JAMIA Open, 00(0), 2021, 1–7
doi: 10.1093/jamiaopen/ooab022
Research and Applications

Research and Applications

Exploiting hierarchy in medical concept embedding*

Anthony Finch [1], Alexander Crowell [1], Mamta Bhatia [1,2], Pooja Parameshwarappa [1], Yung-Chieh Chang [1], Jose Martinez,[1] Michael Horberg [12]

[1]Kaiser Permanente Mid-Atlantic Permanente Medical Group, Rockville, Maryland, USA and [2]Kaiser Permanente Mid-Atlantic Permanente Research Institute, Rockville, Maryland, USA

Corresponding Authors: Anthony Finch, MS, Kaiser Permanente Mid-Atlantic Permanente Medical Group, 2101 E. Jefferson St., Rockville, Maryland 20852, USA; Anthony.J.Finch@kp.org and Michael Horberg, MD, MAS, Kaiser Permanente Mid-Atlantic Permanente Research Institute, 2101 E. Jefferson St., Rockville, Maryland 20852, USA; Michael.Horberg@kp.org

Received 15 December 2020; Revised 2 February 2021; Editorial Decision 20 February 2021; Accepted 26 February 2021

ABSTRACT

Objective: To construct and publicly release a set of medical concept embeddings for codes following the ICD-10 coding standard which explicitly incorporate hierarchical information from medical codes into the embedding formulation.

Materials and Methods: We trained concept embeddings using several new extensions to the Word2Vec algorithm using a dataset of approximately 600,000 patients from a major integrated healthcare organization in the Mid-Atlantic US. Our concept embeddings included additional entities to account for the medical categories assigned to codes by the Clinical Classification Software Revised (CCSR) dataset. We compare these results to sets of publicly released pretrained embeddings and alternative training methodologies.

Results: We found that Word2Vec models which included hierarchical data outperformed ordinary Word2Vec alternatives on tasks which compared naïve clusters to canonical ones provided by CCSR. Our Skip-Gram model with both codes and categories achieved 61.4% normalized mutual information with canonical labels in comparison to 57.5% with traditional Skip-Gram. In models operating on two different outcomes, we found that including hierarchical embedding data improved classification performance 96.2% of the time. When controlling for all other variables, we found that co-training embeddings improved classification performance 66.7% of the time. We found that all models outperformed our competitive benchmarks.

Discussion: We found significant evidence that our proposed algorithms can express the hierarchical structure of medical codes more fully than ordinary Word2Vec models, and that this improvement carries forward into classification tasks. As part of this publication, we have released several sets of pretrained medical concept embeddings using the ICD-10 standard which significantly outperform other well-known pretrained vectors on our tested outcomes.

Key words: concept embedding, medical coding, ICD-10

BACKGROUND

Concept embedding has become an increasingly pervasive technique in applied machine learning. The core conceit of this methodology is that it is significantly more efficient to construct a dense representation of a large set of entities than it is to treat observations of those entities as orthogonal (i.e., sparse). Since its introduction by Mikolov et al., the Word2Vec embedding framework has become one of the most popular methods of constructing dense numerical represen-

LAY SUMMARY

Modern machine learning techniques using electronic medical records frequently construct numerical representations of diagnosis codes to use as inputs into these models. This makes it easier for the machine learning algorithm to interpret patient health information and tends to improve model performance. This technique, called "embedding," has typically been evaluated by comparing distances between codes with known hierarchical structures in diagnosis codes.

In this study, we propose an expansion to established embedding practices. Our technique incorporates hierarchical diagnosis information directly into the algorithm, instead of using it purely as an evaluation metric. We demonstrate that this technique improves the quality of the embeddings and of models that use those embeddings to predict patient outcomes.

tations of sparse feature sets.[1] This model and its competitors have revolutionized applications in Natural Language Processing such as neural machine translation, sentiment analysis, topic modeling, and other traditional NLP mainstays.[2–4]

Within a clinical context, concept embedding has had a growing impact on patient modeling. The first model to employ this concept was introduced by Choi et al, and a growing community of modelers and clinicians have demonstrated that concept embedding can be successfully applied to medical records.[5–7] In most such models, researchers treat patients as "documents" and medical codes (such as ICD-9 or ICD-10 codes) as "words".[5–8] However, this analogy is imperfect. Crucially, medical data occur within the context of a time continuum, as opposed to occurring within a simple sequence (as words do). This adds a layer of complexity, as modelers must choose if and how to incorporate time into their models.[7,8]

Furthermore, there is significantly less data available to clinical researchers than to language modelers, especially in the public domain. This sparsity of data would typically increase the value of techniques such as transfer learning, which allow modelers to "transfer" concepts learned from one dataset to another; however, surprisingly few pretrained concept embeddings have been published.[7] Those that have been published have typically followed the ICD-9 standard, which has been discontinued for several years.[6,7]

On the other hand, there are rich, publicly available datasets which can be used to augment clinical data. While these sources have frequently been used to evaluate concept embeddings or to offer an algorithm a "warm-start," there is limited research into how this data can be used to improve either the construction or use of medical concept embeddings. Patel et al., for example, used the hierarchical code structure of ICD categories to initialize code embeddings with their categories' embeddings when training a model for a medical NLP task.[9] However, this hierarchical training was only used as a "warm start," and has not appeared in other contexts outside of this NLP task. Poincare embedding has been proposed as a modern methodology for training hierarchical embeddings. As it was originally implemented, Poincare embeddings are trained explicitly on a tree of hierarchical relationships (e.g. WordNet),[10,11] as opposed to empirical co-occurrence; however, Beaulieu-Jones et al. demonstrated that the model could alternatively be trained with medical codes using empirical co-occurrence relationships.[11] The chief difficulty with these types of hyperbolic embeddings is that the resulting vectors can be used only sub-optimally with classification models operating in Euclidean space. While there is a growing literature built around hyperbolic spaces in machine learning, there are no mature tools that allow users to build models of this type.[12–14]

In this paper, we propose that modelers explicitly employ hierarchical data when training and using embeddings. We do this by adding clinical groups, as defined by the Clinical Classification Software (Revised), to patient records as additional entities in both the embedding and the modeling steps.[15] We expected this extension to improve the quality of code embeddings and to add valuable information when employing this data in classification tasks. Our technique is simple and requires no extensions to the mathematical formulation of the Word2Vec model; however, it also employs hierarchical data which has remained underexploited in the literature. While this technique does not inherently account for complexities introduced by time distances and code proximity, it is compatible with more sophisticated models which incorporate corrections for such considerations that were out of scope for this study.

We train both Continuous Bag-of-Words and Skip-Gram Word2Vec models on a large, recent dataset from a regional health system and demonstrate how the resulting embeddings perform on clustering and modeling tasks. In contrast to previous research, our data are both extensive and current, allowing us to build embeddings that are practically useful in contemporary health contexts without resorting to imperfect translation between coding standards. We examine how training code categories can improve performance on both metrics. Critically, we also publish the pretrained embedding vectors for all our training methods.

METHODS

Data

To train our models, we employed data from the Kaiser Permanente Mid-Atlantic States (KPMAS) medical system. KPMAS is an integrated medical system serving approximately 780,000 members in Maryland, Virginia, and the District of Columbia. KPMAS has a comprehensive electronic medical record system which includes data from all patient interactions with primary or specialty caregivers, from which all data are derived. The study collected two datasets. One included patient records for the 6 months prior to 1/1/19 (used when training classification models) and the other included patient records for the 12 months after 1/1/19 (used when constructing code embeddings).

To construct a dataset to train embedding models, we included all members of the KPMAS system that were aged 18 or older with active coverage as of January 1, 2019. With these inclusion/exclusion criteria, we had 626,269 members. We present a demographic summary of this population in Table 1.

Once we had identified this set of patients, we gathered all medical codes assigned to patients during the period from January 1, 2019 through December 31, 2019. We eliminated codes with fewer than 5 appearances within the dataset and all patients with 1 or fewer codes assigned. After employing this preprocessing, 537,451 patient records remained with 11.4 million distinct code instances, representing 5,428 distinct codes. To build the code categories, we

Table 1. Demographic details of 626,269 members of KPMAS

Demographic	Result
Average age	48.1 years
Age inter-quartile range	27.9 years
Percentage female	53.8%
Percentage Asian/Pacific Islander	13.0%
Percentage Black/African American	36.3%
Percentage Hispanic/Latino origin	12.0%
Percentage White	28.1%
Percentage other/unknown race	10.4%

employed the 2020 version of the Clinical Classifications Software Refined (CCSR) for ICD-10.[15] Within the CCSR dataset, each ICD-10 code is assigned to one or more categories, with one such category designated as the default. For the purposes of our modeling, each code was considered a member of only its default category.

To build classification models, we employed similar data collected over the 6-month period leading up to January 1, 2019. We limited to this period to ensure that all data was recent, since our model did not include adjustments for older codes. Because mortality and hospitalization events were so rare in younger patients, we included only patients age 45 or older by January 1, 2019 in both of our classification datasets. This yielded 311,179 patients with a total of 4.3 million instances of embedded codes.

Model

The Word2Vec architectures were originally proposed by Mikolov, et al.[1,16] In the classic Continuous Bag-of-Words (CBOW) approach, the average context surrounding a word is used to predict a target word. This objective can be expressed as a softmax which maximizes the network's ability to obtain the missing word. In contrast, the most typical implementation of the Skip-Gram model employs a Negative Sampling technique.[16] In this construction, each word is paired with each of its context words and with several random "negative" samples. The task is then for the model to distinguish between true pairs and randomly sampled ones. The mathematics of these models are described in detail by Rong.[17] For our modeling, we employ model implementations provided by GenSim.[18]

Model training

We trained 12 sets of embeddings with settings differing across three parameters. Our first parameter was the selection of training algorithm, where we trained both CBOW and SG models. Each model was trained using dimension k of 10, 50, and 100. Furthermore, each model-dimension combination was trained with categories and codes trained separately and together (referred to hereafter as "co-trained embeddings" or "co-embeddings"). Each model was trained for 10 iterations. We employed an arbitrarily large context window (100), since all codes necessarily occurred within a short period (1 year).

Evaluation metrics

As Xiang et al., we employ two sets of evaluation metrics. Our first set compares naïve clusters of our code embeddings to the known categories determined by CCSR.[7] Note that category embeddings were not used in any way for evaluating our clustering metrics. This methodology was used successfully in Cai by comparing the resulting clusters using Normalized Mutual Information scores (NMI).[8] NMI scores provide a method to compare two groups of clusters operating on the same set of points. The score is normalized so orthogonal clusters obtain an NMI score of 0 and perfectly correlated clusters obtain a score of 1. As detailed by Vinh et al., we observed that NMI scores tended to increase with the number of clusters used; thus, we also included Adjusted Mutual Information (AMI) scores, which adjusted NMI scores to offer more consistent behavior as we increased the number of clusters.[19]

Finally, we explored the use of silhouette scores when using the CCSR categories as our canonical labels. Silhouette scores compute the average ratio of the distance between a point and its cluster center to the distance between that point and its next-closest cluster center. For the sake of this study, we have employed silhouette scores by choosing a code's canonical cluster as its assignment. While the NMI and AMI scores are standard within the literature, we felt that silhouette scores could offer a useful alternative perspective, since it was possible that codes would be near their canonical clusters while still technically falling into another, similar cluster. For example, it was conceivable that codes from two clusters with similar codes may be randomly distributed within a space that was well-defined and distant from other clusters. In such a case, NMI and AMI scores may show very low scores for these clusters when comparing with the known classifications; however, the silhouette score (as described here) may penalize these clusters less than a raw Mutual Information score would.

For each set of the code embeddings, we produced 20 sets of clusters using the K-Means clustering algorithm, with between 20 and 400 clusters. Then, we compared the labels produced by these clusters to the canonical clusters as labeled by the CCSR categories. We employed NMI and AMI as measures of comparisons. In addition, we constructed artificial cluster centers by assigning each code to its canonical cluster and computing the resulting centers. These canonical centers were then used to compute silhouette scores from a code to its "true" center.[20]

For our second set of evaluations, we built logistic regression models to predict patient mortality and unplanned hospital admission events on patients 45 or older. Models were trained on patient code embeddings using the Scikit-Learn implementations of cross-validation and logistic regression.[21] For each model, we averaged all de-duplicated codes observed for each patient from the period between January 1st, 2018 and December 31st, 2018. With each set of embeddings, we produced three sets of features: one with only the code embeddings, one with only the category embeddings, and one that concatenated both. We used average ROC-AUC scores from a 10-fold cross-validation to evaluate how well a model would perform when predicting the given target. Our two targets were patient mortality throughout the year of 2019 and patient hospital admissions during the first six months of 2019. Cross validation fold membership was consistent across all embedding models. Note that no additional data (e.g. demographics) or deep learning models were used in building this model to isolate the effects of the various embedding algorithms.

Med2Vec comparison

As a baseline comparison, we employed pretrained vectors published by Choi as part of the Med2Vec distribution.[22] These vectors were trained with $k = 200$ dimensions on an external dataset by the original authors. Med2Vec was originally trained using the ICD-9 coding standard; we employed ICD-9 to ICD-10 mapping.[23]

Table 2. Clustering scores by embedding method

Model	Embedding	Embedding dimension	NMI (200)	NMI (400)	AMI (200)	AMI (400)	Silhouette
Med2Vec	Sep	200	0.2011	0.2400	0.0498	0.0480	−0.5822
CBOW	Sep	10	0.4254	0.4774	0.1493	0.1408	−0.4647
CBOW	Sep	50	0.4261	0.4635	0.1776	0.1679	−0.3897
CBOW	Sep	100	0.4011	0.4335	0.1702	0.1622	−0.3884
SG	Sep	10	0.5221	0.5737	0.2052	0.1837	−0.3161
SG	Sep	50	0.5500	0.5905	0.2754	0.2572	−0.1981
SG	Sep	100	0.5288	0.5751	0.2852	0.2797	−0.2001
CBOW	Co	10	0.4313	0.4773	0.1590	0.1429	−0.4639
CBOW	Co	50	0.4576	0.4935	0.2287	0.2133	−0.3549
CBOW	Co	100	0.4478	0.4825	0.2323	0.2154	−0.3448
SG	Co	10	0.5220	0.5798	0.2035	0.1913	−0.3197
SG	Co	50	**0.5726**	**0.6144**	**0.2979**	0.2864	−0.1648
SG	Co	100	0.5605	0.6134	0.2963	**0.3107**	**−0.1615**
Med2Vec*	N/A	200	0.2755	0.3472	0.0524	0.0437	−0.5001
SG*	Co	100	**0.5843**	**0.6448**	0.2559	0.2598	−0.1722

*Models trained on a subsample of codes which occurred in the translated Med2Vec comparison.

Note that the "Co" designation in the embedding column indicates a model which trained category and code embeddings jointly, whereas a "Sep" designation indicates that these embeddings were trained separately.

In order to verify that performance differences between Med2Vec embeddings and our own results were not solely attributable to our new dataset, we also trained the Med2Vec model on our data using its default settings, including the default vector size (200) and a training regime of 10 epochs. We grouped all codes occurring on the same calendar date as Med2Vec "visits." Our Med2Vec model benchmark did not include categorical entities or other novel innovations. Our Med2Vec model was trained using the public repository associated with the original publication.[5]

RESULTS

Cluster evaluation

In the clustering evaluation, the Skip-Gram model consistently outperformed CBOW, given the same dimensionality and co-embedding setting. Furthermore, we found that co-embedding and added embedding dimensionality tended to improve clustering performance. We observed that the Skip-Gram model with co-embedding and either 50 or 100 dimensions were the best two models by a wide margin, although these models were very competitive with one another. On the AMI task with 400 clusters, the 100-dimensional SG with co-trained embeddings achieved a score of 0.3107, comparing favorably with the second-highest score of 0.2864 achieved by the equivalent model with 50-dimensional embeddings. In contrast, the 50-dimensional embedding model achieved a score of 0.6144 on the NMI task with 400 clusters, compared to a score of 0.6134 achieved by the 100-dimensional version. In both cases, the third-best model trailed by a relatively wide margin. The 100-dimensional SG model without co-training achieved an AMI on 400 clusters of 0.2797, and the 50-dimensional SG model without co-training achieved an NMI on 400 clusters of 0.5905. In addition to these numerical results, we also analyzed the qualitative results of embeddings by examining how codes from similar categories interacted (Supplement 1).

As a final check, we also filtered embeddings for the SG model with co-training and 100 dimensions to only include codes which were present in the Med2Vec dataset. In Table 2, these rows are marked with a (*) to indicate that they do not represent the complete set of embedded codes. This subsampling allowed us to perform a direct comparison between the two to verify that performance differences were not attributable to excluding codes. We found that there was very little difference in performance on the clustering tasks due to code exclusions. When excluding these codes, NMI scores both increased (since this increases the ratio of clusters to true labels, this is expected behavior); however, the AMI score with 400 clusters decreased from 0.3107 to 0.2598. This new score was still better than all but three of the other models and was significantly better than the score of 0.0437 achieved by the pretrained Med2Vec embeddings. These results lead us to conclude that Word2Vec models displayed substantial improvement over the pretrained Med2Vec embeddings on the clustering task. Furthermore, we found that training Med2Vec embeddings on our own data produced inferior results to those obtained by the pretrained embeddings, with an AMI score of 0.2400 with 400 clusters and a silhouette score of −0.5822.

Classification evaluation

In almost all cases, combining the code and category embeddings as feature inputs yielded increased performance, although this difference was not always significant. As shown in Table 3, when we control for the embedding model, dimension, and level of co-training, the model which included both categories and codes outperformed the stem-only model in all cases except when training the Skip-Gram model with dimension 50 on the Mortality target. Co-training the code and category embeddings tended to marginally improve performance. When controlling for embedding model, dimensionality, and included data, co-training improved the ROC-AUC score in 77.8% of cases with the mortality target and 55.6% of cases with the hospitalization target.

Overall, the Word2Vec models compared favorably with the Med2Vec model trained on our own data, which produced a ROC-AUC score of 87.09% on the mortality target and 79.13% on the hospitalization target. This performance was substantially lower than the traditional Skip-Gram without co-embedding or categorical features in both cases. The co-trained SG model with dimension $k = 100$ tended to be our best-performing model, although it was marginally outperformed on the hospitalization task by the separately-trained model when our logistic regression was fit only

Table 3. Mortality model performance by embedding method.

Embedding Model	Embedding	Dimension	Code-Only AUC	Category-Only AUC	Combined AUC
Med2Vec	Sep	200	0.8709	N/A	N/A
CBOW	Sep	10	0.8788	0.8632	0.8810
CBOW	Sep	50	0.8824	0.8696	0.8859
CBOW	Sep	100	0.8830	0.8724	0.8903
SG	Sep	10	0.8812	0.8655	0.8865
SG	Sep	50	0.8914	0.8714	0.8929
SG	Sep	100	0.8942	0.8755	0.8951
CBOW	Co	10	0.8736	0.8643	0.8756
CBOW	Co	50	0.8831	0.8710	0.8882
CBOW	Co	100	0.8864	0.8753	0.8937
SG	Co	10	0.8827	0.8652	0.8854
SG	Co	50	0.8937	0.8739	0.8936
SG	Co	100	**0.8951**	**0.8777**	**0.8972**
Med2Vec*	N/A	200	0.7851	N/A	N/A
SG*	Co	100	0.8882	0.8713	0.8905

*Models trained on a subsample of codes which occurred in the translated Med2Vec comparison.

Note that the "Co" designation in the embedding column indicates a model which trained category and code embeddings jointly, whereas a "Sep" designation indicates that these embeddings were trained separately.

Table 4. Hospital admission model performance by embedding method.

Embedding model	Co-embedding	Dimension	Code-only AUC	Category-only AUC	Combined AUC
Med2Vec	Sep	200	0.7913	N/A	N/A
CBOW	Sep	10	0.7912	0.7753	0.7923
CBOW	Sep	50	0.7929	0.7824	0.7940
CBOW	Sep	100	0.7919	0.7827	0.7949
SG	Sep	10	0.7914	0.7770	0.7924
SG	Sep	50	0.7946	0.7822	0.7955
SG	Sep	100	**0.7954**	0.7844	0.7969
CBOW	Co	10	0.7869	0.7791	0.7896
CBOW	Co	50	0.7916	0.7810	0.7940
CBOW	Co	100	0.7929	0.7842	0.7959
SG	Co	10	0.7888	0.7779	0.7905
SG	Co	50	0.7951	0.7842	0.7959
SG	Co	100	0.7951	**0.7852**	**0.7971**
Med2Vec*	N/A	200	0.7107	N/A	N/A
SG*	Co	100	0.7899	0.7808	0.7911

*Models trained on a subsample of codes which occurred in the translated Med2Vec comparison.

Note that the "Co" designation in the embedding column indicates a model which trained category and code embeddings jointly, whereas a "Sep" designation indicates that these embeddings were trained separately.

with code data (Table 4). We also compared this model directly with the pretrained Med2Vec benchmark. To compare against the pretrained Med2Vec vectors, we eliminated all codes from patient records which were not represented in our converted Med2Vec embedding dictionary. We then trained a new model on this reduced dataset (denoted with a * in Tables 3 and 4). We found that this reduction in feature data only minimally decreased model performance.

DISCUSSION

Contributions

We introduced several new approaches tailored to the use of embedding algorithms on medical codes. We began by proposing the use of code categories as additional features in the construction and use of embeddings, whereas previous work had only used categorical labels to evaluate embedding performance. With this basis, we explored how category embeddings could be co-trained with basic code embeddings to improve the quality of both sets of embeddings.

Our results yielded several critical new observations. Combining embeddings from both codes and their high-level categories as feature inputs substantially improves performance on predictive modeling tasks. Furthermore, co-trained hierarchical and specific embedding tasks can improve the performance of both sets of embeddings on clustering and classification tasks.

In addition to these theoretical observations, we note that there are relatively few pretrained sets of medical code embeddings in the literature. Most of those which have been made available employ the ICD-9 standard, which is outdated and not directly translatable to the ICD-10 standard. A major additional contribution from this study is our public release of an additional set of pretrained code vectors which adheres to the latest standards.

Model performance

All embedding models performed very well on the classification tasks, with little practical difference between methods. Our most significant observation was that including category embeddings

consistently improved model performance. Among models with equivalent inputs, we saw that SG outperformed CBOW and that co-trained embeddings tended to outperform independently trained versions of the same model.

We observed significantly more differentiation on the clustering task. Our SG model outperformed the CBOW model by a wide margin in all cases (e.g. SG with dimension 100 and without co-training achieved an AMI with 400 clusters of 28.0% compared to CBOW's 16.2%), and co-training the embeddings consistently improved mutual information scores (e.g. the equivalent SG model with co-trained embeddings achieved an AMI with 400 clusters of 31.1%). While results from this paper are not directly comparable with those achieved by others because of differences in code inclusion and datasets, we note that our margin of outperformance is similar to that achieved by Cai, et al., whose novel attentional extension to Word2-Vec achieved an improvement of 1.5 percentage points in NMI over traditional Skip-Gram (65.46% to 63.96%) on a similarly-sized dataset [7]. This is comparable to our performance improvement of 2.41 percentage points (61.44% to 59.05%) when comparing 50-dimensional Skip-Gram models using a 400-cluster NMI score.

Finally, we observed substantial improvements over public benchmarks. While the pretrained Med2Vec embeddings performed better than our re-trained embeddings on the clustering tasks, the pretrained embeddings performed very poorly on the classification task. Furthermore, both sets of embeddings underperformed relative to their Word2Vec counterparts. Given this contrast in performance, it seems likely that there are sufficient differences between ICD-9 and ICD-10 that embeddings trained in the prior standard should not be used to predict on datasets using the latter.

Limitations and further investigation

Our models were trained with a limited dataset, including members of a single insured health system and in a single year. While this helped us to train many embedding models with limited computational resources, larger datasets may not require the hierarchical information we employed. The limited size of this dataset also eliminated many rare codes because we had insufficient data with which to model them. We have intentionally incorporated a large window size (100) to minimize complications arising from the difference between proximity with respect to sequence and proximity with respect to time. In principle, this could train a code's embedding with less relevant neighbors and introduce noise; however, this decision also simplifies the model definition and ensures that important code pairs are not eliminated by chance. In addition, our predictive evaluations did not make extensive use of the richness of the embedding structure. We employed a simple model to evaluate only the results of the embeddings, which may not have captured all available information in the embeddings. Further investigation is necessary to determine whether various embeddings will behave differently under more sophisticated modeling regimes.

We compared results from our model to results generated by the Med2Vec pretrained vectors.[22] This was not an ideal comparison because we had to eliminate some codes from our dataset which did not have appropriate matches in the Med2Vec dataset. For all evaluations, we only included codes with a direct mapping; all other codes were removed from both evaluation tasks. Given sufficient time and expert clinical input, it may be possible to find approximate matches for most unmapped codes; however, this is not likely to be practical in most settings considering the complexity of mapping between coding standards. This left us with 2,959 embedded codes (54.5%). For our classification task, we were forced to eliminate 965,617 code instances that were included when training classification tasks for our own models (22.5%). This highlights the difficulties of trying to convert vectors from one coding standard to another. The results we obtained when deploying our own Word2-Vec model on this same down-sampled dataset indicate that this may have caused the Med2Vec model to slightly overperform on the clustering task and slightly underperform on the classification task. In comparison, any researcher interested in using our embeddings can always fall back to the category embeddings we have provided, which are much more universal than specific code standards.

When training our Med2Vec benchmark, we did not optimize training parameters or apply our categorical labels as additional entities. We had difficulty training this model, since it depends upon outdated versions of Python and several dependencies. Furthermore, we found that the model was very slow to train (approximately 35 hours) when compared to the GenSim model (20-40 minutes, depending on size and data). To limit the scope of work on our benchmark, we employed the default parameters suggested by the model's original authors.[5] While it may be possible for Med2Vec to achieve results comparable to our own with optimization on our specific modeling tasks, our results suggest that the model could be further improved by incorporating categorical embeddings.

We employed only default categories from the CCSR classification set. This was necessary to limit the scope of our investigation; however, future research may benefit from applying all categories to each of their relevant codes. This may further improve the quality of hierarchical embedding strategies by adding relevant entities to the embedding task, allowing both the code and category embeddings to incorporate secondary diagnostic information. However, adding this data may require careful selection of how secondary categories are weighed in relation to primary categories.

Our algorithms do not include sophisticated corrections for time distance information. While each code's context only includes other codes occurring within one year, this does not allow for proximity to strongly influence our embeddings. Fortunately, it is simple to incorporate the insights from this paper in alternative models.[8]

CONCLUSION

In this study, we introduced the idea of using medical code categories to augment the creation of both code embeddings and models based on those embeddings. We demonstrated that co-training embeddings with their hierarchical information improved performance on a variety of tasks. Finally, we have published pretrained code vectors as part of this research, which are more up-to-date than alternatives that are publicly available.

AUTHOR CONTRIBUTION

Anthony Finch proposed the work, performed the core data analysis, and supervised other technical members of the team. Alexander Crowell assisted in data analysis, data acquisition, experimental design, and literature review. Mamta Bhatia led the literature review. Pooja Parameshwarappa presented some of the research articles that inspired this work and translated pretrained Med2Vec vectors to be used as a baseline. Yung-Chieh Chang helped build the classification model datasets and participated in literature discussions that inspired this research. Jose Martinez managed and supported several of the personnel that participated in this project. Michael Horberg

directly supervised this project and contributed significantly to the research plan.

ACKNOWLEDGMENT

We would like to acknowledge Richard McCarthy, MD, The Permanente Medical Group Associate Executive Director for the Mid-Atlantic States, for his support and critical input.

COMPETING INTERESTS

The authors have no conflict of interests to declare.

SUPPLEMENTARY MATERIAL

Trained embedding dataset is available in the Dryad Digital Repository: https://doi.org/10.5061/dryad.v9s4mw6v0.

DATA AVAILABILITY

All other data underlying this article will be shared on reasonable request to the corresponding author.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

1. Mikolov T, Chen K, Corrado G, *et al.* Efficient estimation of word representations in vector space. *arXiv* 2013; arXiv1301.3781. Accessed March 8, 2021..
2. Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. *Adv Neural Inf Proc Syst* 2017; 30: 5998–6008.
3. Budhkar A, Rudzicz F. Augmenting word2vec with latent Dirichlet allocation within a clinical application. arXiv 2018; arXiv1808.03967. Accessed March 8, 2021.
4. Chen Q, Sokolova M. Word2vec and doc2vec in unsupervised sentiment analysis of clinical discharge summaries. arXiv Preprint arXiv 2018. arXiv1805.00352. Accessed March 8, 2021.
5. Choi E, Bahadori M, Searles E, *et al.* Multi-layer representation learning for medical concepts. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 2016; 1495–1504.
6. Beaulieu-Jones BK, Greene CS, Pooled Resource Open-Access ALS Clinical Trials Consortium. Semi-supervised learning of the electronic health record for phenotype stratification. *J Biomed Inform* 2016; 64: 168–78. doi: 10.1016/j.jbi.2016.10.007. Epub 2016 Oct 12. PMID: 27744022.
7. Xiang Y, Xu J, Si Y, *et al.* Time-sensitive clinical concept embeddings learned from large electronic health records. *BMC Med Inform Decis Making* 2019; 19(Suppl 2): 58. Published 2019 Apr 9. doi:10.1186/s12911-019-0766-3
8. Cai X, Gao J, Ngiam K, *et al.* Medical concept embedding with time-aware attention. arXiv Preprint arXiv 2018; arXiv1806.02873. Accessed March 8, 2021.
9. Patel K, Patel D, Golakiya M, *et al.* Adapting pre-trained word embeddings for use in medical coding. *BioNLP* 2017; 302–306.
10. Nickel M, Kiela D. Poincaré embeddings for learning hierarchical representations. *Adv Neural Inf Proc Syst* 2017; 30: 6338–47.
11. Beaulieu-Jones B, Kohane I, Beam A. Learning contextual hierarchical structure of medical concepts with poincairé embeddings to clarify phenotypes. *Pac Symp Biocomput* 2019; 24: 8–17.
12. Ganea O, Bécigneul G, Hofmann T. Hyperbolic neural networks. *Adv Neural Inf Proc Syst* 2018; 31: 5335–45.
13. Shimizu R, Mukuta Y, Harada T. Hyperbolic Neural Networks++. *arXiv Preprint arXiv* 2020. arXiv:2006.08210. Accessed March 8, 2021.
14. Gulcehre C, Denil M, Malinowski M, *et al.* Hyperbolic attention networks. *arXiv Preprint arXiv* 2018. arXiv1805.09786. Accessed March 8, 2021.
15. Clinical Classifications Software Refined (CCSR) for ICD-10 CM Diagnoses. 2020. Accessed March 8, 2021. Available from: https://www.hcup-us.ahrq.gov/toolssoftware/ccsr/ccs_refined.jsp#download
16. Mikolov T, Sutskever I, Chen K, *et al.* Distributed representations of words and phrases and their compositionality. *Adv Neural Inf Proc Syst* 2013; 26: 3111–9.
17. Rong J. Word2vec parameter learning explained. *arXiv Preprint arXiv* 2014; arXiv1411.2738. Accessed March 8, 2021.
18. GENSIM: Topic Modelling for Humans. 2020. Accessed March 8, 2021. Available from: https://radimrehurek.com/gensim/.
19. Vinh N, Epps J, Bailey J, *et al.* Information theoretic measures for clustering comparison: variants, properties, normalization and correction for chance. *J Mach Learn Res* 2010; 11: 2837–54.
20. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987; 20: 53–65. doi: https://doi.org/10.1016/0377-0427(87)90125-7.
21. Scikit Learn: Machine learning in Python. 2020. Accessed March 8, 2021. Available from: https://scikit-learn.org/stable/.
22. Choi E. Welcome to my humble abode. 2020. Accessed March 8, 2021. Available from: https://mp2893.com/.
23. ICD9CM to ICD10CM. 2020. Accessed March 8, 2021 . Available from: https://github.com/bhanratt/ICD9CMtoICD10CM