# Improving Out-of-Scope Detection in Intent Classification by Using Embeddings of the Word Graph Space of the Classes

**Paulo Cavalin, Victor Henrique Alves Ribeiro, Ana Paula Appel, Claudio Pinhanez**
IBM Research
São Paulo, SP, Brazil
pcavalin@br.ibm.com

## Abstract

This paper explores how intent classification can be improved by representing the class labels not as a discrete set of symbols but as a space where the word graphs associated to each class are mapped using typical graph embedding techniques. The approach, inspired by a previous algorithm used for an inverse dictionary task, allows the classification algorithm to take in account inter-class similarities provided by the repeated occurrence of some words in the training examples of the different classes. The classification is carried out by mapping text embeddings to the word graph embeddings of the classes. Focusing solely on improving the representation of the class label set, we show in experiments conducted in both private and public intent classification datasets, that better detection of out-of-scope examples (OOS) is achieved and, as a consequence, that the overall accuracy of intent classification is also improved. In particular, using the recently-released *Larson dataset*, an error of about 9.9% has been achieved for OOS detection, beating the previous state-of-the-art result by more than 31 percentage points.

## 1 Introduction

Intent classification is usually applied for response selection in conversational systems, such as text-based *chatbots*. For the end-user to have the best possible experience with those systems, it is expected that an *intent classifier* is able not only to map an input utterance to the correct intent but also to detect when the utterance is not related to any of the intents, to which we refer to as *out-of-scope* (OOS)[1] inputs or samples. In the light of this, this paper describes and evaluates a method which tries to capture the complexity of the set of intents by embedding them into a vector space created using *word graphs*, as described later. We show that, although the method in some cases is able to improve the accuracy of a text classifier in *in-scope* examples, it has often a tremendous impact on improving the ability of text classifier to reject OOS text, without relying on OOS examples in the training set.

Notice that the intent classifier is typically implemented using standard text classification algorithms (Weiss et al., 2012; Larson et al., 2019; Casanueva et al., 2020). Consequently, to perform OOS sample detection, methods often rely on one-class classification or *threshold rejection*-based techniques using the probability outputs for each class (Larson et al., 2019) or reconstruction errors (Ryu et al., 2017, 2018).

There also exist approaches based on the assumption that OOS data can be collected and included in the training set (Tan et al., 2019; Larson et al., 2019). However, in practice, collecting OOS data can be a burden for intent classifier creation, which is generally carried out by domain experts and not by machine learning experts. Thus, in the ideal world, one should rely solely on *in-scope* data for this task because it is very difficult to collect a set of data that appropriately represents the space of the very unpredictable OOS inputs.

The classes in a traditional text classifier are generally represented by a discrete set of symbols and the classifier is trained with the help of a finite set of examples, where the classes are assumed to be independent and the set of examples to be disjoint. But, in many cases, the classes are in fact associated with inter-connected higher-level concepts which could be formatted into more meaningful representations and better exploited in the classification process for an enhanced representation of the scope of the classifier.

In particular we explore here the use of *graphs* which represent information by means of nodes

[1] *Out-of-domain* examples is also a common term in the literature.



*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3952–3961,
November 16–20, 2020. 

connected to each other by arcs. Recent research has demonstrated that nodes in a graph can be converted to an *embedding*, that is, projected into a vector space, which can then be mapped to sentences to cope with tasks such as the *reverse dictionary problem* (Hill et al., 2016; Kartsaklis et al., 2018). We propose here an adaptation of those ideas to an intent classifier so it uses such mappings to expand the representation of the class space, its scope, and class inter-dependencies, and thus possibly making the OOS detection task easier.

This paper presents an investigation of exploiting information from *word graphs* associated to the intent classes to improve OOS sample detection in intent classification. By considering that each class is represented by a set of text examples and that different classes can be connected to each other by means of the repeated occurrence of words in their respective examples, we build a word graph where both class labels and words are represented by single nodes. The word nodes are connected to the class label nodes in accordance to their occurrence in the training samples and their respective class labels. Then, a typical graph embedding technique is used to represent classes with the embedding of their corresponding class label node. Instead of finding the classes with the highest probability, the intent classifier search for the class embedding which maps best to the sentence embedding of a given input sample.

We have implemented and tested this idea with different types of base methods for sentence embedding, such as *Long-short Term Memory (LSTM)* neural networks and *Bidirectional Encoder Representations from Transformers (BERT)*, and performed OOS detection by means of a simple threshold-based rejection. We conducted a thorough evaluation on both private and public intent classification datasets, such as the *Larson dataset* for this specific task (Larson et al., 2019).

Our results show that the proposed word-graph based method improves considerably OOS detection, compared against the corresponding traditional classification algorithms, based on combining the sentence embedding algorithm with *softmax* probabilities. In the case of the Larson dataset, where comparison against varied OOS detection methods is available, we show that our proposed approach reduces dramatically the previous state-of-the-art (SOTA) false acceptance rate in more than 30 percentage points, from 41.1% to 9.9%.

## 2 The Word Graph Method

This section presents a formal description of the methodology employed in this work.

### 2.1 Embedding the Set of Classes

An *intent classification* method is a function $D$ which maps a set of sentences (potentially infinite) $S = \{s_1, s_2, ...\}$ into a finite set of classes $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$:

$$D : S \rightarrow \Omega \qquad D(s) = \omega_i \tag{1}$$

To enable a numeric, easier handling of the input text, an embedding $\xi : S \rightarrow \mathbb{R}^n$ is often used, mapping the space of sentences $S$ into a vector space $\mathbb{R}^n$, and defining a classification function $E : \mathbb{R}^n \rightarrow \Omega$ such as $D(s) = E(\xi(s))$. In typical intent classifiers, $E$ is usually composed of a function $C$ which computes the probability of $s$ being in a given class, followed by the *arg max* function. In many intent classifiers, $C$ is the *softmax* function.

$$S \overset{\xi}{\rightarrow} \mathbb{R}^n \overset{C}{\rightarrow} \mathbb{R}^c \overset{argmax}{\rightarrow} \Omega \tag{2}$$

This paper explores how to use embeddings in the other side of the classification functions, that is, by embedding the set $\Omega$ of classes into another vector space $\mathbb{R}^m$. The idea is to use class embedding functions which somehow capture better inter-class relations such as similarities, using, for instance, information from the training sets, as we will show later. Formally, we use a *class embedding* function $\psi : \Omega \rightarrow \mathbb{R}^m$, its inverse $\psi^{-1}$, and a function $M : \mathbb{R}^n \rightarrow \mathbb{R}^m$ to map the two vector spaces so $D(s) = \psi^{-1}(M(\xi(s)))$.

$$S \overset{\xi}{\rightarrow} \mathbb{R}^n \overset{M}{\rightarrow} \mathbb{R}^m \overset{\psi^{-1}}{\rightarrow} \Omega \tag{3}$$

In this paper we use typical sentence embedding methods to implement $\xi$. To approximately construct the function $M$ we employ a basic *Mean Square Error* (MSE) method using the training set composed of sentence examples for each class $\omega_i \in \Omega$. As we will see next, the training set will also be used to construct the embedding function for the set of classes $\psi$ and an approximation for its inverse $\psi^{-1}$.

### 2.2 Adapting *Kartsaklis* Method (LSTM)

In this paper we explore a text classification method proposed for the *inverse dictionary problem*, where text definitions of terms are mapped to the term they define, proposed by Kartsaklis et al. (2018).

The embedding of the class set into the continuous vector space (equivalent to the $\psi$ function in equation 3) is done by expanding the knowledge graph of the dictionary words with nodes corresponding to words related to those terms and performing random walks on the graph to compute graph embeddings related to each dictionary node, using the *DeepWalk* algorithm (Perozzi et al., 2014). Notice that DeepWalk is a two-way function mapping nodes into vectors and back.

An LSTM, composed of two layers and an attention mechanism, is used by Kartsaklis et al. (2018) for mapping the input texts to the output vector space. To map the two continuous vector spaces representing the definition texts and the dictionary terms, a MSE function, learned from the training dataset, is used. This approach achieves SOTA results on the reverse dictionary task and also in other tasks such as document classification and text-to-entity mapping.

In this work, the approach from Kartsaklis et al. (2018) is employed for mapping the classes into a vector space, although we do not use a knowledge graph as described later. Instead, we create a *word graph* $G$ by associating each class to a node and connecting to each of them nodes which correspond to words in the sentences of the training set of each class. We represent this by the function $\zeta$, such as $\zeta(\Omega) = G$, which is also invertible. Substituting this in equation 3,

$$S \overset{LSTM}{\rightarrow} \mathbb{R}^n \overset{MSE}{\rightarrow} \mathbb{R}^m \overset{DeepWalk^{-1}}{\rightarrow} G \overset{\zeta^{-1}}{\rightarrow} \Omega \quad (4)$$

In practice, we compute the mapping from the class embedding space into the class set, called here $InvG : \mathbb{R}^m \rightarrow \Omega$, simply by computing the distance $d$ between a point in $\mathbb{R}^m$ and the inverted projection of each class from $\Omega$, and considering the closest class. That is, for each $w_i \in \Omega$, we consider the associated node in $G$, and compute the mapping in $\mathbb{R}^m$ of that node, as shown here:

$$InvG(x) = \arg\min_{w_i} d(x, DeepWalk(G(w_i)) \quad (5)$$

By substituting this function into equation 4, we obtain the algorithm we call here *LSTM+*:

$$S \overset{LSTM}{\rightarrow} \mathbb{R}^n \overset{MSE}{\rightarrow} \mathbb{R}^m \overset{InvG}{\rightarrow} \Omega \quad (6)$$

For comparison, the traditional corresponding classification method is tested, where the word graph embedding and associated functions are replaced by discrete *softmax* outputs. We call this simply *LSTM*:

$$S \overset{LSTM}{\rightarrow} \mathbb{R}^n \overset{softmax}{\rightarrow} \mathbb{R}^c \overset{argmax}{\rightarrow} \Omega \quad (7)$$

### 2.3 Replacing the LSTM with BERT

The natural language processing community has been recently focusing attention on the novel *transformer* models (Vaswani et al., 2017). This is due to the great performance improvement in several complex tasks, such as machine translation, question answering, and text classification. Moreover, such a performance is achieved without the use of convolutions or recurrence in neural networks. By using only the attention mechanism, models are built with lower computational costs, enabling the rapid development of larger and stronger models, which have been achieving SOTA performance in many different tasks.

BERT is one of such models (Devlin et al., 2019). It is a language representation model pre-trained on unlabeled text and conditioned on both the left and right contexts. Therefore, a simple output layer can be fine-tuned to attain strong results in many different tasks. BERT is employed in this paper with the word graph embedding layer (identical to the one in LSTM+). We call this algorithm *BERT+*:

$$S \overset{BERT}{\rightarrow} \mathbb{R}^n \overset{MSE}{\rightarrow} \mathbb{R}^m \overset{InvG}{\rightarrow} \Omega \quad (8)$$

Like in the previous case, we also use the BERT algorithm with traditional discrete softmax outputs for comparison, called here *BERT*:

$$S \overset{BERT}{\rightarrow} \mathbb{R}^n \overset{softmax}{\rightarrow} \mathbb{R}^c \overset{argmax}{\rightarrow} \Omega \quad (9)$$

### 2.4 Replacing the LSTM with TFIDF

The *term frequency-inverse document frequency* (TFIDF) indicates the importance of a word given its frequency in a document from a corpus (Han et al., 2011). With this statistic, it is possible to detect key words which play important roles in a given document, adjusting to the fact that several words frequently appear in the corpus. Such a technique has been used to generate features in many NLP tasks.

TFIDF is used in this work with an additional output-dense, feed-forward network layer in two different approaches. In the first one, it uses linear outputs for regression with the word graph representation, using the Kartsaklis et al. (2018)-inspired algorithm exactly as we did for the LSTM and

BERT algorithms. We call this algorithm *TFIDF+*. To compare, the feed-forward layer is configured with discrete softmax outputs called *TFIDF*.

### 2.5 Replacing the LSTM with Average Embedding

*Average word embeddings* is also used in this work. In particular, *Glove* (Pennington et al., 2014) is employed for embedding each word of the document. Subsequently, the average of the word embeddings is computed to generate the sentence features. Such an average is computed according to equation 10, where $\overline{x}_j$ is the average embedding for sentence $j$ given the embedding of each of its $N$ words $x_{ij}$. Finally, the computed features are inputted to a regression or a classification dense feed-forward networks, similarly to the previous approaches.

$$\overline{x}_j = \frac{1}{N} \sum_{i=1}^{N} x_{ij} \tag{10}$$

As before, we consider a version where we use average word embeddings in substitution of the LSTM in algorithm LSTM+, called *EMB+*, and also a discrete version using softmax, *EMB*.

### 2.6 Out-of-scope Sample Detection

A rejection mechanism based on a pre-defined threshold is used for OOS detection. This method can be easily applied to all of the methods described previously, without the need neither for any specific training procedure nor OOS training data.

In greater detail, suppose that for each class $\omega_i \in \Omega$ there is a score denoted $\phi_i \in Z$, where $|Z| = |\Omega|$. Given that $\max(Z)$ represents the highest score associated to a class, and that a rejection threshold $\theta$ has been defined on a validation set, samples can be classified as OOS whenever $\max(Z) < \theta$, and they are simply rejected, i.e. no classification output is produced for them. Otherwise, the sample is considered as an *in-scope (IS)* sample and the classification is conducted normally.

In this work, the scores in $Z$ are represented either by the softmax probability computed for LSTM, BERT, TFIDF, and EMB, or by the similarity of sentence and graph embeddings for LSTM+, BERT+, TFIDF+, and EMB+. For the latter, the similarity is computed by means of the dot product between those two embeddings.

## 3 Experimental Evaluation

We performed a comparative evaluation of the performance of the classifiers described in the previous section using a public dataset described in (Larson et al., 2019) called here the *Larson* dataset[2]; a real dataset from a finance chatbot; and a pool of 40 datasets in two different languages from chatbots built using the same platform, to check for the reproducibility of the results from the finance chatbot dataset.

For those experiments, the methods were implemented as follows. For DeepWalk, the embedding size was set to 150, and the walk sizes to 20, for undirected graphs. For LSTM and LSTM+, we considered word embeddings with 200 elements, output sentence embeddings of size 150, and both methods were trained for 50 epochs. For both mapping sentence to graph embeddings and the softmax classifiers, we trained two-layer neural networks with 800 hidden neurons for 1,000 epochs on Larson dataset, and 300 hidden neurons for 20 epochs on the other datasets. Those parameters were set after preliminary evaluations.

### 3.1 Evaluation Metrics

We take into account a commonly-used metric for OOS dectection, i.e. *equal error rate (EER)* (Lane et al., 2007; Ryu et al., 2017, 2018; Tan et al., 2019), which corresponds to the classification error rate when the threshold $\theta$ is set to a value where false acceptance rate (FAR) and false rejection rate (FRR) are the closest. These two metrics are defined as:

$$FAR = \frac{\text{Number of accepted OOS samples}}{\text{Total of OOS samples}} \tag{11}$$

$$FRR = \frac{\text{Number of rejected IS samples}}{\text{Total of IS samples}} \tag{12}$$

In addition, *in-scope error rate (ISER)* is considered to report IS performance, i.e. the accuracy considering only IS samples, as the class error rate in (Tan et al., 2019). This metric is important to evaluate whether the alternative classification methods are able to keep up with the performance of their counterparts in the classification task.

### 3.2 Results on the Larson Dataset

In this section we present an evaluation on the Larson dataset (Larson et al., 2019), a recently proposed dataset which has been specifically designed to cope with intent classification and, most importantly, dealing with rejection of OOS samples, which is referred in the paper as out-of-scope queries. There is a total of 22,500 in-scope samples,

[2]https://github.com/clinc/oos-eval

| Method | EER | FAR | FRR | ISER |
|---|---|---|---|---|
| LSTM | 13.4 | 23.7 | 16.4 | 12.2 |
| BERT | 12.8 | 35.3 | **10.1** | **6.7** |
| TFIDF | 13.7 | 17.7 | 17.3 | 11.7 |
| EMB | 18.0 | 22.8 | 22.8 | 18.1 |
| LSTM+ | 11.7 | 20.8 | 11.9 | 8.4 |
| BERT+ | **9.8** | **9.9** | 12.4 | 7.4 |
| TFIDF+ | 19.2 | 29.5 | 26.4 | 24.2 |
| EMB+ | 31.7 | 29.1 | 34.3 | 58.2 |

Table 1: Results on Larson dataset (in %, the lower the better), for both out-of-scope and in-scope samples: equal error rate (ERR), false acceptance rate (FAR), and false rejection rate (FRR); and only in-scope samples: class error rate (ISER).

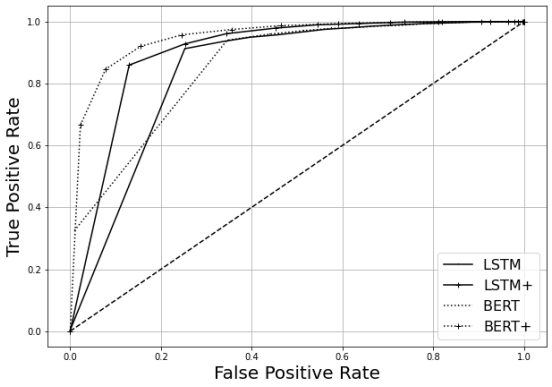


Figure 1: ROC curves on Larson dataset

evenly distributed across 150 classes, and 1,200 out-of-scope samples. From that, the in-scope samples are divided into 18,000 samples for training, and 4,500 samples for test. From the OOS samples, we take only the same 1,000 examples used in (Larson et al., 2019) for test for a direct comparison.

Table 1 presents a summary of the results on this dataset. We observe that the proposed word graph-based methods making use of LSTM and BERT sentence embeddings are able to outperform their corresponding softmax versions, where BERT+ achieves the lowest EER with 9.8% and the FAR value of 9.9%, beating SOTA results by at least 30 percentage points.

In fact, Larson et al. (Larson et al., 2019) reports the best OOS recall as 66.0%, which is equivalent to an FAR of 34%. However, in the setting where no OOS sample is used for training, the reported FAR value is of 41.1% and our approach achieved 31.2 percentage points below that value.

BERT presents the best ISER, meaning that it is the best type of method for classifying in-scope samples. However, the method does not cope well with out-of-scope examples, and results in the highest values for FAR and that negatively affects the final error rate. Overall, as depicted in Figure 1, the graph-based methods tend to produce systems with larger ROC under-the-curve areas, i.e. better systems overall.

We note also that TFIDF+ and EMB+ had poorer performance than TFIDF and EMB, respectively, which we believe owns mainly to the considerably higher ISER presented by the former. We have evaluated several configurations for those methods but we have not been able to achieve lower values for ISER, what may indicate that it might be more difficult to make use of such types of sentence embeddings in the proposed framework.

### 3.3 Results on the Finance Dataset

This dataset uses data extracted from a real chatbot of a large financial institution from Brazil, called here the *Finance* dataset. The chatbot has content related to products and concepts associated both to the institution and finance in general. This set contains a total of 8,823 examples in Brazilian Portuguese language, split into 6,176 for training and 2,647 for test, distributed over a total of 285 classes.

Besides the different language, this dataset allows us to complement the evaluation of the previous section with an unbalanced dataset. The number of samples per class is non uniform, where most of the classes (87%) contain less than 47 samples, but there is one class with a very large number of examples (1,189), and some classes with as few as 2 samples.

In addition, this dataset has not been conceived to deal with OOS samples. For this reason, we had to create a simulation of such scenario by removing 85 randomly-selected classes and their corresponding samples from the training set and then considering all test samples associated to the removed classes as OOS samples. We repeated that procedure five times to come up with five different samplings of OOS classes for a better statistical analysis. The resulting training set sizes vary from 3,383 to 4,796 samples and the corresponding test sets contain about 35% of OOS samples on average. Results hereafter present an average over the results of the five samplings.

The results are presented in Table 2. LSTM+, with the proposed use of word graphs, achieved the best results in the four metrics. It is interesting not only that EER improves compared with LSTM, from 25.7% to 19.2%, but also that ISER

| Method | EER | FAR | FRR | ISER |
|---|---|---|---|---|
| LSTM | 25.7 ±4.6 | 32.2 ±2.4 | 32.0 ±7.8 | 17.5 ±2.7 |
| BERT | 24.6 ±3.3 | 29.2 ±5.3 | 33.4 ±3.3 | 17.9 ±1.9 |
| TFIDF | 22.0 ±1.1 | 26.9 ±4.4 | 26.5 ±3.5 | 43.8 ±8.0 |
| AVG | 25.8 ±2.7 | 30.5 ±4.0 | 33.5 ±5.5 | 48.5 ±8.2 |
| LSTM+ | **19.2 ±1.6** | **20.6 ±5.7** | **22.2 ±2.8** | **9.5 ±1.3** |
| BERT+ | 22.2 ±1.6 | 23.0 ±2.5 | 28.2 ±2.3 | 17.2 ±1.7 |
| TFIDF+ | 24.8 ±0.8 | 28.5 ±2.3 | 35.8 ±1.7 | 53.3 ±7.1 |
| AVG+ | 32.3 ±2.4 | 45.6 ±2.8 | 33.7 ±5.0 | 65.0 ±7.0 |

Table 2: Results on Finance dataset (in %, the lower the better), for both out-of-scope and in-scope samples: equal error rate (ERR), false acceptance rate (FAR), and false rejection rate (FRR); and only in-scope samples: class error rate (ISER).

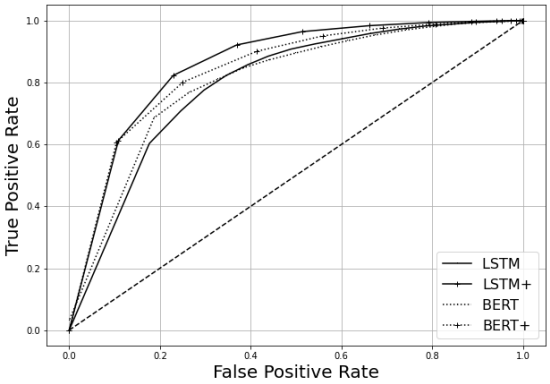


Figure 2: ROC curves on the Finance dataset.

also improves significantly by 6 percentage points. In the case of the BERT-based algorithms, the difference in ISER is much smaller with an improvement of only 0.7 percentage points. In general, the word graph-based BERT+ results in a better system than the softmax counterpart, i.e. BERT, where the former achieves an EER of 22.2% while the latter achieves 24.6%. And the better performance of LSTM+ and BERT+ against LSTM and BERT, respectively, is confirmed by the ROC curve in Figure 2.

Similar to the results on Larson, TFIDF+ and AVG+ presented higher EER than TFIDF and AVG, respectively. The dramatic decreases in ISER show that those word-graph implementations work poorly as classifiers for in-scope samples and we believe that this directly affects the performance on the other metrics. In our opinion, such results indicate that one requirement to benefit from using word graphs to enhance class representations is to make use of sentence embeddings which produce an intent classifier which has an ISER at least comparable to that of softmax-based classifiers. Otherwise, the benefits of the proposed approach are negatively affected by the error which seem be introduced by the in-scope cases.

| Dataset | #Samples | #In-scope classes | Median samples per in-scope class |
|---|---|---|---|
| 1 | 13600 | 966 | 9.0 |
| 2 | 13064 | 75 | 107.0 |
| 3 | 12733 | 76 | 105.5 |
| 4 | 12948 | 206 | 38.0 |
| 5 | 12916 | 205 | 38.0 |
| 6 | 11905 | 196 | 38.0 |
| 7 | 12316 | 75 | 105.0 |
| 8 | 7252 | 63 | 75.0 |
| 9 | 8596 | 64 | 96.0 |
| 10 | 8389 | 91 | 61.0 |
| 11 | 12727 | 76 | 105.5 |
| 12 | 8657 | 63 | 100.0 |
| 13 | 11293 | 137 | 47.0 |
| 14 | 11120 | 137 | 45.0 |
| 15 | 12324 | 75 | 105.0 |
| 16 | 8042 | 307 | 16.0 |
| 17 | 7851 | 302 | 16.0 |
| 18 | 22520 | 20 | 502.0 |
| 19 | 18751 | 91 | 108.0 |
| 20 | 12722 | 76 | 105.5 |

Table 3: Characteristics of the English chatbot datasets.

| Dataset | #Samples | #In-scope classes | Median samples per in-scope class |
|---|---|---|---|
| 21 | 24377 | 33 | 252.0 |
| 22 | 14416 | 31 | 272.0 |
| 23 | 15899 | 271 | 38.0 |
| 24 | 22330 | 384 | 15.0 |
| 25 | 22426 | 468 | 13.0 |
| 26 | 23215 | 530 | 13.0 |
| 27 | 14417 | 31 | 272.0 |
| 28 | 22426 | 468 | 13.0 |
| 29 | 23215 | 530 | 13.0 |
| 30 | 14280 | 169 | 60.0 |
| 31 | 18755 | 351 | 42.0 |
| 32 | 16578 | 393 | 13.0 |
| 33 | 19812 | 397 | 15.0 |
| 34 | 20884 | 390 | 15.0 |
| 35 | 18428 | 336 | 42.0 |
| 36 | 18728 | 425 | 13.0 |
| 37 | 16806 | 390 | 13.0 |
| 38 | 14838 | 6 | 1773.5 |
| 39 | 17046 | 7 | 1732.0 |
| 40 | 19110 | 378 | 39.0 |

Table 4: Characteristics of the Brazilian Portuguese chatbot datasets.

### 3.4 Results in a Pool of Chatbots

Considering the diversity of ways in which intents are defined in professional chatbots, we scaled up our evaluation on multiple chatbots datasets obtained from a dialogue engine platform provider. Those datasets were made available by their developers to be used in improving the performance of the engine but no personal or private information was accessed by us.

In total, 40 datasets, 20 in English (EN) and 20 in Brazilian Portuguese (PT-BR) languages, were used for this experiment. The number of samples per dataset varies from 7,851 to 40,474, while the number of in-scope classes ranges from 6 to 966. For all data sets, the ratio of OOS samples is defined to be close to 20%, resulting in a median num-

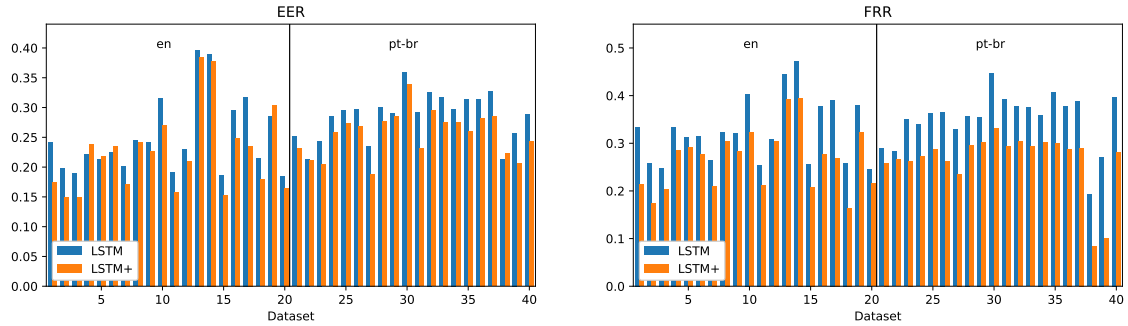


Figure 3: EER and FRR for LSTM and LSTM+ on the Chatbots' datasets.

ber of samples per in-scope class ranging from 9 to 502. In the experiment, we randomly assigned the classes into five different training and test datasets for a better statistical overview. Detailed numbers are provided in Table 3 and Table 4, where it can be noticed the diversity in terms of the number of classes and median samples per class among the the different chatbots.

Figure 3 and Figure 4 present plots of the results comparing LSTM+ vs LSTM and BERT+ vs BERT, respectively, considering two metrics, i.e. EER and FRR, to provide us an idea of the overall performance of the classifiers and the number of examples which are wrongly not rejected. We observe that LSTM+ generally produces lower EER and FRR in general and the statistical significance has been confirmed with the non-parametric paired Wilcoxon's signed rank test (Corder and Foreman, 2009). The mean EER and FRR values presented by LSTM+ were of 24.0% and 26.6%, respectively, and those presented by LSTM were of 26.7% and 33.8%. For BERT+ and BERT, the results show that BERT+ generally produces statistically-significant lower EER and FRR than BERT in the EN datasets, with mean values of 25.5% and 30.2%, respectively, against 26.5% and 34.5%. For PT-BR, though, no statistical difference has been found in EER, with mean EER of 29.3% for BERT+ and 28.6% for BERT. But BERT+ achieves statistically-significant lower FRR than BERT, with a mean of 30.1% of the former versus 38.8% of the latter.

Even though BERT+ has not significantly outperformed BERT in some scenarios, such as with PT-BR chatbots, we can observe a great improvement that the word graphs can bring to intent recognition if we take into account the ISER metric. That is, the difference in ISER of BERT+ against BERT is of 5%, where the former achieved 37% and the latter 32%. In other words, BERT+ can be considered quite worse than BERT for in-scope only intent classification. But, although BERT+ has not been significantly better than BERT with PT-BR chatbots, the proposed word graph-based approach had a great impact in reducing that 5% difference, since both present similar EER values, and still had a huge impact in FRR rates since BERT+ presented significantly better values. Thus, it is likely that by improving the mapping of sentence and graph embeddings for those datasets, and consequently reducing that 5% gap in ISER, BERT+ will stand out as a significantly better approach than BERT.

## 4 Related work

Classification methods, such as those used for intent classification, have been broadly applied to several areas, with the goal of predicting, for an input sample, which of the classes of the problem that sample is associated to. In the case of single-label classification, the training process consists of approximating a probability function, for instance a softmax function for neural networks, by using as reference an one-hot-encoding representation of class labels (Bishop, 2006).

OOS sample detection is a problem which may be critical for intent recognition in chatbots, so that applying rejection mechanisms are important for detecting those cases (Feng and Lin, 2019; Larson et al., 2019; Zheng et al., 2019). Traditional classification can be implemented, for example, by training a specific OOS class to set up a rejection threshold, or even by training a binary classifier (Larson et al., 2019). Given that no specific domain information or structure are taken into account, those methods are roughly the same that have been previously applied for other classification problems (Fumera et al., 2003; Luckner and Homenda, 2014).

Some recent effort has been put specifically for

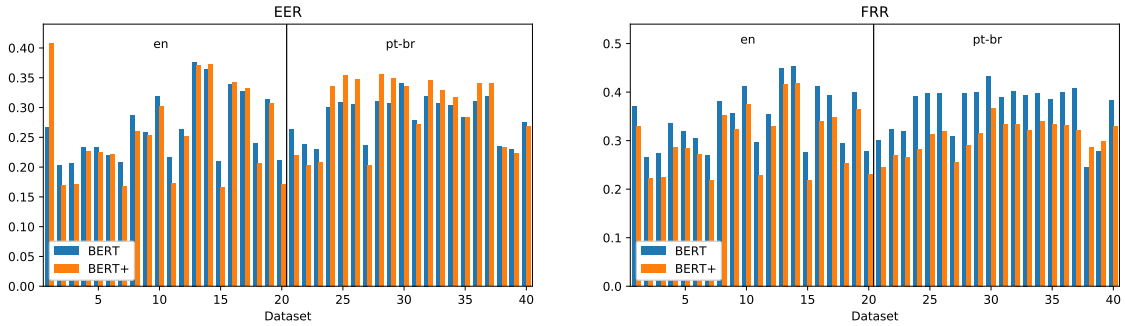


Figure 4: EER and FRR results for BERT and BERT+ on the Chatbots' datasets.

OOS sample detection for intent recognition, either by considering OOS data during the training process (Tan et al., 2019) or solely by improving in-scope sample representation by means of *Auto Encoders* (Ryu et al., 2017) and *Generative Adversarial Neural Networks (GANs)* (Ryu et al., 2018), which is more desirable since there is no reliance on tedious data gathering processes to represent unpredictable OOS inputs. The latter two methods are directly related to ours but, unfortunately, the lack of publicly-available source codes and datasets has made it a challenge to reproduce the methods for a fair direct comparison with ours.

Recently, methods which are able to take advantage of graph information in machine learning models have been proposed. Some of them take advantage at the sample level, such as label propagation (Bui et al., 2018). Others, though, take advantage of graphs at concept level, such as in (Hill et al., 2016; Kartsaklis et al., 2018; Prokhorov et al., 2019). Hill et al. (2016) demonstrate the sentence embeddings could be mapped onto graph embeddings, in reverse dictionary-like problems. Following, Kartsaklis et al. (2018) demonstrated that textual features can improve considerably such a mapping. Those findings have opened an opportunity to enhance class modeling and hopefully better define the scope of a classifier, in special intent classifiers, since classes can be easily represented in a graph space by means of their relationship with individual words extracted from the training samples as we did in this paper.

The previously-mentioned research have been put in practice mostly by advances in sentence embedding (Collobert et al., 2011; Pagliardini et al., 2018) and graph embedding techniques (Cai et al., 2018). Some of them are directly inspired by advances in word embeddings and convolutional neural networks, such as *DeepWalk* (Perozzi et al., 2014) and *Node2Vec* (Grover and Leskovec, 2016).

## 5 Final Remarks

In this paper we propose the use of information from word graphs to enhance intent classification, more specifically, for the detection of out-of-scope examples. Instead of working on the representation of the input text, we enhance the representation of the outputs, i.e. how classes and their corresponding labels are represented. The results demonstrate the approach has a considerable positive impact for the detection of out-of-scope examples when an appropriate sentence embedding such as LSTM and BERT is used. In the publicly-available Larson dataset, the proposed approach beats the previously-published results by a high margin, and particularly enhancing the false acceptance rate (FAR) from 41.1% to 9.9%.

In our view, the improved results are due to a better representation of the higher-level concepts associated to the classes. By connecting the intents to lower-level entities, i.e. the words associated to the intents, and therefore establishing inter-connections between the classes, the word graph space enriches the traditional representation of classes by means of classifier parameters which are learned solely from input examples.

We believe that the approach is general enough to be applied to others areas and presents ideas to develop more accurate classifiers in general, across multiple areas, particularly in contexts where out-of-scope samples are common. In image classification problems, for instance, word graphs related to visual words could be computed. In addition, the proposed word graph method can be improved by exploiting combinations of the proposed expanded class representation with the traditional softmax-based method, what may also provide better accuracy for in-scope samples in some situations.

## References


Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Thang D. Bui, Sujith Ravi, and Vivek Ramavajjala. 2018. Neural graph learning: Training neural networks using graphs. In *Proceedings of 11th ACM International Conference on Web Search and Data Mining (WSDM)*.

Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Gregory W Corder and Dale I Foreman. 2009. *Nonparametric Statistics for Non-Statisticians*. USA: John Wiley & Sons, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yueqi Feng and Jiali Lin. 2019. Enhancing out-of-domain utterance detection with data augmentation based on word embeddings. Arxiv pre-print: 1911.10439.

Giorgio Fumera, Ignazio Pillai, and F. Roli. 2003. Classification with reject option in text categorisation systems. pages 582– 587.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data mining: concepts and techniques*. Elsevier.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Dimitri Kartsaklis, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Mapping text to knowledge graph entities using multi-sense LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1959–1970, Brussels, Belgium. Association for Computational Linguistics.

I. Lane, T. Kawahara, T. Matsui, and S. Nakamura. 2007. Out-of-domain utterance detection using classification confidences of multiple topics. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):150–161.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Marcin Luckner and Wadysaw Homenda. 2014. Pattern recognition with rejection: Application to handwritten digits. In *2014 4th World Congress on Information and Communication Technologies, WICT 2014*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Victor Prokhorov, Mohammad Taher Pilehvar, and Nigel Collier. 2019. Generating knowledge graph paths from textual definitions using sequence-to-sequence models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1968–1976, Minneapolis, Minnesota. Association for Computational Linguistics.

Seonghan Ryu, Seokhwan Kim, Junhwi Choi, Hwanjo Yu, and Gary Geunbae Lee. 2017. Neural sentence

embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems. *Pattern Recogn. Lett.*, 88(C):2632.

Seonghan Ryu, Sangjun Koo, Hwanjo Yu, and Gary Geunbae Lee. 2018. Out-of-domain detection based on generative adversarial network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 714–718, Brussels, Belgium. Association for Computational Linguistics.

Ming Tan, Yang Yu, Haoyu Wang, Dakuo Wang, Saloni Potdar, Shiyu Chang, and Mo Yu. 2019. Out-of-domain detection for low-resource text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3566–3572, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sholom M. Weiss, Nitin Indurkhya, and Tong Zhang. 2012. *Fundamentals of Predictive Text Mining*. Springer Publishing Company, Incorporated.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2019. Out-of-domain detection for natural language understanding in dialog systems. Arxiv pre-print: 1909.03862.