Theoretical and Empirical Advantages of Dense-Vector to One-Hot Encoding of Intent Classes in Open-World Scenarios

Paulo Cavalin and Claudio Pinhanez*
IBM Research, Brazil
{pcavalin,csantosp}@br.ibm.com

Abstract

This work explores the intrinsic limitations of the popular one-hot encoding method in classification of intents when detection of out-of-scope (OOS) inputs is required. Although recent work has shown that there can be significant improvements in OOS detection when the intent classes are represented as dense-vectors based on domain-specific knowledge, we argue in this paper that such gains are more likely due to advantages of the much richer topologies that can be created with dense vectors compared to the equidistant class representation assumed by one-hot encodings. We start by demonstrating how dense-vector encodings are able to create OOS spaces with much richer topologies. Then, we show empirically, using four standard intent classification datasets, that knowledge-free, randomly generated dense-vector encodings of intent classes can yield over 20% gains over one-hot encodings, producing better systems for open-world classification tasks, mostly from improvements in OOS detection.

Keywords: Intent Classification, One-hot Encoding, Out-of-scope Detection, Open-World Classification

1. Introduction

Dense representations of inputs to machine learning (ML) models, often referred to as *input embeddings*, have been one of the key drivers of the massive improvements of performance of most NLP applications in the last 10 years. However, the use of similar *dense-vector* representations for the output classes, or *output embeddings* (Yu and Aloimonos, 2010; Rohrbach et al., 2011; Kankuekul et al., 2012; Akata et al., 2015a), is quite uncommon, except in scenarios of *zero-shot learning* (Romera-Paredes and Torr, 2015) where embeddings are used to encode non-observed latent classes.

The most used method to represent $c > 0$ different intent classes is still to encode them as *one-hot* vectors , i.e. c-dimensional vectors which are all filled with 0s (zeros) except in the position corresponding to the class of the intent label, which is filled with 1 (one). Therefore, classes are represented as equally-distant points in a c-dimensional space and classification is performed by computing the distance to the closest one-hot vector, often using the $softmax$ function to normalize the output.

Recent works (Cavalin et al., 2020; Pinhanez et al., 2021a) have shown that dense-vector representations of intent classes based on domain knowledge can improve intent classification accuracy. In particular, such representations have shown to produce impressive gains in the detection of the inputs to the machine learning system which are outside of the scope of the intent classes, often referred to as *out-of-scope (OOS)* or *out-of-domain (OOD)* samples (Lee et al., 2018b; Vyas et al., 2018; Lee et al., 2018a; Chen et al., 2020). Notice that a classification task without OOS detection in fact assumes a *closed-world* scenario, while the full version, which detects both the correct class for *in-scope (IS)* samples or whether they are OOS samples, corresponds to the *open-world* case. Here we consider only the case where no examples of the OOS class are provided for training, although some are available for testing. Notice also that almost every real, practical use of ML classifiers, such as for intent classification, is in open-world contexts.

The main argument of this paper is that such large accuracy gains in OOS detection are more likely due to advantages of dense-vector to one-hot encoding of intent classes than to the use of domain knowledge. We start by showing that, in open worlds, the complexity of the spaces representable by one-hot encodings is quite limited when compared to dense-vector ones. This argument is made based on the number of different topological spaces enabled by each type of encoding. We show that one-hot encodings of c classes can create only c types of topologically-distinct spaces, while the number of different spaces enabled by dense-vector encodings is larger than c^2.

We follow by presenting the results of some experiments in four intent classification public datasets, a classical open-world scenario, where randomly-generated dense-vectors yielded 8% to 21% improvements in equal error rate (EER), a fundamental metric to compute accuracy in open-world classification, showing that better class topologies can indeed be defined. We observe that better EER values generally came together with decreased false acceptance rates (FAR) but usually lower IS error rates (ISER), indicating that dense-vector encodings present a way to find a balanced

* Both authors contributed equally to this work.

LREC-COLING 2024, pages 16000–16013
20-25 May, 2024.

trade-off between OSS detection and IS accuracy.

In summary, the main contribution of this work is to show that using more powerful representation systems of the output space, i.e., dense-vectors, may have formidable performance impacts in open-world classification tasks and that a likely explanation for this success is their ability to represent more complex topological spaces.

2. Related Work

This paper focuses on text classification tasks which include what is known as *out-of-domain sample detection* (Tan et al., 2019) or *out-of-distribution sample detection* (Lee et al., 2018b; Vyas et al., 2018; Lee et al., 2018a; Chen et al., 2020). A good survey can be found in (Yang et al., 2021). Many of the existing approaches for OOS detection rely on adapting the training algorithm by changing the loss function of a neural network (Lee et al., 2018a); by generating ensembles of classifiers (Vyas et al., 2018; Shalev et al., 2018); by exposing the model to adversarial, crafted inlier and outlier examples (Chen et al., 2020; Li et al., 2021); or by including additional OOS examples either in an unsupervised way (Yu and Aizawa, 2019; Tan et al., 2019) or by generating them (Vernekar et al., 2019). Another approach is to apply some transformation on top the *softmax* outputs to handle better OOS inputs (Hendrycks and Gimpel, 2017; Liang et al., 2018; Techapanurak and Okatani, 2019); or to generate an additional classifier to measure the confidence of the ML classifier, assuming that processing OOS samples have low-confidence scores (Ryu et al., 2017, 2018; DeVries and Taylor, 2018; Lee et al., 2018b).

Instead, our approach focuses on changing the representation of the output layer to try to match it better with the characteristics of the space of intent classes. In many ways, we explore in the output layer one of the most important advances in machine learning, which is the use of *input embeddings* (Turian et al., 2010; Mikolov et al., 2013; Pennington et al., 2014; Dos Santos and Gatti, 2014). In particular, we look into a new use for *output or class embeddings* which have been explored before in other contexts. In particular, in *zero-shot learning* (Romera-Paredes and Torr, 2015), class embeddings have been used as a tool which makes it possible building a solution for the problem. Zero-shot learning is based on the identification and addition of new classes to a classifier with no reliance in input samples. With class embeddings, new classes can be added to the system by simply generating an embedding with the proper configuration of an unseen class, to encapsulate the knowledge of the new concepts (Palatucci et al., 2009; Socher et al., 2013; Akata et al., 2015b,a). Zero-shot learning is a problem closely related to OOS detection but it differs on the criterion of success: in the former, the accuracy of assigning inputs to the previously unknown classes; in the latter, the accuracy of identifying inputs which do not belong to any of the known classes. We believe that some of our arguments may also be valid for zero-shot learning but that is beyond the scope of this work. Here we focus solely in the OOS detection problem and how the utilization of dense encodings affects the classification task.

Recent research has focused on using class embeddings to enhance an ML classifier by encapsulating additional high-level knowledge related to the classes. In Cavalin et al. (2020) the classes were represented by keywords extracted from the class training examples followed by the embedding of the corresponding *word graph*. However, word graphs tend to repeat the class examples with a different structure, thus are far from ideal to produce proper class embeddings. In (Pinhanez et al., 2021a) the hierarchical taxonomy of the classes, as understood by the system developers', was mined from the documentation of the system and used to create class embeddings. Although the latter approach seems promising, such taxonomies might not be available in many cases, limiting the applicability of the method. Notice that those two approaches excelled particularly in OOS detection.

This work explores further the use of class embeddings by looking into the properties of dense-vectors themselves, independent of the presence of knowledge. Our key baseline for comparison is the traditional one-hot encoding methods. Some previous works have explored the difference between one-hot and dense-vectors, such as (Rodríguez et al., 2018), which found higher rates of convergence for the latter. Output embeddings have also been explored in the context of *multi-class classification* problems (Amit et al., 2007; Weinberger and Chapelle, 2009; Weston et al., 2010; Akata et al., 2015a) and *large-scale recognition* (Srivastava and Salakhutdinov, 2013; Deng et al., 2014; Xiao et al., 2014; Yan et al., 2015; Lin et al., 2015). We are not aware of works focusing on OOS detection using knowledge-free class embeddings as described in this paper.

3. Class Encoding for OOS Detection

Following the notation of (Cavalin et al., 2020), 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 $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$:

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

In many practical situations of intent classifica-

tion, it is also necessary to determine whether a sentence s does not belong to any of the classes, what is often referred to as *out-of-scope (OOS) detection* or *out-of-domain (OOD) detection*. This can be represented by expanding Ω to the set $\bar{\Omega} = \Omega \cup \{o\}$ where o represent the class OOS of samples. Following, an *intent classification with OOS detection* method $\bar{D} : S \rightarrow \bar{\Omega}$ is defined by:

$$\bar{D}(s) = \begin{cases} D(s) = \omega_i & \text{if in-scope} \\ o & \text{if out-of-scope} \end{cases} \quad (2)$$

An input 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 $\bar{E} : \mathbb{R}^n \rightarrow \bar{\Omega}$ such as $\bar{D}(s) = \bar{E}(\xi(s))$. In typical intent classifiers, $\bar{E}$ is usually composed of a function M which computes the $z = (z_1, z_2, ..., z_c)$ likelihood of s being in each class ω_i, typically in a finite range such as $[-1, 1]$, followed by a *class encoding* function $\bar{C}$ which maps the likelihood results into the classes in $\bar{\Omega}$.

$$S \xrightarrow{\xi} \mathbb{R}^n \xrightarrow{M} [-1, 1]^c \xrightarrow{\bar{C}} \bar{\Omega} \quad (3)$$

A common way to implement $\bar{C}$, denoted here as $\bar{C}_{max}$, is to verify whether any coordinate z_i is greater than a threshold $0 \leq \theta < 1$ and, if so, to map it into the ω_i associated with the maximum z_i value; otherwise, $\bar{C}_{max}$ maps it into o.

$$\bar{C}_{max}(z, \theta) = \begin{cases} \text{argmax } z_i & \text{if } \max z_i > \theta \\ o & \text{otherwise} \end{cases} \quad (4)$$

Probably the most common method for class encoding is to use the $softmax$ function where the likelihood components are normalized with the exponential funcion, denoted here as $\bar{C}_{softmax}$.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}} \quad \text{thus} \quad \sum_{i=1}^{c} \sigma(z)_i = 1 \quad (5)$$

$$\bar{C}_{softmax}(z, \theta) = \begin{cases} \text{argmax } \sigma(z)_i & \text{if } \max \sigma(z)_i > \theta \\ o & \text{otherwise} \end{cases} \quad (6)$$

Notice that $\bar{C}_{max}$ and $\bar{C}_{softmax}$ can be seen as computing the min-based distance d_{min} of $z = (z_1, z_2, ...z_c)$ to *one-hot* vectors $h_i = (0, 0, ..., 1, ..., 0, 0)$ where the 1 value is in the i-th position of h_i. In this paper we compare those methods with approaches based on Euclidean distance. If instead of min we use the *Euclidean distance* d to the one-hot vectors h_i, we obtain a function which we call $\bar{C}_d$:

$$\bar{C}_d(z, \theta) = \begin{cases} \text{argmin } d(z, h_i) & \text{if } \min d(z, h_i) \leq \theta \\ o & \text{otherwise} \end{cases} \quad (7)$$

Since all previous methods consider which intent class is closer, for a particular distance, to the one-hot vectors, we refer to them as *one-hot class encoding* methods. An alternative approach is to consider each class ω_i as represented by the points closer to a given *dense-vector* $r_i = (r_{i1}, r_{i2}, ...r_{ic}) \in [-1, 1]^c$. We call this function $\bar{C}_r$, defining a typical *dense-vector encoding* method.

$$\bar{C}_r(z, \theta) = \begin{cases} \text{argmin } d(z, r_i) & \text{if } \min d(z, r_i) \leq \theta \\ o & \text{otherwise} \end{cases} \quad (8)$$

The use of dense-vector encoding opens up the exploration of different dimensions beyond c-dimensional encodings as the output of the likelihood function M. In fact, any dimension $p > 0$ can be used, and in this case each class ω_i is represented by the points closer to $r_i^p = (r_{i1}^p, r_{i2}^p, ...r_{ic}^p) \in [-1, 1]^p$, defining the function $\bar{C}_r^p$.

$$\bar{C}_r^p(z^p, \theta) = \begin{cases} \text{argmin } d(z^p, r_i^p) & \text{if } \min d(z^p, r_i^p) \leq \theta \\ o & \text{otherwise} \end{cases} \quad (9)$$

Dense-vector encoding methods can use a variety of ways to generate the r_i^p points to represent the intent classes. We compare experimentally in this paper both random methods to generate such points and the knowledge-informed method based on word graphs described in (Cavalin et al., 2020). But to better understand the differences between using one-hot and dense-vector encodings, we first discuss the different representational expressiveness of each method to handle different cases of component connectiveness of the OOS space.

4. Class Encoding Topologies

We are now ready to demonstrate one of the main contributions of the paper, that is, for any number of classes $c \geq 2$, the one-hot encoding function $\bar{C}_{max}$ defines only one topology, while both $\bar{C}_{softmax}$ and $\bar{C}_d$ define exactly c different topologies. However, for dense-vector encoding methods such as $\bar{C}_r$, the number of different topologies increases at least quadratically with c.

The proof first examines the differences of the class encoding methods in a simplified 2D scenario where the goal is to determine whether a sentence s belongs to one of two classes, ω_1 or ω_2, or is out-of-scope (o). The generalization to high-dimensional spaces is detailed afterwards.

4.1. Class Encoding Topologies in 2D

To simplify the analysis we do not consider here special, limit cases where the intersection of two components degenerates to a single point or to a

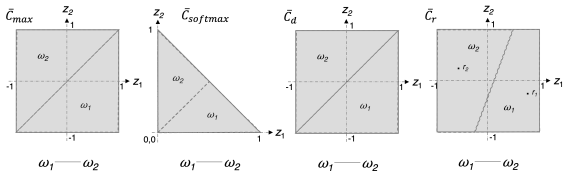

Figure 1: The same topology is generated by $\bar{C}_{max}$, $\bar{C}_{softmax}$, $\bar{C}_d$, and $\bar{C}_r$ in 2D when there is no OOS detection ($\theta = 0$).

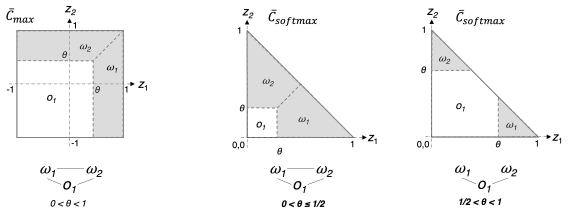

Figure 2: The only $\bar{C}_{max}$ topology and the two topologies of $\bar{C}_{softmax}$ in 2D.

tangent line. Considering this, let us start by examining the simplest situations where there are no OOS detection. This corresponds to the case where $\theta = 0$ in the previous formulas. Figure 1 shows that, in those cases, all the different functions $\bar{C}$ described before can only generate the same topology in which the ω_1 and the ω_2 components are connected, that is, have a non-trivial boundary.

However, when there is OOS detection, that is, $\theta > 0$, a different topological landscape emerges. The leftmost 2D space of fig. 2 illustrates the case of the $\bar{C}_{max}$ function where the class ω_1 maps into a trapezium positioned between $\theta \leq z_1 \leq 1$, bound by the main diagonal of the first quadrant as shown. The class ω_2 similarly maps into a reflected trapezium and the OOS class o_1 occupies a square defined by $-1 \leq z_1 < \theta$ and $-1 \leq z_2 < \theta$. We represent schematically the topology of the connected components defined by $\bar{C}_{max}$ as the leftmost bottom diagram of fig. 2. It shows that the three components are pair-wise connected, for any non-zero value of the threshold θ.

Although quite similar to $\bar{C}_{max}$, the $\bar{C}_{softmax}$ function can represent two distinct topologies, as shown in the central and rightmost parts of fig. 2. This is an effect of the normalization process which maps all points into a triangle in the first quadrant. The first topology is identical to the case of $\bar{C}_{max}$, and it happens if $0 < \theta \leq 1/2$. However, if $\theta > 1/2$, the square of the OOS component o_1 divides the two intent classes into two non-connected triangles, yielding a new configuration where ω_1 and ω_2 are not connected as shown in fig. 2.

Continuing the exploration of one-hot class encodings, fig. 3 shows the effects of substituting max

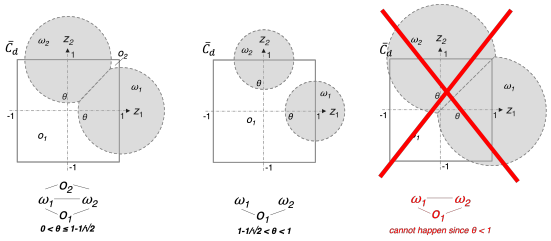

Figure 3: The topologies of $\bar{C}_d$ in 2D. The rightmost topology is not allowed because $\theta \leq 1$.

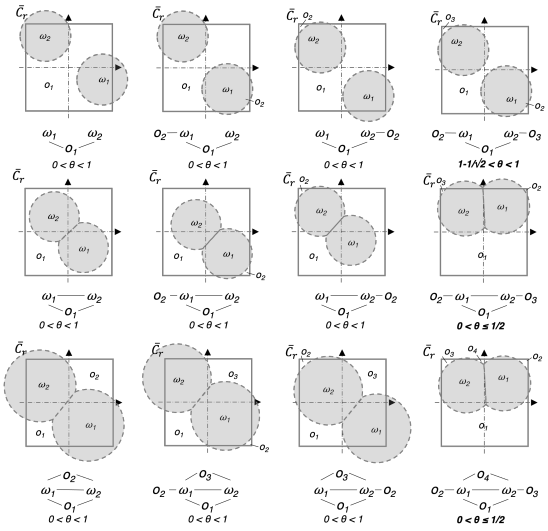

Figure 4: The twelve topologies of $\bar{C}_r$ in 2D.

with the Euclidean distance d, which basically maps w_1 and w_2 into circles of $1-\theta$ radius centered on the one-hot vectors $(1, 0)$ and $(0, 1)$. If $1-1/\sqrt{2} < \theta < 1$ we obtain the case depicted on the center of fig. 3, a topology similar to the second case of $\bar{C}_{softmax}$ where the two intent classes are disconnected and connected to a common OOS component o_1. However, when $0 < \theta \leq 1 - 1/\sqrt{2}$ a new topological configuration is enabled, created by a new unconnected OOS component o_2 corresponding to points with high values of both z_1 and z_2, as seen in the leftmost part of fig. 3. Also, as seen in the rightmost part of fig. 3, it is not possible to have this case a topological configuration where the two classes are pair-wise connected but the OOS space is not split into two components. That would require that the threshold θ to be greater than 1 what is not possible. Notice that although the *sofmax* and Euclidean distance d in one-hot encoding representations allow for different topologies of OOS spaces, both afford just two different topologies which depend on the value of θ.

Figure 4 shows how the use of dense-vector encoding enables many more and richer representations of the output space than the one-hot methods. As depicted, dense-vectors in 2D allow 12 different

topological configurations and the split of the OSS space into 2, 3, and even 4 disconnected components. Also, dense-vector encoding allows for situations where one OOS component is connected to just one of the intent classes, such as in the 6 cases of the two central columns of fig. 4. Notice that each pair corresponds to distinct topologies, since mirroring requires an extra dimension.

This analysis of the 2D scenario shows how potentially limiting is the use of one-hot encoding methods, especially in their ability to represent more complex topological configurations of the OOS space. More important, such analysis holds for higher dimensions of c as we see next.

4.2. Class Encoding Topologies in Higher Dimensions

In this section we show how those results extend to greater dimensions, that is, $c > 2$. As before, we do not consider here special, limit cases where the intersection of two components degenerates to a single point or to a tangent line, and only the cases where OOS detection is needed, that is, $\theta > 0$.

Topologies of $\bar{C}_{max}$: let us consider the $\bar{C}_{max}$ function, with $c > 2$ classes $\omega_1, \omega_2, ...\omega_c$ being recognized, or the symbol o of a OOS input being produced. It is easy to see that the OOS class occupies a c-dimensinal hypercube and each class ω_i a c-dimensional hyper-trapezium. Moreover, each pair of components corresponding to the ω_i and ω_j classes are in contact through the diagonal of the plane defined by the axis of the coordinates z_i and z_j. Similarly, each triad ω_i, ω_j, and ω_k classes share the diagonal of the hyper-cube defined by the coordinates z_i, z_j, and z_k. And the same is valid for every subset $\{\omega_{i_1}, \omega_{i_2}, ...\omega_{i_t}\}$ of Ω, $2 \leq t \leq c$. Therefore, all pair of classes are connected to each other, and so all triads of classes, and so on, in a c-dimensional complete hypergraph, and thus, like in the 2D case, the $\bar{C}_{max}$ function defines only 1 topology for any number of classes $c \geq 2$.

Topologies of $\bar{C}_{softmax}$: let us consider the case of the $\bar{C}_{softmax}$ function. In the 2D case, there were two possible topologies, defined by $0 < \theta \leq 1/2$ and $1/2 < \theta \leq 1$. We show here that, for any c, the $\bar{C}_{softmax}$ defines exactly c different topologies, corresponding to the intervals $0 < \theta \leq 1/c$, $1/c < \theta \leq 1/(c-1)$, ... , $1/2 < \theta \leq 1$. We showed in section 4 that in the case where $c = 2$, there are exactly two topologies, one where all classes are connected when $0 < \theta \leq 1/2$ and one where all classes are disconnected when $1/2 < \theta \leq 1$. Using induction, let us assume that it is true that the number of topologies in the case of $c > 2$ classes is exactly c, defined by the intervals $0 < \theta \leq 1/c$, $1/c < \theta \leq 1/(c-1)$, ... , $1/2 < \theta \leq 1$. Let us consider the case of $c + 1$. First, let us observe that the $1/2 < \theta \leq 1$ in the c case generates a topology where all c classes are disconnected. It is easy to see that this holds in the $c + 1$-dimensional space, and so it goes to all intervals until $1/c < \theta \leq 1/(c-1)$. While the interval $0 < \theta \leq 1/c$ in the c case generated a topology where all classes are fully connected, that is not true in the case in $c + 1$. It is necessary to split the interval $0 < \theta \leq 1/c$ into $0 < \theta \leq 1/(c+1)$ and $1/(c+1) < \theta \leq 1/c$. In the interval $1/(c+1) < \theta \leq 1/c$ all subsets of c components or less are hyperconnected, but only when $0 < \theta \leq 1/(c+1)$, all $c+1$ classes have points in common, corresponding to the additional topology. Since each of the $c + 1$ intervals is associated to a different topology, there are $c + 1$ topologies, completing the induction. Therefore, the $\bar{C}_{softmax}$ function defines c different topologies when the number of classes is c.

To better visualize what is happening in this case, let us examine what happens when $c = 3$. Here O is a cube inscribed in a triangular pyramid where three sides are orthogonal to each other of length 1 and the other face is defined by $x + y + z = 1$. If $\theta > 1/2$ three edges of the cube are outside the $x + y + z = 1$ plane, so the three classes are disconnected. If $\theta < 1/3$, the whole cube is below the $x + y + z = 1$ face, and therefore there are points which belong to the three classes. However, when $1/3 < \theta < 1/2$, exactly one vertex of the cube is outside of the $x + y + z = 1$ plane, so the intersection of the three classes degenerates to a 2D triangle on the $x+y+z = 1$ plane, and therefore there is no 3-class intersection.

Topologies of $\bar{C}_d$: The case of the one-hot softmax is analogous to the one-hot encoding using the Euclidian distance. While in the former the intervals are defined by $1/c$, in the latter they are defined by $1/\sqrt{c}$, or $\sqrt{c}/c$. The same reasoning used in the case of $\bar{C}_{softmax}$ can show that $\bar{C}_d$ defines exactly c different topologies, corresponding to the intervals $0 < \theta \leq 1/\sqrt{c}$, $1/\sqrt{c} < \theta \leq 1/\sqrt{c-1}$, ... , $1/\sqrt{2} < \theta \leq 1$. Therefore, the $\bar{C}_d$ function defines c different topologies when the number of classes is c.

Topologies of $\bar{C}_r$: In section 4 we showed that the $\bar{C}_r$ affords 12 different topologies when $c = 2$. For the sake of the argument of this paper, that is, that dense-vector class encodings enable richer representations of the space of classes, it is not necessary to provide an exact estimation of the topologies defined by $\bar{C}_r$ in the case of c classes. We here simply shows that the number of different topologies in this case grows quadratically with the number of classes by demonstrating that the number of different topologies M_c is equal or greater than the sum $\sum_{m=1}^{c} m = m(m+1)/2$. We have shown that this is true in the case of $c = 2$,

which defines 12 topologies, clearly greater than $\sum_{m=1}^{2} m = 1 + 2 = 3$. Using again induction, let us assume that if the inequality holds for c, that is, if $\bar{C}_r$ defines at least $M_c \geq \sum_{m=1}^{c} m$ different topologies when there are c classes, then it is also true for $c + 1$. To do so, first observe that all M topologies can be extended to the $c + 1$ dimension by considering a value of -1 for the $c+1$ coordinate of each r_i dense vector, $1 \leq i \leq c$. If we define the r_{c+1} to be the one-hot vector of $c + 1$ dimension, it is easy to see that the $c + 1$ component is disjoint with all other components, since $\theta < 1$. Therefore there are at least M_c different topologies defined by the $c + 1$-dimensional $\bar{C}_r$, all of them with the ω_{c+1} component disjoint from the other components. Therefore, we just need to show that there are $c + 1$ other topologies to complete this demonstration. We first show that it is possible to define c different vectors r_{c+1}, denoted here as r^i_{c+1}, in such a way that its associated component is connected to only one of the other vectors. In fact, given $r_i = (r_{i1}, r_{i2}, ...r_{ic})$, $r_{ij} \in [-1, 1]^c$, we can extend all r_i to the $c+1$ dimension by setting the $c+1$ coordinate to $-\theta_i$, $\theta_i < 1$, and for each r_i, consider a vector r^i_{c+1} such as $r^i_{c+1} = (r_{i1}, r_{i2}, ...r_{ic}, \theta_i)$. It is easy to see that each of the c spaces combining the original $r_1, r_2, ...r_c$ dense-vectors to each of the r^i_{c+1} dense-vector has at least one new topology where the component ω_{c+1} is connected to one component $\omega_i, i \leq c$. We can always finding a θ_i which is large enough so the component ω_{c+1} is connected only to the component ω_i, there warranting that all the c encodings constructed this way have different topologies among themselves and also different from any the previous M_c topologies (where the component of ω_{c+1} was always not connected to any other component). We need only to demonstrate that we can create one more topology, different from all of the previous ones, to finish. To do so, let us extend all r_i to the $c + 1$ dimension by setting the $c + 1$ coordinate to 0. If we define r_{c+1} at the center of the space, that is, with all coordinates as 0, it is easy to notice that there is a $0 < \theta < 1$ which the hyper-dimension "ball" of radius θ centered in r_{c+1} intersects with at least two other "balls" of radius θ defined by r_i and r_j, $1 \leq i, j \leq c$. Therefore the set of dense vectors defined this way is guaranteed to define a topology where the component ω_{c+1} is connected to at least two other components, and thus is different from the ones we constructed before. Thus, $M_{c+1} \geq M_c+c+1$, but since $\sum_{i=1}^{c+1} = \sum_{i=1}^{c} +(c+1)$, we obtain $M_{c+1} \geq \sum_{i=1}^{c+1}$. As observed before, $\sum_{i=1}^{c+1} > (c + 1)^2$, yielding $M_{c+1} \geq (c + 1)^2$, and completing the demonstration. Therefore, the $\bar{C}_r$ function defines more than c^2 different topologies when the number of classes is c.

This completes the demonstration that, for any number of classes $c \geq 2$, the one-hot encoding function $\bar{C}_{max}$ defines only one topology; the $\bar{C}_{softmax}$ and $\bar{C}_d$ define exactly c different topologies; and for dense-vector encoding methods such as $\bar{C}_r$, the number of different topologies increases at least quadratically with c, as stated in our analysis in the 2D scenario.

5. Empirical Evaluations of the Encodings

In the previous section we established that dense-vector encoding methods can represent much more complex OOS components and output spaces than one-hot encoding. We now present empirical evidence that dense-vector methods can significantly outperform one-hot encoding methods in OOS detection tasks and that in tasks without OOS detection the gains are small or non-existent.

5.1. The Algorithms

We only used the *Universal Sentence Embeddings (USE)* (Cer et al., 2018) as the main classifier in our experiments. We consider the baseline in our experiments the traditional classification methods which represent a given class symbol in the one-hot encoding format. For simplicity, in our experiments we considered only the *softmax* function $\bar{C}_{softmax}$ for the final step of the classification. We refer to this algorithm as **one-hot softmax**.

Next, we used the **one-hot distance** method which implements one-hot encodings using Euclidean distance, corresponding to the function $\bar{C}_d$. The main idea is to evaluate the impact of switching from *min-softmax* to *Euclidean distance*, which, as we saw, creates different topologically OOS spaces though not increasing the topological count.

As for dense-vector encodings, we evaluated both knowledge-based and random methods. For domain knowledge-based dense-vector encodings, we used the algorithm described by (Cavalin et al., 2020), called here **word graph**. Basically, the graph embedding algorithm *DeepWalk* (Perozzi et al., 2014) was used to generate the dense-vector class embeddings based on a graph composed of nodes to represent the classes linked to keywords extracted from the examples in the training set. The resulting class encodings were the graph embeddings associated to the class nodes.

For the **dense-vector random encodings**, we employed algorithms identical to the one used in *word graph* except that we used N-dimensional random dense-vector class embeddings instead of the word graph embeddings. Classes were represented by N-dimensional vectors filled with values which were randomly sampled from a uniform distribution. We refer to the corresponding algorithms

as **R(N)**. For evaluation purposes, we computed a set of random samples for each *R(N)* and considered both the average and minimum value of the metrics across the samples. This method aims at exploring different topologies that can be defined with randomly-created representations and that can potentially differ from the previous methods.

5.2. Evaluation Metrics

To evaluate the different methods, we used a uniform way to select the key θ value. We considered the fairest way to do so is to use the *equal error rate (EER)* evaluation metrics employed in (Ryu et al., 2017, 2018; Tan et al., 2019; Cavalin et al., 2020; Pinhanez et al., 2021a). In this evaluation metric, the threshold θ is set based on the value where the curves of *false acceptance rate (FAR)* and *false rejection rate (FRR)* intersect. The former metric corresponds to the number of accepted OOS samples divided by the total of OOS samples; the latter represents the ratio between the number of wrongly rejected *in-scope (IS)* cases and the total of IS samples. Additionally, we also took into account *in-scope error rate (ISER)*, which corresponds to the error considering only IS samples and no rejection, i.e. $\theta = 0$, similar to the class error rate in (Tan et al., 2019).

5.3. Experiments and Results

The different algorithms were evaluated in four distinct intent classification datasets, all of them containing tagged OOS examples. These datasets comprise the *HINT3* intent recognition problems (Arora et al., 2020), consisting of three small but unbalanced datasets; and the balanced dataset known as *CLINC150* (Larson et al., 2019), larger than HINT3 both in the number of classes and examples.

5.3.1. HINT3 Datasets

The three HINT3 intent recognition problems (Arora et al., 2020) consist of datasets with actual user queries and real OOS examples. Each dataset is related to a single and unique domain and the datasets possess real-world difficulties such as small and unbalanced training sets and a small number of intents. For the experiments, we considered only the *full* version of each of the three HINT3 datasets: ***SOFTMattress***: with 21 intents, 328 training samples, and 231 IS and 166 OOS test samples; ***Curekart***: with 28 intents, 600 training samples, and 452 IS and 539 OOS test samples; and ***Powerplay11***: with 59 intents, 471 training samples, and 275 IS and 708 OOS test samples.

The neural networks were set with input size of 512, the dimension of the USE vectors; one hidden layer with 1,000 neurons and dropout rate of 0.1; parameters trained with the *Adam optimizer* for 50 epochs; and categorical cross-entropy loss function for *one-hot softmax*, and the mean squared error for the other methods. For the *word graph* method, we created the graph by finding the common words of the training examples as in (Cavalin et al., 2020), and then used *DeepWalk* with the class embedding size set to 200 and walk sizes of 20.

We evaluated 10 different implementations of the $R(N)$ algorithm, with nine N values ranging from 10 to 150, and one N equal to the number of classes c. For each $R(N)$, we trained 500 different randomly-generated class encodings to better explore a relatively large set of dense topologies. Those systems were compared to one training of the *one-hot softmax*, *one-hot distance*, and *word graph* algorithms. We are aware that differences in results can occur since the neural networks weights are randomly initialized but we found, in practice, such differences negligible.

Table 1 shows a summary of the main results, considering the minimum value of $R(N)$ in each metric. The best result for each metric is marked in bold typeface and the best result for the $R(N)$ algorithms in italic. Regarding the minimum EER values reached by the $R(N)$ methods, the results in table 1 show that random sampling is promising towards finding good dense-vector encodings. For SOFTMattress, Curekart, and Powerplay11, the best overall EER values were 0.186, 0.344, and 0.292, respectively from $R(25)$, $R(10)$, and $R(10)$. Those correspond to improvements in EER, compared to *one-hot softmax*, of -28%, -27%, and -23%, respectively.

The improvements in EER seemed to be chiefly due to improvements in the FAR metric. i.e. better OOS detection. Table 1 shows that gains in FAR happened in the three datasets for every N-dimension of $R(N)$, and in $R(30)$, $R(10)$, and $R(10)$ we obtained gains in FAR of -23%, -32%, and -26% to *one-hot softmax*, respectively. The FAR metrics of the best $R(N)$ encodings were not only significantly better than *one-hot softmax*, but also beat easily *one-hot distance* and *word graph* in the three datasets.

The same did not repeat for ISER, as seen in table 1. When considering only the accuracy of classifying IS input in the intent classes, dense-vector encodings were able to perform slightly better in Curekart and Powerplay11, with improvements of -5% and -4% but not in SOFTMattress.

5.3.2. CLINC150 Dataset

We also explored the publicly-available *CLINC150* dataset (Larson et al., 2019), a dataset specifically designed for the evaluation of OOS detection. Unlike the HINT3 datasets, it contains a much larger

	metrics (min)	1-hot encoding		knowledge	dense-vector random encoding								
		softmax	distance	word graph	R(10)	R(15)	R(20)	R(25)	R(30)	R(40)	R(50)	R(100)	R(150)
SOFTMattress	EER	0.259	0.239	0.237	0.212	0.202	0.202	***0.186***	0.199	0.196	0.204	0.202	0.202
	FAR	0.278	0.292	0.271	0.229	0.278	0.229	0.222	***0.215***	0.264	0.229	0.229	0.264
	ISER	**0.198**	0.249	0.213	0.257	0.253	*0.213*	0.285	0.245	0.277	0.249	0.257	0.265
Curekart	EER	0.474	0.432	0.447	***0.344***	0.358	0.371	0.363	0.369	0.364	0.374	0.375	0.379
	FAR	0.522	0.459	0.506	***0.356***	0.375	0.390	0.369	0.388	0.391	0.361	0.386	0.399
	ISER	0.135	0.133	0.163	0.170	0.170	0.144	0.139	0.139	0.148	***0.129***	0.144	0.135
Powerplay11	EER	0.381	0.331	0.318	***0.292***	0.303	0.327	0.304	0.320	0.309	0.324	0.319	0.316
	FAR	0.399	0.341	0.323	***0.295***	0.297	0.329	0.297	0.328	0.310	0.343	0.316	0.307
	ISER	0.337	0.356	0.379	0.421	0.343	0.379	0.333	0.346	0.359	0.359	***0.324***	0.343

Table 1: Summary of the EER, FAR, and ISER results on the HINT3 datasets showing the minimum (min) values. For each metric, the best result is in **bold** typeface; and, among the dense-vector encodings $R(N)$, in *italic*.

metrics (min)	1-hot encoding		knowledge	dense-vector random encoding							
	softmax	distance	word graph	R(10)	R(20)	R(30)	R(40)	R(50)	R(100)	R(150)	R(300)
EER	0.108	0.080	0.080	0.083	***0.070***	0.075	0.075	0.076	0.090	0.090	0.093
FAR	0.165	0.133	0.100	0.191	0.126	***0.095***	0.105	0.119	0.119	0.140	0.161
ISER	0.051	**0.050**	0.050	0.101	0.065	0.058	0.055	0.056	0.053	0.052	*0.052*

Table 2: Summary of the EER, FAR, and ISER results on the CLINC150 dataset showing the minimum (min) values. For each metric, the best result is in **bold** typeface; and, among the dense-vector encodings $R(N)$, in *italic*.

number of samples with a total of 18,000 training samples and 5,500 test samples (4,500 IS and 1,000 OOS), and the number of examples per class is balanced. Moreover, the number of classes in this dataset is quite larger than in the HINT3 with 150 classes, and comprises five different domains unlike the single-domain HINT3 datasets. Also, the error rates reported in CLINC150 are considerably lower, thus it is more challenging to achieve improvements to *one-hot softmax*. Configuration-wise, we used a slightly modified set of values for N, starting in 10 and ending in 300, due to the larger number of classes.

Table 2 shows a summary of the main results, also considering the minimum value of $R(N)$ in each metric. In terms of EER, $R(20)$ was the best encoding, improving *one-hot softmax* by -35% and *one-hot distance* and *word graph* by -13%. The improvement in FAR was even more impressive, from 0.165 of *one-hot softmax* to 0.095 in $R(30)$, a massive gain of -42%, although it was only slightly better than *word graph*. The ISER metric, on the other hand, was 3% worse and for small dimensions it was twice as bad. This indicates that exploiting randomly-defined dense encodings is very likely to make good OOS detectors, although there is some decrease in IS accuracy. Since that is aligned with recent works, arguing that OOS detection and IS accuracy are negatively correlated (Teney et al., 2022), these results provide additional evidence that dense-vectors comprise better topologies for open-world classification settings. But we acknowledge that considerable effort should be put in finding efficiently an optimal topology for each problem, although that is out of the scope of this paper.

6. Conclusions

In this work we provide both theoretical evidence that dense-vector encodings allow much more complex intent spaces in terms of the number of different topologies available to represent the OOS space. We believe our results support the use of dense-vector instead of one-hot encodings in intent classification with OOS detection. Our results also seem to question the need of knowledge-based class encoding as argued in (Cavalin et al., 2020; Pinhanez et al., 2021a). Of course, if such knowledge is available, it may be explored and we have shown here indications that it can be significantly improved. But when knowledge is not present, even the use of a basic random sampling method seems to be a better option than one-hot encodings.

Although we show that there are dense vector

encodings which produce better accuracy (notably in OOS detection), we do not provide an algorithm which actually finds encodings better than one-hot encodings. We would love to have such an algorithm, but so far it has eluded our efforts. We think our research instigates the search for algorithms of such improved/optimal encodings but it may be that finding optimal dense-vector encodings is, in fact, a hard computational problem.

More broadly, our results also suggest that generic ML systems built using one-hot encoding may be improved, perhaps dramatically, in practical applications that need to handle OOS samples by switching to dense-vector encodings, with or without domain knowledge. Investigating further this possibility is imperative in domains other than intent classification, as well as understanding its underlying mechanisms.

7. Limitations

What remains an open question is the way such better topologies can be found. We provided initial results demonstrated that a random-search can lead to improved classifiers, but fail to find better topologies efficiently. Additional search loss functions should be proposed and investigated in the future.

Another limitation is the lack of in-depth evaluation of setting up the architecture and the parameters of our approach, such as the neural architecture which could be tailored to each problem, and a proper value for the threshold θ. In this work the EER copes with the different possibilities of values for θ but in a practical scenario a validation dataset containing both IS and OOS examples is required.

In addition, we acknowledge that there are other baselines which may be stronger that one-hot softmax and have not been included in our empirical evaluation. Nevertheless, they usually include additional mechanisms which affect their performance, requiring those mechanisms to be isolated for a fair comparison. In this work we isolated only the class representation and showed how that impacts the results. The impact of those additional mechanisms, such as the Mahalanobis distance used in (Lee et al., 2018b), should be evaluated further since it can possibly bring additional improvements.

Lastly, given the current ubiquity of Large Language Models (LLMs), we believe that the findings of this paper are applicable to such models, especially when handling classification settings. In this work, we made use of LLMs as off-the-shelf techniques, relying on USE but previous work report comparable results with BERT (Cavalin et al., 2020). Given the greater complexity of such models, additional work should be put on such analysis when fine-tuning is done for new downstream tasks.

8. Bibliographical References

Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. 2015a. Label-embedding for image classification. *IEEE PAMI*, 38(7):1425–1438.

Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. 2015b. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*.

Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. 2007. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*, pages 17–24.

Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. 2020. HINT3: Raising the bar for intent detection in the wild. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 100–105. Association for Computational Linguistics.

Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. 2018. A survey of ontology learning techniques and applications. *Database*, 2018.

Deborah Barreau and Bonnie A. Nardi. 1995. Finding and reminding: File organization from the desktop. *SIGCHI Bulletin*, 27(3):39–43.

Yoshua Bengio. 2017. The consciousness prior. *arXiv:1709.08568*.

Tarek R. Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kuehnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. 2017. Neural-symbolic learning and reasoning: a survey and interpretation. *arXivpreprint arXiv:1711.03902*.

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

John H Boose. 1989. A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition*, 1(1):3–37.

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. 2017. A comprehensive survey of graph embedding: Problems, techniques and applications. *arXivpreprint arXiv:1709.07604*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Paulo Cavalin, Victor Henrique Alves Ribeiro, Ana Appel, and Claudio Pinhanez. 2020. Improving out-of-scope detection in intent classification by using embeddings of the word graph space of the classes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3952–3961. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Shi Kuo Chang. 2001. *Handbook of software engineering and knowledge engineering*, volume 1. World Scientific.

Jiefeng Chen, Xi Wu, Yingyu Liang, Somesh Jha, et al. 2020. Robust out-of-distribution detection in neural networks. *arXiv preprint arXiv:2003.09711*.

Y. Chen, D. Hakkani-Tür, and X. He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *Proc. of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'16)*, pages 6045–6049.

Andrea Civan, William Jones, Predrag Klasnja, and Harry Bruce. 2008. Better to organize personal information by folders or by tags?: The devil is in the details. *Proceedings of the American Society for Information Science and Technology*, 45(1):1–13.

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.

Ernest Davis. 2014. *Representations of commonsense knowledge*. Morgan Kaufmann.

Luc De Raedt, Robin Manhaeve, Sebastijan Dumancic, Thomas Demeester, and Angelika Kimmig. 2019. Neuro-symbolic= neural+ logical+ probabilistic. In *Proc. of the NeSy'19@ IJCAI, the 14th International Workshop on Neural-Symbolic Learning and Reasoning*, Macao, China.

Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. 2014. Large-scale object classification using label relation graphs. In *Proc. of 2014 European Conference on Computer Vision (ECCV'14)*, pages 48–64. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Terrance DeVries and Graham W Taylor. 2018. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

Cicero Dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'14)*, pages 69–78.

Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a definition of knowledge graphs. In *Proc. of 2016 SEMANTiCS (Posters, Demos, SuCCESS)*, volume 48, pages 1–4.

Yueqi Feng and Jiali Lin. 2019. Enhancing out-of-domain utterance detection with data augmentation based on word embeddings. *arXiv preprint arXiv:1911.10439*.

Marco Fossati, Dimitris Kontokostas, and Jens Lehmann. 2015. Unsupervised learning of an extensive and usable taxonomy for dbpedia. In *Proceedings of the 11th International Conference on Semantic Systems (SEM'15)*, Vienna, Austria.

Giorgio Fumera, Ignazio Pillai, and F. Roli. 2003. Classification with reject option in text categorisation systems. In *Proc. of the 12th International Conference on Image Analysis and Processing (ICIAP'03)*, pages 582– 587.

Artur d'Avila Garcez, Marco Gori, Luis C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. 2019. Neural-symbolic computing: An effective methodology for principled integration of

machine learning and reasoning. *arXiv preprint arXiv:1905.06088*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proc.of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, pages 855–864.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, pages 1321–1330, Sydney, Australia. PMLR.

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

Frederick Hayes-Roth. 1984. The industrialization of knowledge engineering. In W. Reitman, editor, *Artificial Intelligence Applications for Business*, pages 159–177. Ablex Norwood, NJ.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of the 2017 International Conference on Learning Representations (ICLR'17)*.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2015. Learning to understand phrases by embedding the dictionary. *arXiv preprint arXiv:1504.00548*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Drew Hudson and Christopher D Manning. 2019. Learning by abstraction: The neural state machine. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS'19)*, pages 5903–5916. Curran Associates, Inc.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*.

William Jones, Ammy Jiranida Phuwanartnurak, Rajdeep Gill, and Harry Bruce. 2005. Don't take my folders away! organizing personal information to get ghings done. In *Proc. of CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA'05)*, page 1505–1508, New York, NY, USA. ACM.

Pichai Kankuekul, Aram Kawewong, Sirinart Tangruamsub, and Osamu Hasegawa. 2012. Online incremental attribute-based zero-shot learning. In *Proc. of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*, pages 3657–3664. IEEE.

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 (EMNLP'18)*, 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'19)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018a. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *Proc. of the Sixth International Conference on Learning Representations (ICLR'18)*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018b. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Proc. of the 2018 Advances in Neural Information Processing Systems (NeurIPS'18)*, pages 7167–7177. Curran Associates, Inc.

Fei-Fei Li and Pietro Perona. 2005. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, page 524–531, USA. IEEE Computer Society.

Xiaoya Li, Jiwei Li, Xiaofei Sun, Chun Fan, Tianwei Zhang, Fei Wu, Yuxian Meng, and Jun Zhang. 2021. kFolden: k-fold ensemble for out-of-distribution detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, pages 3102–3115, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proc. of the Sixth International Conference on Learning Representations (ICLR'18)*.

Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. 2015. Deep learning of binary hash codes for fast image retrieval. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPR'15)*, pages 27–35.

Marcin Luckner and Władysław Homenda. 2014. Pattern recognition with rejection: Application to handwritten digits. In *Proc. of the 4th World Congress on Information and Communication Technologies (WICT'14)*.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. Deepproblog: Neural probabilistic logic programming. In *Proc. of the 2018 Advances in Neural Information Processing Systems (NeurIPS'18)*, pages 3749–3759. Curran Associates, Inc.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, Massachusetts.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *Proc. of the 2019 International Conference on Learning Representations (ICLR'19)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Alessandro Oltramari, Jonathan Francis, Cory Henson, Kaixin Ma, and Ruwan Wickramarachchi. 2020. Neuro-symbolic architectures for context understanding. *arXiv preprint arXiv:2003.04707*.

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 (NAACL'18)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot learning with semantic output codes. In *Proc. of the 2009 Advances in Neural Information Processing Systems (NeurIPS'09)*, volume 22. Curran Associates, Inc.

Emilio Parisotto, Abdel-Rahman Mohamed, Rishabh Singh, Lihong Li, Dengyong Zhou, and Pushmeet Kohli. 2017. Neuro-symbolic program synthesis. In *Proc. of the International Conference on Learning Representations (ICLR'17)*.

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'14)*, 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 (KDD'14)*, pages 701–710.

Claudio Pinhanez, Paulo Cavalin, Victor Henrique Alves Ribeiro, Ana Appel, Heloisa Candello, Julio Nogima, Mauro Pichiliani, Melina Guerra, Maira de Bayser, Gabriel Malfatti, and Henrique Ferreira. 2021a. Using meta-knowledge mined from identifiers to improve intent recognition in conversational systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL'21))*, pages 7014–7027, Online. Association for Computational Linguistics.

Claudio Santos Pinhanez, Heloisa Candello, Paulo Cavalin, Mauro Carlos Pichiliani, Ana Paula Appel, Victor Henrique Alves Ribeiro, Julio Nogima, Maira de Bayser, Melina Guerra, and Henrique Ferreira. 2021b. Integrating machine learning data with symbolic knowledge from collaboration practices of curators to improve conversational systems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI'21)*, pages 1–13.

Victor Prokhorov, Mohammad Taher Pilehvar, and Nigel Collier. 2019. Generating knowledge graph paths from textual definitions using sequence-to-sequence models. *arXiv preprint arXiv:1904.02996*.

Pau Rodríguez, Miguel A Bautista, Jordi Gonzalez, and Sergio Escalera. 2018. Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, 75:21–31.

Marcus Rohrbach, Michael Stark, and Bernt Schiele. 2011. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Proc. of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*, pages 1641–1648. IEEE.

Bernardino Romera-Paredes and Philip Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, pages 2152–2161, Lille, France. PMLR.

Adam Rule, Amanda Birmingham, Cristal Zuniga, Ilkay Altintas, Shih-Cheng Huang, Rob Knight, Niema Moshiri, Mai H Nguyen, Sara Brin Rosenthal, Fernando Pérez, et al. 2018. Ten simple rules for reproducible research in jupyter notebooks. *arXiv preprint arXiv:1810.08055*.

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 Recognitin Letters*, 88(C):26–32.

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 (EMNLP'18)*, pages 714–718, Brussels, Belgium. Association for Computational Linguistics.

Gabi Shalev, Yossi Adi, and Joseph Keshet. 2018. Out-of-distribution detection using multiple semantic label representations. In *Proc. of the 2018 Advances in Neural Information Processing Systems (NeurIPS'18)*, pages 7375–7385. Curran Associates, Inc.

Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *Proc. of the OTM Confederated International Conferences: On the Move to Meaningful Internet Systems*, pages 1223–1237. Springer.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proc. of the 2013 Advances in Neural Information Processing Systems (NeurIPS'13)*. Curran Associates, Inc.

Nitish Srivastava and Ruslan Salakhutdinov. 2013. Discriminative transfer learning with tree-based priors. In *Proc. of the 2013 Advances in Neural Information Processing Systems (NIPS'13)*.

Rudi Studer, V Richard Benjamins, and Dieter Fensel. 1998. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197.

Rudi Studer, Dieter Fensel, Stefan Decker, and V Richard Benjamins. 1999. Knowledge engineering: survey and future directions. In *Proc. of the 1999 German Conference on Knowledge-Based Systems*, pages 1–23. Springer.

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'19)*, pages 3566–3572, Hong Kong, China. Association for Computational Linguistics.

Engkarat Techapanurak and Takayuki Okatani. 2019. Hyperparameter-free out-of-distribution detection using softmax of scaled cosine similarity. *arXiv:1905.10628*.

Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285.

Damien Teney, Yong Lin, Seong Joon Oh, and Ehsan Abbasnejad. 2022. Id and ood performance are sometimes inversely correlated on real-world datasets. *arXiv preprint arXiv:2209.00613*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 384–394.

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 *Proc. of the 2017 Advances in Neural Information Processing Systems (NeurIPS'17)*, pages 5998–6008.

Sachin Vernekar, Ashish Gaurav, Vahdat Abdelzad, Taylor Denouden, Rick Salay, and Krzysztof Czarnecki. 2019. Out-of-distribution detection in classifiers via generation. In *Proc. of the Safety and Robustness in Decision Making Workshop of the 2019 Advances in Neural Information Processing Systems (NeurIPS'19)*.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L. Willke. 2018. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the 2018 European Conference on Computer Vision (ECCV'18)*.

Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. 2019. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–37.

Kilian Q Weinberger and Olivier Chapelle. 2009. Large margin taxonomy embedding for document categorization. In *Proc. of the 2009 Advances in Neural Information Processing Systems (NIPS'09)*, pages 1737–1744.

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

Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine Learning*, 81(1):21–35.

Steve Whittaker, Tara Matthews, Julian Cerruti, Hernan Badenes, and John Tang. 2011. Am I wasting my time organizing email? A study of email refinding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*, page 3449–3458, Vancouver, BC, Canada. ACM.

Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, Yuxin Peng, and Zheng Zhang. 2014. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22nd ACM International Conference on Multimedia (MM'14)*, pages 177–186.

Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. 2015. Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ECCV'15)*, pages 2740–2748.

Bai Yang, Zhang Liping, and Zhao Fengrong. 2019. A survey on research of code comment. In *Proceedings of the 2019 3rd International Conference on Management Engineering, Software Engineering and Service Sciences (ICMSS 2019)*, page 45–51, New York, NY, USA. ACM.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2021. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.

Qing Yu and Kiyoharu Aizawa. 2019. Unsupervised out-of-distribution detection by maximum classifier discrepancy. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV'19)*.

Xiaodong Yu and Yiannis Aloimonos. 2010. Attribute-based transfer learning for object categorization with zero/one training example. In *Proc. of the 2010 European Conference on Computer Vision (ECCV'10)*, pages 127–140. Springer.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP'21)*, pages 870–878, Online. Association for Computational Linguistics.

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

Wenxuan Zhou, Fangyu Liu, and Muhao Chen. 2021. Contrastive out-of-distribution detection for pretrained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, pages 1100–1111, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.