Journal of Intelligent Information Systems (2022) 58:613–640
https://doi.org/10.1007/s10844-021-00693-2

Hierarchy-based semantic embeddings for single-valued & multi-valued categorical variables

Summaya Mumtaz[1] · **Martin Giese**[1]

Received: 12 April 2021 / Revised: 2 December 2021 / Accepted: 2 December 2021 /
Published online: 28 December 2021

Abstract

In low-resource domains, it is challenging to achieve good performance using existing machine learning methods due to a lack of training data and mixed data types (numeric and categorical). In particular, categorical variables with high cardinality pose a challenge to machine learning tasks such as classification and regression because training requires sufficiently many data points for the possible values of each variable. Since interpolation is not possible, nothing can be learned for values not seen in the training set. This paper presents a method that uses prior knowledge of the application domain to support machine learning in cases with insufficient data. We propose to address this challenge by using embeddings for categorical variables that are based on an explicit representation of domain knowledge (KR), namely a hierarchy of concepts. Our approach is to 1. define a semantic similarity measure between categories, based on the hierarchy—we propose a purely hierarchy-based measure, but other similarity measures from the literature can be used—and 2. use that similarity measure to define a modified one-hot encoding. We propose two embedding schemes for single-valued and multi-valued categorical data. We perform experiments on three different use cases. We first compare existing similarity approaches with our approach on a word pair similarity use case. This is followed by creating word embeddings using different similarity approaches. A comparison with existing methods such as Google, Word2Vec and GloVe embeddings on several benchmarks shows better performance on concept categorisation tasks when using knowledge-based embeddings. The third use case uses a medical dataset to compare the performance of semantic-based embeddings and standard binary encodings. Significant improvement in performance of the downstream classification tasks is achieved by using semantic information.

Keywords Embeddings · Categorical data · Knowledge representation

✉ Martin Giese
martingi@ifi.uio.no

Summaya Mumtaz
summayam@ifi.uio.no

[1] University of Oslo, Gaustadalléen 23B, 0373 Oslo, Norway

1 Introduction

In machine learning, standard tasks such as classification and clustering perform well on datasets that contain a large number of training samples. In complex and low-resource domains, the datasets have low sample size and mixed features (numeric and categorical features). Often, categorical features in such scenarios have high cardinality. The combination of these factors makes it challenging to achieve good performance in low-resource domains. A significant amount of prior knowledge is available in many disciplines, often codified in the form of a hierarchy. Our goal is to improve machine learning performance in low-resource scenarios where high-quality structured knowledge (in the form of a hierarchy) is available. This structured knowledge can be easily mapped to the categorical data in the domain.

To process categorical features, they are converted into a vector representation. The statistical literature mostly studies datasets that have categorical variables with low cardinality, such as gender (male, female), weather (sunny, cloudy, rainy), etc. For these datasets, a standard solution is to use one-hot encoding (Potdar et al., 2017) for supervised learning.

One-hot encoding represents a categorical feature with d distinct values $S = \{X_1, \ldots X_d\}$ as d binary features x_i, $i = 1, \ldots, d$. An observation X_j is represented by letting $x_j = 1$ and $x_i = 0$ for $i \neq j$. The vectors resulting from this representation are equidistant in $\mathbb{R}^d$. In contrast, in other vector embedding schemes commonly used in natural language processing (NLP) tasks (Mikolov et al., 2013), the learning process tries to embed the similarities/differences between concepts in the embedding space. There is no contextual or semantic information embedded in one-hot encoding. The existing encoding schemes (Potdar et al., 2017) are not based on the semantic similarity between different values of the categorical variable. This often leads to poor predictions, particularly in use cases where we have a large number d of different values and a small dataset in which the training data does not adequately cover all d possible values. Therefore, such variables tend to be omitted from the standard supervised learning process, which results in models that lack critical information and are not reliable enough (Garchery & Granitzer, 2018).

The key idea of this paper is to use a vector representation for the categorical data that captures part of the prior knowledge about the application domain by exploiting a given structure on the set of categories. Our approach consists of two steps: first, we use the given information about the concepts (typically a hierarchical classification, taken from a standard or a domain ontology) to define a similarity measure on the set of categories S. We then modify the one-hot encoding to take any similarity measure into account in such a way that similar observations lead to similar encoding vectors. This re-establishes the ability to cluster and classify text values that occur rarely or never in the training dataset. Based on the above-mentioned framework, we introduce two embedding schemes for single-valued and multi-valued categorical data (each data instance has multiple feature values or a subset of S), respectively.

Various similarity measures exist in the literature that define the similarity between two domain concepts by using prior domain knowledge. We do not restrict the proposed embedding schemes to a particular similarity measure. Therefore, before demonstrating the viability of the embedding approach for the prediction task, we first describe and compare existing semantic similarity measures.

We demonstrate the viability of the proposed embedding approaches using experiments in two different domains.

– Word Embeddings using the WordNet hierarchy: We create semantic embeddings for words in natural language processing by using various similarity measures. Word

embeddings are evaluated on the task of concept categorisation in NLP on benchmark datasets.
– Patient Mortality Prediction using ICD-9 hierarchy: We take the patient's diagnosis as the main categorical variable in the MIMIC-iii dataset (Johnson et al., 2016) and use ICD-9 classification of diseases as domain knowledge to create categorical encodings.

The paper is organised as follows. Section 2 reviews the state of the art for creating numeric representations from categorical variables. Section 3 introduces the concept of semantic similarity, followed by the definition of the proposed encoding schemes and an overview of existing similarity measures. A comparison of different similarity measures is performed on a standard task of word pair similarity in Section 4. Section 5 defines the scheme for finding word embeddings and evaluation by using the concept categorisation task. In Section 6, we describe the MIMIC dataset, preprocessing steps, and implementation framework. Section 6.6 discusses the experimental study and results for the MIMIC use case. Section 7 discusses the conclusion and future work.

2 Literature review

A survey conducted by Potdar et al. (2017) identified the limited set of encoding schemes available for categorical variables and their impact on machine learning algorithms, particularly neural networks. There are five main types of variable encodings used in practice: one-hot (the most common), ordinal, binary, target, and hashing schemes. These schemes are discussed briefly below.

One-hot encoding creates d input variables for a variable with d distinct categories. It increases the dimensionality of the problem space, and learning rare categories is difficult. The orthogonality of vectors created by the encoding discards any overlap of information that may exist between different values of the categories (Cerda & Varoquaux, 2020). One-hot is unable to assign vectors to new values that may appear in the future or in the test set, even in cases where the test value is similar to the points in the training dataset. Furthermore, it is difficult to interpret the semantics of the categorical variable (Hsu, 2006).

Ordinal encoding assigns a unique integer to each category, provided that the existing categories are known, and there exists some kind of ordering between values. It does not increase the dimensionality of the problem space. However, ordinal encoding is not suitable for categories that do not imply any order (Von Eye & Clogg Clifford, 1996), and a comparative study has shown the negative impact of this encoding scheme on neural network classification (Crone et al., 2006).

Binary encoding first creates ordinal integers and then converts each integer to the equivalent binary code. Digits of binary code are split into columns. This may suit categorical variables that have an order but not variables without inherent order. Even for variables with an inherent order, it leads to many similar binary encodings (i.e. encodings that agree for many of the bits) for semantically very different categorical values (Fitkov-Norris et al., 2012).

Target encoding converts categories to numeric values by estimating the probability of the target attribute. In a classification setting, this probability is equal to the posterior probability of the target, conditioned on the value of the categorical attribute (Micci-Barreca, 2001). This scheme does not increase the dimensionality of the problem space. However, it leaks information from the target variable to the feature set, resulting in overfitting of the data.

Hashing converts strings to a fixed-length vector by using a hash function (Cerda & Varoquaux, 2020). This is useful for high-cardinality variables as it reduces the number of dimensions. The major disadvantage of hashing is the hash collisions, where different categories may fall into the same bucket. If these categories are not related, it affects the accuracy of final predictions. Any similarity or existing inherent order is lost.

Recently, hierarchical couplings-based techniques (Jian et al., 2019; Zhu et al., 2018) have been proposed to learn distance for categorical data. Coupling refers to the interactions between attribute values conditioned on other attributes. Couplings are calculated using conditional probabilities. In data-rich scenarios, these methods perform well to capture relations based on the co-occurrence probabilities of values in a single attribute, between different attributes, and attribute values and target classes. However, in tasks with limited training data and high-cardinality variables, these methods fail because the data do not encompass all possible relationships between categories. This is the case with, for example, industrial applications such as petroleum reservoir recommendation (Mumtaz & Giese, 2020) where the amount of training data is severely limited by the nature of the problem.

Apart from the traditional encoding schemes described, some encoding schemes have been suggested in the context of Natural Language Processing (NLP), based on measuring string similarity between different pairs of categories. Cerda et al. proposed a technique to find common strings in the labels of categories for reducing the high cardinality of dirty nominal data (Cerda et al., 2018). Cerda and Varoquaux proposed two techniques to capture the structural similarity of string entries (Cerda & Varoquaux, 2020). These schemes provide low-dimensional encodings. However, concepts or categories that share common features in a domain may not necessarily have similar string representations.

The existing literature shows the value of adding domain knowledge in various forms to model complex tasks in different machine learning settings. Janusz et al. presented an unsupervised model for learning a semantic similarity measure for documents (Janusz et al., 2012). The model is composed of two main components: the semantic interpreter that maps concepts in the documents to a knowledge base, followed by a similarity function based on derived data. The use of a knowledge base adds predefined semantics to ambiguous words. Marcin Szczuka et al. proposed a strategy to cluster documents based on the content, using the DBpedia knowledge base (Szczuka & Janusz, 2013). The vector representation of documents is improved by using content from the knowledge source, thus resulting in improved clustering.

Domain knowledge in the form of free text can also be used as prior knowledge. Nguyen used a rough approximation framework to add domain knowledge in the form of natural language in classification systems with large feature spaces (Nguyen, 2003). Tarnowska and Ras (2019) and Tarnowska et al. (2020) constructed new categorical attributes from text using folksonomy and sentiment analysis in the business domain.

Several approaches have been suggested to incorporate domain knowledge in the form of a formal domain *ontology*. E.g., Janusz (2014) investigates methods that learn a similarity measure from data (in contrast to our work which takes a similarity measure as input). The 'rule-based similarity model' discussed by Janusz is particularly suited to deal with a high number of interconnected features and it uses an ontology as a model of the relations between the various features attached to a concept (again in contrast to our work which concentrates on the is-a hierarchy between concepts and ignores properties). A learnt similarity measure would be a possible input to our suggested embeddings, but it is not clear whether this would be preferable to using the data directly for a machine learning task. Also, the learning process may still require an amount of data not available in a low-resource situation.

A different way of incorporating an ontology in a classification task was presented by Bazan (2008). Bazan considers learning approximations of concepts (defined through rough set theory). In this approach, the ontology reflects the *hierarchy of definitions* of concepts. E.g. the 'vague' concept of safe driving is defined in terms of other vague concepts like keeping a safe distance, careful overtaking, etc., which can ultimately be described in terms of physical magnitudes and sensor values. This hierarchy of definitions can be used to structure the process of learning the top-level concepts, and should help in dealing with low-resource problems.

Also in the rough set tradition, Midelfart proposed a framework for supervised learning using the Gene Ontology (Midelfart, 2005a, 2005b). Nguyen et al. proposed methods for embedding domain knowledge in the layered learning process to improve the quality of hierarchical classifiers in pattern recognition (Nguyen et al., 2013).

An approach to ontology-based similarity that takes the *is-a* hierarchy into account, like we do, was proposed by d'Amato et al. (2009). While our definitions use only the is-a hierarchy from an ontology, that work defines similarity in terms of the number individuals that belong to a concept, meaning that it relies on the availability of enough data, similarly to the information content based measures described later, and is less suitable in low-resource situations.

The encodings described so far are for features where the value is *one* out of a given set of possibilities. In many applications, the value of a feature is a *set* of categorical values. A recent study (Jia et al., 2019) defined set-level similarity for calculating the distance between sets of clinical taxonomic concepts to measure patient similarity. They divided the entire population of 705 patients into four 'prototypes.' and experimented with different semantic-based similarity techniques to find the correct 'prototypes' group for new patients. This set-based approach takes into account the semantic information for sets of diagnoses. However, it restricts performance by only considering sets of diagnoses and interactions that are present in each 'prototypes' group.

In our current work, we are interested in encoding single and multiple categories based on semantics that can be utilised as input along with other numeric inputs to the existing classification models, such as a neural network. This provides a systematic approach for standard models to find concept interactions for larger datasets.

3 Problem formulation

We first discuss the drawbacks of the traditional encoding schemes in the supervised setting, using the toy example given in Table 1. The table shows data from records of patients admitted to a hospital. The diagnosis column contains values that are taken from a large number N of possible diagnoses, typically more than a thousand. A standard machine learning process will involve the conversion of the diagnosis column into N one-hot encoded columns. This representation models all categories as mutually exclusive. However, the links between various diagnoses are not random. For instance, patients with Angina ad patients with heart failure have a similar set of symptoms and severity of illness and will require similar treatment. Meanwhile, the symptoms of viral pneumonia are entirely different, and such patients would require a different treatment plan.

Human experts (medical doctors) are capable of identifying these similarities by understanding the context and utilising their prior medical knowledge. If a doctor is presented with a diagnosis that she has been taught is similar to one she is well acquainted with, she is

Table 1 Toy example

Patient ID	Diagnosis	Temperature
1	Angina	37
2	Viral Pneumonia	40
3	Heart Failure	37
4	Acute Tonsillitis	39
5	Heart Failure	37

able to use this knowledge for predictions. We aim to utilise such prior knowledge to define the similarity between different concepts in a domain and to improve predictions based on domain knowledge.

Semantic similarity can be made explicit in different ways, and one of the prominent ways is through hierarchies, which we will use in this paper. For finding the semantic similarity in the toy dataset of Table 1, we can map diagnoses to taxonomy as shown in Fig. 1. Based on the evaluation of semantic evidence observed in Fig. 1, it is evident that Heart Failure is more similar to Angina than to Viral Pneumonia.

Motivated by this, we link the notion of similarity based on is-a relationships with the vector representation for categorical data. We develop a framework to use is-a relationships extracted from a concept hierarchy to quantify semantic similarity and propose a semantic embedding technique for categorical variables. We base this on our prior work on semantic similarity measures (Mumtaz & Giese, 2020), which we briefly review in Section 3.3.

3.1 Semantic similarity

Semantic similarity refers to similarity that is based on meaning or semantic content as opposed to form (Smelser & Baltes, 2001). Semantic similarity measures are automated methods for assigning a measure of similarity to a pair of concepts and can be derived from a taxonomy of concepts arranged in is-a relationships (Pedersen et al., 2007). Similarity computation for categorical data can improve the performance of existing machine learning algorithms (Ahmad & Dey, 2007) and may ease the integration of heterogeneous data (Wilson & Martinez, 2000).

Is-a relationships in a concept hierarchy encompass formal classification, properties, and relations between concepts and data. This provides us with a common understanding of the structure of a domain, explicit domain assumptions, and the ability to reuse of domain knowledge. It is vital to consider this information to achieve interpretable and good quality results in machine learning models,

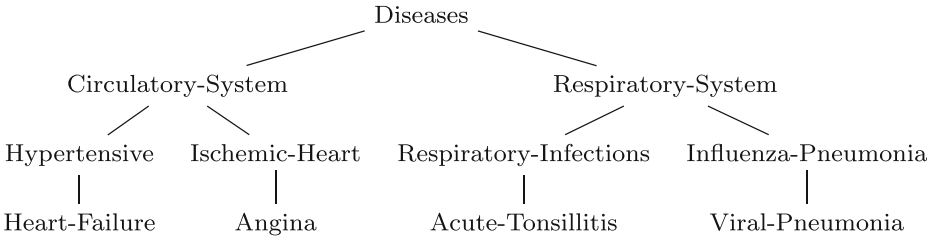

Fig. 1 Hierarchical Classification for Diagnoses

Hierarchies Our similarity measures are based on a given hierarchical structure of the value range of categorical features. Formally, we assume that the categorical values for each feature form a finite, partially ordered set (poset). A poset is a binary relation $\sqsubseteq$ on a set S, that $(\sqsubseteq, S)$ satisfies the following properties, for all $x, y, z \in S$,

- $x \sqsubseteq x$ (Reflexivity)
- If $x \sqsubseteq y$ and $y \sqsubseteq x$, then $x = y$ (Antisymmetry)
- If $x \sqsubseteq y$ and $y \sqsubseteq z$, then $x \sqsubseteq z$ (Transitivity)

If $a \sqsubseteq b$, we call b an ancestor of a, and a a descendant of b. The intention of $a \sqsubseteq b$ is that b is in some way more general, broader, etc. than a. For example, for the diagnoses in Fig. 1, Angina $\sqsubseteq$ Ischemic-Heart disease. A value is called a *leaf value* if it is not the ancestor of any other value (e.g., in Fig. 1, Angina is a leaf value).

A *closest ancestor* or *parent* of a value a is a value a' such that there are no values $x \in S$ with $a \sqsubseteq x \sqsubseteq a'$.

A value $c \in S$ is called a *common ancestor* of two node values $a \in S$ and $b \in S$ if $a \sqsubseteq c$ and $b \sqsubseteq c$. c is called a *lowest common ancestor* of a and b, written $c = a \sqcup b$, if $c \sqsubseteq x$ for any common ancestor x of a and b. It is well known that for general posets, two elements do not necessarily have a least common ancestor, but if it exists, then it is unique.

We call a *hierarchy* a finite poset where there is an element $r \in S$, called the *root*, such that $x \sqsubseteq r$ for all $x \in S$.

A *mono-hierarchy* is a hierarchy in which every value a has at most one parent. In a mono-hierarchy, any two values have the lowest common ancestor. Hierarchies that are not mono-hierarchies are sometimes called poly-hierarchies.

Many hierarchies used in practice are constructed as mono-hierarchies, e.g., Fig. 1, the biological taxonomies of animals and plants, the subdivision hierarchy of geological ages, etc. However, our results are equally valid for poly-hierarchies, and Use Case 2 in this article (see Section 5) is based on a poly-hierarchy.

Hierarchies are easily identified with finite directed acyclic graphs, where there is an edge from b to a if b is a parent of a. We refer to this graph when we talk about the length of a path in the hierarchy, for instance.

In some cases, the categorical values may belong to only the leaves of the hierarchy, while in other cases the values can be both leaves and internal nodes.

3.2 Proposed framework

We formulate two techniques to define semantic embeddings for single-valued and multi-valued categorical variables. A single-valued variable contains only one category for each instance occurring in the dataset, while a multi-valued categorical feature may have multiple categories for each data instance.

3.2.1 Semantic embeddings for single-valued categorical features

We have a set of values $S = \{X_1, X_2, \ldots\}$ that represents unique values occurring for the categorical variable in the data. We assume that these values form a hierarchy[1] $(\sqsubseteq, S)$ and a similarity measure between these values based on hierarchy (see the previous section). Fig. 2 shows one such example.

[1]The hierarchy can be a mono-hierarchy or a poly-hierarchy with a root node.

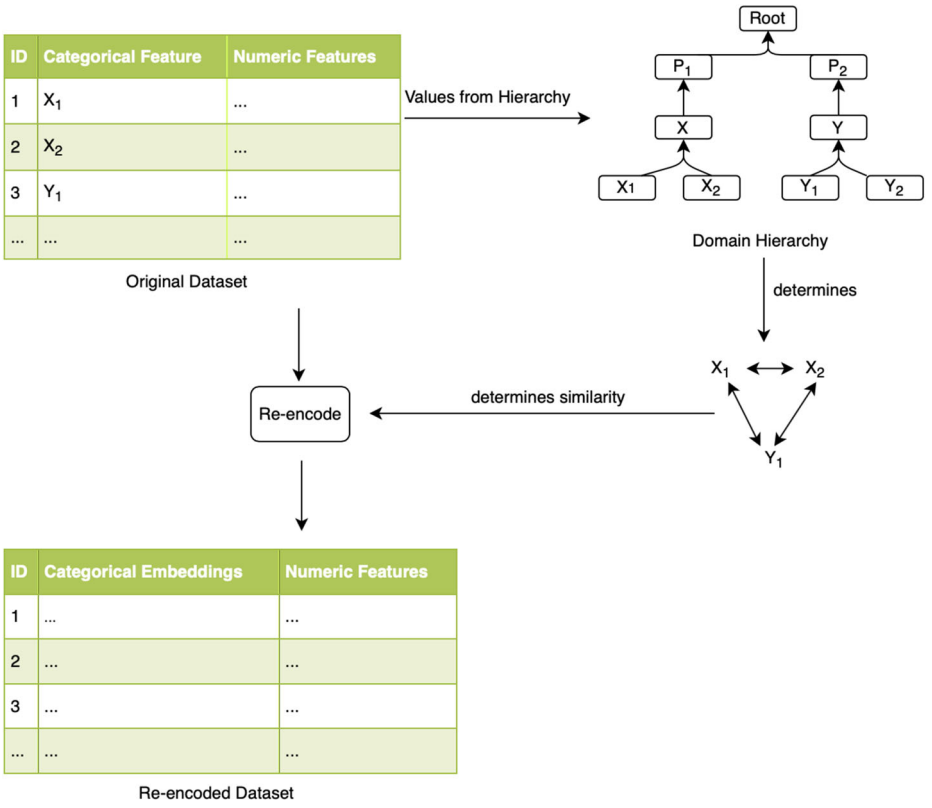

Fig. 2 Embedding Flow

In Fig. 2, X_1 and X_2 are placed close together in the given hierarchy while Y_2 belongs to a different part of the hierarchy. Our key objective is to define embeddings in such a manner that categorical values that are close in the hierarchy and share many similar characteristics, are also close in the embedding space. If $e(X_1)$, $e(X_2)$ are the embedding vectors of X_1 and X_2 respectively, then the key intuition is,

- $sim_{(\sqsubseteq,S)}(X_1, X_2) \approx sim(e(X_1), e(X_2))$
- $sim(e(X_1), e(X_2)) > sim(e(X_1), e(Y_1))$

For creating semantic embeddings, we calculate the similarity between the given value and all other values in the hierarchy. For a pair of values that are identical, the similarity is defined as 1 (representing maximum similarity). For cases where $i \neq j, sim(i, j)$ represents any similarity function that quantifies semantic similarity and should be in range $(0, 1)$. Formally, semantic embedding for a value i is defined as the n dimensional vector with components

$$e_j(i) = sim(i, j) \text{ for } i, j \in \{1, \ldots, n\}, \tag{1}$$

where n represents number of values for the categorical variable. In Section 3.3, we discuss hierarchy-based semantic similarity functions that can be used for sim in this embedding.

3.2.2 Semantic embeddings for multi-valued categorical features

Embeddings for categorical features into a low-dimensional vector space are not easily adapted to multi-valued categorical features. Our embedding supports multiple values in a straightforward way. For a multi-valued categorical variable, we calculate the embedding vector for each category first by using (1). These vectors are then aggregated to get a single vector for the multiple categories. The aggregation operation can be performed in different ways: minimum, maximum, or sum of all the vectors.

In the one-hot encoding, aggregation is performed by placing one for the categories present and zero for the remaining values. In our current setting, we want to have non-zero values for the multiple categories. If the minimum values are selected for combining all the vectors, similarity scores will always be zero, as shown in Fig. 3

The summation option for aggregation is also not suitable in the current setting, as it may lead to similarity scores of greater than one or of one in all dimensions. To aggregate embeddings for all categories, we take the maximum value in each dimension. The maximum operation ensures that the final vector retains maximum similarity for all neighboring values in the hierarchy. Given a set $I \subseteq S$ of categorical values, we define the n-dimensional embedding vector as follows:

$$\bar{e}_j(I) = \max_{i \in I}(e_j(i)) \quad \text{for } j \in \{1, \ldots, n\}, \tag{2}$$

where

$e_j(i)$ is the j-the component of the embedding for the single value i.

3.3 Similarity measures based on hierarchy

In this section, we discuss various semantic-based similarity measures that can be used in (1) for calculating the similarity between two values based on domain hierarchy.

3.3.1 Semantic-based measures

The existing semantic similarity measures are classified into the following categories (Harispe et al., 2015).

- Information-theoretic Approaches: These measures use the relative frequency of values in a corpus in combination with the knowledge source to calculate the semantic similarity.
- Structural Approaches: These measures only utilise the structure of the graph or taxonomy to calculate the semantic similarity between two terms.

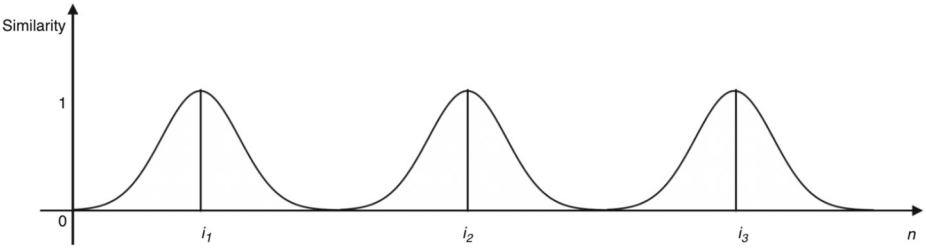

Fig. 3 Similarity for multiple categories

3.3.2 Information-theoretic approaches

These approaches are based on the idea of a) measuring the similarity between two concepts in terms of commonalities and differences defined in terms of the information content (Harispe et al., 2015), and b) quantifying the information content based on Shannon's information theory. Information content (IC) specifies the amount of information embedded in each concept and is based on the idea that abstract concepts will contain less information than specific/concrete entities. The IC of a concept c in this setting is calculated by taking the probability $p(c)$ of instances belonging to c in a text corpus and is formally defined as

$$IC = -\log p(c) \tag{3}$$

where $p(c)$ is estimated by the occurrence of c or instances belonging to c in the text corpus. Below, we state some of the existing information-theoretic measures.

– Resnik's Similarity Measure: Resnik defined the similarity between two conceps x and y as the IC of their lowest common ancestor in the hierarchy: Resnik (1999),

$$Sim_{Res}(x, y) = IC(x \sqcup y) \tag{4}$$

– Lin's Similarity Measure: Lin also uses the idea of IC with the lowest common ancestor, but in a different way, and defines similarity as Lin (1998),

$$Sim_{Lin}(x, y) = \frac{2IC(x \sqcup y)}{IC(x) + Ix(y)} \tag{5}$$

– Jiang & Conrath's Similarity Measure (JCH): This approach defines dissimilarity between two terms as Jiang and Conrath (1997)

$$Dis(x, y) = IC(x) + IC(y) - (2IC(x \sqcup y)) \tag{6}$$

For poly-hierarchy, where there can be several lowest common ancestors, these measures are modified to consider the lowest common ancestor with the highest information content for similarity calculation.

3.3.3 Structural approaches

Structural approaches rely only on the knowledge source, such as a graph or hierarchy, to define the similarity between two concepts. These approaches do not use any text corpus. The focus is the interconnection between concepts in the hierarchy when estimating the similarity (Harispe et al., 2015). Below, we give an overview of the existing structural approaches that we use in our experiments.

– Shortest Path Similarity: This strategy defines similarity based on the shortest path distance between two concepts:

$$Sim_{sp} = \frac{1}{1 + sp(x, y)} \tag{7}$$

where the shortest path (sp) between two nodes or concepts in the hierarchy is calculated by taking into account the ancestor that can be reached by both concepts using the minimum number of traversals (Harispe et al., 2015).

– Wu and Palmer: This measure defines the similarity between two concepts based on their depth in the hierarchy and that of their lowest common ancestor (Harispe et al., 2015). It is defined as

$$Sim_{WUP} = \frac{2depth(x \sqcup y)}{2depth(x \sqcup y) + sp(x, (x \sqcup y)) + sp(y, (x \sqcup y))} \tag{8}$$

– Leacock & Chodorow (LCH): This measure defines similarity based on the shortest path between two concepts and scales it by the maximum depth of the taxonomy (Leacock & Chodorow, 1998). To avoid taking the logarithm of 0 in the LCH measure, 1 is added to the shortest path. The same formulation is used by the standard NLTK library[2] used in our experiments.

$$Sim_{LCH}(x, y) = -\log \frac{sp(x, y) + 1}{2Max_{depth}} \tag{9}$$

In our prior work (Mumtaz & Giese, 2020), we developed a simple method to calculate semantic similarity between two given categorical values belonging to a mono-hierarchy ($\sqsubseteq$, S). As part of the experiments, we extend the same idea and propose a semantic similarity for poly-hierarchy. Below, we give a short overview of the existing measure for mono-hierarchy, followed by the new measure.

3.3.4 Poly-hierarchy semantic similarity (PS)

Semantic similarity between two nodes in the hierarchy is defined as the common information shared between them (Resnik, 1995). Lin quantifies this common information as the lowest common ancestor that subsumes both values in the hierarchy (Lin, 1998). As mentioned earlier, this is based on the idea that any two values having the lowest common ancestor close to leaf nodes, should have high similarity as they share many common characteristics. If the lowest common ancestor is close to the root node, fewer commonalities exist between a given pair of values.

Based on the above idea, hierarchy-based semantic similarity (Mumtaz & Giese, 2020) defines similarity between two values x and y by considering the level of their lowest common ancestor in the hierarchy and is defined as

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ \lambda^{d - level(x \sqcup y)} & \text{if } x \neq y \end{cases} \tag{10}$$

where $x \sqcup y$ denotes the lowest common ancestor of x and y, $0 < \lambda < 1$ is a fixed decay parameter, $level(n)$ is the distance of node n from the root in the hierarchy, and $d = \max_{n \in X} level(n)$ is the maximum depth of the hierarchy. As the level of the lowest common ancestor moves up in the hierarchy (close to the root node), the similarity between nodes decreases.

Equation (10) performs well when all the categorical values can be mapped to the leaf nodes in the hierarchy, and all these leaves are at the same level. It also requires that ($\sqsubseteq$, S) forms a mono-hierarchy.

However, in use cases where the categories are not defined at the same granularity level or some values represent more generic concepts in the hierarchy, (10) may lead to similarity

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

values that contradict the actual similarity in the hierarchy. We modify (10) and propose a new similarity measure between two nodes:

$$\delta^{+}(x, y) = \begin{cases} 1 & \text{if } x = y \\ \lambda^{d(x,y)} & \text{if } x \neq y \end{cases} \tag{11}$$

where $d(x, y)$ represents the distance between nodes in the hierarchy. There are several ways to define distance between nodes in a poly-hierarchy. We define $d(x, y)$ based on the set of common ancestors in the hierarchy as

$$d(x, y) = \min_{x,y \sqsubseteq z} \frac{1}{2}(\bar{l}_z(x) + \bar{l}_z(y)) \tag{12}$$

where $\bar{l}_z(x) = level(x) - level(z)$ is the number of directed edges between node x and the common ancestor z of the nodes x and y. If there is a common ancestor with a small number of edges to the nodes x and y, then a higher similarity value is assigned. For nodes that belong to different parts and different depths in the hierarchy, first we compute edges from both nodes to all the lowest common ancestors separately and then we calculate the average for each common ancestor. The minimum distance is selected for the final similarity calculation. This ensures lower similarity values for node pairs that do not have immediate lowest common ancestors in the hierarchy.

4 Comparison between existing semantic similarity measures

Before utilising semantic similarity measures in embeddings, we compare existing similarity measures. The focus is to analyse the difference between the performance of semantic similarity measures by using a data-based approach as opposed to only the knowledge source (more precisely, the hierarchy).

The assessment of the above-mentioned semantic similarity measures is based on datasets formed by human judgment. These datasets are composed of pairs of words and their similarity scores assigned by human experts. Below, we give an overview of some standard datasets.

4.1 Benchmark datasets and WordNet hierarchy

- **RG-65**: Rubenstein and Goodenough composed a dataset of 65 pairs of nouns (Rubenstein & Goodenough, 1965), and 51 human subjects were asked to rate words on a scale of 0.0–4.0 according to the similarity of meaning. The final dataset contains the average similarity of scores provided by all the participants in the study. The RG65 dataset is largely used for the evaluation of semantic similarity measures.
- **MC-30**: Miller and Charles' benchmark is composed of 30 pairs of nouns extracted from RG65 datasets and the similarity judgments from 38 participants (Miller & Charles, 1991).
- **Wordsim353**: The original dataset is composed of 353 word pairs, and participants rated similarity between pairs on a scale of 0–10 (Finkelstein et al., 2002).
- **RW**: The RW dataset consists of 2034 pairs of rare words rated on a scale of 0–10 (Luong et al., 2013).
- **Card-660**: The Cambridge rare word dataset is composed of 166 pairs of words, rated on a scale of 0–4 (Pilehvar et al., 2018). It covers a wide range of rare words from different domains, including IT, entertainment, politics, medicine, etc.

4.2 WordNet hierarchy

Any knowledge base that encodes relationships such as is-a between words can be utilised for defining word similarity for the datasets mentioned above. In this work, we use WordNet[3] to define word similarity and for conducting experiments. WordNet groups nouns, verbs, adjectives, and adverbs into sets of cognitive synonyms (synsets), each expressing a distinct concept.[4] These distinct concepts are interlinked by means of conceptual-semantic and lexical relations. WordNet lists all the concepts associated with a word based on the most common uses of a word.

There is a slight complication, since the same word can have several meanings, and thus occur in several synsets. For example, the word 'tiger' can designate a large feline or an audacious person, which are quite different concepts, situated in different locations in WordNet's synset hierarchy. Whether 'tiger' is more similar to 'lion' than to 'winner' depends on the intended meaning.

Since we are aiming at a word pair similarity (and not a synset similarity), we have to resolve this ambiguity when computing the similarity between two words. The information available in our case study does not allow us to use, e.g. the word's context. Our approach is to take into account different combinations of synsets of the two words. There could be many synsets per word in general, and we anticipated that the less frequent synsets would have little influence on the outcome of experiments. We ran an initial set of experiments by only considering the top three synsets versus all synsets to confirm this. The initial results suggest that there is little to be gained from including many synsets in the disambiguation. For a fair comparison, we consider the first three synsets for all the datasets in our experiments.

Given a word w, we write $concept_{(i,w)}$ for the i-th most common synset for w. Given two words $word_1$ and $word_2$, for both words, we first compute the similarities

$$\delta^{+}(concept_{(i,word_1)}, concept_{(j,word_2)}) \quad i, j \in \{1 \dots k\}$$

between the pairs of top three synsets using (11). We then aggregate these by picking a maximum strategy (the same approach found in the existing literature Ahu & Iglesias, 2015).

The definition is as follows:

$$sim(w_1, w_2) = \max_{i,j=1\dots3} \delta^{+}(concept_{(i,word_1)}, concept_{(j,word_2)}) \tag{13}$$

4.2.1 Evaluation strategy

The performance of semantic similarity measures is evaluated by using correlation coefficients to find a correlation with the judgment provided by human experts for different datasets. There are two commonly used correlation coefficients for this task: the Pearson correlation coefficient and the Spearman correlation coefficient.

The Pearson correlation coefficient, represented as r, measures how well any semantic similarity measure relates to human similarity scores (Harispe et al., 2015). A score of 1

[3] https://wordnet.princeton.edu/

[4] For our experiments, we only consider word pairs that belong to noun synsets in the WordNet hierarchy.

indicates the perfect correlation between human and measure score, whereas 0 means no correlation. It is defined as

$$r = \frac{n(\sum x_i y_i) - \sum x_i \sum y_i}{\sqrt{n(\sum x_i^2)(\sum x_i)^2}\sqrt{n(\sum y_i^2)(\sum y_i)^2}} \tag{14}$$

where x_i refers to the i-th score in the list of human judgment and y_i refers to the corresponding element in the list of any semantic similarity measure. n corresponds to the total number of word pairs.

The Spearman rank-order coefficient ρ compares the word pair rankings between human judgment and the similarity measures (Spearman, 1987). Like the Pearson correlation coefficient, 1 indicates perfect correlation, while 0 represents no correlation. Given d_i the difference between the ranks of x_i and y_i, ρ is defined as

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \quad . \tag{15}$$

A high Pearson correlation requires a linear relationship between the similarity measurements, while a high ρ indicates that measures agree in the ranking of which inputs are more similar or less similar. Since the scale of human assessment of similarity is somewhat arbitrary, the qualitative approach of the Spearman coefficient makes sense.

4.3 Experiments and results

We perform a set of experiments by using our proposed poly-hierarchy similarity measure (PS), along with the existing similarity measures described in Section 3.3. As some of the existing measures such as Lin and Resnik are based on the IC of the given words, we use three different text corpora:

- Brown: an electronic collection of text samples of American English containing 1.15 million words. This corpus provided the base for the first set of scientific studies for the frequency and distribution of words in everyday language use (Kucera & Francis, 1969).
- Semcor: an English corpus with semantically annotated texts. It is a subset of the Brown corpus consisting of 360,000 words (Landes et al., 1998).
- Genesis: It consists of 200,000 words.

These corpora are used to calculate the IC and similarity of words by using a standard NLP python package, 'NLTK'.[5]

Table 2 shows the results of applying the Resnik, Lin, and JCN measures by using the Brown and Semcor corpora. Table 3 lists the results for Resnik, Lin, and JCN on the Genesis corpus along with structural approaches (WUP, Path, LCH, and PS). Both tables list Spearman and Pearson coefficient values for different datasets. Among IC-based methods, Resnik and Lin show good linear correlation as compared to JCN on the standard datasets (RG-65, MC-30, and WordSim) using all three corpora. However, there is a decrease in performance for Spearman rank correlation on Semcor and Genesis datasets. In addition, there is low coverage for datasets with rare words, particularly for RW datasets with significantly low r and ρ values. This shows that given good quality datasets containing enough representation of word occurrences, Lin and Resnik show the best performance. However, for

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

Table 2 Correlation coefficient values for Semantic similarity task using Brown and Semcor corpus

		IC-based (Brown Corpus)			IC-based (Semcor Corpus)		
Dataset	coffiecient	Resnik	Lin	JCN	Resnik	Lin	JCN
MC-30	r	0.71	0.70	0.49	0.73	0.65	0.65
MC-30	ρ	0.73	0.76	0.47	0.25	0.71	0.47
RG65	r	0.70	0.69	0.48	0.68	0.55	0.52
RG65	ρ	0.80	0.80	0.48	0.19	0.68	0.52
WordSim	r	0.65	0.63	0.31	0.65	0.49	0.46
WordSim	ρ	0.70	0.68	0.31	0.11	0.58	0.31
Card-660	r	0.37	0.56	0.64	0.63	0.60	0.59
Card-660	ρ	0.36	0.63	0.61	0.43	0.63	0.61
RW	r	0.08	0.19	0.18	0.30	0.16	0.14
RW	ρ	0.07	0.18	0.16	0.07	0.20	0.16

rare words, the occurrence frequencies are too low to give an accurate estimate of information quality, which makes it difficult to quantify semantic similarity by using Resnik or Lin. In particular, there is a great degree of fluctuation in rankings (ρ) when moving to Semcor or Genesis data.

All the structural approaches (WUP, Path, LCH, and PS) show equally good performance when using only hierarchy as opposed to IC. Both r and ρ values are consistent for the standard datasets (RG-65, MC-30, and WordSim). Our poly-hierarchy based measure shows performance that is comparable to existing measures and, in some cases, slightly better. In addition, the structural approaches show a significant increase in correlation for rare word datasets (RW and Card-660), except for the Pearson correlation with Resnik on RW, where 0.29 is still better than any of the others.

The initial experiments suggest that given a large corpus and therefore having a good estimate of IC, the information-theoretic approaches (Lin, Resnik) perform well. For scenarios with an insufficient amount of data (low-resource domains, here rare words), the approaches that do not use IC perform better than IC-based approaches.

5 Use case 2: Semantic embeddings for words

Natural Language Processing (NLP) techniques provide embeddings that represent words as dense vectors of real numbers in an embedding space. This vector representation is created by embedding semantic and syntactic similarity by using a large corpus. Existing common methods such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) create embeddings based on the co-occurrences of words and utilise this as a context of a given word. These methods perform well for text datasets that have sufficient co-occurrences. However, their performance is poor for domain-specific terms where domain texts are sparse, or where many important concepts do not frequently occur, such as 'cybersecurity,' 'biomedical,' etc. Roy et al. (2017). Training on a large corpus also requires more resources in terms of time and memory.

We follow an alternate approach for creating word embeddings. We use different similarity measures along with (1) to create word embeddings. Following a similar approach

Table 3 Correlation coefficient values for Semantic similarity task using Genesis corpus and Hierarchy-based Measures

		IC-based (Genesis)			Structural			
Dataset	coffiecient	Resnik	Lin	JCN	Path	WUP	LC	Poly-Hierarchy
MC-30	r	0.76	0.72	0.77	0. 61	0.66	0.61	0.70
MC-30	ρ	0.22	0.77	0.54	0.68	0.70	0.69	0.69
RG65	r	0.68	0.60	0.62	0.67	0.66	0.66	0.76
RG65	ρ	0.35	0.72	0.59	0.72	0.71	0.75	0.78
WordSim	r	0.64	0.33	0.26	0.60	0.65	0.60	0.59
WordSim	ρ	0.24	0.47	0.36	0.59	0.65	0.64	0.61
Card-660	r	0.63	0.65	0.64	0.69	0.66	0.69	0.66
Card-660	ρ	0.53	0.64	0.64	0.72	0.63	0.72	0.69
RW	r	0.29	0.19	0.18	0.28	0.26	0.28	0.25
RW	ρ	0.19	0.22	0.21	0.29	0.25	0.27	0.27

to Section 4.2, the WordNet hierarchy is used in combination with Brown, Semcor, and Genesis datasets to first calculate the similarity between words using information-theoretic approaches and then create embeddings. We also create embeddings using measures based only on the hierarchy (WUP, LCH, Path, and PS).

We choose the task of concept categorisation in NLP for the evaluation of word embeddings on four benchmark datasets. Detailed experiments are discussed below.

5.1 Concept categorisation

Concept categorisation is often used in NLP for evaluating the performance of different embedding schemes. Concept learning involves the process of assigning concepts/words to one or more relevant categories and is also known as concept categorisation (Wang et al., 2019). For instance, given the words $\{milk, tea, bread, cake\}$, the model should group them into two categories.

5.2 Benchmark datasets and evaluation

We use four benchmark datasets for evaluating the task of concept categorisation. Each dataset consists of a list of words and associated categories for each word. The AP dataset consists of 402 words that are divided into 21 categories (Almuhareb, 2006). The BLESS (Baroni & Lenci, 2011) dataset contains 200 words that are assigned to 17 classes. The BM dataset is the largest one, containing 5321 words grouped into 56 categories (Baroni et al., 2010). It contains a large number of duplicate words that belong to more than one category. The clustering task, by definition, requires unique words in each category. In order to make the benchmark fit the clustering task at hand, we have deleted all duplicate words from the input data in this evaluation. The ESSLLI dataset consists of 44 nouns, divided into six semantic categories.[6] All four datasets contain only noun concepts; therefore, we only consider noun hierarchies in the WordNet.

[6] http://www.wordspace.collocations.de/doku.php/data:esslli2008:concrete_nouns_categorisation

For evaluating the task of concept categorisation, first, vector embeddings are created for words in each dataset. Words are then placed into n different clusters by using any clustering algorithm. We use k-means clustering in our experiments and specify k equal to the original number of clusters in each dataset. Clustering purity is used as a performance evaluation measure. Purity is a measure of the extent to which clusters contain a single class (Manning et al., 2008). It shows the percent of the total number of data points that are classified correctly. A purity score of 100% means that all data points are placed in the correct clusters.

Feature compression For datasets that have high cardinality embeddings, compression can be used to get a low-dimensional feature map. We use autoencoders to compress the high-dimensional embedding for the BM dataset into a low 300-dimensional representation. The autoencoder is a deep neural network consisting of encoder and decoder parts. The encoder reduces the input into low-dimension vectors, and the decoder reconstructs the input. Supervised learning is performed to minimise the difference (loss) between the input and output (Yildirim et al., 2018). In our experiments, the purity achieved with feature compression was as good as with the high-dimensional encodings.

5.3 Experiment and results

For each dataset, different embeddings are created based on the WordNet hierarchy using the similarity measures in Section 3.3.1. For comparison, we use vectors for words based on Google's word2vec and GloVe. Google provides vectors for 3 million words pretrained on the Google news dataset (about 100 billion words).[7] For each dataset, we extract 300-dimensional vectors for given words from the existing Google pretrained vectors. Similarly, embeddings pretrained on wikipedia2014 for GloVe (Pennington et al., 2014) were downloaded from the online repository.[8] Our evaluation compares the performance of different existing embeddings on general clustering tasks. An optimisation by further training of the embeddings specifically for these tasks would not be sensible in this context.

Tables 4 and 5 show the purity score for the word2vec (Google), GloVe, and hierarchy-based embeddings. For all the datasets, the purity score of hierarchy-based embeddings is much higher than what is achieved by the Google word2vec and GloVe embeddings.

Among IC-based similarity measures, Resnik shows the best performance by using the Brown Corpus for the word embeddings. However, there is around 20% decrease in clustering purity for Resnik and Lin when embeddings are created using IC from Semcor and Genesis datasets. For clustering tasks, JCN shows some stability in performance even with the change of data corpus. Embeddings based on purely structural approaches (WUP, Path, LCH, and PH) show an equally good purity score.

The experimental results for the task of concept categorisation conform with the results in the previous section of comparing word pair similarities with human judgments. It is observed in both cases that given high-quality data, IC-based measures are good in quantifying semantic similarity; however, for low-resource domains, the structural approaches based only on hierarchy quantify the semantic similarities correctly.

[7] https://code.google.com/archive/p/word2vec/
[8] https://nlp.stanford.edu/projects/glove/

Table 4 Clustering purity in % for word embeddings

	Corpus-based		Hierarchy-based			
Dataset	Google	GloVe	WUP	Path	LCH	PH
AP	79.50	69.42	97	100	99	100
Bless	97.42	85.64	96	96	100	100
BM	88.90	80.04	84	93	92	77
ESSLLI	84.09	75.00	100	100	100	100

6 Use case 2: mortality prediction using MIMIC dataset

In order to evaluate the categorical encoding schemes, we use a mixed-variable real-life dataset containing a prediction task and one high cardinality categorical variable. We convert the categorical variable to an equivalent semantic encoding by using the relevant domain hierarchy. These encodings are combined with numeric features, followed by a standard prediction pipeline, and the results are compared with the one-hot encoding for the prediction task.

6.1 Data description

Mortality prediction of patients admitted in an intensive care unit (ICU) is important for timely intervention to adapt treatments and policies. We aim at predicting the mortality of such patients. We consider mortality as a binary classification task where label 1 represents the death event of a patient. We use MIMIC-iii, a publicly available database (Johnson et al., 2016). This database consists of 53,423 adult patients and 7,870 neonates admitted to the ICU at the Beth Israel Deaconess Medical center in Boston between 2001 and 2012. The database includes information about demographics, diagnosis, vital sign measurements (numeric features), procedures, medications, and mortality. For experiments conducted in this research, we consider diagnosis as the main high cardinality categorical feature and vital signs as numeric features for each patient. The hospital expiry flag acts as a target variable, representing two classes: 1 for expired patients and 0 for alive patients.

For the MIMIC dataset, we observe that for the main categorical variable 'Diagnosis,' 70% of the unique diagnoses occur fewer than 10 times in the dataset. Only 4% of diagnoses frequently appear in the dataset with a frequency of greater than 100. Mostly, the categorical variables follow a long-tail distribution, with some categories being more frequent than others. As observed in the previous two sections, the estimate of IC based on occurrence

Table 5 Clustering purity % for word embeddings (brown and semcor corpus)

	Brown			Semcor			Genesis		
Dataset	Resnik	Lin	JCN	Resnik	Lin	JCN	Resnik	Lin	JCN
AP	100	90	100	84	81	96	80	76	93
Bless	100	98	97	83	86	93	80	75	86
BM	97	73	95	84	68	95	77	69	89
ESSLLI	100	95	100	100	93	100	90	79	90

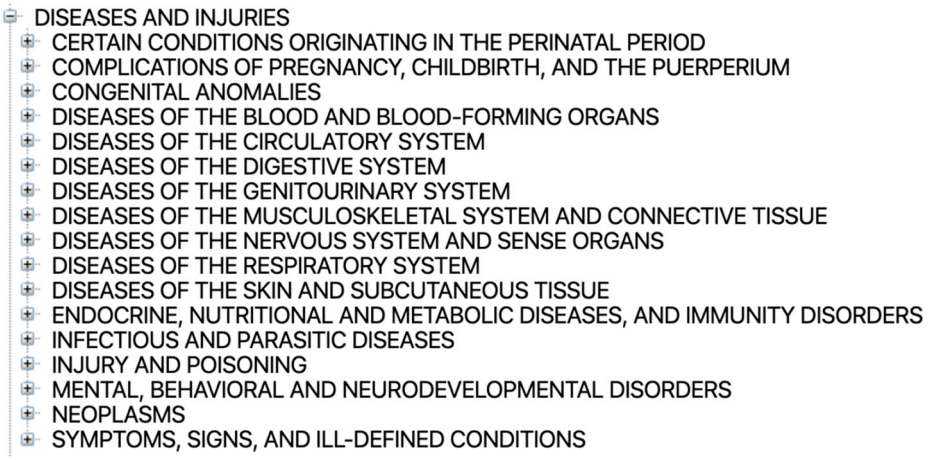

Fig. 4 International Classification of Diseases (ICD9)

probability is not suitable for estimating semantic similarity. Therefore, for the mortality prediction task, we use only structural similarity measures (WUP, LCH, and PS) to create embeddings.

In the next section, we explain how we link 'diagnosis' with the associated diagnosis hierarchy.

6.2 Hierarchy for categorical feature

The diagnosis_icd table consists of patients' identifiers and associated diagnosis codes for each patient. These codes are based on a standard disease ontology called the International Classification of Diseases (ICD-9).[9] ICD-9 contains a description of all known diseases and injuries. Each disease is detailed based on diagnostic characteristics and assigned a unique identifier called the ICD-9 code. ICD-9 contains a standardised classification of around 12000 diseases. All diseases are classified into 17 high-level major disease groups as shown in Fig. 4. Within each major group, further classification of diseases is performed based on similar characteristics.

The MIMIC dataset only contains numeric codes for each diagnosis, but the information regarding the classification of each diagnosis is missing. For mapping the diagnosis column to the ICD-9 ontology, we scraped the ICD-9 ontology from BioPortal and organised each diagnosis with a short title, long title, and associated parents in the hierarchy, as shown in Fig. 5. ICD-9 codes from the ontology contain a decimal point in each numeric code. For instance, the numbers 280 and 28.0 represent two different diagnosis codes. However, the diagnosis codes present in the MIMIC database do not contain decimal points. This creates an ambiguity in matching codes with the BioPortal codes, which have the possibility of having decimal points at several positions. To resolve this issue, we first joined the diagnoses_icd table with the d_icd_diagnoses table. The latter contains diagnosis codes along with their short and long titles. We matched short titles with the BioPortal titles in order

[9] https://bioportal.bioontology.org/ontologies/ICD-9CM/?p=summary

	ICD9_CODE	5	4	3	2	1	0	long_description	SHORT_TITLE
0	3.22	3.2	3	001-009.99	001-139.99	001-999.99	001-999.99	Salmonella pneumonia	Salmonella pneumonia
1	3.21	3.2	3	001-009.99	001-139.99	001-999.99	001-999.99	Salmonella meningitis	Salmonella meningitis
2	3.24	3.2	3	001-009.99	001-139.99	001-999.99	001-999.99	Salmonella osteomyelitis	Salmonella osteomyelitis
3	3.29	3.2	3	001-009.99	001-139.99	001-999.99	001-999.99	Other localized salmonella infections	Local salmonella inf NEC
4	3.23	3.2	3	001-009.99	001-139.99	001-999.99	001-999.99	Salmonella arthritis	Salmonella arthritis

Fig. 5 Example diseases and associated ancestors in ICD-9

to find the correct ICD-9 codes and associated parents in the hierarchy. All codes with no matching short titles were discarded from our experiments.

6.3 Pre-processing for numeric features

For each patient having age $\geq$ 15, only the first admission is considered for analysis, and later admissions are discarded. We extract and clean 18 numeric features for the first 24 hours of a patient's stay by using the benchmark code provided by Purushotham et al. (2017).[10] The features heartrate_max, heartrate_min, sysbp_max, sysbp_min, tempc_max, tempc_min, and urineoutput are taken from the table chartevents, whereas the table labevents contains bun_min, bun_max, wbc_min, wbc_max, potassium_min, potassium_max, sodium_min, sodium_max, bicarbonate_min, bicarbonate_max, and mingcs. The dataset contains measurements for these features in different units. Only data points with the unit used in $\geq$90% of the data are kept; the others are discarded. For features that have multiple recordings at the same time, the average is taken as the final value. For value ranges, the median of the range is used in the analysis. After all the preprocessing and cleaning steps, the cleaned data consists of a total of 25,531 data points, out of which 3114 patients died in the hospital.

6.4 Data imbalance and data augmentation

A dataset is called imbalanced if all classification categories are not equally represented. After preprocessing of the dataset, there are only 3,114 patients who expired in the ICU. This shows a high imbalance in the dataset, and the majority of the instances belong to class 0 (alive patients). Learning classifiers from imbalanced datasets is a difficult task and usually results in labeling all the cases as the majority class (Kotsiantis et al., 2005). A number of solutions have been proposed to handle imbalanced datasets. These techniques include random oversampling with replacement, random undersampling, and oversampling with the generation of new samples. The main drawback of undersampling is that it discards potentially useful data, while oversampling may lead to over-fitting as it replicates the minority class.[11] To avoid drawbacks associated with random over- and undersampling, we perform sampling using the Synthetic Minority Oversampling TEchnique (SMOTE) (Chawla et al., 2002). SMOTE is a data augmentation technique for the minority class that synthesises new points from the minority class.

6.4.1 Sampling train and test set based on the hierarchy

In order to see the effect of domain information for predicting unseen categories, we split our dataset into train and test based on the information from the hierarchy. We grouped

[10] https://github.com/USC-Melady/Benchmarking_DL_MIMICIII

[11] Oversampling and undersampling did not yield good results in our experiments.

all diagnoses based on level 5 in the hierarchy. For each diagnosis group, one diagnosis is selected, and all data points with that diagnosis are chosen for the test data, while data points with all other diagnoses in the group go into the training data. This ensures that only unique diagnoses codes are present in the test set, which the model has not seen during the training phase. After splitting the dataset into training and testing based on diagnosis, we apply the SMOTE strategy only to the numerical features of the training dataset, avoiding information leakage from training into the testing dataset. Further, to see the effect of embedding on the prediction, we do not alter the embeddings using the SMOTE sampling. For the multi-valued attribute, as each patient has multiple diagnoses, the split is performed using the standard sklearn train_test_split utility, followed by SMOTE sampling on only the numeric features of the training dataset.

6.5 Prediction algorithm

In this section, we describe the prediction algorithm and the scoring system used for evaluating the proposed measures. We choose to use the deep learning model for prediction as it provides an automated way of extracting complex data representations (Bengio et al., 2012). The main advantage of the deep learning model is its ability to learn feature representations from raw data and allow generalisation to new combinations of values not seen in the original training dataset (Purushotham et al., 2017). As we are choosing maximum dimensions in our embeddings to encode similarity, the deep learning model fits our need to learn intermediate compressed representations of embeddings along with training for the prediction task.

6.5.1 Bimodal feed forward network

We started our investigation with a set of experiments based on a single neural network architecture without using data augmentation. For single-valued categorical data, the network performance was poor with correct identification of only 15% of expired patients on the test set. Addressing the imbalanced data by means of augmentation (SMOTE) improved the performance to 40%, which is still not satisfactory. The remaining experiments were performed using the augmented training data and a bimodal network which, as the results will show, leads to a significant additional improvement.

We use a bimodal deep learning model to learn shared representations from two different modalities: numeric features and semantic embeddings from the categorical variable. Figure 6 shows an illustration of our bimodal framework.

The key idea here is to first use hidden layers to capture the correlation within each modality separately. For the high-dimensional embeddings, hidden layers also act as a compression component, and only low-dimensional features go into the shared layers. The shared layers then capture the correlation between modalities, which is beneficial in complex datasets. The latent shared representations are learned for the prediction task.

6.5.2 Implementation details

Embeddings We use the Python package `networkx` to create embeddings for each diagnosis. First, a tree hierarchy is created from the ontology database (Fig. 5). Each ICD-9 code is represented as a node in the graph, and we compute its level, i.e. its distance from the root. For each pair of nodes x, y, we calculate $x \sqcup y$ and store $level(x \sqcup y)$. In order to create vector embeddings for all unique nodes, we create a matrix where each column represents

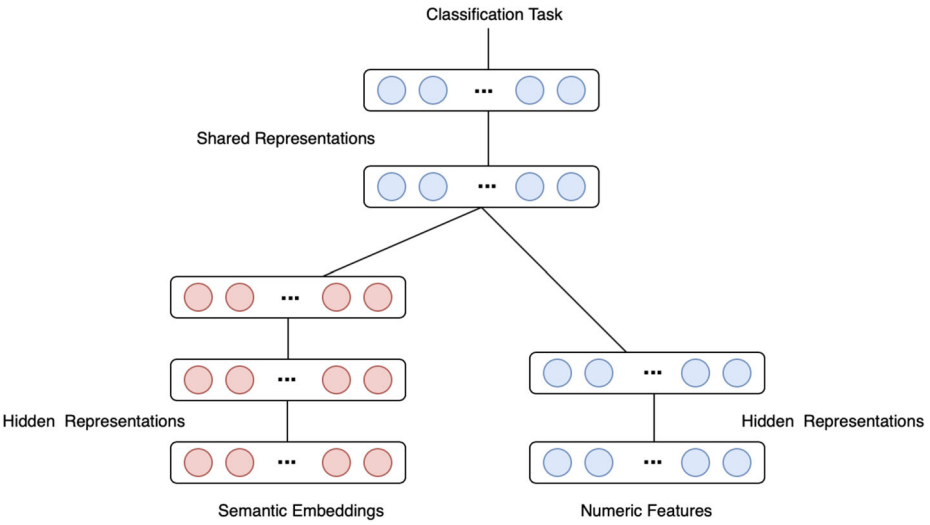

Fig. 6 Bimodal Deep Learning Architecture

a unique diagnosis and each row represents the embedding for one diagnosis by calculating the similarity using (11).

Bimodal deep learning We implement a bimodal network using the Python library `keras` (functional API) and `keras-tuner` for trying different network architectures. For all prediction tasks, we have divided the dataset into three parts: train, validation, and test dataset. We experiment with different optimisers during the training phase and the Adam optimiser with a learning rate 0.001 performs best in our experiments. The batch size is chosen as 256, and the max epoch is 100. Early stopping with a neuron drop rate of 0.2 % is performed during the training phase. The final model uses two hidden layers in total: the first hidden layer learns the compressed representation for embeddings, which is then combined with vital measurements. This is further passed through a second hidden layer, followed by the output layer. The goal of our experiments is to compare our embeddings to one-hot *under similar conditions* rather than tuning the architecture for the best possible performance for each embedding. Therefore, we use the same model architecture for training one-hot encoded data and semantic-based encoded data to compare all methods' performance in a fair way. The final evaluation is performed on the models that achieve minimum validation loss during the training phase. These experiments' key focus is to show that a performance improvement can be achieved using prior knowledge in limited experimental settings. Therefore, we did not perform an exhaustive optimisation of bimodal network architecture in terms of increasing the depth of architecture.

6.6 Experiments

We perform two sets of experiments for single-valued and multi-valued embeddings, respectively. The MIMIC dataset contains multiple diagnoses per patient, along with a priority number for each diagnosis. In order to evaluate single-valued embeddings, we consider only one diagnosis for each patient, namely the one with the highest priority number in the

dataset. For multi-valued embeddings, we consider all diagnoses related to a patient in the database.

For performance comparison, pre-trained embeddings based on NLP techniques such as GloVe or word2Vec can be used. We downloaded pre-trained embeddings from a Google news dataset that contains 300-dimensional vectors for 3 million words[12] and GloVe-based pre-trained embeddings by Stanford.[13] The vocabulary present in both datasets does not contain the diagnosis concepts present in the MIMIC dataset. This is justifiable as both the datasets are trained on the documents containing only words used in day-to-day vocabulary. Pubmed contains documents in the biomedical scientific literature, and word2vec-based embeddings for PubMed articles are provided by the NLP lab.[14] However, these embeddings also do not cover all the concepts present in the MIMIC dataset. Therefore, we do not use any of these pre-trained embeddings in our experiments.

We compare the performance of the proposed embedding scheme with the standard one-hot encoding technique.

Below, we explain the evaluation metrics, followed by results reported in Section 6.7.

6.6.1 Performance evaluation

We use recall score and Area under the ROC curve (AUC-ROC) to evaluate the performance of our prediction models. Recall specifies the model's ability to find all positive cases (patients who died) in the dataset. For our use case, we are more interested in predicting critical patients correctly; hence we focus on recall score as compared to the overall accuracy of the models. The recall is calculated as

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{16}$$

True positives is the patients who died in ICU and were predicted to die, while false negatives are patients that died although the model predicted they would survive.

For evaluating the overall performance of models, we also use the AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) curve. It is used to measure the performance in a classification setting, and it specifies how good a model is in distinguishing between different classes. Higher values show that a model is good at predicting $0s$ as $0s$ and $1s$ as $1s$.

6.7 Results

Table 6 shows the results for mortality prediction for one-hot and semantic-based schemes. We observe that semantic-based embedding schemes for categorical variables perform better than the traditional one-hot encoding of high-dimensional categorical variables in both settings: single-valued and multi-valued encodings. The highest recall score of 79% is achieved for single-valued embeddings using the PS technique, which is significantly better than one-hot.[15] The one-hot model has a recall of 31% and 51% for both settings, which shows that the model is randomly classifying data and is unable to learn the relationship

[12] https://code.google.com/archive/p/word2vec/

[13] https://nlp.stanford.edu/projects/glove/

[14] https://bio.nlplab.org/

[15] The results are based on (11) with $\lambda = 0.7$ and (12) for calculating $d(x, y)$. We manually tune different values of λ, and the final value reported here is based on the best results achieved.

Table 6 Results for multi-valued embeddings

			Similarity-based		
Variable type	Evaluation measure	One-Hot	WUP	LCH	PS
Single-valued	Recall%	31	67	65	67
	AUC-ROC%	65	79	79	79
Multi-valued	Recall%	51	75	81	84
	AUC-ROC%	80	80	89	89

between data points. For single-valued embeddings, the test data does not contain the same categories that occur in the training set; instead, it is based on siblings of the categories occurring in the hierarchy. The results show that the trained model is able to learn correlations based on semantics and performs well for unseen categories in the test set. In this case, it is also clearly not so important which similarity measure is chosen as the basis for the embedding: the crucial factor is to include the semantics in the embedding, not the exact way of doing so.

For multiple diagnoses, hierarchy-based embeddings outperform all other schemes by achieving the highest recall of 84% and AUC-ROC score of 89%. We surmise that the improvement in the performance for multiple diagnoses depends on the hierarchy-based similarity measure. The final embeddings are aggregated in such a way that each dimension picks up the maximum similarity score based on (2). The hierarchy-based similarity measures favor the idea that for similar diagnoses, the respective dimensions will have high similarity and vice versa. This enables the network to learn better representations and perform better.

7 Conclusion and future work

It is common practice in Data Science and Machine Learning processes to accommodate non-numeric data by means of vector space embeddings. Such embeddings are either pre-trained or trained for the task at hand, using large volumes of data. This approach fails if not enough data are available to train the embeddings. On the other hand, in many areas of human endeavor, there is ample domain knowledge available that could be used to improve or abbreviate the data-based generation of embeddings.

In this work, we suggest an approach for encoding high-dimensional categorical variables based on domain knowledge in the form of hierarchies using semantic similarity. First, we compare the performance of existing semantic-based similarity measures using different datasets of word pairs. The results suggest that semantic similarity measures based on notions of information content (IC) depend on the training dataset for IC calculation. Given large amounts of training data, IC-based measures performed better than purely hierarchy-based measures in our experiments. Overall, Resnik's similarity measure (which is based on IC and hierarchy) achieves better performance on most of the example use-cases. For cases where we do not have enough training data, as in the low-resource cases we are targeting, purely hierarchy-based similarity measures achieve better performance in our experiments.

In our concept categorisation experiment, we compare our approach of defining an embedding from a hierarchy to existing data-based methods and show that the hierarchy-based embedding outperforms the data-based ones in most cases we considered. In

particular, the clustering task is dependent on the semantic similarity of the words. In such scenarios, hierarchy-based similarity and embeddings explicitly specify the semantic similarity, resulting in improved performance. The data-based methods (Google and Glove) encode the co-occurrence statistics of the training corpus, which do not necessarily capture the concepts' semantic relatedness. Such methods are more suitable for tasks that rely on the context of a given word. For instance, in sentiment classification, all words occurring in the same context have great significance in the final performance. Therefore, our results confirm our intuition that tasks that depend on semantic similarity can be performed better using a hierarchy, while for context-dependent tasks, data-based methods are more suitable.

The experimental results show that for the MIMIC use case with an imbalanced dataset, the proposed embedding schemes can aid the learning process and enhance the model performance. The model is able to classify categories that do not occur in the training set. The semantic embeddings calculated using any hierarchy-based similarity measure outperform the traditional one-hot encodings.

We should point out that the approach has limitations: intuitively adding *relevant* domain knowledge, e.g. in the form of a hierarchy, should improve the performance of ML tasks, as seen in our experiments. However, if the hierarchy is not relevant to task, and the grouping does not correspond to a notion of similarity that is useful for this task, there will not be any improvement. The addition of irrelevant domain information may even lead to worse results.

In the case where no relevant domain information is readily available, and it first needs to be generated, the approach is also less interesting, since the effort to construct a good hierarchy may be considerable. E.g. the disease hierarchy used in our study is based on centuries of medical science.

For future research, it would be interesting to combine context and semantic similarity-based embeddings to evaluate the performance in NLP tasks. The idea of semantic similarity can be extended to knowledge sources in the form of complex graphs that contain multiple relations for nodes in the graph. Also, our work concentrates on single (single-valued or multi-valued) categorical features, which can be embedded in isolation; incorporating domain knowledge about the connections between features, e.g. along the lines of Janusz (2014) might be very effective. Following this direction, using domain knowledge to understand the numeric variables and creating a unified framework for handling both numeric and categorical data in low-resource domains would be an interesting future direction.

Acknowledgements The authors would like to thank Egor V. Kostylev, Christos Dimitrakakis, and the anonymous reviewers for their useful comments and suggestions on drafts of this paper. This work was supported by the Norwegian Research Council via the SIRIUS Centre for Research-Based Innovation, Grant Nr. 237898.

Funding Open access funding provided by University of Oslo (incl Oslo University Hospital).

Availability of data and material MIMIC-iii is an openly available dataset developed and maintained by the MIT Lab for Computational Physiology (Johnson et al., 2016) and can be downloaded from their official website. Prior to requesting access to the dataset, the user should complete the 'Data or Specimens Only Research' course. Therefore we are not publishing the MIMIC dataset. The remaining datasets for WordNet are publicly available.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Ahmad, A., & Dey, L. (2007). A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters*, *28*, 110–118.

Almuhareb, A. (2006). *Attributes in lexical acquisition*. Ph.D. thesis, University of Essex.

Baroni, M., & Lenci, A. (2011). How we BLESSed distributional semantic evaluation. In *Roceedings of the GEMS 2011 workshop on GEometrical models of natural language semantics* (pp. 1–10). Association for computational linguistics.

Baroni, M., Murphy, B., Barbu, E., & Poesio, M. (2010). Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science, 34*(2), 222–254. https://doi.org/10.1111/j.1551-6709.2009.01068.x.

Bazan, J. G. (2008). Hierarchical classifiers for complex spatio-temporal concepts. In *Transactions on Rough Sets IX* (pp. 474–750). Berlin: Springer. https://doi.org/10.1007/978-3-540-89876-4_26.

Bengio, Y., Courville, A., & Vincent, P. (2012). Unsupervised feature learning and deep learning: a review and new perspectives. CoRR arXiv:1206.5538.

Cerda, P., & Varoquaux, G. (2020). Encoding high-cardinality string categorical variables. *IEEE Transactions on Knowledge and Data Engineering*.

Cerda, P., Varoquaux, G., & Kégl, B. (2018). Similarity encoding for learning with dirty categorical variables. *Machine Learning*, 107.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, *16*, 321–357.

Crone, S. F., Lessmann, S., & Stahlbock, R. (2006). The impact of preprocessing on data mining: an evaluation of classifier sensitivity in direct marketing. *European Journal of Operational Research*, *173*(3), 781–800.

d'Amato, C., Fanizzi, N., & Esposito, F. (2009). A semantic similarity measure for expressive description logics. CoRR arXiv:0911.5043.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. ACM Trans. *Information Systems*, *20*(1), 116–131. https://doi.org/10.1145/503104.503110.

Fitkov-Norris, E., Vahid, S., & Hand, C. (2012). Evaluating the impact of categorical data encoding and scaling on neural network classification performance: The case of repeat consumption of identical cultural goods. *Communications in Computer and Information Science*, *311*, 343–352.

Garchery, M., & Granitzer, M. (2018). On the influence of categorical features in ranking anomalies using mixed data. *Procedia Computer Science*, *126*, 77–86.

Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. (2015). *Semantic similarity from natural language and ontology analysis*. Morgan and Claypool Publishers.

Hsu, C. C. (2006). Generalizing self-organizing map for categorical data. *IEEE Transactions on Neural Networks*, *17*, 294–304.

Janusz, A. (2014). Algorithms for similarity relation learning from high dimensional data. In *Transactions on Rough Sets XVII* (pp. 174–292). Berlin: Springer. https://doi.org/10.1007/978-3-642-54756-0_7.

Janusz, A., Slezak, D., & Nguyen, H. S. (2012). Unsupervised similarity learning from textual data. *Fundamenta Informaticae*, *119*, 319–336.

Jia, Z., Lu, X., Duan, H., & Li, H. (2019). Using the distance between sets of hierarchical taxonomic clinical concepts to measure patient similarity. *BMC Medical Informatics and Decision Making*, *19*, 91.

Jian, S., Pang, G., Cao, L., Lu, K., & Gao, H. (2019). Cure flexible categorical data representation by hierarchical coupling learning. *IEEE Transactions on Knowledge and Data Engineering*, *31*, 853–866.

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference* (pp. 19–33).

Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R.G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*, 160035.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2005). Handling imbalanced datasets: a review. GESTS. *Int. Transactions on Computer Science and Engineering*, *30*, 25–36.

Kucera, H., & Francis, W. N. (1969). Computational analysis of present-day american english. *International Journal of American Linguistics*, 35.

Landes, S., Leecock, C., & Tengi, R. I. (1998). Building semantic concordances. In C. Fellbaum (Ed.) *Wordnet: an electronic lexical database* (pp. 197–216). MIT press.

Leacock, C., & Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In C. Fellbaum (Ed.) *Wordnet: an electronic lexical database., chap. 13* (pp. 265–283). MIT press.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98* (pp. 296-304). USA: Morgan Kaufmann Publishers Inc., San Francisco, CA.

Luong, M. T., Socher, R., & Manning, C. D. (2013). Better word representations with recursive neural networks for morphology. CoNLL.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *SIGKDD*, *3*, 27–32.

Midelfart, H. (2005). Supervised learning in the gene ontology part I: a rough set framework. In *Transactions on rough sets IV* (pp. 69–97). Berlin: Springer.

Midelfart, H. (2005). Supervised learning in the gene ontology part II: a bottom-up algorithm. In *Transactions on rough sets IV* (pp. 98–124). Springer.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.

Miller, G., & Charles, W. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, *6*, 1–28.

Mumtaz, S., & Giese, M. (2020). Frequency-based vs. knowledge-based similarity measures for categorical data. In *Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice, AAAI-MAKE 2020, CEUR Workshop Proceedings*, Vol. 2600. CEUR-WS.org. http://ceur-ws.org/Vol-2600/paper16.pdf.

Nguyen, S. H., Nguyen, T. T., Szczuka, M., & Nguyen, H. S. (2013). An approach to pattern recognition based on hierarchical granular computing. *Fundamenta Informaticae*, *127*(1–4), 369–384.

Nguyen, T. T. (2003). Rough set approach to domain knowledge approximation. *Fundam. Inf.*, *59*(2–3), 261–270.

Pedersen, T., Pakhomov, S. V., Patwardhan, S., & Chute, C. G. (2007). Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, *40*(3), 288–299.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014* (pp. 1532–1543). ACL. https://doi.org/10.3115/v1/d14-1162.

Pilehvar, M. T., Kartsaklis, D., Prokhorov, V., & Collier, N. (2018). Card-660: Cambridge rare word dataset - a reliable benchmark for infrequent word representation models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 1391–1401). Association for Computational Linguistics.

Potdar, K., Pardawala, T., & Pai, C. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, *175*, 7–9.

Purushotham, S., Meng, C., Che, Z., & Liu, Y. (2017). Benchmark of deep learning models on large healthcare mimic datasets. *Journal of Biomedical Informatics*, 83.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'95* (pp. 448–453).

Resnik, P. (1999). Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, *11*, 95–130.

Roy, A., Park, Y., & Pan, S. (2017). Learning domain-specific word embeddings from sparse cybersecurity texts. CoRR arXiv:1709.07470.

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, *8*(10), 627–633. https://doi.org/10.1145/365628.365657.

Smelser, N. J., & Baltes, P. B. (2001). *International encyclopedia of the social & behavioral sciences*. Elsevier.

Spearman, C. (1987). The proof and measurement of association between two things. *The American Journal of Psychology*, *100*, 441–471.

Szczuka, M., & Janusz, A. (2013). Semantic Clustering of Scientific Articles Using Explicit Semantic Analysis, 83–102.

Tarnowska, K., & Ras, Z. W. (2019). Sentiment analysis of customer data. *Web Intelligence Journal*, *17*, 343–363.

Tarnowska, K., Ras, Z. W., & Lynn, D. (2020). *Recommender System for Improving Customer Loyalty* Vol. 55. Berlin: Springer.

Von Eye, A., & Clogg Clifford, C. (1996). *Categorical variables in developmental research: Methods of analysis*. Elsevier Science.

Wang, B., Wang, A., Chen, F., Wang, Y., & Jay Kuo, C. C. (2019). Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, 8.

Wilson, D., & Martinez, T. (2000). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6.

Yildirim, O., Tan, R. S., & Acharya, U. R. (2018). An efficient compression of ECG signals using deep convolutional autoencoders. *Cognitive Systems Research*, *52*, 198–211.

Zhu, C., Cao, L., Liu, Q., Yin, J., & Kumar, V. (2018). Heterogeneous metric learning of categorical data with hierarchical couplings. *IEEE Transactions on Knowledge and Data Engineering*, *30*, 1254–1267.

Zhu, G., & Iglesias, C. A. (2015). Sematch semantic entity search from knowledge graph. In *Joint Proceedings of the 1st International Workshop on Summarizing and Presenting Entities and Ontologies and the 3rd International Workshop on Human Semantic Web Interfaces (SumPre 2015, HSWI 2015) co-located with the 12th Extended Semantic Web Conference (ESWC 2015)*, Vol. 1556. Portoroz: CEUR Workshop Proceedings, CEUR-WS.org.