Journal of the American Medical Informatics Association, 00(0), 2021, 1–8
doi: 10.1093/jamia/ocab030
Research and Applications

AMIA INFORMATICS PROFESSIONALS. LEADING THE WAY.
OXFORD

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocab030/6275415 by University of Cambridge user on 01 June 2021

Research and Applications

Deep-learning-based automated terminology mapping in OMOP-CDM

Byungkon Kang,[1,*] Jisang Yoon,[2,*] Ha Young Kim,[2,*] Sung Jin Jo,[3] Yourim Lee,[4] and Hye Jin Kam[5]

[1]Department of Computer Science, State University of New York, Incheon, South Korea, [2]Graduate School of Information, Yonsei University, Seoul, South Korea, [3]Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, North Gyeongsang,South Korea, [4]RWE Analytics, EvidNet, Seongnam-si, Gyeonggi-do, South Korea, and [5]Healthcare, Life Solution Cluster, New Business Unit, Hanwha Life, Seoul, South Korea

*These authors contributed equally.
Corresponding Author: Hye Jin Kam, PhD, Healthcare, Life Solution Cluster, New Business Unit, Hanwha Life, 50, 63-Ro, Yeongdeungpo-gu, Seoul, South Korea (kam.hyejin@gmail.com)

Received 26 November 2020; Revised 7 January 2021; Accepted 5 February 2021; Editorial Decision 29 January 2021

ABSTRACT

Objective: Accessing medical data from multiple institutions is difficult owing to the interinstitutional diversity of vocabularies. Standardization schemes, such as the common data model, have been proposed as solutions to this problem, but such schemes require expensive human supervision. This study aims to construct a trainable system that can automate the process of semantic interinstitutional code mapping.
Materials and Methods: To automate mapping between source and target codes, we compute the embedding-based semantic similarity between corresponding descriptive sentences. We also implement a systematic approach for preparing training data for similarity computation. Experimental results are compared to traditional word-based mappings.
Results: The proposed model is compared against the state-of-the-art automated matching system, which is called Usagi, of the Observational Medical Outcomes Partnership common data model. By incorporating multiple negative training samples per positive sample, our semantic matching method significantly outperforms Usagi. Its matching accuracy is at least 10% greater than that of Usagi, and this trend is consistent across various top-k measurements.
Discussion: The proposed deep learning-based mapping approach outperforms previous simple word-level matching algorithms because it can account for contextual and semantic information. Additionally, we demonstrate that the manner in which negative training samples are selected significantly affects the overall performance of the system.
Conclusion: Incorporating the semantics of code descriptions more significantly increases matching accuracy compared to traditional text co-occurrence-based approaches. The negative training sample collection methodology is also an important component of the proposed trainable system that can be adopted in both present and future related systems.

Key words: deep-learning, terminology mapping, embedding, common data model, automated mapping

© The Author(s) 2021. Published by Oxford University Press on behalf of the American Medical Informatics Association.
All rights reserved. For permissions, please email: journals.permissions@oup.com

INTRODUCTION

In the past, many researchers gathered long-term patient data from a single institution, or from cohorts that have been created by investing large amounts of resources. However, in recent years, a number of methods for integrating multiorganization or multinational data have been developed to obtain large numbers of patient datasets to ensure statistical significance. This trend is partially influenced by the high demand for large datasets used to train machine learning algorithms, which have become prevalent. However, a more fundamental motivation is that there are certain biases among patients who visit different institutions, depending on the geological and cultural traits of those institutions.[1]

Despite the need for enhancing the connectivity and integration of data to reduce such biases, the process of multi-institute research faces several obstacles. In the medical domain, multiple codes and terminologies are often used to describe the same concept depending on the contextual purpose and function of that concept.[2–4] Because health information system and electronic medical record databases use subsets of the full medical vocabulary set, it is necessary to perform mapping operations to bridge individual vocabularies to form standard or common vocabularies.[4,5]

One effort to standardize medical data has been presented in the form of a common data model (CDM) that provides a conceptual basis for data integration.[6–8] CDMs encompass both the external structures of data and the process of internal vocabulary standardization. The use of CDMs is important in terms of research quality because standardizing data is more difficult when performing lossless mapping between vocabulary sets than when mapping structures. Despite various efforts, standardizing many datasets with differing backgrounds across multiple institutions is still a highly challenging task requiring significant time and manual effort.[9–11]

We will discuss the automation and efficiency of terminology mapping in the Observational Medical Outcomes Partnership (OMOP)-CDM,[12–15] which is one of the fastest-growing CDMs in medical informatics. The OMOP-CDM is operated by the Observational Health Data Sciences and Informatics (OHDSI) consortium that develops and provides various tools for data standardization. Many services and tools provided by OHDSI use a component called Usagi,[16] which we will briefly introduce in the next section.

Usagi

Usagi is an open-source software used to perform automatic vocabulary mapping. The core of Usagi's engine is term frequency-inverse document frequency (TF-IDF)-based similarity computation. TF-IDF is a simple statistical technique that is used to compute the importance of each word in a document numerically. The TF component reflects the frequency of a given term in a particular document and the IDF component measures the uniqueness of a term over all documents in a set. A high TF-IDF value for a word indicates that the word is a good candidate for characterizing the given document. Usagi's approach to performing vocabulary mapping is to treat the descriptive sentence of each vocabulary as a 1-sentence document and compute a vector of TF-IDF scores for each vocabulary. Each vocabulary is then matched with the vocabulary in the standard set having the highest cosine similarity between score vectors. Although TF-IDF is easy to compute and somewhat interpretable, it does not incorporate many useful types of textual information, such as word co-occurrence statistics, positions, and semantics.

Word and sentence embedding

We have developed a single framework that can solve the aforementioned problems using deep-learning. Specifically, we propose an embedding technique that is well suited for semantic representation and comparison. Word embedding is a machine learning technique that aims to compute vector representations of words. The goal of representing words as vectors is to reflect the semantics of words: semantically related words are expected to be in close proximity, whereas dissimilar words should be separated. The most frequently used technique for word embedding is statistical analysis of the co-occurrence patterns of words in a large training corpus. The rationale behind this technique is that more similar words tend to appear more frequently than less similar words, which should correspond to nearby Euclidean vectors. Most learning algorithms proceed by assigning higher scores to nearby vectors that correspond to co-occurring words. Prevalent algorithms include Word2Vec,[17] fastText,[18] and ELMo.[19] Based on word embeddings, sentences can also be represented as vectors. The overall concept is similar to that of word embedding computation, whereby sentences with similar semantics can be employed to predict similar adjacent sentences. There have been many implementations of this concept, including SkipThought,[20] BERT,[21] and InferSent.[22]

OBJECTIVE

We aim to tackle the problem of automated concept mapping using deep-learning-based natural language processing (NLP). By using deep NLP, we not only can perform simple word matching but also reflect the underlying semantics of concepts. The proposed framework also allows us to incorporate supplementary information, such as hierarchies of concepts, to extend our main approach into a technique that reflects the structural characteristics of concept sets. This mapping framework allows specific techniques to be translated to generalized settings, such as when mapping between different terms is required.

MATERIALS AND METHODS

Data collection and preprocessing

Among the multiple domains ("Condition," "Drug," "Procedure," "Visit," "Device," "Specimen") of OMOP-CDM,[12,23,24] this work focuses on the "Condition" domain, which is related to patient conditions (diagnoses or symptoms observed by a provider or reported by the patient), for the testing application of our algorithm. Our dataset includes manual mappings performed by medical experts from 5 tertiary hospitals in South Korea based on the OMOP-CDM. This dataset is a real-world data-mapping record containing approximately 83 000 terms. The target concepts in our mapping belong to the SNOMED clinical terms (CT) code and the source concepts were paired with natural language descriptions in English (source names) through a manual mapping process consisting of initial mapping, cross-checking, and third-party review.[25] The mapping managers are nurses with years of experience who are familiar with clinical terminology and have worked on the basis of the terminology system and the operation manual for mapping, and the mapping of the same term is performed by 2 people as separate work. The final mapping is decided through a 3-person review and discussion.

We used ICDO3-mapped data provided by OHDSI for training and validation to increase the scale of the training data. Mapping between a source and target entails a 1-to-many relationship; there-

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocab030/6275415 by University of Cambridge user on 01 June 2021

Table 1. Breakdown of training data counts

	Before (SNOMED-CT only)	After (+ICDO3, synonyms)
Number of source codes	83 113	83 113
Number of concept ids	67 079	67 418
Number of rows	86 820	95 471

fore, there are fewer unique source codes compared to the total number of rows in the mapping table. In addition to raw data, we included synonym data from OHDSI and hierarchy-related information. Table 1 summarizes the changes in the number of concepts after this augmentation was applied. Adding ICDO3 data is intended to make the model learn additional semantic representation.

Each term in a code set is typically composed of alphanumeric characters, meaning the code itself does not carry any semantics. However, all codes are paired with corresponding definition sentences, which provide natural language descriptions of the codes. Because definition sentences are intended to be semantically equivalent to their corresponding codes, we can leverage this relationship to measure the similarity between any given pair of codes, even if they come from different institutions.

Dataset preparation for deep-learning

The dataset used in this work consists of 2 sets of vocabularies. One is the source set $S = \{(t_i^S, d_i^S)\}_{i=1}^{N}$, whose codes t_i^S are paired with corresponding description sentences d_i^S. Our goal is to map these source terms to target codes in the target set $T = \{(t_i^T, d_i^T)\}_{i=1}^{M}$, where t_i^T and d_i^T are a target term and corresponding description, respectively. This target set is the standard OMOP-CDM set. Our approach to computing vocabulary mapping uses sentence embedding. Similar to word embedding, general embedding is a deep-learning technique that aims to represent an object of interest (eg, word or sentence) as a Euclidean vector. Euclidean embedding is performed in a manner that reflects the semantics of target objects. Objects with similar meanings will be assigned to vectors that are close to each other, while dissimilar objects will be far apart. The merit of embedding-based representation is that the (dis)similarity between any pair of objects can be quantified based on distances and inner products between the corresponding vectors. This semantic representation feature is exploited in our proposed system.

Formally, we are given a set of N training data points $D = \{(t_1^S, d_1^S, t_1^T, d_1^T, y_1), \ldots, (t_N^S, d_N^S, t_N^T, d_N^T, y_N) | (t_i^S, d_i^S) \in S, (t_i^T, d_i^T) \in T, y_i \in \{0,1\}\}$, where the tuple $(t_i^S, d_i^S, t_i^T, d_i^T, y_i)$ is the i^{th} data point containing a source term and description, target term and description, and binary label, respectively. The binary label indicates whether the given pair is a positive sample (1) or negative sample (0). The sample being positive or negative depends on whether the sample is a correct source-to-target mapping or not. The necessity of negative samples will be discussed further in the Training section. We present the following 3 strategies for collecting negative samples.

The random sampling scheme: We randomly match an arbitrary description to a given source concept name. For each description sentence, we randomly sample a fixed number of concept names from the dataset and pair them with a label of zero. This process is illustrated in Figure 2.

The false-positive (FP) sampling scheme: The second approach is based on a similarity measure. This approach performs negative sampling based on the mapping results of the deep neural network model trained using the random sampling scheme. Specifically, based on the similarity scores of mapping results from the pretrained deep neural network model, FP concept names in the top-100 list for each description that were not target names are extracted as negative samples for training. The rationale behind this approach is that seemingly similar samples (ie, FP samples) with high similarity should be separated by a greater distance.

The hierarchical sampling scheme: The final approach uses hierarchical information regarding concepts. Negative samples are selected according to the hierarchical relationships between target concepts. The standard SNOMED-CT codes form a hierarchy through which we move downward and upward to select negative samples. Specifically, in the hierarchy of target concepts, descendants and ancestors that are at most 2 steps away from the given code are considered. For each positive sample, we select an average of 3 descendants and 4.5 ancestors of the target code to form a negative sample set. A visual summary of the negative sample collection process is presented in Figure 1.

Once the negative samples are gathered, we combine them with the positive samples to form a full dataset. The split configuration used in our experiments is 7:1:2 for training, validation, and test, respectively.

Learning architecture

The main concept of our approach is to construct a mapping that pairs codes with the highest description similarities. For a given code t_i^S in the source set, we assign t_j^T as its mapping result such that d_i^S is the most similar to d_j^T in terms of their semantics. To compute the semantic similarity between 2 natural language sentences, we use the embedding of each descriptive sentence, which is fed into a neural network to derive final results.

Specifically, we propose using the probability of matching as a measure of similarity between codes. This probability is computed from the embeddings of the descriptive sentences of the corresponding codes by inputting them into a neural network. Our neural network architecture takes the form of a dual-input network, as shown in Figure 2. Our main architecture takes the descriptive sentences for 2 concept names we wish to compare as inputs. One of the concepts comes from the institution code list and the other comes from the standard SNOMED-CT code list. These 2 sentences are processed separately until they are merged to compute the probability of a match.

The descriptive sentence for each code is represented as a sequence of words, each of which is represented as an embedded vector. We use fastText word embeddings, which yielded similar performance compared to other embeddings. The sequence of vectors is fed into a recurrent neural network (RNN)[26,27] used to encode a sentence. We use a pretrained InferSent model as an embedding RNN. This RNN produces a sentence embedding vector of length d, which we use to compute similarities with respect to other codes.

As a side note, there are more domain-specific embedding models like BioBERT. However, the reason we choose Infersent and fastText as an embedding model is that we think the published InferSent model with fastText might be more suitable for our task in the sense that it implements the generic natural language inference problem. Moreover, using a model that has already been pretrained on a similar task seemed more cost-efficient. Given 2 sentence encodings u and v, the process for computing their actual similarity (ie, probability of a match) is defined as follows:

First, we create a vector of length 4d that is a concatenation of u, v, and the following 2 vectors:

- $|u - v|$: absolute element-wise difference between u and v
- $u * v$: element-wise product of u and v

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocab030/6275415 by University of Cambridge user on 01 June 2021

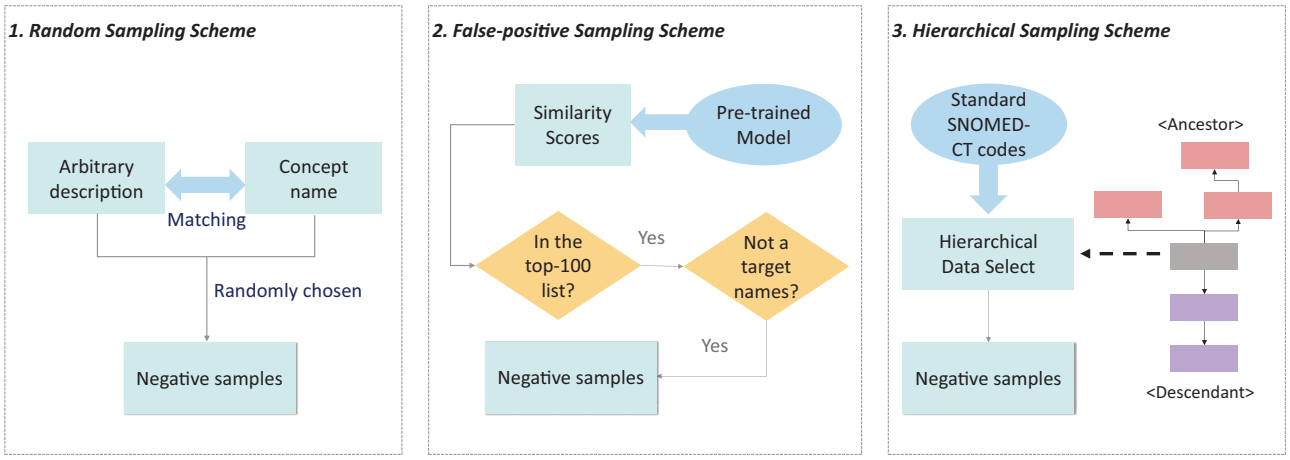

Figure 1. Overview of 3 sampling schemes for collecting negative training samples.

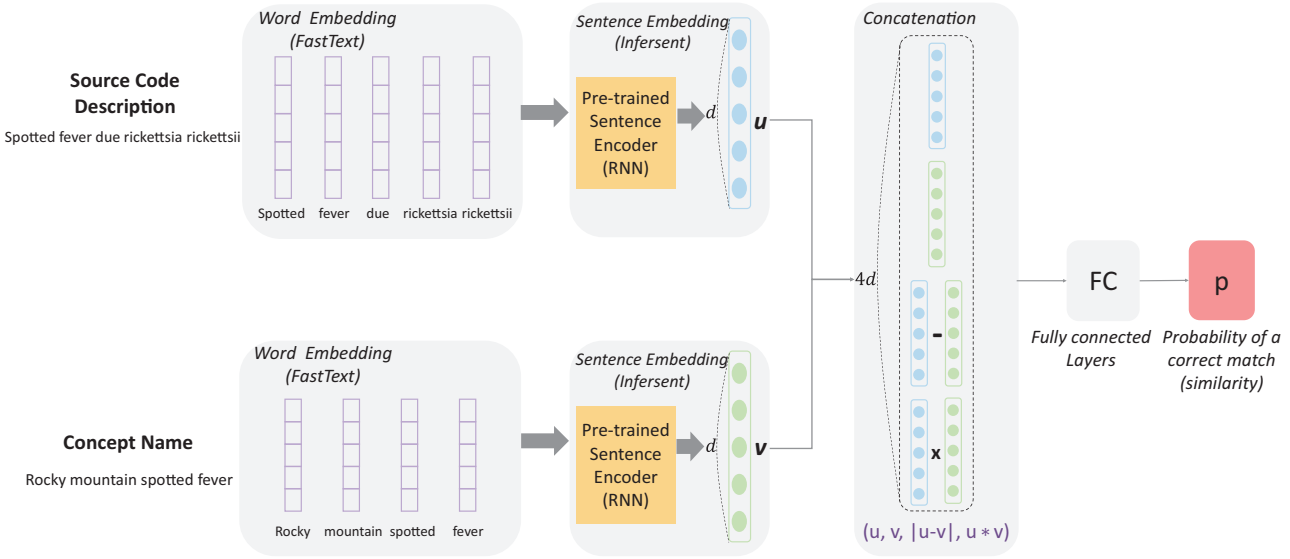

Figure 2. Schematic overview of the proposed architecture. The output is the semantic similarity or probability of matching between the 2 given descriptions.

The motivation for this composition is to incorporate as much information regarding the "closeness" between the 2 vectors as possible. The former indicates how far apart each component in the 2 vectors is, while the latter indicates how aligned each component is (eg, components with the same signs will become positive when multiplied).

Second, we pass the concatenated vector through 2 fully connected layers to produce a scalar output. Given a concatenated vector of length 4d, the final output is computed as follows:

$$\Pr(\text{u matches v}) = p_{uv} = \sigma(V^\top f(W^\top x)).$$

This formula represents 2 fully connected layers, where $x \in \mathbb{R}^{4d}$ is the input of the fully connected layers computed from u and v, $W \in \mathbb{R}^{4d \times 512}$ and $V \in \mathbb{R}^{512}$ are the layer weights, $f(x) = \max(0, x)$ is the rectified linear unit (ReLU) activation function, and $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function. We convert the 4d-dimensional vector into a 512-dimensional representation W prior to applying the ReLU function. We multiply this intermediate result by V to acquire a scalar that is converted into a probability by the sigmoid function. The values of the weight matrices W and V are tuned using the training procedure outlined below such that the probability of a correct match is maximized. Based on its text-based comparison mechanism, we call our system Text-based OMOP Knowledge Integration (TOKI, which is also a pronunciation of the Korean word for rabbit).

Training

Recall that we have a small set of human-labeled mappings that can serve as positive samples (ie, samples that are known to have correct mappings). The training of the proposed network proceeds by maximizing the probabilities of the positive samples. However, training our model not only involves maximizing positive probabilities but minimizing negative probabilities to avoid training a degenerate model. The technique used for constructing the negative sample set had been discussed in the Dataset preparation section.

Our goal is to maximize the probability of matching for positive pairs and minimize the probabilities for all negative pairs. This goal can be achieved by using a binary cross-entropy loss function. Let $h($

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocab030/6275415 by University of Cambridge user on 01 June 2021

$x, y; \Theta$) denote the neural network architecture depicted in Figure 1, which accepts the source and target definitions x and y as inputs. The only trainable part of this network is the final sequence of fully connected layers parameterized by Θ. Our goal is to minimize the following binary cross-entropy loss with respect to Θ:

$$\min_{\Theta} \frac{1}{N} \sum_{i}^{N} y_i \log\left(h\left(d_i^S, d_i^T; \Theta\right)\right) + (1 - y_i) \log\left(1 - h\left(d_i^S, d_i^T; \Theta\right)\right).$$

Because this optimization problem does not yield a closed-form solution, we use stochastic gradient descent over multiple epochs to find a local minimum. Training epochs are implemented using 3 types of negative samples over a 3-stage fine-tuning procedure.

Test procedure

To measure the performance of the proposed system, we use the top-k accuracy metric. Top-k accuracy is computed by counting the number of test instances that contain true mappings among the top k most similar items. For each vocabulary t that we wish to standardize (ie, find a correct mapping to an OMOP-CDM term), we compute the set of all possible pairwise similarities between t and all OMOP-CDM terms $C_t = \{p_{tv} | (v, d_v) \in T\}$. We then sort C_t in descending order and examine the first k terms with the highest similarity values. If this set of k terms contains at least 1 true mapping, we count this instance as a correct instance. The final top-k accuracy is computed as follows:

$$\text{top} - \text{k} = \frac{1}{N} \sum_{i=1}^{N} \delta_i(k),$$

where $\delta_i(k)$ is an indicator variable that takes a value of 1 if the i^{th}instance has a true match in the top-k set and a value of 0 otherwise. We use a value of 100 for k based on opinions from domain experts with extensive experience performing manual mapping. The motivation behind this choice is that once the mapping algorithm suggests a set of 100 candidates that is likely to contain an answer, manual verification becomes much easier. This setting represents a semiautomated scenario in which the system proposes a small set of candidates from which a human operator will select an answer. We experimented on a variety of cases ranging from the fully automated case of k = 1 to final value of k = 100.

RESULTS

We applied our algorithm to a real-world dataset gathered from 5 tertiary hospitals in South Korea. The total number of codes in the source domain is 83 113. This dataset contains manually constructed and verified ground-truth mappings to the target OMOP concepts. Training the final TOKI architecture required 70% of these mappings, while another 10% are used to fine-tune learning hyperparameters, such as the learning rate. The remaining data are used to derive the results reported in Table 2. As mentioned in the dataset description, we incorporate random, FP, and hierarchical negative samples. For the random negative samples, the included numbers are different multiples of the positive samples (N = 1, 50, 100, 150, 200, and 250). The results of these settings are listed in columns 2–7 in Table 2. The notation "*N*POS" indicates that we used N times more negative random samples than positive samples to train the model.

To compare in a sense of both semiautomatic and automatic system, we report precision@Top 1 and recall@Top 1 as well as precision@Top 100 and recall@Top 100. Under both the semiautomatic and automatic schemes, TOKI outperforms Usagi as the number of samples increases. FP samples are used in a 2-step training process. In the first step, TOKI is pretrained using 150 random negative samples. We then select 50 FP negative samples based on the outcomes of the pretrained model. We collect 50 additional random negative samples and combine them with the 50 FP samples to form 100 new negative samples for fine-tuning the pretrained model. The results of this FP-based fine-tuning are presented in column 8 in Table 2.

For hierarchical negative samples, we make use of both the ancestors and descendants of the target concepts. Similar to FP-based fine-tuning, we also use hierarchical samples for fine-tuning the model from the previous operation (which was also fine-tuned from a random-sample model). However, this final fine-tuning step involves a much smaller set of random negative samples to balance the scarcity of hierarchical samples. Instead of the range N = 1 to

Table 2. Results for top-k accuracies, precision @Top 1/Top 100, and recall @Top 1/Top 100, and F1-score @ Top 1/Top 100

Model	Usagi	1POS	50POS	100POS	150POS	200POS	250POS	150POS + FP	150POS + FP + HIER
Top 1	0.4210	0.3300	0.4520	0.4640	0.4760	0.4780	0.4800	0.5100	0.5740
Top 5	0.5800	0.3820	0.5980	0.6400	0.6680	0.6660	0.6700	0.7000	0.7780
Top 10	0.6220	0.4260	0.6760	0.7140	0.7480	0.7460	0.7440	0.7720	0.8220
Top 20	0.6390	0.4620	0.7260	0.7780	0.7900	0.7940	0.7920	0.8260	0.8520
Top 50	0.6590	0.5200	0.8120	0.8480	0.8580	0.8620	0.8560	0.8640	0.8900
Top 100	0.6590	0.5760	0.8480	0.8760	0.8860	0.8900	0.8880	0.8920	0.9100
Precision @Top 1	0.4210	0.3300	0.4520	0.4640	0.4760	0.4780	0.4800	0.5100	0.5740
Recall @Top 1	0.3763	0.3190	0.4087	0.4153	0.4253	0.4263	0.4293	0.4493	0.4903
F1-score @Top 1	0.3973	0.3244	0.4292	0.4383	0.4492	0.4506	0.4532	0.4777	0.5288
Precision @Top 100	0.0089	0.0058	0.0099	0.0103	0.0105	0.0106	0.0106	0.0107	0.0113
Recall @Top 100	0.5663	0.5273	0.7830	0.8090	0.8220	0.8263	0.8247	0.8323	0.8713
F1-score @Top 100	0.0175	0.0114	0.0195	0.0203	0.0207	0.0209	0.0209	0.0211	0.0223

Note: The first column represents the results from Usagi, and the remaining columns were calculated for our proposed model

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocab030/6275415 by University of Cambridge user on 01 June 2021

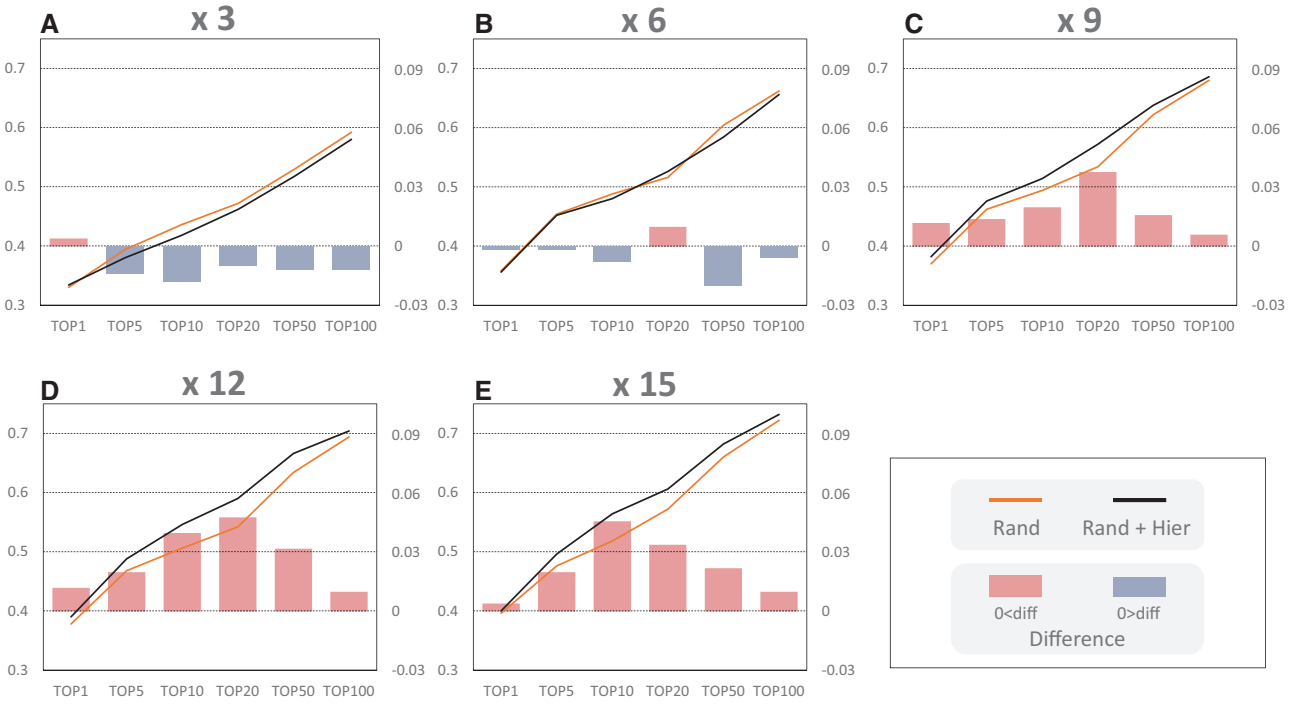

Figure 3. Accuracies achieved by adding different numbers of hierarchical negative samples. The x*M* notation above each graph indicates that M times more negative samples have been used with respect to the positive samples.

250, we use N = 3, 6, 9, 12, and 15 for the multiples of the number of positive samples. The results of this final fine-tuning step are presented in the last column of Table 2.

In addition to this main experiment, we also test the effects of using hierarchy-based samples in the absence of FP negative samples. In this experiment, there are 2 considerations to be made when selecting the N negative samples. First, we can use only random negative samples for the N choices ("Rand"). Second, N/2 samples have to be random and the remaining N/2 have to be hierarchical ("Hier+Rand"). The goal of this experiment is to examine the effects of hierarchical negative samples when excluding FP samples. The top-k results for each value of N are presented in Figure 3.

Next, we present 2 cases of actual matching: 1 successful and 1 mismatch. We present these cases to demonstrate the advantages and disadvantages of the proposed model. The first match is the case where TOKI finds a successful match and the second is where Usagi finds a successful match.

DISCUSSION

The results of the main experiment (Table 2) demonstrate that TOKI outperforms Usagi by orders of magnitude. Notably, the top-k accuracy steadily increases as we incorporate additional negative samples. This trend is expected because using a small number of negative samples will result in the model being unable to reject false matches. This intuition is also supported by the extra increase in accuracy after performing FP- and hierarchy-based fine-tuning in rows 8 (150POS+FP) and 9 (150POS+FP+HIER) in Table 2.

The experimental results demonstrate that using multiple negative samples is necessary to outperform simple count-based similarity models. In general, it is necessary to use many negative samples to train a matching system properly because positive samples only account for a very small portion of the data. Whether the negative samples are random or FP also seems to have a meaningful influence on the final outcome. As a reference, analysis of the types of negative samples has been conducted in other embedding-related studies.[17,28]

Random negative samples are the easiest and cheapest samples to collect, meaning they are the most frequently used. Therefore, we tested various multiples of random negative samples relative to positive samples to find the combination producing the best results. When we use more than N = 50 times more random negative samples, the TOKI model consistently outperforms Usagi. This advantage is most pronounced when random negative samples are 150 times more prevalent than positive samples.

Using FP samples in addition to random negative samples yields an additional improvement in performance. For example, the results in Table 2 reveal that for a given number of negative samples, including some FP samples is more beneficial than having purely random negative samples. This trend is expected because FP sampling explicitly attempts to correct the mistakes made by the model by providing feedback in the training loop. In contrast, random negative sampling blindly incorporates randomly selected samples as negatives.

The results above show that the matching accuracy monotonically increases with the number of candidates (ie, top-k). Although our method significantly outperforms Usagi for every value of k, the results for the top-1 accuracy are relatively poor. In an ideal case, it would be desirable to have the matching candidate be the top-1 candidate, which is often perceived as "the answer." However, top-1 matching is difficult to achieve, so it is more realistic to expect a practical compromise. In practice, automated matching recommendations are verified by human operators who typically report that having the correct candidate in the top-100 list is acceptable in terms of ease of operation (personal communication with EvidNet, July 2019). This suggests that although the top-1 accuracies seem low, the high top-100 accuracies demonstrate the practical value of the proposed system.

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocab030/6275415 by University of Cambridge user on 01 June 2021

Table 3. Two sample matches. The first row is a successful match for TOKI and the second row is a successful match for Usagi

Source	Target	
	TOKI	Usagi
Spinal osteochondrosis lumbar region	Spinal stenosis lumbar region	Spinal stenosis of lumbar region
	Disorder fetal abdominal region	Juvenile osteochondritis
	Disorder lumbar spine	Familial Scheuermann disease
	Disorder spinal region	Adult osteochondrosis of spine
	Spinal stenosis lumbar region disorder	Multiple congenital exostosis
	Disorder fetal abdominal region	Juvenile osteochondrosis of spine
	Spinal stenosis cervical region	Juvenile osteochondrosis of acetabulum
	Spinal stenosis thoracic region	Calvé's vertebral osteochondrosis
	Disorder orbital region	Synovial osteochondromatosis
	Disorder hip region	Regional osteoporosis
Ventricular septal defect as current complication following acute myocardial infarction	Ventricular aneurysm current complication following acute myocardial infraction	**Atrial septal defect due to and following acute myocardial infarction**
	Past history clinical finding	Ventricular aneurysm due to and following acute myocardial infarction
	Ventricular aneurysm due to following acute myocardial infraction disorder	Hemopericardium due to and following acute myocardial infarction
	Degenerative vascular disorders ear	Arrhythmia due to and following acute myocardial infarction
	Ventricular septal aneurysm disorder	Post-infarction ventricular septal defect
	Ventricular ectopic complex	Pulmonary embolism due to and following acute myocardial infarction
	Ventricular interpolated complexes disorder	Rupture of papillary muscle as current complication following acute myocardial infarction
	Finding related pregnancy	Rupture of chordae tendineae due to and following acute myocardial infarction
	Ventricular myocardial noncompaction cardiomyopathy disorder	Rupture of cardiac wall without hemopericardium as current complication following acute myocardial infarction
	Ventricular dysphonia disorder	Thrombosis of atrium, auricular appendage, and ventricle due to and following acute myocardial infarction

Qualitatively, our model can better match concepts with few or no overlapping words. Table 3 contains a successful match by TOKI and a failed match by Usagi. TOKI's results show a good balance between literal word matching (containing the same words) and semantic matching (different words with related meanings). In contrast, Usagi largely focuses on particular keywords such as "osteochondrosis," likely based on its large IDF. TOKI's result is not surprising because its matching is performed on a semantic level using learned embeddings. Leveraging contextual information seems to aid in prediction, as has been demonstrated in other deep-learning-based models.[20,22,29]

The results in the second case, where TOKI fails and Usagi succeeds, suggest that these 2 models are not necessarily in a competitive relationship. Instead, TOKI's semantic approach can be applied in conjunction with Usagi to form an ensemble model that considers both context and semantics. Usagi's straightforward emphasis on matching words and phrases yields a higher retrieval rate. TOKI sometimes searches excessively based on semantics and drifts toward unrelated phrases. We believe that both models have the potential to complement each other by counteracting the negative aspects of the other. Therefore, exploiting the merits of both models to improve on each weakness can be satisfied when we make an ensemble model, which will be considered in future work. In addition to an ensemble model, using various negative sampling techniques based on domain-specific knowledge would be needed. Moreover, adopting a hierarchical penalty to the mapping model using soft label instead of hard label in calculation of loss function and other word-embedding models are possible future topics to pursue.

CONCLUSION

The goal of this study is to develop a trainable system that can match the medical terminologies of individual institutions to those of the OMOP standard. To map disparate sets of codes to standard codes, one must consider the semantics of each term. Such semantics are given in the form of code descriptions, which we convert into latent representations that can be matched to the most relevant description in the standard set. By performing 1-time training of this semantic representation extraction module, our system can provide high-accuracy mapping for any given institution's code set using their descriptive sentences. We verified that the presence of negative training samples affects the outcomes significantly. The manner in which such samples are collected is also of significance and can be applied to similar trainable text-matching systems.

FUNDING

This work was supported by the Ministry of Science and ICT (MSIT), Korea, under the ICT Consilience Creative Program (IITP-2020-2011-1-00783), which is supervised by the Institute for Information & Communications Technology Planning & Evaluation

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocab030/6275415 by University of Cambridge user on 01 June 2021

(IITP). It was also supported by a Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korean government (MOTIE) (20202020800030, Development of Smart Hybrid Envelope Systems for Zero-Energy Buildings through Holistic Performance Testing and Evaluation Methods and Field Verifications).

AUTHOR CONTRIBUTIONS

HJK initiated the study and oversaw the technical details regarding OMOP-CDM. BK and HYK provided the main architectural design and prepared the manuscript draft. JSY implemented the main system and conducted most experiments, while receiving assistance from SJJ. YL assisted in minor implementation tasks as well as data preparation. All authors contributed to editing and revising the manuscript. The manuscript has been read and approved by all named authors and there are no other persons who satisfy the criteria for authorship that are not listed. The order of authors listed in the manuscript has been approved by all authors.

ACKNOWLEDGMENTS

The authors thank Aelan Park (RN, CMD, PhD) and Soo-Yeon Cho (RN, MPH) of EvidNet for their technical support for terminology mapping and testing with Usagi.

DATA AVAILABILITY STATEMENT

The data underlying this article were provided by EvidNet under license.

COMPETING INTERESTS STATEMENT

The authors have no competing interests to declare.

REFERENCES

1. McMurry AJ, Murphy SN, MacFadden D, *et al.* SHRINE: enabling nationally scalable multi-site disease Studies. *PLoS ONE* 2013; 8 (3): e55811.
2. Burrows EK, Razzaghi H, Utidjian L, *et al.* Standardizing clinical diagnoses: evaluating alternate terminology selection. *AMIA Summits Transl Sci Proc* 2020; 2020: 71–9.
3. Wermuth C, Verplaetse H. (2019). Medical terminology in the Western world: current situation. In: Alsulaiman, A, Allaithy, A, eds. *Handbook of Terminology Volume* 2. Amsterdam, The Netherlands: John Benjamins Publishing; 2019: 84–108.
4. Awaysheh A, Wilcke J, Elvinger F, *et al.* A review of medical terminology standards and structured reporting. *J Vet Diagn Invest* 2018; 30 (1): 17–25.
5. Luna D, Otero C, Gambarte ML, Frangella J. Terminology Services: Standard Terminologies to Control Medical Vocabulary. "Words are Not What they Say but What they Mean." *eHealth: Making Health Care Smarter*, 2018. doi: 10.5772/intechopen.75781.
6. Klann JG, Phillips LC, Herrick C, *et al.* Web services for data warehouses: OMOP and PCORnet on i2b2. *J Am Med Inform Associ* 2018; 25 (10): 1331–8.
7. Tabano DC, Cole E, Holve E, *et al.* Distributed data networks that support public health information needs. *J Public Health Manag Pract* 2017; 23 (6): 674–83.
8. Garza M, Del Fiol G, Tenenbaum J, *et al.* Evaluating common data models for use with a longitudinal community registry. *Journal of Biomedical Informatics* 2016; 64: 333–41.
9. Klann JG, Joss MA, Embree K, *et al.* Data model harmonization for the All of Us Research Program: Transforming i2b2 data into the OMOP common data model. *PloS One* 2019; 14 (2): e0212463.
10. Candore G, Hedenmalm K, Slattery J, *et al.* Can we rely on results from IQVIA medical research data UK converted to the observational medical outcome partnership common data model? A validation study based on prescribing codeine in children. *Clin Pharmacol Ther* 2020; 107 (4): 915–25.
11. Ji H, Kim S, Yi S, *et al.* Converting clinical document architecture documents to the common data model for incorporating health information exchange data in observational health studies: CDA to CDM. *J Biomed Inform* 2020; 107: 103459.
12. https://www.ohdsi.org/data-standardization/the-common-data-model/ Accessed July, 2020
13. Yoon D, Ahn EK, Park M, *et al.* Conversion and data quality assessment of electronic health record data at a Korean tertiary teaching hospital to a common data model for distributed network research. *Healthc Inform Res* 2016; 22 (1): 54–8.
14. Lamer A, Depas N, Doutreligne M, *et al.* Transforming French electronic health records into the Observational Medical Outcome Partnership's common data model: a feasibility study. *Appl Clin Inform* 2020; 11 (01): 13– 22.
15. Lynch KE, Deppen SA, DuVall SL, *et al.* Incrementally transforming electronic medical records into the Observational Medical Outcomes Partnership common data model: a multidimensional quality assurance approach. *Appl Clin Inform* 2019; 10 (05): 794–803.
16. https://www.ohdsi.org/software-tools/Accessed April, 2020
17. Mikolov T. Distributed representations of words and phrases and their compositionality. In proceedings of *Advances in Neural Information Processing Systems* 2013.
18. Joulin A, Grave E, Bojanowski P, *et al.* Bag of tricks for efficient text classification. In proceedings of the *Conference of the European Chapter of the Association for Computational Linguistics*; April 3–7, 2017; Valencia, Spain.
19. Peters M, Neumann M, Iyyer M, *et al.* Deep contextualized word representations. In proceedings of the *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; June 1–6, 2018; New Orleans.
20. Kiros R, Zhu Y, Salakhutdinov R, *et al.* Skip-thought vectors. In proceedings of *Neural Information Processing Systems*; December 7–10, 2015; Montreal, Canada.
21. Devlin JCChang M-W, Lee K, *et al.* BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; June 2–7, 2019; Minneapolis.
22. Conneau A, Kiela D, Schwenk H, *et al.* Supervised learning of universal sentence representations from natural language inference data. In Proceedings of the *Conference on Empirical Methods in Natural Language Processing;* September 7–11, 2017; Copenhagen, Denmark.
23. https://ohdsi.github.io/TheBookOfOhdsi/CommonDataModel.html#fn20Accessed July, 2020
24. https://athena.ohdsi.org/search-terms/startAccessed July, 2020
25. Kim S, Park A, Cho S-Y, *et al.* Evaluation of a semi-automated code mapping and management system, OHDSI symposium 2019. https://www.ohdsi.org/2019-us-symposium-showcase-13/
26. Williams RJ, Hinton GE, Rumelhart DE. Learning representations by back-propagating errors. *Nature* 1986; 323 (6088): 533–6.
27. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997; 9 (8): 1735–80.
28. Pennington J, Socher R, Manning CD. GloVe: Global vectors for word representation. In proceedings of the Conference on Empirical Methods in Natural Language Processing; 2014; Doha, Qatar.
29. Bengio Y, Ducharme R, Vincent P, Janvin C. A neural probabilistic language model. *J Mach Learn Res* 2003; 3: 1137–55.

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocab030/6275415 by University of Cambridge user on 01 June 2021