4, respectively, which may also contribute to the misclassifications observed.

We also observed that a sentence's mean attribute score may provide an alternative means to help evaluate the quality of a prediction. In figure 4.26, one random class

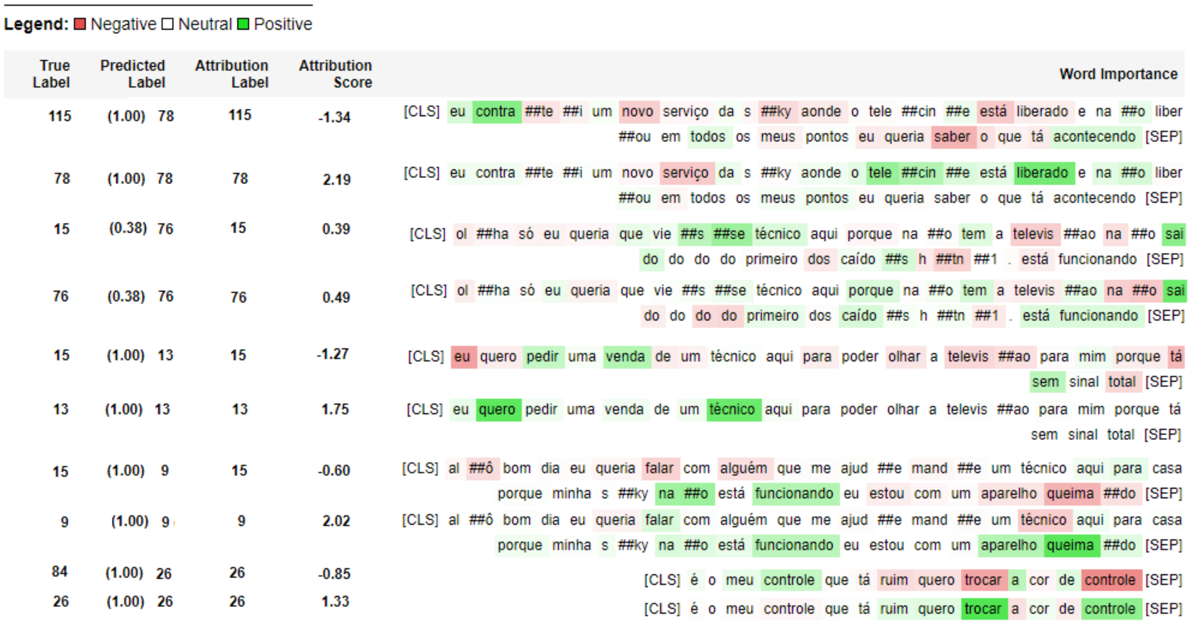


Figure 4.19: List of example sentences from *Virtual Operator*, showing token importances and mean attribution scores calculated with respect to the sentence's true class (first sentence) and predicted class (second sentence) for classes where TAPT was detrimental. Attribution scores were in general higher for the predicted class than the scores of the respective true class.

Table 4.11: *Mercado Livre* classes selected for investigation. Classes 107, 91 and 105 had the highest improvement on their accuracy when TAPT was used. Classes 15, 84 and 115, conversely, had their accuracy degraded.

ID	Class Name	Accuracy		%
		BERT BASE	BERT BASE + TAPT	
999	IGNITION_CONTROL_MODULES	0.222	0.429	170.27%
584	TABLET_KEYBOARDS	0.333	0.667	100.30%
928	KEYBOARD_CONTROLLERS	0.154	0.308	100.00%
765	MEAT_GRINDERS	0.800	0.000	-100.00%
994	NECK_GAITERS_MASKS_AND_BALACLAVAS	0.667	0.000	-100.00%
996	MAKEUP_TRAIN_CASES	0.667	0.000	-100.00%

was selected for each one of the three investigated datasets. We presented two scattered plots for each class - the first one, using the softmax output from the predicted class, and the second one, the mean attribute score from the classified sample with respect to the predicted class. A threshold could be set on the mean attribute score plot for all three cases that would result in better separation between correct and incorrect samples than using softmax. Regarding *NLU-Evaluation* class *calendar_query_event* is not even possible to define a threshold on the softmax plot, whereas in the attribute score plot, a threshold close to 0.9 separates most of the correct samples from the incorrect ones. This

Legend: ■ Negative □ Neutral ■ Positive

True Label	Predicted Label	Attribution Score	Word Importance
999	73 (0.91)	0.03	[CLS] ch ##ico ##te distribu ##idor eletro ##nico 2 vias - fus ##ca g ##m fi ##at 147 [SEP]
999	999 (0.62)	0.49	[CLS] ch ##ico ##te distribu ##idor eletro ##nico 2 vias - fus ##ca g ##m fi ##at 147 [SEP]
584	100 (0.64)	1.69	[CLS] teclado i ##pad pro 12 . 9 s ##mart k ##ey ##board + ap ##ple pen ##cil lac ##rado [SEP]
584	584 (0.77)	1.95	[CLS] teclado i ##pad pro 12 . 9 s ##mart k ##ey ##board + ap ##ple pen ##cil lac ##rado [SEP]
928	240 (1.00)	-0.27	[CLS] controlado ##r mi ##di nova ##tion m ##ki ##i 49 ##s ##l [SEP]
928	928 (0.50)	1.73	[CLS] controlado ##r mi ##di nova ##tion m ##ki ##i 49 ##s ##l [SEP]

Figure 4.20: Example sentences from the 3 classes that most benefited from TAPT on *Mercado Livre*, showing, for each example, the true and predicted labels, the attribution score and the features that most influenced the classifier prediction. Each sentence is listed twice - the first occurrence shows the result from the BERT Base classifier, and the second one, from the BERT + TAPT classifier.

Legend: ■ Negative □ Neutral ■ Positive

True Label	Predicted Label	Attribution Score	Word Importance
996	996 (1.00)	1.70	[CLS] mal ##eta fras ##queira p pro - maquia ##gem profissional fre ##te gra ##tis [SEP]
996	774 (0.88)	0.56	[CLS] mal ##eta fras ##queira p pro - maquia ##gem profissional fre ##te gra ##tis [SEP]
996	996 (1.00)	2.19	[CLS] mal ##eta fras ##queira alu ##min ##io porta jo ##ias maquia ##gem [SEP]
996	774 (0.98)	0.19	[CLS] mal ##eta fras ##queira alu ##min ##io porta jo ##ias maquia ##gem [SEP]
994	994 (0.69)	0.62	[CLS] bala ##cla ##va 2 - pac ##k mas ##cara fa ##cial cas ##cos de motocic ##leta for ##ro [SEP]
994	399 (0.22)	0.63	[CLS] bala ##cla ##va 2 - pac ##k mas ##cara fa ##cial cas ##cos de motocic ##leta for ##ro [SEP]
765	765 (0.98)	1.92	[CLS] mo ##edor de carne manual ma ##quina profissional corte no 32 [SEP]
765	320 (0.96)	0.17	[CLS] mo ##edor de carne manual ma ##quina profissional corte no 32 [SEP]
765	765 (1.00)	2.03	[CLS] mo ##edor de carne mo ##ida fazer lingu ##ica ma ##quina manual 10 a [SEP]
765	320 (0.85)	0.41	[CLS] mo ##edor de carne mo ##ida fazer lingu ##ica ma ##quina manual 10 a [SEP]

Figure 4.21: Example sentences from the 3 classes which accuracy degraded when TAPT was applied on *Mercado Livre*, showing, for each example, the true and predicted labels, the attribution score and the features that most influenced the classifier prediction. Each sentence is listed twice - the first occurrence shows the result from the BERT Base classifier, and the second one, from the BERT + TAPT classifier.

property of the mean attribute may allow it to be used as a confidence measure to help to decide if a prediction can be considered reliable or not.

4.5 Case Study

The ULMFit LM pretrained on Wikipedia BR, fine-tuned on the *Virtual Operator* dataset, and trained on a user intent classification task was further applied in a real case scenario on a large Cable TV and content provider. The ULMFit model was chosen for this case

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
996	774 (0.88)	996	0.56	[CLS] mal ##eta fras ##queira p pro - maquia ##gem profissional fre ##te gra ##tis [SEP]
996	774 (0.88)	774	1.37	[CLS] mal ##eta fras ##queira p pro - maquia ##gem profissional fre ##te gra ##tis [SEP]
996	774 (0.98)	996	0.19	[CLS] mal ##eta fras ##queira alu ##min ##io porta jo ##ias maquia ##gem [SEP]
996	774 (0.98)	774	2.00	[CLS] mal ##eta fras ##queira alu ##min ##io porta jo ##ias maquia ##gem [SEP]
994	399 (0.22)	994	0.63	[CLS] bala ##cla ##va 2 - pac ##k mas ##cara fa ##cial cas ##cos de motocic ##leta for ##ro [SEP]
994	399 (0.22)	399	1.75	[CLS] bala ##cla ##va 2 - pac ##k mas ##cara fa ##cial cas ##cos de motocic ##leta for ##ro [SEP]
765	320 (0.96)	765	0.17	[CLS] mo ##edor de carne manual ma ##quina profissional corte no 32 [SEP]
765	320 (0.96)	320	1.82	[CLS] mo ##edor de carne manual ma ##quina profissional corte no 32 [SEP]
765	320 (0.85)	765	0.41	[CLS] mo ##edor de carne mo ##ida fazer lingu ##ica ma ##quina manual 10 a [SEP]
320	320 (0.85)	320	1.39	[CLS] mo ##edor de carne mo ##ida fazer lingu ##ica ma ##quina manual 10 a [SEP]

Figure 4.22: List of example sentences from *Mercado Livre*, showing token importances and mean attribution scores calculated with respect to the sentence's true class (first sentence) and predicted class (second sentence) for classes where TAPT was detrimental. Attribution scores were in general higher for the predicted class than the scores of the respective true class.

study because it was the first model trained for the investigations presented in this study. This provider offers a telephonic technical support service covering the intents represented by the 121 classes found in the *Virtual Operator* dataset. An automated support service using ASR to collect users' input was implemented to capture their intent. The transcribed text output by the ASR is fed into a rule-based classifier that uses regular expressions to identify the intent. In this case, the application delivers the call to a more specific workflow but still handles the call automatically through ASR. If there is no matching expression, the classification fails, and the call is diverted to a human operator with no intent information. The percentual amount of users that could be serviced automatically by the automated system is called *Retention Rate*, and it is the primary metric used to evaluate the performance of such automated systems. Another metric commonly used is the *Net Promoter Score (NPS)*, which is employed as a measure of customer satisfaction. NPS is computed from scores provided by users at the end of the call after the automated system services them.

We decided to test the ULMFit classifier in those situations where the rule-based classifier cannot identify the user's intent. After a non-matching result is returned by the rule-based classifier, the same sentence is sent to the ULMFit classifier for intent classification. We referred to this implementation as *hybrid classifier*. We comparatively tested the performance of this classifier by designating 20% of the incoming calls to it. After four weeks, we compared the retention rates from both classifiers. The hybrid

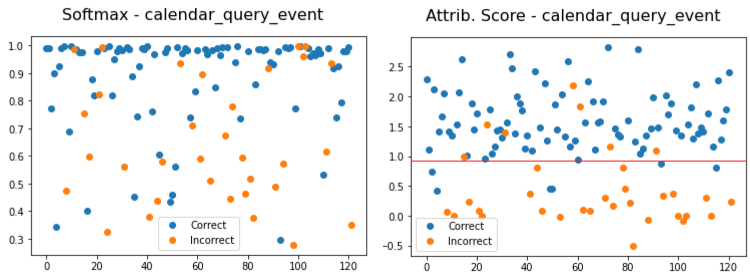


Figure 4.23: NLU-Evaluation

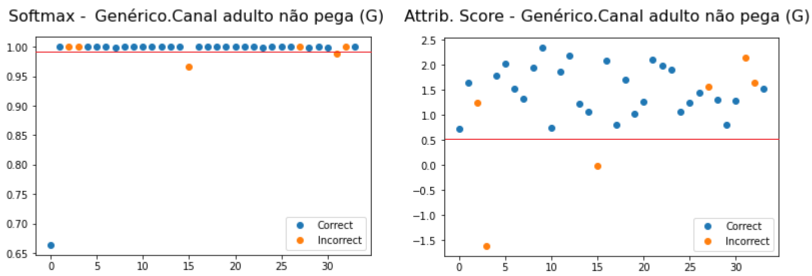


Figure 4.24: Virtual Operator

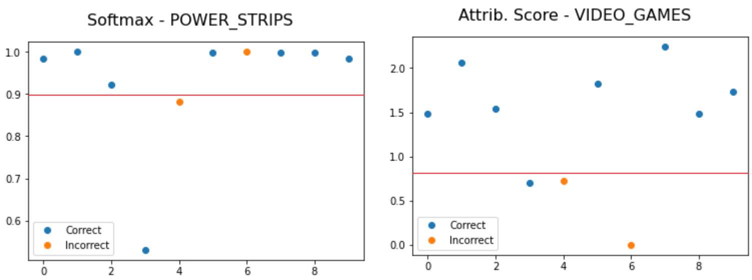


Figure 4.25: Mercado Livre

Figure 4.26: Comparison between class softmax scores and attribution scores plotted for each dataset classified using BERT + TAPT. Attribution scores allow for a clearer separation between correctly and incorrectly classified samples than softmax.

classifier could retain 4% more calls, with no perceived loss in NPS, meaning that the ULMFit classifier helped to recover situations that would otherwise lead to transferring the call to a human operator.

Chapter 5

Conclusions

The recent advances in the field of machine learning provided an increasingly vast amount of approaches that can be applied in NLP tasks such as text classification. We aimed to investigate the intent classification of short text sentences and which neural language models and classifiers could be efficiently applied to such classification tasks. The datasets used in this research contained sentences that were directly inputted by a user through typing or by means of conversion from voice to text using Automatic Speech Recognition (ASR). Such sentences can carry noise such as spelling or grammatical errors produced by the user, ASR errors induced by environmental noise or even by unusual idiomatic expressions. To numerically represent the short sentences, sparse and dense vectors are taken into account. In the first case, we rely on Bag-of-Words (BOW) features extracted from a sparse-vectors representation. In the second case, we consider low-dimensional dense vectors extracted from different embedding language models, including embeddings induced from shallow neural networks, namely, Word2Vec and FastText, and embeddings induced from deep architectures, namely, ELMo and BERT. These embeddings come from distinguished training mechanisms; namely, they are collected from pretrained publicly available resources, pretrained on the dataset vocabulary, or jointly trained with the classifier. Conversely, to generate the classification models from sentences, this dissertation focused on neural network classifiers ranging from a shallow Feed-Forward Neural Network (FFNN) to deep learning models, namely, Convolutional Neural Network (CNN) and Bidirectional Long-Short Term Memory (LSTM). Furthermore, we also investigated whether such classification tasks could benefit from fine-tuning the pretrained language models using the strategies conveyed by two methods, namely, ULMFit and BERT. Fine-tuning is conducted from the downstream classification task following Task-Adaptive PreTraining (TAPT). Lastly, we tested TAPT on BERT LMs but including an additional pretraining step that used sentences from the target datasets. Experiments were conducted with three datasets. *Virtual Operator* contained 669,929 examples in Brazilian Portuguese and 121 classes; *25,578* sentences in English and 64 classes; and *Mercado Livre*, 692,750 samples and 1,048 classes.

In regards to question one, formulated on section 1.1, the experimental results given by this dissertation pointed out that BERT LMs fine-tuned a downstream classification task including an intermediate TAPT step provided the best overall performance, achieving superior accuracy on two datasets from the three we tested. ULMFit provided a slightly lower performance when compared to BERT. Regarding ULMFit models, LMs pretrained on random tweets had superior performance on *NLU-Evaluation*, a result we believe can be related to the smaller mean sentence size on this dataset. When comparing only LMs for features extraction, BERT-classifier with sentence features extracted from BERT also had superior performance, followed by the BiLSTM classifier with jointly trained embeddings. This BiLSTM was only outperformed by the Word2Vec tweets LM on *NLU-Evaluation*. Again, we believe this result was also related to the concise, short, and command-like sentences, which are characteristic of this dataset. This investigation also demonstrated that an FFFN trained on BOW features extracted from sparse-vectors representations can achieve reasonable performance, in some cases comparable to some state-of-the-art approaches. The BOW classifier was superior to ULMFit trained on Wikipedia on both *NLU-Evaluation* and *Mercado Livre*. We also showed that stop-words convey relevant information which is learned by the classifier, and its removal can be detrimental to the classifier's performance. Both *NLU-Evaluation* and *Virtual Operator* experienced a drop on accuracy after removal of stop-words. The use of Captum and its feature attribution score method allowed us to visualize and understand the influence of stop-words in the output of a classifier. Some of them figured amongst the topmost influential tokens which define the outcome of a prediction, and its removal sometimes led to loss of information, and as a consequence, misclassification.

Regarding question two of our investigation, we can conclude that the language model approaches investigated here, despite having the English language as their primary research focus, and be successfully applied on Portuguese language corpora. The results achieved by the classifiers trained in this research on PT-BR datasets have comparable performances to their EN counterparts.

In relation to question three, the analysis of the TAPT approach on BERT demonstrated that, while this approach could provide an overall improvement on classification performance, not all classes in the investigated datasets benefited from this strategy. In fact, in *Mercado Livre*, the lack of improvement when TAPT was used was related to the mean gain on accuracy of classes that benefited from TAPT being compensated by the losses on accuracy of classes in which that strategy impaired performance. Although the reason that led some classes to have worse results on BERT + TAPT remains to be further investigated, we were able to identify some situations that might have contributed to this behavior. On *Virtual Operator*, we demonstrated that the BERT + TAPT classifier correctly predicted samples that were wrongly labeled on the test set, thus leading to a false reduction in the computed accuracy. It is possible that such TAPT-induced reduction on accuracy can be used as an indicator of classes with a higher percentage of mislabelled samples. Moreover, some classes may also represent conflicting intents. On *NLU-Evaluation*, we identified that classes 0 *calendar_notification* and 62 *reminder_set* represented similar ideas and shared some tokens which had high importance according to our feature importance analysis. The mechanism behind this behaviour demands further investigation, but one explanation for this may reside in the fact that class 62 has a support that is 64.4% higher than class 0 support. TAPT may favor classes with higher support when there is a significant level of semantic conflict between them.

Lastly, we identified that a sentence's mean attribute score might be used as an alternative means to evaluate prediction quality. A comparative analysis of both Softmax and Attribute Scores of randomly selected classes on all three datasets showed that the latter provides better separation between correct and incorrect sentences. In this way, a confidence level threshold could be defined, which would allow a classifier to decide whether an output could be reliable or not.

5.1 Limitations and Threats to Validity

All three datasets selected for this investigation have a high degree of class imbalance. It is possible that applying techniques such as oversampling or undersampling could affect the results presented here. Also, due to hardware and time limitations, all sentence examples selected for Captum attribution scores analysis were selected based on their true label. This limitation can hide samples belonging to other classes, which were eventually predicted under the analyzed class, impairing precision. The results concerning the best strategies are based on an analysis of their absolute score values. To better assess the best values, it would be appropriate to rely on a statistical significance test. All the conclusions are taken from only a small set of three datasets due to the lack of publicly available user-intent data. A larger set of datasets could lead the conclusions to a different path. Also, hyperparameters were defined following a greedy search heuristics, which does not guarantee that the best possible values were used during training.

5.2 Future Work

Using curated versions of the datasets to eliminate labeling errors could help identify the role of mislabelled samples in those classes which presented lower accuracy when TAPT was used. Furthermore, one could use our results as motivation to design strategies that automatically adjust mislabelled examples or better learn from them. Also, using techniques such as data augmentation to increase the sample of misrepresented classes or employing weighted loss functions during the model training is worth investigating. User intents have a noisy nature that is not directly contemplated in pretrained language models. Further investigating fine-tuning and pretraining from such a noisy environment could also help to contribute to other classification tasks from noisy data, such as the ones from social media and calls to other types of services, such as 911 (190) service. An extension of this investigation could rely on methods to access the quality of data, as proposed by works such as [59].

References

[1] Cs224n word vectors 2 and word senses. https://programmersought.com/article/1824 4563325/. Accessed: 2021-01-19.

[2] Akbik, A.; Bergmann, T.; Blythe, D.; Rasul, K.; Schweter, S.; Vollgraf, R. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)* (2019), pp. 54–59.

[3] Almeida, F.; Xexéo, G. Word embeddings: A survey, 2019.

[4] Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate, 2016.

[5] Basheer, I.; Hajmeer, M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods 43*, 1 (2000), 3–31. Neural Computting in Micrbiology.

[6] Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. A neural probabilistic language model. *J. Mach. Learn. Res. 3*, null (Mar. 2003), 1137–1155.

[7] Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* (2016).

[8] Braun, D.; Hernandez Mendez, A.; Matthes, F.; Langen, M. Evaluating natural language understanding services for conversational question answering systems. In *Proc. of the 18th Annual SIGdial Meeting on Discourse and Dialogue* (2017), Association for Computational Linguistics, pp. 174–185.

[9] Chelba, C.; Mikolov, T.; Schuster, M.; Ge, Q.; Brants, T.; Koehn, P.; Robinson, T. One billion word benchmark for measuring progress in statistical language modeling, 2014.

[10] Chen, Q.; Zhuo, Z.; Wang, W. Bert for joint intent classification and slot filling, 2019.

[11] Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[12] Collobert, R.; Weston, J. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML '08* (2008).

[13] Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; Primet, M.; Dureau, J. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces, 2018.

[14] Deng, J.; Dong, W.; Socher, R.; Li, L.; Kai Li; Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 248–255.

[15] Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805* (2018).

[16] Goldberg, Y.; Hirst, G. *Neural Network Methods in Natural Language Processing.* Morgan & Claypool Publishers, 2017.

[17] Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks, 2020.

[18] Haykin, S. 1 feedforward neural networks : An introduction. In *Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives* (2004).

[19] Hinton, G. E.; McClelland, J. L.; Rumelhart, D. E. Distributed representations. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. MIT Press, 1986, pp. 77–109.

[20] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation 9*, 8 (1997), 1735–1780.

[21] Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw. 2*, 5 (July 1989), 359–366.

[22] Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018), pp. 328–339.

[23] Joshi, P. A step-by-step nlp guide to learn elmo for extracting features from text. https://www.analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/. Accessed: 2021-01-20.

[24] Kalchbrenner, N.; Blunsom, P. Recurrent convolutional neural networks for discourse compositionality. In *Proc. of the 2013 Workshop on Continuous Vector Space Models and their Compositionality* (2013).

[25] Kokhlikyan, N.; Miglani, V.; Martin, M.; Wang, E.; Alsallakh, B.; Reynolds, J.; Melnikov, A.; Kliushkina, N.; Araya, C.; Yan, S.; Reblitz-Richardson, O. Captum: A unified and generic model interpretability library for pytorch, 2020.

[26] Krogh, A. What are artificial neural networks? *Nature Biotechnology 26*, 2 (Feb 2008), 195–197.

[27] Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[28] Lee, J. Y.; Dernoncourt, F. Sequential short-text classification with recurrent and convolutional neural networks. In *Proc. of the 2016 Conference of the NAACL: Human Language Technologies* (2016), pp. 515–520.

[29] Litman, D. J.; Walker, M. A.; Kearns, M. S. Automatic detection of poor speech recognition at the dialogue level. In *Proc. of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* (1999), ACL, pp. 309–316.

[30] Mahgoub, A.; Shahin, Y.; Mansour, R.; Bagchi, S. SimVecs: Similarity-based vectors for utterance representation in conversational AI systems. In *Proc. of the 23rd Conference on Computational Natural Language Learning (CoNLL)* (2019), Association for Computational Linguistics, pp. 708–717.

[31] Martínez-Cámara, E.; Martín-Valdivia, M. T.; Urena-López, L. A.; Montejo-Ráez, A. R. Sentiment analysis in twitter. *Natural Language Engineering 20*, 1 (2014), 1–28.

[32] McCulloch, W. S.; Pitts, W. *A Logical Calculus of the Ideas Immanent in Nervous Activity.* MIT Press, Cambridge, MA, USA, 1943, p. 15–27.

[33] Merity, S.; Keskar, N. S.; Socher, R. Regularizing and optimizing lstm language models, 2017.

[34] Merity, S.; Xiong, C.; Bradbury, J.; Socher, R. Pointer sentinel mixture models, 2016.

[35] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space, 2013.

[36] Mikolov, T.; Grave, E.; Bojanowski, P.; Puhrsch, C.; Joulin, A. Advances in pre-training distributed word representations. In *Proc. of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (Miyazaki, Japan, May 2018), European Language Resources Association (ELRA).

[37] Minsky, M.; Papert, S. *Perceptrons.* MIT Press, Cambridge, MA, 1969.

[38] Ng, A. Y. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-First International Conference on Machine Learning* (New York, NY, USA, 2004), ICML '04, Association for Computing Machinery, p. 78.

[39] Nigam, A.; Sahare, P.; Pandya, K. Intent detection and slots prompt in a closed-domain chatbot. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)* (2019), pp. 340–343.

[40] Pak, A.; Paroubek, P. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* (Valletta, Malta, May 2010), European Language Resources Association (ELRA).

[41] Pascanu, R.; Mikolov, T.; Bengio, Y. Understanding the exploding gradient problem. *CoRR abs/1211.5063* (2012).

[42] Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, Oct. 2014), ACL, pp. 1532–1543.

[43] Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In *Proc. of NAACL* (2018).

[44] Prabhavalkar, R.; Rao, K.; Sainath, T. N.; Li, B.; Johnson, L.; Jaitly, N. A comparison of sequence-to-sequence models for speech recognition. In *Proc. Interspeech 2017* (2017), pp. 939–943.

[45] Ribeiro, E.; Ribeiro, R.; de Matos, D. M. A study on dialog act recognition using character-level tokenization. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications* (2018), Springer, pp. 93–103.

[46] Robertson, S.; Zaragoza, H. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr. 3*, 4 (Apr. 2009), 333–389.

[47] Rodrigues, R. C.; Rodrigues, J.; de Castro, P. V. Q.; da Silva, N. F. F.; Soares, A. Portuguese language models and word embeddings: Evaluating on semantic similarity tasks. In *Computational Processing of the Portuguese Language* (Cham, 2020), P. Quaresma, R. Vieira, S. Aluísio, H. Moniz, F. Batista, and T. Gonçalves, Eds., Springer International Publishing, pp. 239–248.

[48] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* (1958), 65–386.

[49] Ruder, S. *Neural Transfer Learning for Natural Language Processing*. Tese de Doutorado, National University of Ireland, Galway, 2019.

[50] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Learning Internal Representations by Error Propagation*. MIT Press, Cambridge, MA, USA, 1986, p. 318–362.

[51] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Learning Representations by Back-Propagating Errors*. MIT Press, Cambridge, MA, USA, 1988, p. 696–699.

[52] Salton, G.; Wong, A.; Yang, C. S. A vector space model for automatic indexing. *Commun. ACM 18*, 11 (Nov. 1975), 613–620.

[53] Saxena, S. Artificial neuron networks(basics) - introduction to neural networks. https://becominghuman.ai/artificial-neuron-networks-basics-introduction-to-neural-networks-3082f1dcca8c. Accessed: 2021-02-23.

[54] Seth, Y. Understanding the working of universal language model fine tuning (ulmfit). https://yashuseth.blog/2018/06/17/understanding-universal-language-model-fine-tuning-ulmfit/. Accessed: 2020-12-10.

[55] Shaw, G. L. Donald hebb: The organization of behavior. In *Brain Theory* (Berlin, Heidelberg, 1986), G. Palm and A. Aertsen, Eds., Springer Berlin Heidelberg, pp. 231–233.

[56] Souza, F.; Nogueira, R.; Lotufo, R. Bertimbau: Pretrained bert models for brazilian portuguese. In *Intelligent Systems* (Cham, 2020), R. Cerri and R. C. Prati, Eds., Springer International Publishing, pp. 403–417.

[57] Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res. 15*, 1 (Jan. 2014), 1929–1958.

[58] Subedi, N. Fasttext: Under the hood. https://towardsdatascience.com/fasttext-under-the-hood-11efc57b2b3. Accessed: 2021-01-19.

[59] Swayamdipta, S.; Schwartz, R.; Lourie, N.; Wang, Y.; Hajishirzi, H.; Smith, N. A.; Choi, Y. Dataset cartography: Mapping and diagnosing datasets with training dynamics, 2020.

[60] Taylor, W. L. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly 30*, 4 (1953), 415–433.

[61] Turney, P. D.; Pantel, P. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research 37* (Feb 2010), 141–188.

[62] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need, 2017.

[63] Wagner Filho, J. A.; Wilkens, R.; Idiart, M.; Villavicencio, A. The brWaC corpus: A new open resource for Brazilian Portuguese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (Miyazaki, Japan, May 2018), European Language Resources Association (ELRA).

[64] Wen, T.-H.; Miao, Y.; Blunsom, P.; Young, S. Latent intention dialogue models. In *Proc. of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 3732–3741.

[65] Werbos, P. J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Tese de Doutorado, Harvard University, 1974.

[66] Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE 78*, 10 (1990), 1550–1560.

[67] Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Łukasz Kaiser; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G.; Hughes, M.; Dean, J. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[68] Xingkun Liu, Arash Eshghi, P. S.; Rieser, V. Benchmarking natural language understanding services for building conversational agents. In *Proc. of the 10th International Workshop on Spoken Dialogue Systems Technology (IWSDS)* (2019), Springer.

[69] Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification, 2016.

[70] Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)* (Dec 2015), pp. 19–27.

APPENDIX A – Datasets Labels Distribution

A.1 NLU-Evaluation

Label	**Sentences**
music_play	1218
IOT_hue	1068
QA_factoid	973
calendar_set_event	959
email_query	887
weather_request	839
general_conversation	824
calendar_delete_event	729
news_query	729
radio_play	697
general_feedback	696
datetime_query	674
QA_definition	618
calendar_query_event	610
QA_open_query	599
email_send_email	582
social_post	581
QA_celebrity	539
podcasts_play	480
lists_query	477
transport_train	477
weather_question	439
music_preferences	416
lists_remove	403

Label	**Sentences**
reminder_set	372
game_play	279
audiobook_play	273
contacts_query	267
audio_volume	255
QA_stock	252
music_settings	252
IOT_coffee	248
general_confusion	244
general_mistake	243
alarm_set	241
IOT_cleaning	240
takeaway_query	233
social_query	229
reminder_query	228
general_joke	225
cooking_recipe	225
cooking_question	225
calendar_notification	225
music_question	220
takeaway_order	220
recommendation_events	217
datetime_question	216
recommendation_movies	216
transport_traffic	215
recommendation_locations	215
email_reply	213
general_confirmation	213
news_set_notification	210
calendar_question	210
lists_creating	209
transport_taxi	208
IOT_wemo	205

Label	**Sentences**
transport_directions	204
alarm_remove	203
audio_mute	191
QA_maths	189
alarm_query	186
datetime_convert	177
lists_adding	171

A.2 Virtual Operator

Label	**Sentences**
Genérico.Sem sinal	72762
Qualificado.Ausência de sinal	50791
Genérico.Problema com equipamento	48223
Genérico.Serviço funciona	41785
Genérico.Falar com atendente	41309
Genérico.Problema com imagem	33027
Genérico.Canal não pega	29835
Genérico.Troca de equipamento	18686
Genérico.Problema com canal	17736
Qualificado.Mudança de endereço	16715
Qualificado.Banda larga	12948
Genérico.Mudança de endereço G	12503
Genérico.Equipamento não funciona G	11592
Genérico.Problema com visita técnica	10963
Genérico.Equipamento queimado G	10531
Qualificado.NãoTéc ponto adicional	10246
Qualificado.Mudança de cômodo	9873
Qualificado.Operadora Online	9629
Qualificado.Cancelamento	9236
Qualificado.Equipamento não liga	9080
Genérico.Texto ou código na tela	8470
Qualificado.NãoTéc_plano	8415

Label	**Sentences**
Genérico.Problema Controle2	8305
Qualificado.Cabos e conectores	7791
Qualificado.Técnico não veio	7371
Qualificado.NãoTéc_fatura	7255
Genérico.Equipamento quebrado G	7203
Genérico.Canal comum não pega (G)	6000
Qualificado.Priorizar atendimento	5700
Qualificado.Código 77	5471
Qualificado.Canal PPV não está disponível	5078
Qualificado.Gravação	4983
Qualificado.Código 4	4898
Qualificado.Irritação ou Anatel	4752
Qualificado.Informações e confirmação de visita técnica	4552
Qualificado.Código 6	4525
Genérico.Mudança	4453
Qualificado.Tela preta	4305
Qualificado.Aplicativo Operadora	4044
Qualificado.Outros problemas	3647
Genérico.Canal HD não pega G	3483
Qualificado.Código 1-2-25	3291
Qualificado.Mudança de posição antena	3268
Genérico.Instalação	3234
Qualificado.Código 56	2923
Qualificado.Apenas imagem, sem áudio	2811
Qualificado.Travado no canal do cliente	2528
Genérico.Canal travado	2385
Qualificado.Equipamento liga e desliga sozinho	2277
Genérico.Não sei	2199
Qualificado.NãoTéc upgrade hd	2093
Qualificado.Guia de programação	2045
Qualificado.Guia de programação	2045
Qualificado.Programação local	1901
Genérico.Mudança de antena	1860

Label	Sentences
Qualificado.Canal fora da grade	1711
Genérico.Entendimento errado	1610
Qualificado.Agendar visita técnica	1580
Qualificado.Controle perdido	1532
Genérico.Canal Globo não pega	1521
Genérico.Problema com áudio	1506
Qualificado.NãoTéc_cadastro	1461
Genérico.Problema com senha	1453
Qualificado.TV é HD, mas receptor é SD	1339
Genérico.Problema com legenda	1259
Qualificado.Tela com chuvisco	1168
Genérico.Canal opcional não pega	1109
Qualificado.Resolvido com sinal booster	1059
Qualificado.Alterar áudio	1018
Genérico.Tela monocromática	935
Qualificado.NãoTéc_outros	928
Qualificado.Equipamento queimado	736
Genérico.Sem sinal nem código	732
Qualificado.Novo Controle Pedido	731
Qualificado.Problema tudo	723
Qualificado.Controle quebrado	689
Qualificado.Tela azul	671
Qualificado.Evento indisponível	650
Qualificado.NãoTéc_compra	647
Qualificado.NãoTéc Operadora livre	633
Qualificado.Controle não funciona para receptor	625
Qualificado.Código diagnóstico	611
Qualificado.Senha - padrão	601
Qualificado.Reset de senha padrão	577
Qualificado.Código 14	554
Qualificado.Controle não funciona para tv	534
Qualificado.Numeração nova	506
Genérico.Mudança de instalação	489

Label	**Sentences**
Qualificado.Imagem preto e branco	477
Qualificado.Equipamento travado	466
Qualificado.Código 109	463
Qualificado.Chip do equipamento	458
Qualificado.Técnico não resolveu	457
Genérico.Problema de antena	416
Qualificado.Habilitar recurso de senha	391
Genérico.Atualização de endereço G	381
Qualificado.Legenda não aparece na tela	350
Genérico.Código sim	336
Qualificado.Equipamento superaquecido	334
Qualificado.Reativar programação	298
Qualificado.Cancelar Serviço	249
Qualificado.Código 19	247
Genérico.Canal adulto não pega (G)	171
Qualificado.Cliente está longe	161
Qualificado.Procurando sinal sintonizador terrestre	153
Qualificado.Legenda incorreta	151
Qualificado.Recarga	133
Qualificado.Código 13	113
Qualificado.Equipamento com ruído	107
Qualificado.Ausência sinal geral	106
Qualificado.Criar senha padrão	99
Genérico.Problema com troca de canal	94
Qualificado.Atualização crítica de endereço	86
Qualificado.Travado exceto 200	85
Genérico.Promessa de oferta	64
Qualificado.Código 9	46
Qualificado.Lentidão trocar canal	43
Qualificado.Ativar closed caption	33
Qualificado.Áudio atrasado	32
Genérico.Problema com closed caption	20
Qualificado.Msg carregando conteúdo	14

Label	Sentences
Qualificado.Número da OS	11

A.3 Mercado Livre

Label	Sentences
CAR_SEAT_COVERS	942
AUTOMOTIVE_SHIFT_LEVER_KNOBS	938
CAR_ANTENNAS	934
FOOTBALL_SHIRTS	921
SURVEILLANCE_CAMERAS	909
VIDEO_GAMES	908
WALLPAPERS	885
WRISTWATCHES	876
SUNGLASSES	875
CARPETS	857
HANDBAGS	850
DOLLS	843
BOOKS	834
LIGHT_BULBS	829
RAM_MEMORY_MODULES	822
JACKETS_AND_COATS	815
MOBILE_DEVICE_CHARGERS	804
ACTION_FIGURES	800
PANTS	799
COMPUTER_PROCESSORS	794
AUTOMOTIVE_WEATHERSTRIPS	788
ELECTRIC_GUITARS	778
DIGITAL_VOICE_RECORDERS	774
ENGINE_OILS	770
MUSICAL_KEYBOARD_CASES_AND_BAGS	760
T_SHIRTS	742
FISHING_REELS	740
EYESHADOWS	738
AUTOMOTIVE_SIDE_VIEW_MIRRORS	737

Label	**Sentences**
FOOTBALL_SHOES	734
TELEVISIONS	728
SPARK_PLUGS	726
SMARTWATCHES	724
AUTOMOTIVE_MOLDINGS	720
CAR_WHEELS	720
AUTOMOTIVE_CLUTCH_KITS	719
MOTORCYCLE_HELMETS	715
HAIR_CLIPPERS	714
DECORATIVE_VINYLS	709
FOUNDATIONS	706
PUREBRED_DOGS	705
COMPUTER_MONITORS	701
BACKPACKS	699
PEDAL_EFFECTS	695
DRESSES	692
STUFFED_TOYS	679
DESKTOP_COMPUTER_POWER_SUPPLIES	679
CELL_BATTERIES	670
MEMORY_CARDS	669
WALLETS	665
AUTOMOTIVE_AMPLIFIERS	662
BOARD_GAMES	654
DRONES	653
TABLETS	624
GAMEPADS_AND_JOYSTICKS	612
FLASHLIGHTS	606
DIECAST_VEHICLES	606
FANS	597
STOOLS	594
CAR_AV_RECEIVERS	589
ROLLER_SKATES	586
FISHING_LINES	584

Label	**Sentences**
SUITCASES	582
SUSPENSION_BALL_JOINTS	579
COFFEE_MAKERS	567
LIPSTICKS	562
CAMERA_BATTERIES	554
MOTORCYCLE_JACKETS	551
BABY_CAR_SEATS	550
WHEELS_BEARINGS	547
TV_AND_MONITOR_MOUNTS	546
TABLECLOTHS	544
NOTEBOOKS	543
PARKING_SENSORS	539
AUTOMOTIVE_SIDE_VIEW_MIRROR_GLASSES	536
CALCULATORS	536
COMICS	534
MAKEUP_BRUSHES	534
MATTRESSES	534
VEHICLE_STICKERS	530
SPEAKERS	525
REFRIGERATORS	524
AUTOMOTIVE_EMBLEMS	524
BATHROOM_FAUCETS	521
MUSICAL_KEYBOARDS	511
WOMEN_SWIMWEAR	500
PORTABLE_CELLPHONE_CHARGERS	500
ARTIFICIAL_FLOWERS	497
OUTER_TIE_ROD_ENDS	496
KITCHEN_POTS	485
WALL_CLOCKS	480
HOVERBOARDS	480
SPORT_WATCHES	478
CEILING_LIGHTS	472
BABY_STROLLERS	468

Label	**Sentences**
BASS_GUITARS	467
MICROPHONES	465
FLOOD_LIGHTS	462
ANALOG_CAMERAS	457
DEEP_FRYERS	455
BLENDERS	452
DVD_RECORDERS	450
RANGES	450
INSTRUMENT_AMPLIFIERS	449
SHAVING_MACHINES	449
FREEZERS	444
CV_JOINTS	443
SCULPTURES	437
IRONS	433
KITCHEN_FAUCETS	431
SUPPLEMENTS	430
ROOF_RACKS	428
BATHROOM_SINKS	427
CAMERA_TRIPODS	427
MALE_UNDERWEAR	422
VEHICLE_SPEAKERS	421
STREAMING_MEDIA_DEVICES	415
ENGINE_CONTROL_MODULES	415
ELECTRIC_DRILLS	414
COOKING_SCALES	412
HOME_APPLIANCE_CONTACTORS_AND_RELAYS	408
REAR_WHEEL_HUBS_BEARING_ASSEMBLY	401
PRINTERS	400
WATER_RADIATORS	399
AUTOMOTIVE_WATER_PUMPS	391
AM_FM_RADIOS	383
SOLDERING_MACHINES	383
DRINKING_GLASSES	382

Label	**Sentences**
BODY_SKIN_CARE_PRODUCTS	380
FABRICS	379
BABY_DIAPERS	375
ENGINE_INTAKE_HOSES	373
WRENCHES	371
CAR_POWER_STEERING_PUMPS	370
ELECTRIC_SAWS	363
SERVING_AND_HOME_TRAYS	354
STARTERS	354
AIR_COMPRESSORS	353
MOTORCYCLE_FAIRINGS	353
VR_HEADSETS	352
MIXERS	350
VEHICLE_BRAKE_PADS	350
GAME_CONSOLES	347
BRACELETS_AND_ANKLE_BRACES	346
DESKTOP_COMPUTER_COOLERS_AND_FANS	345
ELECTRIC_PRESSURE_WASHERS	343
FACIAL_SKIN_CARE_PRODUCTS	341
AUTOMOTIVE_DOORS	332
MUGS	329
CELLPHONES	329
ENGINE_BEARINGS	327
PLANTS	313
AUDIO_AMPLIFIERS	311
ACCORDIONS	311
TV_ANTENNAS	310
KITCHEN_RANGE_HOODS	309
AUDIO_INTERFACES	300
GATE_MOTORS	298
GLASSES_FRAMES	296
ALARMS_AND_SENSORS	295
BODYWEIGHT_SCALES	293

Label	**Sentences**
FOG_LIGHTS	292
SEWING_MACHINES	292
SANDER_MACHINES	292
CD_AND_DVD_PLAYERS	291
EMERGENCY_LIGHTS	288
CAMERA_CHARGERS	285
WALKIE_TALKIES	283
TV_SMPS	279
EROTIC_CREAMS	279
KITCHEN_TOWELS	277
COSTUMES	276
KEYBOARD_AND_MOUSE_KITS	275
DECORATIVE_VASES	274
SHORTS	274
OPERATING_SYSTEMS	270
TURNTABLES	270
CAR_GEARBOXES	267
WHISKEYS	267
TOOTHBRUSHES	265
WATCH_BANDS	265
TABLE_AND_DESK_LAMPS	260
AUTOMOTIVE_SUSPENSION_CONTROL_ARMS	254
BAR_CODE_SCANNERS	254
MARTIAL_ARTS_AND_BOXING_GLOVES	254
KITCHEN_SINKS	253
ADHESIVE_TAPES	253
ELECTRICAL_CABLES	250
TOILET_RUGS	249
TOY_BUILDING_SETS	249
WATER_HEATERS	247
INTERACTIVE_GAMING_FIGURES	246
AIRSOFT_GUNS	245
BUMPER_IMPACT_ABSORBERS	245

Label	**Sentences**
CABIN_FILTERS	244
CAR_STEREOS	243
CRIBS	242
MOTORCYCLE_CLUTCH_COVERS	239
SHOWER_HEADS	235
HOME_HEATERS	234
ULTRABOOKS	232
SPORT_AND_BAZAAR_BOTTLES	229
HEADBOARDS	228
WATER_DISPENSERS	225
MOTORCYCLE_CASES	222
MOTORCYCLE_TURN_SIGNAL_LIGHTS	221
ANTI_THEFT_STUDS	220
BABY_MONITORS	220
CAMERA_LENSES	219
LED_STAGE_LIGHTS	216
HABERDASHERY_RIBBONS	216
AQUARIUM_FILTERS	214
CUSHIONS	213
DRUMS	212
ELECTRONIC_ENTRANCE_INTERCOMS	211
REMOTE_CONTROL_TOY_VEHICLES	209
POSTERS	206
CAR_DISTRIBUTOR_CAPS	205
KITCHEN_KNIVES	202
DJ_EFFECTS_PROCESSORS	202
SIDEBOARDS	201
MOUSE_PADS	201
DRAWERS	198
TOILET_SEATS	191
WINDSHIELD_WIPERS	190
GRAPHICS_TABLETS	190
NETWORK_CABLES	190

Label	**Sentences**
LIP_GLOSSES	190
HEADPHONES	189
ALL_IN_ONE	189
AV_RECEIVERS	188
DISPOSABLE_CUPS	188
BINOCULARS	186
TRAILER_HITCHES	184
BICYCLES	184
PENCIL_CASES	183
WINES	182
RESISTANCE_BANDS	180
BLANK_DISCS	178
BLU_RAY_PLAYERS	175
DJ_CONTROLLERS	175
HUMIDIFIERS_AND_VAPORIZERS	173
WALL_LIGHTS	172
OVENS	171
BEERS	170
HOOKAHS	168
FACE_MASKS	166
TACTICAL_AND_SPORTING_KNIVES_AND_BLADES	164
INDOOR_CURTAINS_AND_BLINDS	161
CATS_AND_DOGS_FOODS	161
CYCLING_COMPUTERS	161
HAIRDRESSING_SCISSORS	160
HOME_SHELVES	157
CACHACAS	157
FOOTBALL_BALLS	156
SCREEN_PRINTERS	156
CELLPHONE_TABLET_AND_GPS_SCREEN_PROTECTORS	155
AIR_MATTRESSES	154
SKIRTS	154
NOTEBOOKS_AND_WRITING_PADS	153

Label	**Sentences**
SOUVENIRS	151
VIOLINS	151
PICTURE_FRAMES	151
MOTORCYCLE_GLOVES	151
AUTOMOTIVE_DOOR_PANELS	150
HOME_OFFICE_DESKS	149
CIRCUIT_BREAKERS	149
DISHWASHERS	148
LUMBAR_AND_ABDOMINAL_BRACES	147
MOTORCYCLE_TIRES	147
FURNITURE_KNOBS	147
LATHES	146
FISH_TANKS	145
NETBOOKS	145
MAGAZINES	143
STEERING_COLUMNS	143
DRILL_BITS	143
YARNS	142
CONTINUOUS_INK_SYSTEMS	140
ABS_SENSORS	140
PENDRIVES	139
BRAKE_BOOSTERS	139
AUTOMOTIVE_POWER_WINDOW_REGULATORS	138
SUSPENSION_CONTROL_ARM_BUSHINGS	138
PORTABLE_EVAPORATIVE_AIR_COOLERS	138
FOOD_PROCESSORS	135
NECKTIES	134
ENGINE_PISTONS	132
PORTABLE_GENERATORS	132
AUTOMOTIVE_HEADLIGHTS	131
NAIL_DRYERS	131
BLANKETS	131
AUTOMOTIVE_WHEEL_COVERS	129

Label	**Sentences**
KNEE_BRACES_SUPPORTS	129
FLEA_AND_TICK_TREATMENTS	129
FLUTES	128
ELLIPTICAL_MACHINES	128
CRIB_BEDDING_SETS	127
MARKERS_AND_HIGHLIGHTERS	126
AUTOMOTIVE_FENDERS	124
TORSION_BARS	123
MODEMS	123
ELECTRIC_SCREWDRIVERS	122
ELECTRICAL_OUTLETS	122
MAGNIFYING_GLASSES	122
BABY_SWIMWEAR	121
AUTOMOTIVE_SPRING_SUSPENSIONS	121
COMPUTER_AND_TV_FLEX_CABLES	120
AUTOMOTIVE_TRUNK_LIDS	118
SHIRTS	118
SWEATSHIRTS_AND_HOODIES	118
MOTORCYCLE_PANTS	118
ELECTRONIC_DRUMS	117
VIBRATORS	117
TV_REPLACEMENT_BACKLIGHT_LED_STRIPS	115
RACKS_AND_PINIONS	113
HANDICRAFT_BOXES	112
BRUSH_CUTTERS	110
BABY_PLAYARDS	110
TOOL_BOXES	109
HABERDASHERY_LACE_EDGINGS	109
ENGINE_CRANKSHAFT_PULLEYS	107
JUMPSUITS_AND_OVERALLS	106
GUITAR_STRINGS	106
PARTY_DECORATIVE_BACKDROPS	105
BOOTS	104

Label	**Sentences**
FUEL_INJECTORS	103
DIAPER_BAGS	103
HORSE_SADDLES	102
CRAYONS	101
THERMOSES	100
BABY_BOTTLES	100
SUBMERSIBLE_PUMPS	100
KITCHEN_PLAYSETS	100
SOFAS	100
LIQUORS	100
SOFA_AND_FUTON_COVERS	99
CAR_AIR_FRESHENERS	99
SWAY_BARS	98
OFFICE_CHAIRS	98
DOORS	97
VESTS	97
PAINTBALL_MARKERS	97
CAR_AC_CONDENSERS	96
BABIES_FOOTWEAR	96
EPILATORS	94
VODKAS	94
HEAT_GUNS	94
TOY_TRAINS	94
PERMANENT_EPILATORS	94
CLEANING_CLOTHS	93
MASCARAS	92
CLOTHES_HANGERS	92
CAMERAS	92
CELLPHONE_AND_TABLET_CASES	91
GAZEBOS	91
NECKLACES	91
ROUTERS	90
WHEELCHAIRS	90

Label	**Sentences**
SOCKS	89
EROTIC_PUMPS	89
LIQUID_HAND_AND_BODY_SOAPS	89
HAIR_TREATMENTS	89
SWAY_BAR_LINKS	89
BABY_HIGH_CHAIRS	88
BABIES_FORMULA	88
CONCEALERS	88
POWERED_RIDE_ON_TOYS	86
CHARMS_AND_MEDALS	86
MIRRORS	86
PAINTBALLS	85
MOTORCYCLE_BATTERIES	85
ANIMAL_CLIPPERS	85
COMBUSTION_CHAINSAWS	84
ENGINE_COOLING_FAN_SHROUDS	83
KEYCHAINS	83
SEWING_THREADS	83
HAND_AND_FOOT_CREAMS	83
PORTABLE_ELECTRIC_MASSAGERS	83
OFFICE_SOFTWARE	82
HAMMOCKS	82
DATA_CABLES_AND_ADAPTERS	80
DJ_TURNTABLES	80
AUTOMOTIVE_TIRES	80
SLATWALL_PANELS	80
BATHROOM_ACCESSORIES_SETS	78
TELEVISION_MAIN_PLATE_REPLACEMENTS	78
MULTIGAME_MACHINES	77
SWIMMING_GOGGLES	77
COOKIES_CUTTERS	77
ORTHOPEDIC_WRIST_BRACES	77
CUSHION_COVERS	76

Label	**Sentences**
BABY_CLOTHING_SETS	76
IDLER_ARMS	75
BED_SHEETS	75
LENS_FILTERS	75
DISC_PACKAGINGS	74
PUZZLES	74
STATIONARY_BICYCLES	74
MOTORCYCLE_JERSEYS	74
INDUSTRIAL_AND_COMMERCIAL_SCALES	73
SHOCK_MOUNT_INSOLATORS	73
SNEAKERS	72
INTEGRATED_CIRCUITS	72
CRASHED_CARS	71
MOVIES	71
VINYL_ROLLS	70
PARTY_MASKS	70
MICRO_ROTARY_TOOLS	69
VEHICLE_CV_AXLES	69
ENGINE_VALVES_SPRING_RETAINERS	68
LAPTOP_CHARGERS	68
UMBRELLAS	68
VIDEO_GAME_PREPAID_CARDS	68
TABLE_RUNNERS	67
GYM_GLOVES	67
LATEX_ENAMEL_AND_ACRYLIC_PAINTS	67
EROTIC_BOOKS	67
FISHING_LURES	67
AUTOMOTIVE_SHOCK_ABSORBERS	66
SAFETY_FOOTWEAR	66
ESSENTIAL_OILS	65
TREADMILLS	65
HATS_AND_CAPS	65
UPS_BATTERIES	64

Label	**Sentences**
FUEL_INJECTION_RAILS	64
ENGINE_CYLINDER_HEAD_BOLTS	64
THERMOMETERS	63
WELDING_MASKS	63
TOOTHPASTES	62
PARKING_BRAKE_HANDLES	61
PUZZLE_CUBES	61
STRING_TRIMMERS	60
BAR_SOAPS	60
DISHES_PLATES	59
CLOTHING_PATCHES	58
SCREWS	57
DOG_CARRIERS_AND_CARRYING_BAGS	57
GARDEN_HOSES	56
LONGBOARDS	56
GLOW_PLUG_CONTROLLERS	56
THERMAL_CUPS_AND_TUMBLERS	56
INDUSTRIAL_BLENDERS	56
PAJAMAS	56
AUTOMOTIVE_AIR_FILTERS	55
LASER_MEASURES	55
AUTOMOTIVE_ARMRESTS	55
CELLPHONE_COVERS	54
MICROMETERS	54
TRANSISTORS	54
PROJECTOR_SCREENS	53
LAPTOP_LCD_SCREENS	52
TV_STORAGE_UNITS	52
RICE_COOKERS	52
UKULELES	52
MOTORCYCLE_RAIN_SUITS	52
PERFUMES	52
DINING_SETS	52

Label	**Sentences**
PAPER_CLIPS	51
ENGINE_INTAKE_MANIFOLDS	51
CELLPHONE_REPLACEMENT_CAMERAS	51
EMBROIDERY_MACHINES	51
BODY_SHAPERS	50
EYELINERS	50
AUTOMOTIVE_THROTTLE_BODIES	50
WASHING_AND_DRYER_MACHINE_COVERS	50
DISHES_RACKS	49
SELF_ADHESIVE_LABELS	49
NEBULIZERS	49
CAMERA_MONOPODS	49
BELTS	49
PANTIES	49
ALTERNATORS	48
TABLE_DRILLS	48
SAFES	48
LUGGAGE_TAGS	47
3D_PRINTERS	47
CARDS_AND_INVITATIONS	47
BIRD_TOYS	46
UNIVERSAL_HOME_GYMS	45
TRADING_CARD_GAMES	45
SANDALS_AND_FLIP_FLOPS	45
CAKE_STANDS	45
DECORATIVE_BASKETS	44
EARRINGS	44
ENGINE_VALVES	44
PADLOCKS	44
HAIR_SHAMPOOS_AND_CONDITIONERS	44
AUTOMOTIVE_AC_COMPRESSORS	43
BEDS	43
AUTOMOTIVE_OIL_FILTERS	43

Label	**Sentences**
RINGS	43
FRAME_POOLS	43
STICKY_NOTES	43
AIR_FRESHENERS	43
STYLUSES	42
CURLING_IRONS	42
AIRBAGS	42
HARD_DRIVES_AND_SSDS	42
EROTIC_BALLS	42
BABY_SAFETY_LOCKS	41
COMPUTER_MOTHERBOARDS	41
TOILETRY_BAGS	40
BARBECUE_TOOL_SETS	40
IRRIGATION_VALVES	40
GAS_LIFT_SUPPORTS	39
MANGA	39
HEARING_PROTECTORS	39
JUMP_ROPES	39
HOSPITAL_BEDS	39
CELLPHONE_BATTERIES	38
WORKOUT_BENCHES	38
CASH_DRAWERS	38
EROTIC_MALE_UNDERWEAR	38
AUTOMOTIVE_NERF_BARS	38
STIMULATING_PILLS_AND_CAPSULES	38
KITCHEN_FURNITURE	38
BLOUSES	38
ELECTRONIC_MUSCLE_STIMULATORS	38
EXTERNAL_LAPTOP_COOLERS	38
POOL_INFLATABLES	37
SIDE_TABLES	37
CHRISTMAS_TREES	37
AUTOMOTIVE_SHOCK_ABSORBER_BUMP_STOPS	37

Label	**Sentences**
MOTORCYCLE_SUITS	37
VIDEO_CAMERAS	37
VASES	37
DESKTOP_COMPUTERS	36
STETHOSCOPES	36
GARDENING_AND_AGRICULTURE_SEEDS	36
WARDROBES	36
DINING_CHAIRS	36
DIFFERENTIALS	35
ENGINE_TAPPET_GUIDE_HOLDS	35
CONTINUOUS_LIGHTING	35
BOOKCASES	35
BUTT_PLUGS	34
SAXOPHONES	34
DENTAL_PLIERS	34
SUITS	33
TEQUILAS	33
SEX_TOY_KITS	32
POWER_STEERING_FLUID_RESERVOIRS	32
LAPTOP_BATTERIES	32
SPARK_PLUG_WIRESETS	32
GRAPHICS_CARDS	32
PUSH_AND_RIDING_TOYS	32
COMMERCIAL_LIGHT_SIGNS	32
MUSIC_STANDS	32
VIDEO_CAPTURE_DEVICES	32
HAND_FANS	32
CAR_WINDOW_SWITCHES	31
PILLOWS	31
CHAMPAGNES	31
FOOD_CARTS	31
SUNSCREENS	31
DECORATIVE_BOXES	31

Label	**Sentences**
NETWORK_CARDS	31
FLATWARE_SETS	31
INK_CARTRIDGES	30
PLAYING_CARDS	30
BLOOD_PRESSURE_MONITORS	30
FIRE_EXTINGUISHERS	30
PLACEMATS	30
BATTERY_CHARGERS	30
CLUTCH_SLAVE_CYLINDERS	30
COLLECTIBLE_CANS_BOTTLES_AND_SODA_SIPHONS	30
LEGGINGS	30
HEEL_CUPS	29
VOLTAGE_DETECTORS	29
LASER_PRINTER_DRUMS	29
FOOTBALL_JACKETS	29
SPORTS_CONES	29
MOTORCYCLE_IGNITION_COILS	29
NIGHTSTANDS	29
BABY_BLANKETS	29
THERMAL_REFRIGERATORS_AND_BAGS	28
IP_TELEPHONES	28
SPICE_RACKS	28
FOLDERS_AND_EXPANDING_FILES	28
MIRROR_BALLS	28
HAND_BRAKE_CABLES	28
LAPTOP_KEYBOARDS	27
ARTIFICIAL_PLANTS	27
PENS	27
CONDOMS	27
BABY_BODYSUITS	27
KITCHEN_APRONS	27
TOILETS	27
PC_KEYBOARDS	26

Label	**Sentences**
COIN_PURSES	26
HAIR_STRAIGHTENING_BRUSHES	26
COFFEE_TABLES	26
PACKAGING_ROLLS	26
BATHROOM_GRAB_BARS	26
SOLDERING_STATIONS	26
AUTOMOTIVE_MANUAL_TRANSMISSION_SHIFT_LEVERS	25
ELECTRIC_BATHROOM_FAUCETS	25
GIFT_CARDS	25
TOILET_PAPER_HOLDERS	25
INTERCOOLER_HOSES	24
SIM_CARDS	24
DRILLS_SCREWDRIVERS	24
DRIVE_SHAFTS	24
DRUM_PEDALS	24
NON_CORRECTIVE_CONTACT_LENSES	24
BEER_DISPENSERS	24
FINGERPRINT_READERS	24
PREAMPLIFIERS	23
WORLD_GLOBES	23
KIDS_TABLES_AND_CHAIRS_SETS	23
CHALKBOARD_AND_WHITEBOARD_ERASERS	23
WINDOWS	23
SECURITY_SEALS	23
LABEL_MAKERS	23
AIR_CONDITIONERS	23
STATUES	23
PERSONAL_LUBRICANTS_AND_GELS	23
BABY_STERILIZERS	22
LUNCHBOXES	22
CALIPERS	22
FOOD_SLICERS	22
KITCHEN_BOWLS	22

Label	**Sentences**
LIFE_JACKETS	22
BEAUTY_WIGS	22
CUT_OFF_AND_GRINDING_WHEELS	22
POOL_COVERS	22
ELECTRIC_GRILLS	21
MOTORCYCLE_FENDERS	21
MOTORCYCLE_CRASH_BARS	21
HEATER_CORES	21
VEHICLE_BRAKE_DISCS	21
EGR_VALVES	21
FOOTBALL_CAPS	21
CRANKSHAFTS	21
SWIMMING_POOL_HEATERS	21
TELEPHONES	21
SANDPAPERS	21
DRINK_PITCHERS	21
WATER_PURIFIERS_FILTERS	20
XENON_KITS	20
COMFORTERS	20
ENGINE_CRANKSHAFT_POSITION_SENSORS	20
SAFETY_GOGGLES	20
MDF_BOARDS	20
FISHING_VESTS	20
INDUSTRIAL_ICE_CREAM_MACHINES	20
INSTANT_COFFEE	20
WETSUITS	19
VEHICLE_LED_BULBS	19
VACUUM_TUBES	19
CATS	19
LOAFERS_AND_OXFORDS	19
FABRIC_SOFTENERS	19
MOTORCYCLE_DISTRIBUTION_CHAINS	19
SOLAR_PANELS	19

Label	**Sentences**
STEAM_CLEANERS	19
FISHING_RODS	19
MEN_SWIMWEAR	18
BABY_BOUNCERS	18
CELLPHONE_REPAIR_TOOL_KITS	18
BILLIARD_TABLES	18
VIBRATION_PLATFORMS	18
HAIR_STRAIGHTENERS	18
AUTOMOBILE_FENDER_LINERS	18
ELECTRIC_DEMOLITION_HAMMERS	18
TV_RECEIVERS_AND_DECODERS	18
NOTEBOOK_CASES	17
CAR_AC_HOSE_ASSEMBLIES	17
CARD_PAYMENT_TERMINALS	17
WASTE_BASKETS	17
HAND_FILES	17
BEDROOM_SETS	17
VARNISHES	17
MAP_SENSORS	17
ALTERNATOR_PULLEYS	17
BRAKE_LIGHTS	17
GUITAR_PICKS	17
ENGINE_GASKET_SETS	17
TOY_GARAGES_AND_GAS_STATIONS	17
EROTIC_MAGAZINES	16
MARKING_AND_WARNING_TAPES	16
FOOTBALL_GOALKEEPER_GLOVES	16
VACUUM_CLEANERS	16
ANTIVIRUS_AND_INTERNET_SECURITY	16
ORTHOTICS	16
POOL_LIGHTS	16
BEDLINERS	16
CAMERA_BATTERY_GRIPS	16

Label	**Sentences**
HONEY	16
EMBROIDERY_DESIGNS	16
BAR_CLAMPS	16
DINING_TABLES	16
ORTHOPEDIC_ANKLE_BRACES	15
JEWELRY_DISPLAYS	15
FLOUR	15
CAR_ENGINE_CAMSHAFTS	15
CAT_SCRATCHERS	15
BASKETBALL_JERSEYS	15
SCALEXTRIC_CARS	15
HAIR_DRYERS	15
PILATES_BALLS	15
BABY_PACIFIERS	15
MALE_MASTURBATORS	15
EQUALIZERS	15
TOY_ROBOTS	15
CAR_LIGHT_BULBS	14
ENGINE_COOLING_FAN_MOTORS	14
GARDEN_BENCHES	14
PET_COLLARS	14
MINI_PCS	14
SCREEN_PRINTING_MACHINES	14
IGNITION_SWITCH_ACTUATORS	14
HEDGE_TRIMMERS	14
DISTRIBUTION_KITS	14
HAND_POLISHERS	14
ORTHOPEDIC_WALKER_BOOTS	14
TELEPHONE_CABLES	14
CATS_AND_DOGS_TREATS	14
LIVING_ROOM_SETS	14
PIPES_AND_TUBES	13
NETWORK_SWITCHES	13

Label	**Sentences**
BABY_WALKERS	13
CERAMIC_TILES	13
CAR_DOOR_HINGES	13
POOL_WATERFALLS	13
BICYCLE_FRAMES	13
TACTICAL_VESTS	13
TREADMILL_RUNNING_BELTS	13
MICROWAVES	13
PNEUMATIC_STAPLERS	13
KATANA_SWORDS	13
INDUSTRIAL_DOUGH_KNEADERS	13
PLAYGROUND_SLIDES	13
RUBBER_FLOORS	13
POWER_GRINDERS	13
AUTOMOTIVE_MIRROR_COVERS	12
SOAP_HOLDERS	12
PENCILS	12
SPARKLING_WINES	12
KIDS_WALKIE_TALKIES	12
SCOOTERS	12
SHADE_CLOTHS	12
CATS_LITTER	12
GARAGE_DOORS	12
POOL_PUMPS	12
WASHING_MACHINES	12
WASTE_CONTAINERS	12
BRAKE_MASTER_CYLINDERS	12
FLOOR_LAMPS	11
AUTOMOTIVE_TRANSMISSION_GEARS	11
FITNESS_TRAMPOLINES	11
PAINT_ROLLERS	11
COOKTOPS	11
RADIO_FREQUENCY_MICROPHONES	11

Label	**Sentences**
SUNBATHING_CHAIRS	11
SKIN_REPELLENTS	11
MATE_GOURDS	11
TENTS	11
BREAST_FEEDING_PILLOWS	11
WINE_CELLARS	11
KITCHEN_MOLDS	10
POWER_STRIPS	10
OUTDOOR_TABLES	10
OSCILLOSCOPES	10
VEHICLE_CLUTCH_CABLES	10
SALT	10
CAR_SCREENS	10
MEDICAL_WALKERS	10
CAN_OPENERS	10
DOG_LEASHES	10
BRAKE_DRUMS	10
AB_ROLLER_WHEELS	10
HEARING_AIDS	10
TEA	10
SOLID_SWEET_PASTES	10
SCHOOL_AND_OFFICE_GLUES	10
POUFS	10
MINI_COMPONENT_SYSTEMS	10
TV_REMOTE_CONTROLS	9
HOME_THEATERS	9
GPS	9
LAPTOP_BRIEFCASES	9
BOX_SPRING_AND_MATTRESS_SETS	9
PENIS_SLEEVES	9
TOWEL_HOLDERS	9
FISHES	9
DEHUMIDIFIERS	9

Label	**Sentences**
VEGETABLES_AND_FRUITS_CHOPPERS	9
ACOUSTIC_PANELS	9
GARDEN_SOIL	9
DRUM_BRAKE_SHOES	9
PADDLE_TENNIS_RACKETS	9
LINGERIE_SETS	9
CARABINERS	9
INFLATABLE_POOLS	9
ELBOW_SUPPORTS	9
ISOPROPYL_ALCOHOLS	9
VEHICLE_BRAKE_HYDRAULIC_HOSES	9
NAPKIN_HOLDERS	9
BICYCLE_PEDALS	9
POPCORN_MACHINES	9
GOLF_CLUBS_SETS	9
PORTABLE_DVD_PLAYERS	9
MEGAPHONES	9
LAWN_MOWER_BLADES	9
AUTOMOTIVE_CLUTCH_MASTER_CYLINDERS	8
CLEANING_SPONGES	8
ELECTRIC_AIR_PUMPS	8
CYMBALS	8
DRONE_BATTERIES	8
AIRBRUSHES	8
EXHAUST_MANIFOLDS	8
BATHROOM_VANITIES	8
ORAL_IRRIGATORS	8
FREEZER_BAGS	8
AUDIO_AND_VIDEO_CABLES_AND_ADAPTERS	8
MAKEUP_VANITIES	8
TOY_PLANES	8
COMPOSTERS	8
MERCHANDISER_REFRIGERATORS	8

Label	**Sentences**
DIVING_MASKS	8
LASER_POINTERS	8
PHOTO_ALBUMS	8
TABLE_CLOCKS	8
HOOD_HINGES	8
MOUTHWASHES	8
HAMMER_DRILLS	8
STRAWS	8
TORQUE_WRENCHES	8
SWEETENERS	8
PLUNGE_ROUTERS	8
STOVETOP_POPCORN_POPPERS	8
WAFFLE_MAKERS	8
ESPADRILLES	8
DRYER_MACHINES	8
PARTY_HATS	8
HAIRDRESSING_CAPS	8
CUPCAKE_STANDS	8
PATIO_FURNITURE_SETS	8
SCHOOL_AND_OFFICE_PAPERS	8
DILDOS	8
LASER_LEVELS	8
KITCHEN_CABINET_ORGANIZERS	7
DOG_BEDS	7
ENERGETIC_STONES	7
ANTIQUE_CHAIRS	7
SAFETY_HELMETS	7
VINYL_FLOORINGS	7
COTTON_CANDY_MACHINES	7
HOLE_PUNCHES	7
CAMERA_CASES	7
MOTORCYCLE_CHEST_PROTECTORS	7
ELECTRIC_BLOWERS	7

Label	**Sentences**
INFLATABLE_SOFAS	7
BICYCLE_AND_MOTORCYCLE_ALARMS	7
ECT_SENSORS	7
ELECTRIC_HAND_PLANERS	7
FETAL_DOPPLERS	7
BALL_PIT_BALLS	7
LIGHT_STANDS	7
VARIABLE_FREQUENCY_DRIVES	7
CAMERA_REPLACEMENT_DISPLAYS	7
ELECTROLYTIC_CAPACITORS	7
IGNITION_CONTROL_MODULES	7
LAMINATORS	7
AUTOMOTIVE_CV_JOINT_BOOTS	7
DRUM_STANDS	7
WOOD_BURNING_MACHINES	7
TANDEM_CHAIRS	7
ICE_BUCKETS	7
JEWELRY_BOXES	6
COAT_RACKS	6
KNITTING_NEEDLES	6
PINBALLS	6
CHOCOLATE_WATERFALLS	6
CAR_CENTER_CONSOLES	6
ENGINE_COOLING_FAN_SWITCHES	6
MICRODERMABRASION_MACHINES	6
CAR_SCANNERS	6
SNARE_DRUMS	6
LAPTOP_HOUSINGS	6
RACQUETS	6
BABY_GYMS	6
MULTIMETERS	6
TABLE_TENNIS_TABLES	6
MAGNETIC_WELDING_HOLDERS	6

Label	**Sentences**
MOTORCYCLE_LEVERS	6
CYCLING_HELMETS	6
POWER_STEERING_HOSES	6
LAUNDRY_BASKETS	6
RADIO_BASE_STATIONS	6
WHEEL_STUDS	6
STAPLERS	6
BABY_JUMPERS	6
SAFETY_GLOVES	6
VIDEO_CASSETTES	6
DRONE_PROPELLERS	6
ARCHERY_BOWS	6
HAND_SAWS	6
MAGNETIC_COMPASSES	6
AUTOMOTIVE_SEATS	6
GAUZES	6
ELECTRICAL_TIMERS	6
CUTTING_BOARDS	6
AUTOMOTIVE_CELLPHONE_AND_GPS_MOUNTS	6
BICYCLE_WHEELS	6
FLATWARE_ORGANIZERS	6
APERITIFS	5
INDUSTRIAL_PULLEYS	5
JUICERS	5
MOTORCYCLE_CARBURETORS	5
PROJECTOR_MOUNTS	5
TELESCOPES	5
SHOE_RACKS	5
BEER_FAUCETS	5
DOLLHOUSES	5
PAPER_SHREDDERS	5
KITES	5
BASEBALL_AND_SOFTBALL_BATS	5

Label	**Sentences**
PORCELAIN_TILES	5
REFLECTIVE_VESTS	5
VEHICLE_TRACKERS	5
AUTOMOTIVE_DEFLECTORS	5
ELECTRIC_SHOWER_HEADS	5
YOGURT_MAKERS	5
POOL_CLEANERS	5
KITCHEN_GRATERS	5
POTENTIOMETERS	5
COFFEE_CAPSULES	5
BABY_PACIFIER_CLIPS	5
DEODORANTS	5
BILL_COUNTERS	5
AUTOMOTIVE_BATTERIES	5
MENSTRUAL_CUPS	5
RUBBER_STAMPS	5
CAMERA_FLASHES	5
SOUND_CARDS	5
BICYCLE_HANDLEBARS	5
WIRELESS_ANTENNAS	5
KEYBOARD_CONTROLLERS	5
FANNY_PACKS	4
MOTORCYCLE_SPEEDOMETERS	4
SLEEPING_BAGS	4
LAMP_HOLDERS	4
KIDS_TRICYCLES	4
MAKEUP_TRAIN_CASES	4
SHOWER_CURTAINS	4
SPHYGMOMANOMETERS	4
KEY_RACKS	4
WALL_ANCHOR_PLUGS	4
STEPPERS	4
ELECTRIC_LAWN_MOWERS	4

Label	**Sentences**
RECEPTION_DESKS	4
KITCHEN_MORTARS	4
TROLLEY_AND_FURNITURE_CASTERS	4
TABLET_KEYBOARDS	4
ENGINE_COOLING_FAN_CLUTCHES	4
AXES	4
DENTAL_CHAIRS	4
VIDEOCASSETTE_PLAYERS	4
RUM	4
HARMONICAS	4
UNIVERSAL_CAR_REMOTES	4
PUPPETS	4
CRUTCHES	4
GROOVE_JOINT_PLIERS	4
HAND_TRUCKS	4
SAFETY_HARNESSES	4
SYRINGES	4
OTOSCOPES	4
AUDIO_AND_VIDEO_CONNECTORS	4
CHIP_AND_DIP_SERVERS	4
AIRGUN_PELLETS	4
MOTORCYCLE_TRANSMISSION_CROWNS	4
MUSIC_ALBUMS	4
SCREEN_PRINTING_KITS	4
ELECTRICITY_METERS	4
MASSAGE_SOFAS	4
LED_STRIPS	4
STORE_SHOPPING_CARTS	4
TRUMPETS	4
GINS	4
PENIS_RINGS	4
MEDICINE_BALLS	4
GATE_GEAR_RACKS	4

Label	**Sentences**
AUTOMOTIVE_BUMPER_GRILLES	3
EDIBLE_SEEDS	3
SELF_TANNERS	3
MONEY_BOXES	3
CHESTS	3
DESKTOP_COMPUTER_CASES	3
COMPRESSION_SLEEVES	3
RICE	3
MEAT_GRINDERS	3
PAINTBALL_O_RINGS	3
TENNIS_BALLS	3
MANUAL_HAMMERS	3
EROTIC_ANAL_AND_VAGINAL_DOUCHES	3
CLUTCH_FORKS	3
CLUTCH_BEARINGS	3
CAMERA_STRAPS	3
TURNTABLE_NEEDLES	3
MOTORCYCLE_GRAB_BARS	3
CAMERA_AND_CELLPHONE_STABILIZERS	3
BREAD_MAKERS	3
LINEMAN_PLIERS	3
PUNCHING_BAGS	3
SCREWDRIVERS_SETS	3
AFTERSHAVES	3
AIRBAG_MODULES	3
HAND_BLENDERS	3
CEREAL_BARS	3
MICROWAVE_KEYPADS	3
CAR_HOODS	3
SODS	3
METAL_DETECTORS	3
ELECTRIC_CHAINSAWS	3
ENGINE_OIL_PRESSURE_SENSORS	3

Label	**Sentences**
BICYCLE_SEATS	3
VOLLEYBALL_BALLS	3
HOME_BOTTLE_STANDS	3
CNC_LATHES	3
UNIVERSAL_REMOTE_CONTROLS	3
DOOR_AND_WINDOW_LOCKS	3
DISPOSABLE_GLOVES	3
MEMORY_CARD_READERS	3
DRIED_FRUITS	2
STABILIZERS_AND_UPS	2
COUNTERFEIT_MONEY_DETECTOR_MACHINE	2
MEAT_HOOKS	2
SHIN_GUARDS	2
READY_TO_DRINK_COCKTAILS	2
BASKET_BALLS	2
SWIMMING_NOSE_CLIPS	2
NECK_GAITERS_MASKS_AND_BALACLAVAS	2
SANDWICH_MAKERS	2
DENTAL_FLOSSES	2
DOG_NAIL_CLIPPERS	2
SWIMMING_EARPLUGS	2
TOOTHBRUSH_HOLDERS	2
STYLING_CHAIRS	2
BINDING_SPINES	2
DIGITAL_WEATHER_STATIONS	2
BOXING_HEADGEARS	2
CAR_FRONT_MASKS	2
DOORBELLS	2
TABLE_TENNIS_BALLS	2

APPENDIX B – Sparse Vector (BOW) Per Class Performances With and Without Stop-Words

B.1 NLU-Evaluation

ID	Class	F1-Score		Suport	%
		With Stop-words	Without Stop-words		
50	datetime_question	0.526	0.55	43	4.56%
45	alarm_remove	0.639	0.667	40	4.38%
34	alarm_query	0.575	0.600	37	4.35%
31	IOT_wemo	0.927	0.950	41	2.48%
49	lists_creating	0.825	0.843	42	2.18%
27	lists_adding	0.783	0.794	34	1.40%
48	social_post	0.916	0.928	116	1.31%
6	cooking_question	0.525	0.529	45	0.76%
35	recommendation_movies	0.563	0.563	43	0.00%
39	transport_traffic	0.843	0.843	43	0.00%
40	audiobook_play	0.865	0.865	55	0.00%
43	IOT_coffee	0.949	0.949	50	0.00%
51	QA_stock	0.900	0.900	50	0.00%
58	IOT_cleaning	0.936	0.936	48	0.00%
61	transport_taxi	0.962	0.962	41	0.00%
11	transport_train	0.878	0.876	95	-0.23%
36	calendar_delete_event	0.872	0.870	146	-0.23%
56	recommendation_events	0.621	0.619	43	-0.32%
23	IOT_hue	0.974	0.968	214	-0.62%

ID	Class	F1-Score		Suport	%
		With Stop-words	Without Stop-words		
16	lists_query	0.851	0.845	95	-0.71%
1	transport_directions	0.605	0.600	41	-0.83%
14	QA_factoid	0.777	0.769	195	-1.03%
63	datetime_convert	0.800	0.788	35	-1.50%
25	email_reply	0.889	0.875	43	-1.57%
3	radio_play	0.858	0.844	139	-1.63%
44	podcasts_play	0.854	0.840	96	-1.64%
4	lists_remove	0.876	0.859	81	-1.94%
8	general_joke	0.933	0.913	45	-2.14%
30	game_play	0.877	0.857	56	-2.28%
5	news_query	0.774	0.755	146	-2.45%
47	reminder_query	0.692	0.675	46	-2.46%
37	social_query	0.776	0.753	46	-2.96%
19	calendar_set_event	0.825	0.798	192	-3.27%
20	weather_request	0.731	0.705	168	-3.56%
33	music_settings	0.574	0.553	50	-3.66%
15	email_query	0.930	0.893	177	-3.98%
60	calendar_query_event	0.609	0.582	122	-4.43%
29	music_play	0.802	0.765	244	-4.61%
18	recommendation_locations	0.742	0.706	43	-4.85%
41	email_send_email	0.883	0.840	116	-4.87%
2	cooking_recipe	0.705	0.667	45	-5.39%
46	alarm_set	0.752	0.708	48	-5.85%
62	reminder_set	0.524	0.491	74	-6.30%
7	contacts_query	0.768	0.718	53	-6.51%
53	calendar_question	0.557	0.518	42	-7.00%
55	news_set_notification	0.553	0.514	42	-7.05%
24	datetime_query	0.864	0.801	135	-7.29%
26	QA_maths	0.769	0.711	38	-7.54%
57	music_preferences	0.689	0.636	83	-7.69%
42	general_feedback	0.739	0.680	139	-7.98%

ID	Class	F1-Score		Suport	%
		With Stop-words	Without Stop-words		
13	music_question	0.695	0.638	44	-8.20%
12	weather_question	0.620	0.563	88	-9.19%
21	QA_definition	0.863	0.779	124	-9.73%
22	takeaway_query	0.911	0.822	47	-9.77%
28	QA_celebrity	0.804	0.721	108	-10.32%
59	general_confirmation	0.519	0.464	43	-10.60%
54	audio_volume	0.720	0.633	51	-12.08%
38	takeaway_order	0.854	0.750	44	-12.18%
9	audio_mute	0.753	0.659	38	-12.48%
17	general_conversation	0.497	0.433	165	-12.88%
52	general_confusion	0.653	0.563	49	-13.78%
0	calendar_notification	0.388	0.319	45	-17.78%
32	general_mistake	0.500	0.378	49	-24.40%
10	QA_open_query	0.409	0.295	120	-27.87%

B.2 Virtual Operator

ID	Class	F1-Score		Support	%
		With Stop-words	Without Stop-words		
103	Qualificado.Número da OS	0.000	0.000	2	-
114	Qualificado.Ativar closed caption	0.000	0.000	6	-
115	Genérico.Promessa de oferta	0.000	0.222	13	-
111	Qualificado.Código 13	0.829	0.955	23	15.20%
118	Qualificado.Lentidão trocar canal	0.571	0.625	8	9.46%
12	Qualificado.Controle não funciona para tv	0.695	0.731	107	5.18%
116	Qualificado.Ausência sinal geral	0.581	0.611	21	5.16%
91	Qualificado.Recarga	0.720	0.750	27	4.17%
107	Genérico.Problema com troca de canal	0.320	0.333	19	4.06%
43	Qualificado.Controle não funciona para operadora	0.624	0.640	125	2.56%

ID	Class	F1-Score		Support	%
		With Stop-words	Without Stop-words		
105	Qualificado.Habilitar recurso de senha	0.815	0.835	78	2.45%
93	Qualificado.Equipamento queimado	0.707	0.724	147	2.40%
109	Qualificado.Procurando sinal sintonizador terrestre	0.875	0.892	31	1.94%
85	Qualificado.Código 14	0.941	0.955	111	1.49%
79	Qualificado.Código 19	0.947	0.958	49	1.16%
30	Qualificado.Equipamento liga e desliga sozinho	0.813	0.819	455	0.74%
60	Qualificado.Código diagnóstico	0.960	0.967	122	0.73%
86	Qualificado.Guia de programação	0.962	0.968	409	0.62%
67	Qualificado.Senha - padrão	0.761	0.765	120	0.53%
75	Qualificado.Código 109	0.952	0.957	93	0.53%
17	Qualificado.NãoTéc_outros	0.795	0.799	186	0.50%
26	Qualificado.Controle perdido	0.844	0.848	306	0.47%
3	Genérico.Equipamento não funciona G	0.902	0.906	2318	0.44%
70	Genérico.Problema de antena	0.852	0.855	83	0.35%
32	Qualificado.Equipamento não liga	0.932	0.935	1816	0.32%
47	Qualificado.Código 56	0.972	0.975	585	0.31%
54	Qualificado.Informações e confirmação de visita técnica	0.868	0.870	910	0.23%
95	Qualificado.NãoTéc_cadastro	0.769	0.770	292	0.13%
92	Qualificado.Novo Controle Pedido	0.821	0.822	146	0.12%
72	Genérico.Canal travado	0.851	0.852	477	0.12%
49	Genérico.Problema Controle2	0.894	0.895	1661	0.11%
104	Genérico.Atualização de endereço G	0.932	0.933	76	0.11%
71	Qualificado.Aplicativo Operadora	0.993	0.994	809	0.10%
28	Genérico.Mudança de endereço G	0.984	0.984	2501	0.00%
41	Qualificado.Tela preta	0.841	0.841	861	0.00%
55	Qualificado.Código 4	0.832	0.832	980	0.00%
57	Qualificado.Chip do equipamento	0.963	0.963	92	0.00%
62	Genérico.Entendimento errado	0.727	0.727	322	0.00%
65	Qualificado.Atualização crítica de endereço	0.727	0.727	17	0.00%
89	Genérico.Sem sinal nem código	0.551	0.551	146	0.00%

ID	Class	F1-Score		Support	%
		With Stop-words	Without Stop-words		
101	Qualificado.Código 9	0.750	0.750	9	0.00%
119	Qualificado.Msg carregando conteúdo	1.000	1.000	3	0.00%
120	Genérico.Problema com closed caption	1.000	1.000	4	0.00%
18	Qualificado.Banda larga	0.985	0.984	2590	-0.10%
20	Qualificado.Ausência de sinal	0.978	0.977	10158	-0.10%
9	Genérico.Equipamento queimado G	0.965	0.964	2106	-0.10%
13	Genérico.Falar com atendente	0.963	0.962	8262	-0.10%
94	Genérico.Problema com senha	0.993	0.991	291	-0.20%
21	Qualificado.Cabos e conectores	0.981	0.979	1558	-0.20%
56	Qualificado.Agendar visita técnica	0.901	0.899	316	-0.22%
78	Genérico.Canal opcional não pega	0.721	0.719	222	-0.28%
80	Genérico.Mudança de antena	0.959	0.956	372	-0.31%
59	Qualificado.Irritação ou Anatel	0.943	0.940	950	-0.32%
5	Qualificado.Cancelamento	0.909	0.906	1847	-0.33%
88	Qualificado.TV é HD. mas equip é SD	0.800	0.797	268	-0.38%
1	Genérico.Instalação	0.966	0.962	647	-0.41%
22	Qualificado.Técnico não veio	0.876	0.872	1474	-0.46%
83	Qualificado.Reset de senha padrão	0.829	0.825	115	-0.48%
100	Qualificado.Problema tudo	0.818	0.814	145	-0.49%
77	Genérico.Mudança de instalação	0.995	0.990	98	-0.50%
24	Qualificado.Mudança de endereço	0.950	0.945	3343	-0.53%
84	Qualificado.Controle quebrado	0.689	0.685	138	-0.58%
51	Qualificado.Reativar programação	0.959	0.952	60	-0.73%
40	Qualificado.Código 1-2-25	0.831	0.824	658	-0.84%
35	Qualificado.Gravação	0.933	0.924	997	-0.96%
82	Qualificado.NãoTéc Op livre	0.920	0.911	127	-0.98%
16	Genérico.Troca de equipamento	0.949	0.939	3737	-1.05%
58	Qualificado.Equipamento superaquecido	0.884	0.874	67	-1.13%
52	Qualificado.Código 77	0.840	0.829	1094	-1.31%
34	Qualificado.Programação local	0.853	0.841	380	-1.41%
33	Qualificado.NãoTéc_plano	0.780	0.768	1683	-1.54%
36	Qualificado.Operadora Online	0.906	0.892	1926	-1.55%
48	Qualificado.Priorizar atendimento	0.817	0.804	1140	-1.59%

ID	Class	F1-Score		Support	%
		With Stop-words	Without Stop-words		
8	Qualificado.Mudança de cômodo	0.931	0.916	1975	-1.61%
19	Genérico.Mudança	0.958	0.942	891	-1.67%
42	Qualificado.Travado no canal do cliente	0.718	0.706	506	-1.67%
7	Qualificado.NãoTéc_fatura	0.829	0.815	1451	-1.69%
2	Genérico.Canal não pega	0.880	0.865	5967	-1.70%
102	Qualificado.Imagem preto e branco	0.807	0.793	95	-1.73%
46	Qualificado.Evento indisponível	0.850	0.835	130	-1.76%
0	Genérico.Operadora não funciona	0.902	0.886	8357	-1.77%
73	Qualificado.Resolvido com sinal booster	0.646	0.634	212	-1.86%
63	Genérico.Não sei	0.748	0.734	440	-1.87%
45	Qualificado.Canal PPV não está disponível	0.896	0.877	1016	-2.12%
68	Qualificado.Tela azul	0.844	0.826	134	-2.13%
39	Genérico.Texto ou código na tela	0.800	0.782	1694	-2.25%
98	Qualificado.Canal fora da grade	0.793	0.775	342	-2.27%
4	Genérico.Sem sinal	0.924	0.901	14552	-2.49%
14	Genérico.Problema com canal	0.936	0.911	3547	-2.67%
10	Genérico.Problema com equipamento	0.946	0.920	9645	-2.75%
50	Genérico.Canal HD não pega G	0.816	0.792	697	-2.94%
99	Genérico.Problema com legenda	0.960	0.930	252	-3.12%
38	Genérico.Problema com visita técnica	0.993	0.960	2193	-3.32%
6	Qualificado.Outros problemas	0.718	0.694	729	-3.34%
27	Qualificado.Tela com chuvisco	0.836	0.808	234	-3.35%
31	Qualificado.Código 6	0.843	0.814	905	-3.44%
37	Genérico.Canal comum não pega (G)	0.761	0.734	1200	-3.55%
29	Genérico.Problema com imagem	0.928	0.893	6605	-3.77%
23	Qualificado.NãoTéc ponto adicional	0.829	0.796	2049	-3.98%
69	Qualificado.Mudança de posição antena	0.838	0.799	654	-4.65%
15	Qualificado.Técnico não resolveu	0.532	0.506	91	-4.89%
61	Genérico.Tela monocromática	0.769	0.726	187	-5.59%
110	Qualificado.Equipamento com ruído	0.778	0.722	21	-7.20%

ID	Class	F1-Score		Support	%
		With Stop-words	Without Stop-words		
44	Qualificado.Apenas imagem. sem áudio	0.839	0.773	562	-7.87%
66	Genérico.Problema com áudio	0.927	0.850	301	-8.31%
113	Qualificado.Legenda não aparece na tela	0.828	0.753	70	-9.06%
53	Qualificado.NãoTéc upgrade hd	0.874	0.793	419	-9.27%
106	Genérico.Canal adulto não pega (G)	0.765	0.694	34	-9.28%
74	Qualificado.Legenda incorreta	0.710	0.643	30	-9.44%
87	Qualificado.Numeração nova	0.696	0.622	101	-10.63%
25	Genérico.Canal Globo não pega	0.525	0.468	304	-10.86%
96	Qualificado.Criar senha padrão	0.645	0.571	20	-11.47%
81	Qualificado.NãoTéc_compra	0.765	0.664	129	-13.20%
11	Genérico.Equipamento quebrado G	0.859	0.734	1441	-14.55%
117	Qualificado.Áudio atrasado	0.667	0.500	6	-25.04%
76	Qualificado.Equipamento travado	0.494	0.353	93	-28.54%
108	Qualificado.Cliente está longe	0.348	0.244	32	-29.89%
64	Genérico.Código sim	0.449	0.213	67	-52.56%
112	Qualificado.Travado exceto 200	0.250	0.091	17	-63.60%
90	Qualificado.Cancelar Operadora	0.356	0.098	50	-72.47%

APPENDIX C – Class Performance Comparison - BERT and BERT + TAPT

C.1 NLU-Evaluation

ID	Class	F1-Score		Support	%
		BERT	**BERT + TAPT**		
34	alarm_query	0.727	0.800	37	10.04%
1	transport_directions	0.576	0.629	41	9.20%
18	recommendation_locations	0.713	0.764	43	7.15%
37	social_query	0.804	0.857	46	6.59%
38	takeaway_order	0.753	0.800	44	6.24%
47	reminder_query	0.675	0.716	46	6.07%
30	game_play	0.857	0.895	56	4.43%
62	reminder_set	0.521	0.537	74	3.07%
33	music_settings	0.624	0.642	50	2.88%
60	calendar_query_event	0.674	0.693	122	2.82%
53	calendar_question	0.725	0.744	42	2.62%
39	transport_traffic	0.867	0.889	43	2.54%
56	recommendation_events	0.641	0.654	43	2.03%
51	QA_stock	0.902	0.920	50	2.00%
11	transport_train	0.868	0.885	95	1.96%
29	music_play	0.785	0.800	244	1.91%
14	QA_factoid	0.786	0.798	195	1.53%
3	radio_play	0.827	0.839	139	1.45%
35	recommendation_movies	0.563	0.571	43	1.42%
9	audio_mute	0.769	0.779	38	1.30%
61	transport_taxi	0.937	0.949	41	1.28%
49	lists_creating	0.818	0.828	42	1.22%
50	datetime_question	0.659	0.667	43	1.21%
48	social_post	0.934	0.945	116	1.18%
43	IOT_coffee	0.942	0.951	50	0.96%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
52	general_confusion	0.712	0.717	49	0.70%
55	news_set_notification	0.521	0.524	42	0.58%
21	QA_definition	0.874	0.879	124	0.57%
36	calendar_delete_event	0.882	0.887	146	0.57%
16	lists_query	0.886	0.890	95	0.45%
23	IOT_hue	0.977	0.979	214	0.20%
5	news_query	0.735	0.736	146	0.14%
31	IOT_wemo	0.925	0.925	41	0.00%
27	lists_adding	0.825	0.825	34	0.00%
22	takeaway_query	0.909	0.909	47	0.00%
8	general_joke	0.923	0.923	45	0.00%
4	lists_remove	0.874	0.874	81	0.00%
12	weather_question	0.720	0.719	88	-0.14%
15	email_query	0.936	0.933	177	-0.32%
19	calendar_set_event	0.859	0.856	192	-0.35%
63	datetime_convert	0.722	0.718	35	-0.55%
24	datetime_query	0.849	0.844	135	-0.59%
40	audiobook_play	0.837	0.832	55	-0.60%
17	general_conversation	0.637	0.633	165	-0.63%
25	email_reply	0.857	0.850	43	-0.82%
45	alarm_remove	0.761	0.754	40	-0.92%
54	audio_volume	0.759	0.752	51	-0.92%
46	alarm_set	0.842	0.833	48	-1.07%
57	music_preferences	0.675	0.667	83	-1.19%
41	email_send_email	0.905	0.894	116	-1.22%
26	QA_maths	0.769	0.759	38	-1.30%
6	cooking_question	0.606	0.596	45	-1.65%
58	IOT_cleaning	0.957	0.938	48	-1.99%
59	general_confirmation	0.430	0.421	43	-2.09%
44	podcasts_play	0.874	0.851	96	-2.63%
42	general_feedback	0.791	0.770	139	-2.65%
10	QA_open_query	0.482	0.469	120	-2.70%
20	weather_request	0.805	0.780	168	-3.11%
2	cooking_recipe	0.744	0.716	45	-3.76%
28	QA_celebrity	0.826	0.794	108	-3.87%
7	contacts_query	0.800	0.763	53	-4.63%
32	general_mistake	0.608	0.569	49	-6.41%
13	music_question	0.711	0.639	44	-10.13%
0	calendar_notification	0.415	0.352	45	-15.18%

C.2 Virtual Operator

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
114	Qualificado.Ativar closed caption	0.000	0.000	6	-
103	Qualificado.Número da OS	0.000	0.000	2	-
107	Genérico.Problema com troca de canal	0.485	0.588	19	21.24%
91	Qualificado.Recarga	0.808	0.926	27	14.60%
105	Qualificado.Habilitar recurso de senha	0.836	0.919	78	9.93%
106	Genérico.Canal adulto não pega (G)	0.761	0.824	34	8.28%
87	Qualificado.Numeração nova	0.874	0.931	101	6.52%
90	Qualificado.Cancelar Operadora	0.739	0.769	50	4.06%
17	Qualificado.NãoTéc_outros	0.852	0.886	186	3.99%
70	Genérico.Problema de antena	0.869	0.903	83	3.91%
110	Qualificado.Equipamento com ruído	0.895	0.927	21	3.58%
117	Qualificado.Áudio atrasado	0.429	0.444	6	3.50%
102	Qualificado.Imagem preto e branco	0.869	0.899	95	3.45%
95	Qualificado.NãoTéc_cadastro	0.878	0.904	292	2.96%
65	Qualificado.Atualização crítica de endereço	0.778	0.800	17	2.83%
54	Qualificado.Inf. e conf. visita técnica	0.927	0.950	910	2.48%
12	Qualificado.Controle não func. p/ tv	0.869	0.889	107	2.30%
79	Qualificado.Código 19	0.938	0.959	49	2.24%
53	Qualificado.NãoTéc upgrade hd	0.914	0.933	419	2.08%
27	Qualificado.Tela com chuvisco	0.889	0.907	234	2.02%
104	Genérico.Atualização de endereço G	0.948	0.967	76	2.00%
83	Qualificado.Reset de senha padrão	0.881	0.897	115	1.82%
62	Genérico.Entendimento errado	0.882	0.897	322	1.70%
75	Qualificado.Código 109	0.979	0.995	93	1.63%
7	Qualificado.NãoTéc_fatura	0.921	0.936	1451	1.63%
22	Qualificado.Técnico não veio	0.926	0.941	1474	1.62%
47	Qualificado.Código 56	0.976	0.989	585	1.33%
57	Qualificado.Chip do equipamento	0.951	0.963	92	1.26%
63	Genérico.Não sei	0.876	0.887	440	1.26%
41	Qualificado.Tela preta	0.909	0.920	861	1.21%
46	Qualificado.Evento indisponível	0.917	0.928	130	1.20%
113	Qualificado.Legenda não aparece na tela	0.925	0.936	70	1.19%
82	Qualificado.NãoTéc Op livre	0.951	0.962	127	1.16%
31	Qualificado.Código 6	0.936	0.946	905	1.07%
56	Qualificado.Agendar visita técnica	0.957	0.966	316	0.94%
42	Qualificado.Travado no canal do cliente	0.855	0.863	506	0.94%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
45	Qualificado.Canal PPV não está disponível	0.955	0.963	1016	0.84%
6	Qualificado.Outros problemas	0.885	0.892	729	0.79%
34	Qualificado.Programação local	0.925	0.932	380	0.76%
5	Qualificado.Cancelamento	0.965	0.972	1847	0.73%
33	Qualificado.NãoTéc_plano	0.891	0.897	1682	0.67%
109	Qualificado.Procurando sinal sint. terrestre	0.906	0.912	31	0.66%
52	Qualificado.Código 77	0.928	0.934	1094	0.65%
58	Qualificado.Equipamento superaquecido	0.956	0.962	67	0.63%
19	Genérico.Mudança	0.972	0.978	891	0.62%
67	Qualificado.Senha - padrão	0.872	0.877	120	0.57%
36	Qualificado.Operadora Online	0.950	0.955	1926	0.53%
40	Qualificado.Código 1-2-25	0.949	0.953	658	0.42%
43	Qualificado.Controle não funciona para op	0.805	0.808	125	0.37%
100	Qualificado.Problema tudo	0.940	0.943	145	0.32%
23	Qualificado.NãoTéc ponto adicional	0.942	0.945	2049	0.32%
55	Qualificado.Código 4	0.953	0.956	980	0.31%
16	Genérico.Troca de equipamento	0.978	0.981	3737	0.31%
21	Qualificado.Cabos e conectores	0.987	0.990	1558	0.30%
80	Genérico.Mudança de antena	0.988	0.991	372	0.30%
71	Qualificado.Aplicativo Operadora	0.993	0.996	809	0.30%
61	Genérico.Tela monocromática	0.872	0.874	187	0.23%
44	Qualificado.Apenas imagem. sem áudio	0.941	0.943	562	0.21%
0	Genérico.Operadora não funciona	0.960	0.962	8357	0.21%
86	Qualificado.Guia de programação	0.980	0.982	409	0.20%
32	Qualificado.Equipamento não liga	0.981	0.983	1815	0.20%
1	Genérico.Instalação	0.993	0.995	647	0.20%
94	Genérico.Problema com senha	0.993	0.995	291	0.20%
98	Qualificado.Canal fora da grade	0.926	0.927	342	0.11%
97	Qualificado.Alterar áudio	0.942	0.943	204	0.11%
35	Qualificado.Gravação	0.977	0.978	997	0.10%
24	Qualificado.Mudança de endereço	0.984	0.985	3343	0.10%
10	Genérico.Problema com equipamento	0.991	0.992	9645	0.10%
20	Qualificado.Ausência de sinal	0.993	0.994	10158	0.10%
38	Genérico.Problema com visita técnica	0.998	0.999	2193	0.10%
4	Genérico.Sem sinal	0.974	0.974	14552	0.00%
13	Genérico.Falar com atendente	0.985	0.985	8262	0.00%
2	Genérico.Canal não pega	0.952	0.952	5967	0.00%
14	Genérico.Problema com canal	0.980	0.980	3547	0.00%
9	Genérico.Equipamento queimado G	0.988	0.988	2106	0.00%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
49	Genérico.Problema Controle2	0.958	0.958	1661	0.00%
50	Genérico.Canal HD não pega G	0.934	0.934	697	0.00%
77	Genérico.Mudança de instalação	0.995	0.995	98	0.00%
111	Qualificado.Código 13	0.978	0.978	23	0.00%
120	Genérico.Problema com closed caption	1.000	1.000	4	0.00%
119	Qualificado.Msg carregando conteúdo	1.000	1.000	3	0.00%
28	Genérico.Mudança de endereço G	0.995	0.994	2501	-0.10%
18	Qualificado.Banda larga	0.988	0.987	2590	-0.10%
69	Qualificado.Mudança de posição antena	0.921	0.920	654	-0.11%
68	Qualificado.Tela azul	0.896	0.895	134	-0.11%
37	Genérico.Canal comum não pega (G)	0.892	0.891	1200	-0.11%
99	Genérico.Problema com legenda	0.986	0.984	252	-0.20%
29	Genérico.Problema com imagem	0.982	0.980	6605	-0.20%
8	Qualificado.Mudança de cômodo	0.977	0.975	1975	-0.20%
11	Genérico.Equipamento quebrado G	0.981	0.978	1441	-0.31%
3	Genérico.Equipamento não funciona G	0.975	0.972	2318	-0.31%
74	Qualificado.Legenda incorreta	0.900	0.897	30	-0.33%
72	Genérico.Canal travado	0.930	0.926	477	-0.43%
92	Qualificado.Novo Controle Pedido	0.925	0.921	146	-0.43%
96	Qualificado.Criar senha padrão	0.923	0.919	20	-0.43%
116	Qualificado.Ausência sinal geral	0.833	0.829	21	-0.48%
26	Qualificado.Controle perdido	0.973	0.966	306	-0.72%
88	Qualificado.TV é HD. mas equip é SD	0.937	0.930	268	-0.75%
39	Genérico.Texto ou código na tela	0.918	0.911	1694	-0.76%
59	Qualificado.Irritação ou Anatel	0.985	0.977	950	-0.81%
51	Qualificado.Reativar programação	0.984	0.976	60	-0.81%
66	Genérico.Problema com áudio	0.969	0.961	301	-0.83%
48	Qualificado.Priorizar atendimento	0.958	0.950	1140	-0.84%
85	Qualificado.Código 14	0.987	0.977	111	-1.01%
30	Qualificado.Equipamento liga e desliga sozinho	0.954	0.942	455	-1.26%
25	Genérico.Canal Globo não pega	0.789	0.777	304	-1.52%
60	Qualificado.Código diagnóstico	1.000	0.984	122	-1.60%
93	Qualificado.Equipamento queimado	0.905	0.889	147	-1.77%
78	Genérico.Canal opcional não pega	0.865	0.847	222	-2.08%
64	Genérico.Código sim	0.874	0.855	67	-2.17%
81	Qualificado.NãoTéc_compra	0.925	0.900	129	-2.70%
73	Qualificado.Resolvido com sinal booster	0.869	0.845	212	-2.76%
76	Qualificado.Equipamento travado	0.760	0.728	93	-4.21%
89	Genérico.Sem sinal nem código	0.819	0.772	146	-5.74%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
101	Qualificado.Código 9	1.000	0.941	9	-5.90%
118	Qualificado.Lentidão trocar canal	0.632	0.588	8	-6.96%
15	Qualificado.Técnico não resolveu	0.662	0.611	91	-7.70%
108	Qualificado.Cliente está longe	0.462	0.426	32	-7.79%
84	Qualificado.Controle quebrado	0.857	0.786	138	-8.28%
115	Genérico.Promessa de oferta	0.267	0.235	13	-11.99%
112	Qualificado.Travado exceto 200	0.629	0.552	17	-12.24%

C.3 Mercado Livre

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
693	COUNTERFEIT_MONEY_DETECTOR_MACHINE	0.000	0.667	2	-
954	READY_TO_DRINK_COCKTAILS	0.000	0.000	2	-
973	AUDIO_AND_VIDEO_CONNECTORS	0.000	0.000	4	-
1001	HAND_BLENDERS	0.000	0.333	3	-
1025	ELECTRICITY_METERS	0.000	0.000	4	-
1029	SWIMMING_EARPLUGS	0.000	0.000	2	-
1035	ENGINE_OIL_PRESSURE_SENSORS	0.000	0.000	3	-
999	IGNITION_CONTROL_MODULES	0.222	0.600	7	170.27%
584	TABLET_KEYBOARDS	0.333	0.667	4	100.30%
928	KEYBOARD_CONTROLLERS	0.154	0.308	5	100.00%
569	MONEY_BOXES	0.400	0.667	3	66.75%
913	WHEEL_STUDS	0.182	0.286	6	57.14%
581	FLATWARE_ORGANIZERS	0.444	0.667	6	50.23%
930	CAMERA_AND_CELLPHONE_STABILIZERS	0.667	1.000	3	49.93%
910	DILDOS	0.353	0.522	8	47.88%
982	RUBBER_STAMPS	0.615	0.889	5	44.55%
893	POWER_STRIPS	0.571	0.800	10	40.11%
844	CHOCOLATE_WATERFALLS	0.667	0.909	6	36.28%
576	SHOE_RACKS	0.500	0.667	5	33.40%
612	JEWELRY_BOXES	0.500	0.667	6	33.40%
862	LAUNDRY_BASKETS	0.500	0.667	6	33.40%
912	UNIVERSAL_REMOTE_CONTROLS	0.500	0.667	3	33.40%
802	GPS	0.714	0.941	9	31.79%
1031	STEPPERS	0.571	0.750	4	31.35%
867	HAMMER_DRILLS	0.545	0.714	8	31.01%
572	FOOTBALL_JACKETS	0.488	0.630	29	29.10%
205	STATUES	0.450	0.578	23	28.44%
856	HOOD_HINGES	0.556	0.714	8	28.42%
736	VACUUM_TUBES	0.421	0.529	19	25.65%
564	TELESCOPES	0.727	0.909	5	25.03%
671	CAR_SCANNERS	0.727	0.909	6	25.03%
849	LAPTOP_BRIEFCASES	0.700	0.875	9	25.00%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
485	RECEPTION_DESKS	0.800	1.000	4	25.00%
1010	MICROWAVE_KEYPADS	0.800	1.000	3	25.00%
1026	METAL_DETECTORS	0.800	1.000	3	25.00%
970	DEODORANTS	0.667	0.833	5	24.89%
1036	ICE_BUCKETS	0.667	0.833	7	24.89%
940	COAT_RACKS	0.462	0.571	6	23.59%
723	INFLATABLE_SOFAS	0.750	0.923	7	23.07%
876	BATHROOM_VANITIES	0.714	0.875	8	22.55%
168	GARDEN_BENCHES	0.621	0.759	14	22.22%
870	LAWN_MOWER_BLADES	0.778	0.941	9	20.95%
977	CHIP_AND_DIP_SERVERS	0.333	0.400	4	20.12%
837	SOAP_HOLDERS	0.500	0.600	12	20.00%
684	SNARE_DRUMS	0.667	0.800	6	19.94%
939	LINEMAN_PLIERS	0.667	0.800	3	19.94%
358	PUSH_AND_RIDING_TOYS	0.600	0.714	32	19.00%
917	GINS	0.750	0.889	4	18.53%
770	ELECTRICAL_TIMERS	0.769	0.909	6	18.21%
513	LIFE_JACKETS	0.732	0.864	22	18.03%
188	PREAMPLIFIERS	0.571	0.667	23	16.81%
969	KEY_RACKS	0.571	0.667	4	16.81%
1012	DRUM_STANDS	0.714	0.833	7	16.67%
933	CRANKSHAFTS	0.800	0.930	21	16.25%
374	PIPES_AND_TUBES	0.581	0.667	13	14.80%
943	SAFETY_GLOVES	0.250	0.286	6	14.40%
763	AUTOMOTIVE_TRANSMISSION_GEARS	0.700	0.800	11	14.29%
985	BICYCLE_PEDALS	0.875	1.000	9	14.29%
512	TV_RECEIVERS_AND_DECODERS	0.737	0.842	18	14.25%
332	COOKTOPS	0.609	0.692	11	13.63%
518	CLUTCH_SLAVE_CYLINDERS	0.735	0.831	30	13.06%
998	LAMP_HOLDERS	0.444	0.500	4	12.61%
717	AXES	0.889	1.000	4	12.49%
748	BRAKE_MASTER_CYLINDERS	0.786	0.880	12	11.96%
739	MOTORCYCLE_FENDERS	0.800	0.895	21	11.88%
587	LAPTOP_KEYBOARDS	0.877	0.981	27	11.86%
592	KITCHEN_MOLDS	0.632	0.706	10	11.71%
843	CYCLING_HELMETS	0.600	0.667	6	11.17%
353	PROJECTOR_MOUNTS	0.800	0.889	5	11.13%
715	DEHUMIDIFIERS	0.800	0.889	9	11.13%
938	POOL_CLEANERS	0.800	0.889	5	11.13%
703	IRRIGATION_VALVES	0.714	0.791	40	10.78%
942	WATER_PURIFIERS_FILTERS	0.552	0.611	20	10.69%
842	SCOOTERS	0.833	0.917	12	10.08%
848	ELBOW_SUPPORTS	0.818	0.900	9	10.02%
812	MOTORCYCLE_LEVERS	0.909	1.000	6	10.01%
901	PLUNGE_ROUTERS	0.800	0.875	8	9.37%
983	DRYER_MACHINES	0.857	0.933	8	8.87%
846	MAP_SENSORS	0.867	0.941	17	8.54%
465	BABY_BOUNCERS	0.810	0.878	18	8.40%
885	MATE_GOURDS	0.846	0.917	11	8.39%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
628	CUTTING_BOARDS	0.923	1.000	6	8.34%
993	HAND_SAWS	0.923	1.000	6	8.34%
490	HOLE_PUNCHES	0.769	0.833	7	8.32%
630	MOTORCYCLE_CHEST_PROTECTORS	0.714	0.769	7	7.70%
978	ARCHERY_BOWS	0.857	0.923	6	7.70%
478	CAR_AIR_FRESHENERS	0.825	0.887	99	7.52%
9	SWIMMING_POOL_HEATERS	0.909	0.976	21	7.37%
780	MARKING_AND_WARNING_TAPES	0.933	1.000	16	7.18%
808	FETAL_DOPPLERS	0.933	1.000	7	7.18%
688	SAFETY_HELMETS	0.800	0.857	7	7.12%
835	AIRBRUSHES	0.800	0.857	8	7.12%
838	INFLATABLE_POOLS	0.667	0.714	9	7.05%
841	PATIO_FURNITURE_SETS	0.667	0.714	8	7.05%
859	POWER_STEERING_HOSES	0.667	0.714	6	7.05%
735	LIQUID_HAND_AND_BODY_SOAPS	0.791	0.844	89	6.70%
716	COMPOSTERS	0.875	0.933	8	6.63%
627	KIDS_TABLES_AND_CHAIRS_SETS	0.769	0.818	23	6.37%
474	FREEZER_BAGS	0.824	0.875	8	6.19%
706	TOY_PLANES	0.824	0.875	8	6.19%
772	ACOUSTIC_PANELS	0.824	0.875	9	6.19%
813	PADDLE_TENNIS_RACKETS	0.824	0.875	9	6.19%
945	VEHICLE_BRAKE_HYDRAULIC_HOSES	0.824	0.875	9	6.19%
246	MEN_SWIMWEAR	0.778	0.824	18	5.91%
538	BAR_SOAPS	0.853	0.903	60	5.86%
925	GOLF_CLUBS_SETS	0.889	0.941	9	5.85%
948	NAPKIN_HOLDERS	0.889	0.941	9	5.85%
775	WINE_CELLARS	0.833	0.880	11	5.64%
836	CARABINERS	0.947	1.000	9	5.60%
352	ARTIFICIAL_PLANTS	0.633	0.667	27	5.37%
462	HOME_THEATERS	0.600	0.632	9	5.33%
509	AUTOMOTIVE_SHOCK_ABSORBERS	0.784	0.825	66	5.23%
755	FISHING_VESTS	0.851	0.895	20	5.17%
533	MALE_MASTURBATORS	0.815	0.857	15	5.15%
211	BATHROOM_ACCESSORIES_SETS	0.811	0.852	78	5.06%
725	VEGETABLES_AND_FRUITS_CHOPPERS	0.667	0.700	9	4.95%
309	SNEAKERS	0.914	0.959	72	4.92%
12	SPORT_AND_BAZAAR_BOTTLES	0.897	0.941	229	4.91%
677	RUBBER_FLOORS	0.917	0.960	13	4.69%
476	BAR_CLAMPS	0.857	0.897	16	4.67%
951	CAR_CENTER_CONSOLES	0.588	0.615	6	4.59%
472	BLOUSES	0.744	0.778	38	4.57%
532	EPILATORS	0.857	0.896	94	4.55%
906	GARAGE_DOORS	0.783	0.818	12	4.47%
666	ENGINE_TAPPET_GUIDE_HOLDS	0.917	0.957	35	4.36%
327	SIM_CARDS	0.898	0.936	24	4.23%
249	HAIR_TREATMENTS	0.826	0.860	89	4.12%
598	FRAME_POOLS	0.907	0.944	43	4.08%
236	THERMAL_CUPS_AND_TUMBLERS	0.883	0.919	56	4.08%
491	DJ_TURNTABLES	0.756	0.786	80	3.97%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
503	GARDENING_AND_AGRICULTURE_SEEDS	0.895	0.930	36	3.91%
571	RADIO_FREQUENCY_MICROPHONES	0.700	0.727	11	3.86%
729	ENGINE_CYLINDER_HEAD_BOLTS	0.780	0.810	64	3.85%
392	PLAYGROUND_SLIDES	0.963	1.000	13	3.84%
200	RICE_COOKERS	0.945	0.981	52	3.81%
604	IDLER_ARMS	0.876	0.909	75	3.77%
831	EMBROIDERY_DESIGNS	0.933	0.968	16	3.75%
425	ALARMS_AND_SENSORS	0.930	0.964	295	3.66%
14	FACE_MASKS	0.860	0.891	166	3.60%
526	AUTOMOTIVE_SHOCK_ABSORBER_BUMP_STOPS	0.912	0.943	37	3.40%
618	SPORTS_CONES	0.951	0.983	29	3.36%
646	HEEL_CUPS	0.951	0.983	29	3.36%
670	SOLAR_PANELS	0.872	0.900	19	3.21%
452	FLOUR	0.909	0.938	15	3.19%
622	CELLPHONE_REPAIR_TOOL_KITS	0.941	0.971	18	3.19%
929	ALTERNATOR_PULLEYS	0.941	0.971	17	3.19%
380	MIRRORS	0.936	0.965	86	3.10%
222	SOUVENIRS	0.908	0.936	151	3.08%
481	FOOD_CARTS	0.909	0.937	31	3.08%
525	SANDPAPERS	0.878	0.905	21	3.08%
768	LOAFERS_AND_OXFORDS	0.944	0.973	19	3.07%
508	ENGINE_GASKET_SETS	0.522	0.538	17	3.07%
676	SAFETY_GOGGLES	0.947	0.976	20	3.06%
767	FOOTBALL_CAPS	0.900	0.927	21	3.00%
582	MIRROR_BALLS	0.769	0.792	28	2.99%
42	HANDICRAFT_BOXES	0.805	0.829	112	2.98%
501	CARD_PAYMENT_TERMINALS	0.914	0.941	17	2.95%
807	BRAKE_LIGHTS	0.914	0.941	17	2.95%
683	AUTOMOTIVE_MIRROR_COVERS	0.833	0.857	12	2.88%
220	MOVIES	0.806	0.829	71	2.85%
365	MOTORCYCLE_TIRES	0.913	0.939	147	2.85%
624	VEHICLE_LED_BULBS	0.848	0.872	19	2.83%
414	WETSUITS	0.889	0.914	19	2.81%
805	BILLIARD_TABLES	0.889	0.914	18	2.81%
37	THERMOSES	0.863	0.887	100	2.78%
172	EYELINERS	0.863	0.887	50	2.78%
193	LASER_MEASURES	0.946	0.972	55	2.75%
97	ULTRABOOKS	0.910	0.935	232	2.75%
149	CUPCAKE_STANDS	0.556	0.571	8	2.70%
654	ANIMAL_CLIPPERS	0.893	0.917	85	2.69%
660	INDUSTRIAL_BLENDERS	0.645	0.662	56	2.64%
662	ABS_SENSORS	0.950	0.975	140	2.63%
659	ENGINE_CRANKSHAFT_POSITION_SENSORS	0.769	0.789	20	2.60%
914	FABRIC_SOFTENERS	0.923	0.947	19	2.60%
915	MOTORCYCLE_DISTRIBUTION_CHAINS	0.923	0.947	19	2.60%
535	CONCEALERS	0.936	0.960	88	2.56%
394	DIAPER_BAGS	0.937	0.961	103	2.56%
596	WORKOUT_BENCHES	0.824	0.845	38	2.55%
732	MDF_BOARDS	0.950	0.974	20	2.53%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
233	AUTOMOTIVE_AC_COMPRESSORS	0.953	0.977	43	2.52%
792	CUT_OFF_AND_GRINDING_WHEELS	0.878	0.900	22	2.51%
517	PARKING_BRAKE_HANDLES	0.960	0.984	61	2.50%
191	EMERGENCY_LIGHTS	0.942	0.965	288	2.44%
704	BEER_DISPENSERS	0.917	0.939	24	2.40%
98	ENGINE_BEARINGS	0.963	0.986	327	2.39%
66	DISPOSABLE_CUPS	0.856	0.876	188	2.34%
143	PERMANENT_EPILATORS	0.901	0.922	94	2.33%
690	BIRD_TOYS	0.944	0.966	46	2.33%
762	DRINK_PITCHERS	0.864	0.884	21	2.31%
680	WORLD_GLOBES	0.978	1.000	23	2.25%
253	FINGERPRINT_READERS	0.936	0.957	24	2.24%
470	CLEANING_CLOTHS	0.936	0.957	93	2.24%
711	LABEL_MAKERS	0.936	0.957	23	2.24%
616	ROUTERS	0.854	0.873	90	2.22%
185	STREAMING_MEDIA_DEVICES	0.952	0.973	415	2.21%
593	WINDOWS	0.958	0.979	23	2.19%
697	SAFES	0.958	0.979	48	2.19%
464	CHALKBOARD_AND_WHITEBOARD_ERASERS	0.913	0.933	23	2.19%
310	AUTOMOTIVE_THROTTLE_BODIES	0.959	0.980	50	2.19%
636	ELECTRIC_GRILLS	0.826	0.844	21	2.18%
87	EROTIC_BOOKS	0.971	0.992	67	2.16%
277	DISC_PACKAGINGS	0.901	0.920	74	2.11%
936	STOVETOP_POPCORN_POPPERS	0.857	0.875	8	2.10%
567	HABERDASHERY_RIBBONS	0.915	0.934	216	2.08%
315	PANTIES	0.920	0.939	49	2.07%
451	PERFUMES	0.923	0.942	52	2.06%
202	NETBOOKS	0.924	0.943	145	2.06%
294	CAR_DISTRIBUTOR_CAPS	0.933	0.952	205	2.04%
361	HATS_AND_CAPS	0.884	0.902	65	2.04%
610	ELECTRONIC_MUSCLE_STIMULATORS	0.795	0.811	38	2.01%
289	AUTOMOTIVE_WATER_PUMPS	0.903	0.921	391	1.99%
266	PARKING_SENSORS	0.977	0.996	539	1.94%
302	HAIR_STRAIGHTENING_BRUSHES	0.981	1.000	26	1.94%
488	HABERDASHERY_LACE_EDGINGS	0.935	0.953	109	1.93%
583	KEYCHAINS	0.836	0.852	83	1.91%
834	DINING_SETS	0.953	0.971	52	1.89%
32	AUDIO_INTERFACES	0.904	0.921	300	1.88%
216	HAND_FANS	0.866	0.882	32	1.85%
547	COMPUTER_MOTHERBOARDS	0.814	0.829	41	1.84%
494	OVENS	0.954	0.971	171	1.78%
588	CURLING_IRONS	0.900	0.916	42	1.78%
83	KITCHEN_SINKS	0.908	0.924	253	1.76%
103	CD_AND_DVD_PLAYERS	0.924	0.940	291	1.73%
483	ENGINE_CRANKSHAFT_PULLEYS	0.936	0.952	107	1.71%
407	WATER_DISPENSERS	0.946	0.962	225	1.69%
597	MOTORCYCLE_BATTERIES	0.894	0.909	85	1.68%
80	BATHROOM_SINKS	0.955	0.971	427	1.68%
159	DRINKING_GLASSES	0.872	0.886	382	1.61%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
342	SPORT_WATCHES	0.939	0.954	478	1.60%
86	CYCLING_COMPUTERS	0.954	0.969	161	1.57%
124	TREADMILLS	0.977	0.992	65	1.54%
163	VIDEO_CAMERAS	0.932	0.946	37	1.50%
316	ELECTRICAL_OUTLETS	0.933	0.947	122	1.50%
338	SHORTS	0.957	0.971	274	1.46%
477	CHRISTMAS_TREES	0.972	0.986	37	1.44%
343	PORTABLE_EVAPORATIVE_AIR_COOLERS	0.921	0.934	138	1.41%
187	SWAY_BAR_LINKS	0.868	0.880	89	1.38%
125	SPARK_PLUGS	0.955	0.968	726	1.36%
586	BOOTS	0.977	0.990	104	1.33%
344	DOG_CARRIERS_AND_CARRYING_BAGS	0.904	0.916	57	1.33%
128	BLANK_DISCS	0.934	0.946	178	1.28%
340	POOL_INFLATABLES	0.935	0.947	37	1.28%
396	KITCHEN_FURNITURE	0.935	0.947	38	1.28%
364	AUTOMOTIVE_OIL_FILTERS	0.860	0.871	43	1.28%
389	MOTORCYCLE_CLUTCH_COVERS	0.947	0.959	239	1.27%
305	NETWORK_CABLES	0.957	0.969	190	1.25%
432	NAIL_DRYERS	0.969	0.981	131	1.24%
46	BABIES_FORMULA	0.977	0.989	88	1.23%
537	SANDALS_AND_FLIP_FLOPS	0.977	0.989	45	1.23%
288	DATA_CABLES_AND_ADAPTERS	0.897	0.908	80	1.23%
422	SLATWALL_PANELS	0.988	1.000	80	1.21%
336	SOFAS	0.908	0.919	100	1.21%
77	BATTERY_CHARGERS	0.750	0.759	30	1.20%
345	ENGINE_INTAKE_MANIFOLDS	0.951	0.962	51	1.16%
272	KITCHEN_POTS	0.954	0.965	485	1.15%
437	BEDS	0.955	0.966	43	1.15%
527	GAZEBOS	0.967	0.978	91	1.14%
413	CLOTHES_HANGERS	0.972	0.983	92	1.13%
468	LUGGAGE_TAGS	0.905	0.915	47	1.10%
339	SWIMMING_GOGGLES	0.915	0.925	77	1.09%
585	BABIES_FOOTWEAR	0.943	0.953	96	1.06%
26	SMARTWATCHES	0.945	0.955	724	1.06%
69	AUTOMOTIVE_POWER_WINDOW_REGULATORS	0.946	0.956	138	1.06%
649	FUEL_INJECTION_RAILS	0.857	0.866	64	1.05%
34	LIGHT_BULBS	0.953	0.963	829	1.05%
741	LONGBOARDS	0.955	0.965	56	1.05%
72	PROJECTOR_SCREENS	0.990	1.000	53	1.01%
443	SCREWS	0.897	0.906	57	1.00%
5	BATHROOM_FAUCETS	0.903	0.912	521	1.00%
742	PARTY_MASKS	0.919	0.928	70	0.98%
615	HAIR_DRYERS	0.929	0.938	15	0.97%
247	PORTABLE_CELLPHONE_CHARGERS	0.962	0.971	500	0.94%
70	HOOKAHS	0.967	0.976	168	0.93%
371	UKULELES	0.972	0.981	52	0.93%
459	WATER_HEATERS	0.973	0.982	247	0.92%
18	DEEP_FRYERS	0.975	0.984	455	0.92%
405	BINOCULARS	0.975	0.984	186	0.92%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
553	COMPUTER_AND_TV_FLEX_CABLES	0.979	0.988	120	0.92%
449	ALTERNATORS	0.905	0.913	48	0.88%
964	WAFFLE_MAKERS	0.933	0.941	8	0.86%
403	CLOTHING_PATCHES	0.938	0.946	58	0.85%
669	EXTERNAL_LAPTOP_COOLERS	0.824	0.831	38	0.85%
35	CELL_BATTERIES	0.969	0.977	670	0.83%
409	TELEPHONES	0.857	0.864	21	0.82%
651	UMBRELLAS	0.985	0.993	68	0.81%
878	DINING_TABLES	0.875	0.882	16	0.80%
136	BABY_SWIMWEAR	0.889	0.896	121	0.79%
544	BABY_PLAYARDS	0.934	0.941	110	0.75%
292	INSTRUMENT_AMPLIFIERS	0.936	0.943	449	0.75%
423	TOOL_BOXES	0.938	0.945	109	0.75%
165	KNEE_BRACES_SUPPORTS	0.939	0.946	129	0.75%
356	PUZZLES	0.947	0.954	74	0.74%
457	LUMBAR_AND_ABDOMINAL_BRACES	0.962	0.969	147	0.73%
460	DISHWASHERS	0.966	0.973	148	0.72%
100	TABLETS	0.967	0.974	624	0.72%
157	MALE_UNDERWEAR	0.973	0.980	422	0.72%
350	THERMOMETERS	0.977	0.984	63	0.72%
49	CARDS_AND_INVITATIONS	0.870	0.876	47	0.69%
573	UNIVERSAL_HOME_GYMS	0.907	0.913	45	0.66%
17	NOTEBOOKS	0.949	0.955	543	0.63%
61	ARTIFICIAL_FLOWERS	0.954	0.960	497	0.63%
129	SPEAKERS	0.956	0.962	525	0.63%
399	MASCARAS	0.956	0.962	92	0.63%
107	HEADPHONES	0.957	0.963	189	0.63%
376	DESKTOP_COMPUTER_COOLERS_AND_FANS	0.966	0.972	345	0.62%
699	MOTORCYCLE_JERSEYS	0.966	0.972	74	0.62%
134	BAR_CODE_SCANNERS	0.976	0.982	254	0.61%
142	MICROPHONES	0.981	0.987	465	0.61%
782	VIDEO_CAPTURE_DEVICES	0.862	0.867	32	0.58%
258	FACIAL_SKIN_CARE_PRODUCTS	0.864	0.869	341	0.58%
126	HAIR_CLIPPERS	0.881	0.886	714	0.57%
0	FISHING_LINES	0.908	0.913	584	0.55%
359	SHOWER_HEADS	0.909	0.914	235	0.55%
577	DISHES_PLATES	0.926	0.931	59	0.54%
520	COOKIES_CUTTERS	0.930	0.935	77	0.54%
372	TOOTHPASTES	0.937	0.942	62	0.53%
92	PANTS	0.950	0.955	799	0.53%
377	SWAY_BARS	0.959	0.964	98	0.52%
274	VEHICLE_SPEAKERS	0.961	0.966	421	0.52%
348	ELECTRONIC_ENTRANCE_INTERCOMS	0.964	0.969	211	0.52%
141	CHARMS_AND_MEDALS	0.965	0.970	86	0.52%
219	BABY_STROLLERS	0.966	0.971	468	0.52%
325	GATE_MOTORS	0.968	0.973	298	0.52%
498	RACKS_AND_PINIONS	0.973	0.978	113	0.51%
257	AUTOMOTIVE_SIDE_VIEW_MIRRORS	0.977	0.982	737	0.51%
208	BABY_CAR_SEATS	0.979	0.984	550	0.51%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
349	CRAYONS	0.980	0.985	101	0.51%
264	KITCHEN_TOWELS	0.986	0.991	277	0.51%
152	REAR_WHEEL_HUBS_BEARING_ASSEMBLY	0.989	0.994	401	0.51%
28	AUTOMOTIVE_CLUTCH_KITS	0.990	0.995	719	0.51%
788	SUNSCREENS	0.875	0.879	31	0.46%
740	NIGHTSTANDS	0.889	0.893	29	0.45%
790	TOILET_PAPER_HOLDERS	0.898	0.902	25	0.45%
15	KITCHEN_FAUCETS	0.904	0.908	431	0.44%
223	VODKAS	0.938	0.942	94	0.43%
446	MOTORCYCLE_PANTS	0.941	0.945	118	0.43%
196	CAR_AV_RECEIVERS	0.944	0.948	589	0.42%
312	PEDAL_EFFECTS	0.958	0.962	695	0.42%
112	BRUSH_CUTTERS	0.964	0.968	110	0.41%
127	SUITCASES	0.970	0.974	582	0.41%
275	HOME_OFFICE_DESKS	0.970	0.974	149	0.41%
251	BUMPER_IMPACT_ABSORBERS	0.971	0.975	245	0.41%
170	FANS	0.978	0.982	597	0.41%
164	COFFEE_MAKERS	0.979	0.983	567	0.41%
424	BABY_DIAPERS	0.984	0.988	375	0.41%
182	HOME_APPLIANCE_CONTACTORS_AND_RELAYS	0.985	0.989	408	0.41%
297	CABIN_FILTERS	0.988	0.992	244	0.40%
212	CAR_STEREOS	0.851	0.854	243	0.35%
204	AUTOMOTIVE_TIRES	0.870	0.873	80	0.34%
774	TOILETRY_BAGS	0.897	0.900	40	0.33%
429	INK_CARTRIDGES	0.900	0.903	30	0.33%
176	DESKTOP_COMPUTERS	0.904	0.907	36	0.33%
904	SOLDERING_STATIONS	0.923	0.926	26	0.33%
557	DIFFERENTIALS	0.925	0.928	35	0.32%
354	PORTABLE_ELECTRIC_MASSAGERS	0.931	0.934	83	0.32%
381	BLU_RAY_PLAYERS	0.963	0.966	175	0.31%
206	T_SHIRTS	0.964	0.967	742	0.31%
29	WRISTWATCHES	0.965	0.968	876	0.31%
330	LIPSTICKS	0.966	0.969	562	0.31%
215	AIRSOFT_GUNS	0.971	0.974	245	0.31%
224	PUREBRED_DOGS	0.975	0.978	705	0.31%
228	BACKPACKS	0.975	0.978	699	0.31%
530	INDOOR_CURTAINS_AND_BLINDS	0.975	0.978	161	0.31%
53	TABLECLOTHS	0.978	0.981	544	0.31%
279	DRESSES	0.978	0.981	692	0.31%
13	STARTERS	0.979	0.982	354	0.31%
167	WALL_LIGHTS	0.980	0.983	172	0.31%
62	OUTER_TIE_ROD_ENDS	0.981	0.984	496	0.31%
81	MEMORY_CARDS	0.981	0.984	669	0.31%
286	SOLDERING_MACHINES	0.983	0.986	383	0.31%
190	DRONES	0.984	0.987	653	0.30%
290	FOOTBALL_BALLS	0.984	0.987	156	0.30%
529	AUTOMOTIVE_DOOR_PANELS	0.990	0.993	150	0.30%
88	MOTORCYCLE_CASES	0.995	0.998	222	0.30%
417	SUSPENSION_CONTROL_ARM_BUSHINGS	0.844	0.846	138	0.24%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
504	MODEMS	0.883	0.885	123	0.23%
209	WATCH_BANDS	0.916	0.918	265	0.22%
320	FOOD_PROCESSORS	0.917	0.919	135	0.22%
213	CELLPHONE_TABLET_AND_GPS_SCREEN_PROTECTORS	0.921	0.923	155	0.22%
696	POWER_STEERING_FLUID_RESERVOIRS	0.921	0.923	32	0.22%
499	BOOKCASES	0.930	0.932	35	0.22%
411	OPERATING_SYSTEMS	0.937	0.939	270	0.21%
82	HANDBAGS	0.958	0.960	850	0.21%
701	BLOOD_PRESSURE_MONITORS	0.966	0.968	30	0.21%
410	AIR_COMPRESSORS	0.970	0.972	353	0.21%
301	CV_JOINTS	0.972	0.974	443	0.21%
50	ELECTRIC_GUITARS	0.973	0.975	778	0.21%
378	AQUARIUM_FILTERS	0.977	0.979	214	0.20%
7	IRONS	0.981	0.983	433	0.20%
94	FISHING_REELS	0.983	0.985	740	0.20%
217	SIDEBOARDS	0.983	0.985	201	0.20%
45	DESKTOP_COMPUTER_POWER_SUPPLIES	0.985	0.987	679	0.20%
232	INTERACTIVE_GAMING_FIGURES	0.988	0.990	246	0.20%
31	AUTOMOTIVE_SIDE_VIEW_MIRROR_GLASSES	0.989	0.991	536	0.20%
146	ROLLER_SKATES	0.990	0.992	586	0.20%
179	KITCHEN_RANGE_HOODS	0.990	0.992	309	0.20%
415	TOILET_SEATS	0.990	0.992	191	0.20%
57	ENGINE_CONTROL_MODULES	0.994	0.996	415	0.20%
140	FOOTBALL_SHOES	0.995	0.997	734	0.20%
362	MAGAZINES	0.768	0.769	143	0.13%
148	BOOKS	0.857	0.858	834	0.12%
111	HOME_SHELVES	0.890	0.891	157	0.11%
123	DOLLS	0.938	0.939	843	0.11%
41	CELLPHONES	0.953	0.954	329	0.10%
73	AUTOMOTIVE_EMBLEMS	0.953	0.954	524	0.10%
171	FOUNDATIONS	0.956	0.957	706	0.10%
556	HEARING_PROTECTORS	0.961	0.962	39	0.10%
30	WALLPAPERS	0.963	0.964	885	0.10%
160	DRUMS	0.963	0.964	212	0.10%
119	PENCIL_CASES	0.966	0.967	183	0.10%
11	TELEVISIONS	0.968	0.969	728	0.10%
750	LAPTOP_BATTERIES	0.968	0.969	32	0.10%
75	AUTOMOTIVE_SPRING_SUSPENSIONS	0.979	0.980	121	0.10%
480	ACCORDIONS	0.981	0.982	311	0.10%
76	ENGINE_OILS	0.982	0.983	770	0.10%
60	VIDEO_GAMES	0.984	0.985	908	0.10%
395	ELECTRIC_SAWS	0.985	0.986	363	0.10%
93	STOOLS	0.986	0.987	594	0.10%
8	MATTRESSES	0.988	0.989	534	0.10%
197	STEERING_COLUMNS	0.989	0.990	143	0.10%
303	WHEELS_BEARINGS	0.991	0.992	547	0.10%
226	HOVERBOARDS	0.993	0.994	480	0.10%
132	CAR_SEAT_COVERS	0.996	0.997	942	0.10%
346	MOTORCYCLE_HELMETS	0.996	0.997	715	0.10%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
19	TORSION_BARS	0.996	0.996	123	0.00%
24	HEADBOARDS	0.993	0.993	228	0.00%
38	HUMIDIFIERS_AND_VAPORIZERS	0.968	0.968	173	0.00%
58	LATEX_ENAMEL_AND_ACRYLIC_PAINTS	1.000	1.000	67	0.00%
59	MUSICAL_KEYBOARD_CASES_AND_BAGS	0.985	0.985	760	0.00%
63	EXHAUST_MANIFOLDS	0.941	0.941	8	0.00%
67	EYESHADOWS	0.978	0.978	738	0.00%
102	HORSE_SADDLES	1.000	1.000	102	0.00%
106	ROOF_RACKS	0.986	0.986	428	0.00%
114	NECKTIES	0.985	0.985	134	0.00%
117	WHEELCHAIRS	0.989	0.989	90	0.00%
154	CALCULATORS	0.991	0.991	536	0.00%
156	PAPER_CLIPS	0.980	0.980	51	0.00%
162	WHISKEYS	0.969	0.969	267	0.00%
166	WALL_CLOCKS	0.986	0.986	480	0.00%
169	SCULPTURES	0.944	0.944	437	0.00%
180	DRAWERS	0.987	0.987	198	0.00%
189	FISH_TANKS	0.965	0.965	145	0.00%
194	CASH_DRAWERS	0.973	0.973	38	0.00%
198	SKIRTS	0.974	0.974	154	0.00%
234	HAIRDRESSING_SCISSORS	0.981	0.981	160	0.00%
245	PAINT_ROLLERS	0.952	0.952	11	0.00%
261	CAR_ANTENNAS	0.998	0.998	934	0.00%
262	CACHACAS	0.981	0.981	157	0.00%
265	CALIPERS	1.000	1.000	22	0.00%
269	STETHOSCOPES	0.957	0.957	36	0.00%
273	FUEL_INJECTORS	0.913	0.913	103	0.00%
276	GUITAR_STRINGS	0.977	0.977	106	0.00%
280	BASS_GUITARS	0.971	0.971	467	0.00%
295	MOUSE_PADS	0.985	0.985	201	0.00%
321	TRAILER_HITCHES	0.997	0.997	184	0.00%
322	SOFA_AND_FUTON_COVERS	0.995	0.995	99	0.00%
324	VEHICLE_CLUTCH_CABLES	1.000	1.000	10	0.00%
328	BABY_MONITORS	0.984	0.984	220	0.00%
329	VEHICLE_CV_AXLES	0.978	0.978	69	0.00%
334	FIRE_EXTINGUISHERS	0.983	0.983	30	0.00%
347	COMBUSTION_CHAINSAWS	1.000	1.000	84	0.00%
357	WASHING_AND_DRYER_MACHINE_COVERS	0.970	0.970	50	0.00%
363	ENGINE_COOLING_FAN_SHROUDS	0.994	0.994	83	0.00%
367	PILLOWS	0.968	0.968	31	0.00%
379	CELLPHONE_REPLACEMENT_CAMERAS	0.980	0.980	51	0.00%
384	ENGINE_VALVES_SPRING_RETAINERS	0.978	0.978	68	0.00%
387	COTTON_CANDY_MACHINES	1.000	1.000	7	0.00%
388	ORAL_IRRIGATORS	0.933	0.933	8	0.00%
398	MINI_PCS	0.800	0.800	14	0.00%
400	AUTOMOTIVE_WHEEL_COVERS	0.977	0.977	129	0.00%
404	WOOD_BURNING_MACHINES	1.000	1.000	7	0.00%
412	CAR_ENGINE_CAMSHAFTS	0.882	0.882	15	0.00%
420	AUTOMOTIVE_NERF_BARS	0.987	0.987	38	0.00%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
426	MOTORCYCLE_SUITS	0.958	0.958	37	0.00%
428	CAT_SCRATCHERS	0.966	0.966	15	0.00%
433	CAMERA_MONOPODS	0.902	0.902	49	0.00%
435	SALT	1.000	1.000	10	0.00%
439	AUTOMOTIVE_MANUAL_TRANSMISSION_SHIFT_LEVERS	0.868	0.868	25	0.00%
450	MAGNIFYING_GLASSES	0.954	0.954	122	0.00%
461	ELECTRIC_LAWN_MOWERS	0.857	0.857	4	0.00%
469	HEATER_CORES	0.833	0.833	21	0.00%
482	HEAT_GUNS	0.995	0.995	94	0.00%
486	PICTURE_FRAMES	0.968	0.968	151	0.00%
487	OSCILLOSCOPES	0.947	0.947	10	0.00%
489	BEERS	0.979	0.979	170	0.00%
493	KITCHEN_MORTARS	1.000	1.000	4	0.00%
511	TROLLEY_AND_FURNITURE_CASTERS	0.857	0.857	4	0.00%
515	IGNITION_SWITCH_ACTUATORS	1.000	1.000	14	0.00%
516	BABY_STERILIZERS	0.957	0.957	22	0.00%
523	ELLIPTICAL_MACHINES	0.948	0.948	128	0.00%
528	XENON_KITS	0.976	0.976	20	0.00%
536	SCHOOL_AND_OFFICE_GLUES	0.783	0.783	10	0.00%
542	HEDGE_TRIMMERS	1.000	1.000	14	0.00%
543	STABILIZERS_AND_UPS	1.000	1.000	2	0.00%
570	LASER_PRINTER_DRUMS	0.947	0.947	29	0.00%
580	HAND_FILES	0.944	0.944	17	0.00%
595	BEER_FAUCETS	0.800	0.800	5	0.00%
603	PERSONAL_LUBRICANTS_AND_GELS	0.556	0.556	23	0.00%
625	CHESTS	0.857	0.857	3	0.00%
626	JUMP_ROPES	0.947	0.947	39	0.00%
639	CRIB_BEDDING_SETS	0.964	0.964	127	0.00%
640	ORTHOTICS	0.933	0.933	16	0.00%
641	MEDICAL_WALKERS	0.947	0.947	10	0.00%
643	BABY_SAFETY_LOCKS	0.941	0.941	41	0.00%
644	CAR_AC_CONDENSERS	0.989	0.989	96	0.00%
645	CAN_OPENERS	0.818	0.818	10	0.00%
648	VEHICLE_BRAKE_DISCS	0.905	0.905	21	0.00%
650	DOG_LEASHES	1.000	1.000	10	0.00%
656	TRUMPETS	0.889	0.889	4	0.00%
661	DOLLHOUSES	0.667	0.667	5	0.00%
664	MAKEUP_VANITIES	0.857	0.857	8	0.00%
667	MOTORCYCLE_RAIN_SUITS	0.962	0.962	52	0.00%
668	BEAUTY_WIGS	0.977	0.977	22	0.00%
672	ELECTRIC_BLOWERS	0.875	0.875	7	0.00%
673	PAPER_SHREDDERS	1.000	1.000	5	0.00%
675	DESKTOP_COMPUTER_CASES	1.000	1.000	3	0.00%
678	KITES	0.889	0.889	5	0.00%
691	TOILETS	0.920	0.920	27	0.00%
702	FITNESS_TRAMPOLINES	1.000	1.000	11	0.00%
709	RICE	1.000	1.000	3	0.00%
719	RACQUETS	0.923	0.923	6	0.00%
720	CAR_DOOR_HINGES	0.963	0.963	13	0.00%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
726	VARNISHES	0.971	0.971	17	0.00%
730	BABY_BOTTLES	0.975	0.975	100	0.00%
731	BICYCLE_AND_MOTORCYCLE_ALARMS	0.857	0.857	7	0.00%
733	BASEBALL_AND_SOFTBALL_BATS	1.000	1.000	5	0.00%
737	BABY_GYMS	0.909	0.909	6	0.00%
738	CNC_LATHES	1.000	1.000	3	0.00%
745	POUFS	0.857	0.857	10	0.00%
747	DRIVE_SHAFTS	0.957	0.957	24	0.00%
751	POOL_WATERFALLS	0.960	0.960	13	0.00%
752	ECT_SENSORS	1.000	1.000	7	0.00%
758	DENTAL_CHAIRS	1.000	1.000	4	0.00%
759	SUNBATHING_CHAIRS	0.952	0.952	11	0.00%
760	BICYCLE_FRAMES	0.889	0.889	13	0.00%
764	MULTIMETERS	0.727	0.727	6	0.00%
769	PAINTBALL_O_RINGS	0.800	0.800	3	0.00%
773	ELECTRIC_HAND_PLANERS	1.000	1.000	7	0.00%
777	TABLE_TENNIS_TABLES	1.000	1.000	6	0.00%
778	POOL_COVERS	0.930	0.930	22	0.00%
779	GARDEN_SOIL	0.800	0.800	9	0.00%
781	CYMBALS	0.933	0.933	8	0.00%
783	DRUM_BRAKE_SHOES	1.000	1.000	9	0.00%
785	CONDOMS	1.000	1.000	27	0.00%
786	TREADMILL_RUNNING_BELTS	1.000	1.000	13	0.00%
787	HAND_BRAKE_CABLES	0.982	0.982	28	0.00%
789	MAGNETIC_WELDING_HOLDERS	1.000	1.000	6	0.00%
794	CLEANING_SPONGES	1.000	1.000	8	0.00%
795	SKIN_REPELLENTS	0.909	0.909	11	0.00%
797	MERCHANDISER_REFRIGERATORS	0.167	0.167	8	0.00%
798	TENNIS_BALLS	0.800	0.800	3	0.00%
800	TOWEL_HOLDERS	0.889	0.889	9	0.00%
803	HONEY	0.970	0.970	16	0.00%
804	MANUAL_HAMMERS	0.667	0.667	3	0.00%
810	DIVING_MASKS	0.750	0.750	8	0.00%
814	PORCELAIN_TILES	1.000	1.000	5	0.00%
815	BALL_PIT_BALLS	0.923	0.923	7	0.00%
816	HARMONICAS	1.000	1.000	4	0.00%
818	LINGERIE_SETS	0.824	0.824	9	0.00%
823	UNIVERSAL_CAR_REMOTES	1.000	1.000	4	0.00%
824	DRUM_PEDALS	0.913	0.913	24	0.00%
827	HAIR_STRAIGHTENERS	0.914	0.914	18	0.00%
828	MICROWAVES	0.889	0.889	13	0.00%
830	REFLECTIVE_VESTS	1.000	1.000	5	0.00%
832	MICROMETERS	1.000	1.000	54	0.00%
840	POOL_LIGHTS	0.848	0.848	16	0.00%
845	HOSPITAL_BEDS	1.000	1.000	39	0.00%
853	VEHICLE_TRACKERS	0.909	0.909	5	0.00%
854	BRAKE_DRUMS	1.000	1.000	10	0.00%
858	CLUTCH_BEARINGS	1.000	1.000	3	0.00%
860	MOUTHWASHES	0.941	0.941	8	0.00%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
863	LIVING_ROOM_SETS	0.741	0.741	14	0.00%
871	PNEUMATIC_STAPLERS	1.000	1.000	13	0.00%
874	STRAWS	0.857	0.857	8	0.00%
879	TURNTABLE_NEEDLES	0.800	0.800	3	0.00%
881	ORTHOPEDIC_WALKER_BOOTS	0.720	0.720	14	0.00%
883	TORQUE_WRENCHES	1.000	1.000	8	0.00%
884	CATS_LITTER	0.957	0.957	12	0.00%
887	AUTOMOTIVE_DEFLECTORS	1.000	1.000	5	0.00%
888	VARIABLE_FREQUENCY_DRIVES	0.923	0.923	7	0.00%
889	SWEETENERS	1.000	1.000	8	0.00%
890	AUTOMOTIVE_CELLPHONE_AND_GPS_MOUNTS	0.833	0.833	6	0.00%
891	RADIO_BASE_STATIONS	0.222	0.222	6	0.00%
892	CRUTCHES	1.000	1.000	4	0.00%
894	KATANA_SWORDS	0.800	0.800	13	0.00%
897	TELEPHONE_CABLES	0.963	0.963	14	0.00%
898	SOLID_SWEET_PASTES	0.889	0.889	10	0.00%
899	DISPOSABLE_GLOVES	0.800	0.800	3	0.00%
900	MOTORCYCLE_GRAB_BARS	1.000	1.000	3	0.00%
902	GROOVE_JOINT_PLIERS	0.889	0.889	4	0.00%
903	BICYCLE_HANDLEBARS	0.727	0.727	5	0.00%
905	TANDEM_CHAIRS	0.667	0.667	7	0.00%
907	SPARK_PLUG_WIRESETS	0.871	0.871	32	0.00%
908	ELECTRIC_SHOWER_HEADS	0.333	0.333	5	0.00%
911	INDUSTRIAL_DOUGH_KNEADERS	0.880	0.880	13	0.00%
916	DOOR_AND_WINDOW_LOCKS	0.500	0.500	3	0.00%
919	STAPLERS	1.000	1.000	6	0.00%
920	APERITIFS	1.000	1.000	5	0.00%
921	SHOWER_CURTAINS	1.000	1.000	4	0.00%
922	ANTIQUE_CHAIRS	0.444	0.444	7	0.00%
924	SHIN_GUARDS	0.667	0.667	2	0.00%
931	BABY_JUMPERS	0.800	0.800	6	0.00%
934	BREAD_MAKERS	0.857	0.857	3	0.00%
944	ISOPROPYL_ALCOHOLS	0.889	0.889	9	0.00%
949	PUNCHING_BAGS	0.500	0.500	3	0.00%
950	ESPADRILLES	0.933	0.933	8	0.00%
956	DRONE_PROPELLERS	0.769	0.769	6	0.00%
957	TENTS	0.880	0.880	11	0.00%
958	SAFETY_HARNESSES	0.667	0.667	4	0.00%
959	SYRINGES	0.889	0.889	4	0.00%
960	BEDLINERS	0.800	0.800	16	0.00%
961	ELECTROLYTIC_CAPACITORS	1.000	1.000	7	0.00%
962	BASKET_BALLS	1.000	1.000	2	0.00%
963	OTOSCOPES	1.000	1.000	4	0.00%
967	COFFEE_CAPSULES	0.750	0.750	5	0.00%
968	BABY_PACIFIER_CLIPS	0.833	0.833	5	0.00%
971	INDUSTRIAL_PULLEYS	0.909	0.909	5	0.00%
972	BILL_COUNTERS	1.000	1.000	5	0.00%
975	ENGINE_COOLING_FAN_SWITCHES	0.769	0.769	6	0.00%
980	MENSTRUAL_CUPS	1.000	1.000	5	0.00%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
986	MOTORCYCLE_CARBURETORS	0.800	0.800	5	0.00%
989	STORE_SHOPPING_CARTS	0.857	0.857	4	0.00%
990	SWIMMING_NOSE_CLIPS	1.000	1.000	2	0.00%
992	AIRBAG_MODULES	1.000	1.000	3	0.00%
995	MAGNETIC_COMPASSES	1.000	1.000	6	0.00%
997	POPCORN_MACHINES	1.000	1.000	9	0.00%
1000	CAR_FRONT_MASKS	1.000	1.000	2	0.00%
1002	KIDS_TRICYCLES	1.000	1.000	4	0.00%
1003	AIRGUN_PELLETS	0.400	0.400	4	0.00%
1004	AUTOMOTIVE_SEATS	0.714	0.714	6	0.00%
1005	MOTORCYCLE_TRANSMISSION_CROWNS	1.000	1.000	4	0.00%
1006	LAMINATORS	0.833	0.833	7	0.00%
1008	MUSIC_ALBUMS	0.857	0.857	4	0.00%
1011	WALL_ANCHOR_PLUGS	0.667	0.667	4	0.00%
1013	PET_COLLARS	0.929	0.929	14	0.00%
1014	GATE_GEAR_RACKS	0.857	0.857	4	0.00%
1015	CAR_HOODS	1.000	1.000	3	0.00%
1017	LED_STRIPS	0.889	0.889	4	0.00%
1018	SANDWICH_MAKERS	0.800	0.800	2	0.00%
1019	DENTAL_FLOSSES	1.000	1.000	2	0.00%
1028	KNITTING_NEEDLES	0.750	0.750	6	0.00%
1033	TOOTHBRUSH_HOLDERS	0.667	0.667	2	0.00%
1034	TABLE_TENNIS_BALLS	1.000	1.000	2	0.00%
1037	MASSAGE_SOFAS	1.000	1.000	4	0.00%
1038	STYLING_CHAIRS	0.667	0.667	2	0.00%
1039	BICYCLE_SEATS	1.000	1.000	3	0.00%
1040	VOLLEYBALL_BALLS	0.800	0.800	3	0.00%
1042	BINDING_SPINES	0.667	0.667	2	0.00%
1043	DIGITAL_WEATHER_STATIONS	1.000	1.000	2	0.00%
1045	DOORBELLS	0.500	0.500	2	0.00%
1046	DRIED_FRUITS	1.000	1.000	2	0.00%
1047	BOXING_HEADGEARS	0.667	0.667	2	0.00%
120	AUTOMOTIVE_SHIFT_LEVER_KNOBS	1.000	0.999	938	-0.10%
260	TV_AND_MONITOR_MOUNTS	0.995	0.994	546	-0.10%
155	TV_ANTENNAS	0.990	0.989	310	-0.10%
101	MAKEUP_BRUSHES	0.982	0.981	534	-0.10%
95	WALLETS	0.981	0.980	665	-0.10%
259	PACKAGING_ROLLS	0.981	0.980	26	-0.10%
707	BATHROOM_GRAB_BARS	0.981	0.980	26	-0.10%
351	CAMERA_TRIPODS	0.979	0.978	427	-0.10%
608	SAXOPHONES	0.971	0.970	34	-0.10%
89	COSTUMES	0.967	0.966	276	-0.10%
285	FLOOD_LIGHTS	0.967	0.966	462	-0.10%
229	ADHESIVE_TAPES	0.957	0.956	253	-0.10%
727	STYLUSES	0.952	0.951	42	-0.11%
56	AM_FM_RADIOS	0.945	0.944	383	-0.11%
563	CELLPHONE_COVERS	0.927	0.926	54	-0.11%
25	YARNS	0.920	0.919	142	-0.11%
255	SUPPLEMENTS	0.877	0.876	430	-0.11%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
99	CAR_POWER_STEERING_PUMPS	0.993	0.991	370	-0.20%
110	FLASHLIGHTS	0.988	0.986	606	-0.20%
113	COMPUTER_PROCESSORS	0.986	0.984	794	-0.20%
227	SUSPENSION_BALL_JOINTS	0.980	0.978	579	-0.20%
458	AIR_MATTRESSES	0.977	0.975	154	-0.20%
218	VEHICLE_STICKERS	0.975	0.973	530	-0.21%
1	MOBILE_DEVICE_CHARGERS	0.972	0.970	804	-0.21%
122	WALKIE_TALKIES	0.967	0.965	283	-0.21%
199	SANDER_MACHINES	0.964	0.962	292	-0.21%
306	CRIBS	0.942	0.940	242	-0.21%
138	JACKETS_AND_COATS	0.941	0.939	815	-0.21%
192	BLENDERS	0.938	0.936	452	-0.21%
4	CAR_WHEELS	0.995	0.992	720	-0.30%
225	VIOLINS	0.993	0.990	151	-0.30%
133	AUTOMOTIVE_SUSPENSION_CONTROL_ARMS	0.981	0.978	254	-0.31%
135	AUTOMOTIVE_MOLDINGS	0.978	0.975	720	-0.31%
84	CAMERA_BATTERIES	0.977	0.974	554	-0.31%
299	MIXERS	0.977	0.974	350	-0.31%
326	FOG_LIGHTS	0.976	0.973	292	-0.31%
23	AUTOMOTIVE_WEATHERSTRIPS	0.975	0.972	788	-0.31%
243	CAMERA_LENSES	0.975	0.972	219	-0.31%
48	CARPETS	0.974	0.971	857	-0.31%
130	DVD_RECORDERS	0.973	0.970	450	-0.31%
68	RANGES	0.970	0.967	450	-0.31%
96	MUSICAL_KEYBOARDS	0.967	0.964	511	-0.31%
270	PRINTERS	0.966	0.963	400	-0.31%
21	CEILING_LIGHTS	0.963	0.960	472	-0.31%
239	AUTOMOBILE_FENDER_LINERS	0.947	0.944	18	-0.32%
307	HOME_HEATERS	0.946	0.943	234	-0.32%
195	CAMERA_CHARGERS	0.927	0.924	285	-0.32%
241	STUFFED_TOYS	0.891	0.888	679	-0.34%
601	EROTIC_MALE_UNDERWEAR	0.889	0.886	38	-0.34%
40	PORTABLE_GENERATORS	1.000	0.996	132	-0.40%
418	VEHICLE_BRAKE_PADS	0.994	0.990	350	-0.40%
118	RAM_MEMORY_MODULES	0.993	0.989	822	-0.40%
242	MARTIAL_ARTS_AND_BOXING_GLOVES	0.988	0.984	254	-0.40%
51	COMPUTER_MONITORS	0.976	0.972	701	-0.41%
441	AUTOMOTIVE_HEADLIGHTS	0.973	0.969	131	-0.41%
85	WATER_RADIATORS	0.971	0.967	399	-0.41%
144	WOMEN_SWIMWEAR	0.970	0.966	500	-0.41%
300	MOTORCYCLE_FAIRINGS	0.953	0.949	353	-0.42%
370	FLUTES	0.952	0.948	128	-0.42%
296	FABRICS	0.946	0.942	379	-0.42%
552	MOTORCYCLE_GLOVES	0.938	0.934	151	-0.43%
606	AUTOMOTIVE_ARMRESTS	0.899	0.895	55	-0.44%
447	INTEGRATED_CIRCUITS	0.886	0.882	72	-0.45%
500	TOY_TRAINS	0.872	0.868	94	-0.46%
427	KITCHEN_BOWLS	0.837	0.833	22	-0.48%
575	CONTINUOUS_LIGHTING	0.829	0.825	35	-0.48%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
366	MOTORCYCLE_TURN_SIGNAL_LIGHTS	0.998	0.993	221	-0.50%
502	SUBMERSIBLE_PUMPS	0.985	0.980	100	-0.51%
337	GRAPHICS_TABLETS	0.984	0.979	190	-0.51%
796	SOCKS	0.983	0.978	89	-0.51%
445	POSTERS	0.976	0.971	206	-0.51%
434	BICYCLES	0.975	0.970	184	-0.51%
298	SEWING_MACHINES	0.969	0.964	292	-0.52%
115	MOTORCYCLE_JACKETS	0.968	0.963	551	-0.52%
10	KITCHEN_KNIVES	0.960	0.955	202	-0.52%
153	DIECAST_VEHICLES	0.958	0.953	606	-0.52%
244	EROTIC_PUMPS	0.953	0.948	89	-0.52%
308	BRACELETS_AND_ANKLE_BRACES	0.944	0.939	346	-0.53%
121	MUGS	0.943	0.938	329	-0.53%
256	HAMMOCKS	0.994	0.988	82	-0.60%
181	FURNITURE_KNOBS	0.983	0.977	147	-0.61%
173	DRILL_BITS	0.982	0.976	143	-0.61%
2	SUNGLASSES	0.981	0.975	875	-0.61%
116	WRENCHES	0.977	0.971	371	-0.61%
145	PARTY_DECORATIVE_BACKDROPS	0.977	0.971	105	-0.61%
506	PENDRIVES	0.971	0.965	139	-0.62%
214	GLASSES_FRAMES	0.961	0.955	296	-0.62%
442	PAINTBALLS	0.936	0.930	85	-0.64%
865	HAND_POLISHERS	0.774	0.769	14	-0.65%
574	SCALEXTRIC_CARS	0.903	0.897	15	-0.66%
721	BODY_SHAPERS	0.872	0.866	50	-0.69%
201	BRAKE_BOOSTERS	0.993	0.986	139	-0.70%
492	VR_HEADSETS	0.986	0.979	352	-0.71%
594	LATHES	0.983	0.976	146	-0.71%
79	CONTINUOUS_INK_SYSTEMS	0.982	0.975	140	-0.71%
27	FOOTBALL_SHIRTS	0.980	0.973	921	-0.71%
183	SHAVING_MACHINES	0.838	0.832	449	-0.72%
391	TABLE_RUNNERS	0.964	0.957	67	-0.73%
505	OFFICE_CHAIRS	0.940	0.933	98	-0.74%
319	DECORATIVE_VASES	0.913	0.906	274	-0.77%
473	SAFETY_FOOTWEAR	1.000	0.992	66	-0.80%
455	PAJAMAS	0.974	0.966	56	-0.82%
313	ENGINE_PISTONS	0.966	0.958	132	-0.83%
687	STRING_TRIMMERS	0.957	0.949	60	-0.84%
177	DECORATIVE_VINYLS	0.956	0.948	709	-0.84%
238	ENGINE_INTAKE_HOSES	0.954	0.946	373	-0.84%
311	COOKING_SCALES	0.949	0.941	412	-0.84%
539	ELECTRIC_BATHROOM_FAUCETS	0.816	0.809	25	-0.86%
369	TOILET_RUGS	0.926	0.918	249	-0.86%
268	HAIR_SHAMPOOS_AND_CONDITIONERS	0.892	0.884	44	-0.90%
416	AUTOMOTIVE_DOORS	0.994	0.985	332	-0.91%
237	TV_REPLACEMENT_BACKLIGHT_LED_STRIPS	0.983	0.974	115	-0.92%
578	CIRCUIT_BREAKERS	0.983	0.974	149	-0.92%
382	AUTOMOTIVE_FENDERS	0.976	0.967	124	-0.92%
16	TACTICAL_AND_SPORTING_KNIVES_AND_BLADES	0.966	0.957	164	-0.93%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
65	REFRIGERATORS	0.957	0.948	524	-0.94%
475	SHOCK_MOUNT_INSOLATORS	0.850	0.842	73	-0.94%
74	PAINTBALL_MARKERS	0.953	0.944	97	-0.94%
90	BODYWEIGHT_SCALES	0.948	0.939	293	-0.95%
431	JUMPSUITS_AND_OVERALLS	0.926	0.917	106	-0.97%
521	NEBULIZERS	1.000	0.990	49	-1.00%
230	ELECTRICAL_CABLES	0.994	0.984	250	-1.01%
284	CAR_GEARBOXES	0.991	0.981	267	-1.01%
267	ELECTRIC_PRESSURE_WASHERS	0.987	0.977	343	-1.01%
91	GAMEPADS_AND_JOYSTICKS	0.976	0.966	612	-1.02%
360	ANTI_THEFT_STUDS	0.965	0.955	220	-1.04%
507	NECKLACES	0.950	0.940	91	-1.05%
679	VASES	0.657	0.650	37	-1.07%
20	AUDIO_AMPLIFIERS	0.844	0.835	311	-1.07%
341	DOORS	0.937	0.927	97	-1.07%
524	ENGINE_VALVES	0.930	0.920	44	-1.08%
712	CATS	0.833	0.824	19	-1.08%
139	CHAMPAGNES	0.903	0.893	31	-1.11%
558	FLEA_AND_TICK_TREATMENTS	0.992	0.981	129	-1.11%
402	LIQUORS	0.975	0.964	100	-1.13%
22	BELTS	0.970	0.959	49	-1.13%
559	AUTOMOTIVE_TRUNK_LIDS	0.970	0.959	118	-1.13%
545	RESISTANCE_BANDS	0.967	0.956	180	-1.14%
71	AUTOMOTIVE_AMPLIFIERS	0.955	0.944	662	-1.15%
175	EROTIC_CREAMS	0.929	0.918	279	-1.18%
638	TABLE_DRILLS	0.928	0.917	48	-1.19%
36	SURVEILLANCE_CAMERAS	0.921	0.910	909	-1.19%
466	RINGS	0.988	0.976	43	-1.21%
55	PLANTS	0.987	0.975	313	-1.22%
589	EROTIC_BALLS	0.892	0.881	42	-1.23%
254	BLANKETS	0.889	0.878	131	-1.24%
695	ORTHOPEDIC_WRIST_BRACES	0.966	0.954	77	-1.24%
496	EARRINGS	0.956	0.944	44	-1.26%
718	PADLOCKS	0.955	0.943	44	-1.26%
54	VIBRATORS	0.871	0.860	117	-1.26%
390	GAME_CONSOLES	0.938	0.926	347	-1.28%
561	CATS_AND_DOGS_FOODS	0.994	0.981	161	-1.31%
240	DJ_CONTROLLERS	0.915	0.903	175	-1.31%
184	TOOTHBRUSHES	0.989	0.976	265	-1.31%
271	ELECTRIC_DRILLS	0.973	0.960	414	-1.34%
131	KEYBOARD_AND_MOUSE_KITS	0.972	0.959	275	-1.34%
331	SERVING_AND_HOME_TRAYS	0.967	0.954	354	-1.34%
158	ALL_IN_ONE	0.965	0.952	189	-1.35%
3	FREEZERS	0.938	0.925	444	-1.39%
554	MANGA	0.865	0.853	39	-1.39%
314	FLATWARE_SETS	0.931	0.918	31	-1.40%
108	ELECTRIC_SCREWDRIVERS	0.858	0.846	122	-1.40%
419	REMOTE_CONTROL_TOY_VEHICLES	0.918	0.905	209	-1.42%
613	TRANSISTORS	0.902	0.889	54	-1.44%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
756	SIDE_TABLES	0.958	0.944	37	-1.46%
681	PUZZLE_CUBES	0.932	0.918	61	-1.50%
278	LIP_GLOSSES	0.929	0.915	190	-1.51%
430	MARKERS_AND_HIGHLIGHTERS	0.917	0.903	126	-1.53%
368	TEQUILAS	0.970	0.955	33	-1.55%
293	MUSIC_STANDS	1.000	0.984	32	-1.60%
607	SEX_TOY_KITS	0.871	0.857	32	-1.61%
64	DIGITAL_VOICE_RECORDERS	0.924	0.909	774	-1.62%
566	TELEVISION_MAIN_PLATE_REPLACEMENTS	0.922	0.907	78	-1.63%
531	CAR_WINDOW_SWITCHES	0.921	0.906	31	-1.63%
467	AUTOMOTIVE_AIR_FILTERS	0.973	0.957	55	-1.64%
611	SEWING_THREADS	0.901	0.886	83	-1.66%
250	DJ_EFFECTS_PROCESSORS	0.809	0.795	202	-1.73%
235	CUSHIONS	0.971	0.954	213	-1.75%
438	BODY_SKIN_CARE_PRODUCTS	0.852	0.837	380	-1.76%
281	TURNTABLES	0.907	0.891	270	-1.76%
619	SPICE_RACKS	0.897	0.881	28	-1.78%
947	KITCHEN_GRATERS	0.727	0.714	5	-1.79%
448	GARDEN_HOSES	0.982	0.964	56	-1.83%
568	POWERED_RIDE_ON_TOYS	0.976	0.958	86	-1.84%
333	GRAPHICS_CARDS	0.970	0.952	32	-1.86%
746	CUSHION_COVERS	0.961	0.943	76	-1.87%
323	SHIRTS	0.934	0.916	118	-1.93%
304	AV_RECEIVERS	0.878	0.861	188	-1.94%
252	NOTEBOOKS_AND_WRITING_PADS	0.974	0.955	153	-1.95%
602	HAND_AND_FOOT_CREAMS	0.922	0.904	83	-1.95%
137	TABLE_AND_DESK_LAMPS	0.960	0.941	260	-1.98%
658	CRASHED_CARS	0.986	0.966	71	-2.03%
714	SECURITY_SEALS	0.936	0.917	23	-2.03%
549	VESTS	0.984	0.964	97	-2.03%
811	LASER_POINTERS	0.875	0.857	8	-2.06%
591	LAPTOP_LCD_SCREENS	0.971	0.951	52	-2.06%
522	COMFORTERS	0.723	0.708	20	-2.07%
291	LENS_FILTERS	0.959	0.939	75	-2.09%
724	STATIONARY_BICYCLES	0.953	0.933	74	-2.10%
373	MOTORCYCLE_CRASH_BARS	0.950	0.930	21	-2.11%
283	BED_SHEETS	0.927	0.907	75	-2.16%
221	MICRO_ROTARY_TOOLS	0.971	0.950	69	-2.16%
771	TACTICAL_VESTS	0.818	0.800	13	-2.20%
33	KITCHEN_PLAYSETS	0.946	0.925	100	-2.22%
78	SWEATSHIRTS_AND_HOODIES	0.942	0.921	118	-2.23%
632	PLACEMATS	0.897	0.877	30	-2.23%
105	SCREEN_PRINTERS	0.962	0.940	156	-2.29%
282	TOY_BUILDING_SETS	0.916	0.895	249	-2.29%
454	CELLPHONE_AND_TABLET_CASES	0.912	0.891	91	-2.30%
178	BOARD_GAMES	0.945	0.923	654	-2.33%
713	WINES	0.978	0.955	182	-2.35%
674	STICKY_NOTES	0.976	0.953	43	-2.36%
150	UPS_BATTERIES	0.859	0.838	64	-2.44%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
766	INDUSTRIAL_ICE_CREAM_MACHINES	0.842	0.821	20	-2.49%
749	GAS_LIFT_SUPPORTS	0.961	0.937	39	-2.50%
263	TV_SMPS	0.936	0.912	279	-2.56%
546	LAPTOP_CHARGERS	0.933	0.909	68	-2.57%
463	LED_STAGE_LIGHTS	0.890	0.867	216	-2.58%
497	WARDROBES	0.946	0.921	36	-2.64%
401	WINDSHIELD_WIPERS	0.982	0.956	190	-2.65%
857	DENTAL_PLIERS	0.867	0.844	34	-2.65%
514	SELF_ADHESIVE_LABELS	0.863	0.840	49	-2.67%
408	GLOW_PLUG_CONTROLLERS	0.964	0.938	56	-2.70%
6	ACTION_FIGURES	0.776	0.755	800	-2.71%
440	TV_STORAGE_UNITS	0.951	0.925	52	-2.73%
318	STEAM_CLEANERS	0.857	0.833	19	-2.80%
665	EGR_VALVES	0.878	0.851	21	-3.08%
710	STIMULATING_PILLS_AND_CAPSULES	0.810	0.785	38	-3.09%
231	COFFEE_TABLES	0.898	0.870	26	-3.12%
421	ESSENTIAL_OILS	0.896	0.868	65	-3.13%
406	COMMERCIAL_LIGHT_SIGNS	0.955	0.925	32	-3.14%
647	EROTIC_MAGAZINES	1.000	0.968	16	-3.20%
864	BABY_PACIFIERS	1.000	0.968	15	-3.20%
109	VACUUM_CLEANERS	0.933	0.903	16	-3.22%
548	COIN_PURSES	0.868	0.840	26	-3.23%
540	VOLTAGE_DETECTORS	0.983	0.951	29	-3.26%
161	WELDING_MASKS	0.942	0.911	63	-3.29%
635	INDUSTRIAL_AND_COMMERCIAL_SCALES	0.759	0.734	73	-3.29%
397	BABY_HIGH_CHAIRS	0.817	0.790	88	-3.30%
52	MULTIGAME_MACHINES	0.865	0.836	77	-3.35%
793	JEWELRY_DISPLAYS	1.000	0.966	15	-3.40%
634	HARD_DRIVES_AND_SSDS	0.965	0.932	42	-3.42%
385	BABY_WALKERS	0.929	0.897	13	-3.44%
386	ANTIVIRUS_AND_INTERNET_SECURITY	0.968	0.933	16	-3.62%
248	ANALOG_CAMERAS	0.967	0.932	457	-3.62%
839	GIFT_CARDS	0.906	0.873	25	-3.64%
633	NETWORK_SWITCHES	0.960	0.923	13	-3.85%
926	FOOTBALL_GOALKEEPER_GLOVES	0.933	0.897	16	-3.86%
620	PENCILS	0.880	0.846	12	-3.86%
866	SHADE_CLOTHS	1.000	0.960	12	-4.00%
565	WASTE_BASKETS	0.774	0.743	17	-4.01%
147	COMICS	0.838	0.804	534	-4.06%
203	FISHING_LURES	0.876	0.840	67	-4.11%
728	LUNCHBOXES	0.957	0.917	22	-4.18%
757	BARBECUE_TOOL_SETS	0.883	0.846	40	-4.19%
761	IP_TELEPHONES	0.906	0.868	28	-4.19%
393	3D_PRINTERS	1.000	0.958	47	-4.20%
722	MINI_COMPONENT_SYSTEMS	0.522	0.500	10	-4.21%
541	ORTHOPEDIC_ANKLE_BRACES	0.929	0.889	15	-4.31%
151	CAKE_STANDS	0.956	0.913	45	-4.50%
453	FISHING_RODS	0.905	0.864	19	-4.53%
937	BREAST_FEEDING_PILLOWS	0.857	0.818	11	-4.55%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
317	BABY_CLOTHING_SETS	0.877	0.837	76	-4.56%
495	DISHES_RACKS	0.752	0.717	49	-4.65%
872	THERMAL_REFRIGERATORS_AND_BAGS	0.792	0.755	28	-4.67%
590	GYM_GLOVES	0.913	0.870	67	-4.71%
653	INSTANT_COFFEE	0.976	0.930	20	-4.71%
104	VIDEO_GAME_PREPAID_CARDS	0.864	0.823	68	-4.75%
700	EQUALIZERS	0.750	0.714	15	-4.80%
609	DECORATIVE_BOXES	0.892	0.848	31	-4.93%
375	FOOD_SLICERS	0.930	0.884	22	-4.95%
44	CAMERAS	0.840	0.798	92	-5.00%
621	CELLPHONE_BATTERIES	0.959	0.909	38	-5.21%
657	DINING_CHAIRS	0.875	0.829	36	-5.26%
579	CAR_SCREENS	0.889	0.842	10	-5.29%
456	BUTT_PLUGS	0.831	0.787	34	-5.29%
753	FOLDERS_AND_EXPANDING_FILES	0.943	0.893	28	-5.30%
819	MEGAPHONES	0.941	0.889	9	-5.53%
822	DRONE_BATTERIES	0.941	0.889	8	-5.53%
623	PLAYING_CARDS	0.877	0.828	30	-5.59%
877	VIBRATION_PLATFORMS	1.000	0.944	18	-5.60%
821	TRADING_CARD_GAMES	0.899	0.848	45	-5.67%
637	CAMERA_BATTERY_GRIPS	0.667	0.629	16	-5.70%
287	ELECTRONIC_DRUMS	0.944	0.890	117	-5.72%
510	VINYL_ROLLS	0.750	0.707	70	-5.73%
560	GUITAR_PICKS	0.970	0.914	17	-5.77%
617	AIRBAGS	0.965	0.909	42	-5.80%
605	PENIS_SLEEVES	0.875	0.824	9	-5.83%
979	VINYL_FLOORINGS	0.875	0.824	7	-5.83%
550	HAIRDRESSING_CAPS	1.000	0.941	8	-5.90%
484	PARTY_HATS	0.625	0.588	8	-5.92%
692	NOTEBOOK_CASES	0.938	0.882	17	-5.97%
868	POWER_GRINDERS	0.867	0.815	13	-6.00%
614	KIDS_WALKIE_TALKIES	0.833	0.783	12	-6.00%
355	NON_CORRECTIVE_CONTACT_LENSES	0.894	0.840	24	-6.04%
43	EMBROIDERY_MACHINES	0.900	0.844	51	-6.22%
599	AUDIO_AND_VIDEO_CABLES_AND_ADAPTERS	0.800	0.750	8	-6.25%
799	AIR_CONDITIONERS	0.889	0.833	23	-6.30%
471	AIR_FRESHENERS	0.730	0.684	43	-6.30%
855	AB_ROLLER_WHEELS	0.952	0.889	10	-6.62%
833	MOTORCYCLE_IGNITION_COILS	0.967	0.903	29	-6.62%
479	BASKETBALL_JERSEYS	0.857	0.800	15	-6.65%
952	VIDEO_CASSETTES	0.857	0.800	6	-6.65%
600	ENGINE_COOLING_FAN_MOTORS	0.929	0.867	14	-6.67%
817	PHOTO_ALBUMS	1.000	0.933	8	-6.70%
955	FLOOR_LAMPS	0.818	0.762	11	-6.85%
744	BABY_BODYSUITS	0.867	0.807	27	-6.92%
436	SCREEN_PRINTING_MACHINES	0.933	0.867	14	-7.07%
682	BEDROOM_SETS	0.875	0.811	17	-7.31%
754	CAR_LIGHT_BULBS	0.720	0.667	14	-7.36%
873	LIGHT_STANDS	1.000	0.923	7	-7.70%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
932	CAMERA_REPLACEMENT_DISPLAYS	1.000	0.923	7	-7.70%
631	BOX_SPRING_AND_MATTRESS_SETS	0.800	0.737	9	-7.88%
776	ELECTRIC_DEMOLITION_HAMMERS	0.900	0.829	18	-7.89%
875	SUITS	0.759	0.697	33	-8.17%
708	NETWORK_CARDS	0.781	0.717	31	-8.19%
652	MICRODERMABRASION_MACHINES	0.909	0.833	6	-8.36%
984	ENERGETIC_STONES	0.875	0.800	7	-8.57%
935	SCHOOL_AND_OFFICE_PAPERS	0.625	0.571	8	-8.64%
851	DISTRIBUTION_KITS	0.833	0.759	14	-8.88%
207	OFFICE_SOFTWARE	0.770	0.701	82	-8.96%
1009	AUTOMOTIVE_CV_JOINT_BOOTS	0.824	0.750	7	-8.98%
444	TOY_ROBOTS	0.759	0.690	15	-9.09%
694	LAPTOP_HOUSINGS	1.000	0.909	6	-9.10%
880	HEARING_AIDS	1.000	0.909	10	-9.10%
923	YOGURT_MAKERS	0.800	0.727	5	-9.13%
39	KITCHEN_APRONS	0.846	0.764	27	-9.69%
974	WASHING_MACHINES	0.870	0.783	12	-10.00%
686	COLLECTIBLE_CANS_BOTTLES_AND_SODA_SIPHONS	0.915	0.820	30	-10.38%
941	POOL_PUMPS	0.960	0.857	12	-10.73%
685	ENGINE_COOLING_FAN_CLUTCHES	0.750	0.667	4	-11.07%
927	LASER_LEVELS	0.875	0.778	8	-11.09%
966	ELECTRIC_AIR_PUMPS	0.875	0.778	8	-11.09%
801	RUM	1.000	0.889	4	-11.10%
861	WIRELESS_ANTENNAS	1.000	0.889	5	-11.10%
953	POTENTIOMETERS	1.000	0.889	5	-11.10%
895	CATS_AND_DOGS_TREATS	0.846	0.750	14	-11.35%
210	PENS	0.862	0.764	27	-11.37%
174	PC_KEYBOARDS	0.840	0.741	26	-11.79%
882	TEA	0.667	0.588	10	-11.84%
1020	CAMERA_FLASHES	0.909	0.800	5	-11.99%
1023	GAUZES	0.909	0.800	6	-11.99%
186	CAR_AC_HOSE_ASSEMBLIES	0.733	0.645	17	-12.01%
47	MOTORCYCLE_SPEEDOMETERS	0.857	0.750	4	-12.49%
791	VIDEOCASSETTE_PLAYERS	0.857	0.750	4	-12.49%
519	FISHES	0.800	0.700	9	-12.50%
689	CERAMIC_TILES	0.929	0.800	13	-13.89%
743	WASTE_CONTAINERS	0.857	0.737	12	-14.00%
1016	SCREEN_PRINTING_KITS	1.000	0.857	4	-14.30%
829	PUPPETS	0.667	0.571	4	-14.39%
1030	SOUND_CARDS	0.889	0.750	5	-15.64%
534	DRILLS_SCREWDRIVERS	0.593	0.500	24	-15.68%
784	DECORATIVE_BASKETS	0.644	0.543	44	-15.68%
909	PORTABLE_DVD_PLAYERS	0.875	0.737	9	-15.77%
705	COMPRESSION_SLEEVES	0.800	0.667	3	-16.63%
886	MEAT_HOOKS	0.800	0.667	2	-16.63%
1007	CEREAL_BARS	0.800	0.667	3	-16.63%
1021	DOG_NAIL_CLIPPERS	0.800	0.667	2	-16.63%
1024	SODS	0.800	0.667	3	-16.63%
1027	ELECTRIC_CHAINSAWS	0.800	0.667	3	-16.63%

ID	Class	F1-Score		Support	%
		BERT	BERT + TAPT		
1032	FANNY_PACKS	0.800	0.667	4	-16.63%
1044	HOME_BOTTLE_STANDS	0.800	0.667	3	-16.63%
896	MEMORY_CARD_READERS	0.400	0.333	3	-16.75%
642	PILATES_BALLS	0.966	0.800	15	-17.18%
383	SPARKLING_WINES	0.818	0.667	12	-18.46%
555	SELF_TANNERS	1.000	0.800	3	-20.00%
825	TABLE_CLOCKS	0.500	0.400	8	-20.00%
847	CLUTCH_FORKS	0.500	0.400	3	-20.00%
850	OUTDOOR_TABLES	0.667	0.526	10	-21.14%
698	TOY_GARAGES_AND_GAS_STATIONS	0.688	0.541	17	-21.37%
852	JUICERS	0.727	0.571	5	-21.46%
734	DOG_BEDS	0.923	0.714	7	-22.64%
1041	SPHYGMOMANOMETERS	0.750	0.571	4	-23.87%
988	EDIBLE_SEEDS	0.667	0.500	3	-25.04%
820	TV_REMOTE_CONTROLS	0.571	0.421	9	-26.27%
1022	PINBALLS	0.909	0.667	6	-26.62%
551	CAMERA_CASES	0.833	0.600	7	-27.97%
562	BABY_BLANKETS	0.613	0.429	29	-30.02%
655	AUTOMOTIVE_BUMPER_GRILLES	0.857	0.571	3	-33.37%
946	MEDICINE_BALLS	0.857	0.571	4	-33.37%
965	PENIS_RINGS	0.667	0.444	4	-33.43%
629	INTERCOOLER_HOSES	0.486	0.323	24	-33.54%
869	CAMERA_STRAPS	0.800	0.500	3	-37.50%
987	SCREWDRIVERS_SETS	0.800	0.500	3	-37.50%
809	LEGGINGS	0.186	0.114	30	-38.71%
981	SLEEPING_BAGS	0.545	0.286	4	-47.52%
918	HAND_TRUCKS	0.667	0.333	4	-50.07%
976	AUTOMOTIVE_BATTERIES	0.750	0.286	5	-61.87%
806	BICYCLE_WHEELS	0.800	0.250	6	-68.75%
335	KITCHEN_CABINET_ORGANIZERS	0.400	0.000	7	-100.00%
663	AUTOMOTIVE_CLUTCH_MASTER_CYLINDERS	0.200	0.000	8	-100.00%
826	EROTIC_ANAL_AND_VAGINAL_DOUCHES	0.800	0.000	3	-100.00%
991	AFTERSHAVES	0.500	0.000	3	-100.00%
765	MEAT_GRINDERS	0.800	0.000	3	-100.00%
994	NECK_GAITERS_MASKS_AND_BALACLAVAS	0.667	0.000	2	-100.00%
996	MAKEUP_TRAIN_CASES	0.667	0.000	4	-100.00%