Hierarchical Multi-Task Word Embedding Learning for Synonym Prediction

Hongliang Fei, Shulong Tan, Ping Li
Cognitive Computing Lab (CCL), Baidu Research USA
{hongliangfei,shulongtan,liping11}@baidu.com

ABSTRACT

Automatic synonym recognition is of great importance for entity-centric text mining and interpretation. Due to the high language use variability in real-life, manual construction of semantic resources to cover all synonyms is prohibitively expensive and may also result in limited coverage. Although there are public knowledge bases, they only have limited coverage for languages other than English. In this paper, we focus on medical domain and propose an automatic way to accelerate the process of medical synonymy resource development for Chinese, including both formal entities from healthcare professionals and noisy descriptions from end-users. Motivated by the success of distributed word representations, we design a multi-task model with hierarchical task relationship to learn more representative entity/term embeddings and apply them to synonym prediction. In our model, we extend the classical skip-gram word embedding model by introducing an auxiliary task "neighboring word semantic type prediction" and hierarchically organize them based on the task complexity. Meanwhile, we incorporate existing medical term-term synonymous knowledge into our word embedding learning framework. We demonstrate that the embeddings trained from our proposed multi-task model yield significant improvement for entity semantic relatedness evaluation, neighboring word semantic type prediction and synonym prediction compared with baselines. Furthermore, we create a large medical text corpus in Chinese that includes annotations for entities, descriptions and synonymous pairs for future research in this direction.

ACM Reference Format:
Hongliang Fei, Shulong Tan, Ping Li. 2019. Hierarchical Multi-Task Word Embedding Learning for Synonym Prediction. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330914

1 INTRODUCTION

Synonym prediction has become an important task for entity-centric text mining and interpretation [28, 32]. With the aid of synonym prediction, informal mentions of an entity can be normalized into its standard form, which will significantly reduce the communication gap between end-users and downstream applications. Such examples include but not limited to question & answering [9], information retrieval [39] and medical diagnosis [20].

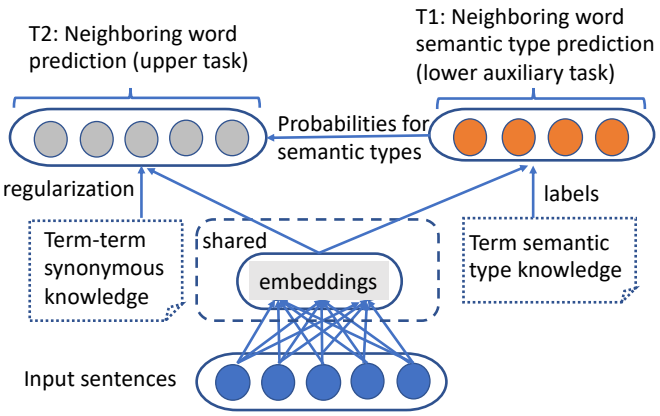

Figure 1: Overview of the proposed hierarchical multi-task word embedding model. Semantic type knowledge of terms and term-term synonymous knowledge are both utilized at different layers in different ways.

From the resource perspective, the major difficulty in synonym prediction is high variability of language use [5] but low coverage of knowledge base (KB) [13], especially for languages other than English. For example in Chinese medical domain, the concept "食欲不振 (translation: loss of appetite)" has more than 20 synonyms[1], but most of them are missing in existing medical KB [1] because they are mainly used by patients without much medical knowledge. Although one can leverage state-of-the-art named entity recognition tools [21] to discover more entities, there is little work to construct labeled data with annotations for informal descriptions and synonyms for training.

From the modeling perspective, the key question for synonym prediction is how to learn more effective representations for entities and descriptions. With a high-quality semantic representation, any off-the-shelf classifiers can be applied to predict synonymous relation. Recently, word and entity embedding methods [16, 17, 23, 24], which learn distributed vector representation of words from a large corpus, have been prevalent in data mining communities. For English, a few word or character embedding based synonym prediction methods have been proposed [11, 15, 32].

For example, Wang et al. [32] integrated the semantic type knowledge of terms into word embedding learning and combined the

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330914

[1] "Loss of appetite" synonym examples in Chinese. {胃口差, 吃不下东西, 胃口不好, 东西吃不下, 饭吃不下, 不爱吃饭}(translation: no desire for food); {食欲低下, 食欲下降, 食欲不太好, 缺乏食欲, 食欲差, 食欲减退}(translation: decreased appetite)

learned embeddings with other syntactic features for synonym prediction. Although the model leveraged semantic type knowledge, it ignored the rich relational information among entities. Hasan et al. [11] used character embeddings as term features and cast the synonym prediction task as a neural machine translation problem, in which a target synonym was generated by a bi-directional RNN given the source term. A limitation of such a complex model is that it requires a large amount of labeled data (synonym pairs) from UMLS [18], but there are no such public resources in Chinese.

We hypothesize that incorporating semantic knowledge will learn more representative word embeddings, and hence lead to a more accurate synonym prediction. Herein semantic knowledge includes both entity's semantic type information and semantic relatedness information among entities. Motivated by Søgaard and Goldberg [29] and Hashimoto et al. [12] who showed the power of predicting two increasingly complex but related tasks at successive layers, we propose a hierarchical multi-task word embedding model as shown in Figure 1. At the lower layer, we introduce an auxiliary task that predicts neighboring word semantic types given the target word. At the upper layer, we extend the skip-gram model [23] to incorporate existing synonymy knowledge among entities and the lower level task's outcomes. Such a hierarchical structure allows us to not only utilize entities' semantic types and semantic relation but mutually enhance the two tasks in the training stage.

Though our methodology is generic, our paper is particularly motivated by the medical domain in Chinese, which has very high language use variability, rich semantic knowledge but low knowledge base coverage. Our model can also be applied to any other domains where external knowledge is tremendous, and language use variability is very high. Experimental results show that our model can learn more representative embeddings and generate better accuracy for entity semantic relatedness evaluation, neighboring word semantic type prediction and synonym prediction.

To summarize, our contributions of this paper are as follows:

- We present a hierarchical multi-task word embedding model that fully leverages medical domain knowledge. By introducing an auxiliary task of neighboring word semantic type prediction, we provide more information to the word embedding objective. We have designed an alternative optimization algorithm for the model and achieved better performance compared with existing methods.
- We collect a large Chinese medical corpus (around 10M sentences) from professional medical textbooks, wikis, and forums with the purpose of identifying more informal medical descriptions and synonymous pairs. From the corpus, we identified and annotated 151K medical entities and descriptions covering 18 categories with 185K high-quality synonymous pairs. The annotated dataset will help other researchers to discover more noisy and informal medical descriptions. To our best knowledge, this corpus is the first Chinese benchmark with both entities annotated and synonyms labeled.
- We apply our model to 400M pairs of medical terms and obtained around 1M synonym candidates unseen in any previous medical resources. The newly discovered synonyms can enrich existing knowledge bases in Chinese. Furthermore, we perform a thoughtful error analysis of our approach, which sheds light on future work in this direction.

2 RELATED WORK

The importance of synonym extraction has been well recognized in the biomedical and clinical research community [14, 22]. Early approaches are typically non-neural based methods. Conventional techniques include the use of lexical and syntactic features [10], bilingual alignment-based methods [31] and random walk on the term graph [25]. For simplicity, we do not cover them in details.

For neural based methods, word embedding techniques have been widely adopted for synonym prediction [11, 15, 32]. Recently, there is a growing interest to enhance word embedding by incorporating domain semantic knowledge. The enhancement either changes the objective of word embedding by adding relation regularization during the training phase [34, 35] or takes a post-processing step on the trained word vectors to accommodate the semantic relation [7]. For either case, only the term-term relation is used, but semantic type information of terms is ignored. In Table 1, we summarize the characteristics of related methods and ours.

Table 1: Characteristics for each method. ST means semantic type, SR means synonymous relation, PP means post-processing and MT means multi-task. "x" indicates a method has a certain property.

Method	ST	SR	PP	MT
Our method	x	x		x
Yu and Dredze [37]		x		
Wang et al. [32]	x			x
Faruqui et al. [7]		x	x	

Among all the embedding based methods, the most similar works to ours are Wang et al. [32] and Yu and Dredze [37]. In Wang et al. [32], semantic types of terms were incorporated as extra-label information in the word embedding training process. Such a semi-supervised method enables word embedding model to consider the "desired type" when generating the "desired word", which is a special case of multi-task learning with two tasks on the same level. In our model, we leverage not only the semantic type of terms but also the term-term synonymous relation. In Yu and Dredze [37], a relation constrained word embedding model is presented, in which the term-term synonymous relation is utilized by maximizing the log-likelihood of all synonymous pairs. Although we also use the synonymous relation among terms, there are two major differences between our work and theirs. The first difference is that our word embedding model is a hierarchical multi-task learning framework with an auxiliary task of predicting semantic types of terms. The second difference is that we employ a different regularization strategy to enforce the synonymous pairs to share similar embeddings instead of maximizing their log-likelihood.

Another line of related research is multi-task learning (MTL), which learns multiple related tasks simultaneously to improve generalization performance. MTL has been applied to a wide range of applications including healthcare informatics [8], speech recognition [30] and natural language processing [12, 29]. In particular, our work is motivated by Søgaard and Goldberg [29] and Hashimoto et al. [12], which demonstrate the strength of positioning different tasks at different layers by considering the linguistic hierarchies. For example, Hashimoto et al. [12] built a many-task model

in which tasks were incrementally growing according to their complexity (e.g. POS tagging → entity chunking → dependency parsing). The key difference between their work and ours is that our hierarchical multi-task model not only solves the two predictive tasks but also leverages two types of semantic knowledge.

3 METHODOLOGY

In this section, we first describe the original skip-gram model [23], then explain our hierarchical multi-task word embedding model. Before introducing them in details, we outline the notation of this paper in Table 2.

Table 2: Notation table.

Notation	Meaning
n	number of words in the vocabulary
m	number of semantic types
d	word embedding dimension
x_i	ith input word
$\boldsymbol{V}$	Word embedding matrix of size $n \times d$
$\boldsymbol{U}$	Parameters for semantic type prediction layer with size $m \times d$
$\boldsymbol{W}$	Parameters for word prediction layer with size $n \times (d + m)$
$\mathbb{C}$	The set of all semantic types with size m
$\mathbb{X}$	The vocabulary of size n
c_i	The ith semantic type in $\mathbb{C}$
$\boldsymbol{A}_i$	The ith row of matrix $\boldsymbol{A}$
$\sigma(.)$	Sigmoid function: $\sigma(x) = 1/(1 + \exp(-x))$
$\|\boldsymbol{A}\|_F$	F-norm of matrix $\boldsymbol{A}$

3.1 Skip-gram Embedding Model

The goal of skip-gram model [23] is to optimize word embeddings that are effective to predict neighboring words given the target word. More formally, it minimizes the following objective function:

$$L_{sg} = \frac{1}{n} \sum_{t=1}^{n} \sum_{-c \le j \le c, j \ne 0} -\log p(x_{j+t}|x_t) \tag{1}$$

where x_t is the target word, c is the context window size. The probability $p(x_O|x_I)$ is calculated using the softmax function:

$$p(x_O|x_I) = \frac{\exp\left(\boldsymbol{V}_{x_I}^T \boldsymbol{W}_{x_O}\right)}{\sum_{x' \in \mathbb{X}} \exp\left(\boldsymbol{V}_{x_I}^T \boldsymbol{W}_{x'}\right)} \tag{2}$$

Skip-gram model alternatively updates $\boldsymbol{V}$ and $\boldsymbol{W}$ and outputs the hidden representation $\boldsymbol{V}$ as final word embeddings, where the ith row of $\boldsymbol{V}_i$ is the word x_i's embedding vector.

3.2 Hierarchical Multi-task Word Embedding

We extend the skip-gram model [23] by introducing an auxiliary task of neighboring word semantic type prediction. The key insight is that knowing the semantic types of neighboring words will benefit neighboring word prediction. For example in the medical domain, symptom terms are often surrounded by other symptom terms or disease terms. In this paper, we assume each input sentence has been segmented into a sequence of words/phrases, and medical entities are annotated. The advantage of the preprocessing is that we can directly train embeddings for medical entities and descriptions like other ordinary words.

It is obvious that there are three ways to organize the two tasks:

- Two tasks are organized in parallel and share the common hidden embedding layer, which amounts to ordinary multi-task learning with shared hidden layers in neural networks.
- Two tasks are hierarchically organized, wherein the neighboring word prediction task is positioned lower, and the neighboring word semantic type prediction task is placed upper.
- The hierarchical structure proposed in our paper as shown in Figure 1. It enables the neighboring word prediction to leverage the outcomes of the neighboring word semantic type prediction and the shared word embeddings.

We choose the last structure for two reasons. First, predicting neighboring words is more complex than predicting their semantic types. The cardinality of the set of all possible neighboring words equals to the vocabulary size, which is much larger than that of semantic types. Second, from a linguistic perspective, knowing the possible semantic types will help neighboring word prediction task to focus on the words belonging to those types.

In Figure 2, we show our proposed model framework. During training, the target word and its neighboring words are first fed into the input layer to perform embedding lookup. Meanwhile, the neighboring words are queried against an external medical knowledge base (KB) to determine their corresponding semantic types. The target word embedding together with its neighboring words' types will be the task T1's training data. Note that only the neighboring words with valid semantic types (e.g. the words in red color) will be fed into T1. The task T2's input are the concatenation of the probability distribution of semantic types from T1 and the target word's embedding together with the neighboring words. Besides, the target word's synonyms are fed into T2 as external knowledge.

3.2.1 T1: Neighboring Word Semantic Type Prediction. Given the input word and its embedding vector, this task is to predict its neighboring words' possible semantic types within a context window. For example in Figure 2, the input term "runny_nose" is surrounded by two symptom terms and one disease term with context window size 7. This model is expected to assign higher probabilities to the semantic types of symptom and disease.

We cast the task T1 as a multi-label classification problem, in which the number of labels equals the number of semantic types. Although there are many complicated multi-label classification algorithms [38, 40], we use binary relevance [27], which amounts to independently training one binary classifier for each label. The reason for choosing binary relevance is that it is not only computationally effective but can induce optimal models when the loss function is a macro-averaged measure [19]. In particular, we minimize the following regularized weighted cross entropy objective:

$$L_{T1} = -\frac{1}{n} \sum_{t=1}^{n} \sum_{j=1}^{|\mathbb{C}|} \{w_j y_{tj} \log p(y_{tj}|x_t) + (1 - y_{tj}) \times (1 - \log p(y_{tj}|x_t))\} + \lambda \|\boldsymbol{V} - \boldsymbol{V_0}\|_F^2 \tag{3}$$

where $y_{tj} = 1$ when the input word x_t has a neighboring word with type c_j in the training set, and $y_{tj} = 0$ otherwise. w_j is

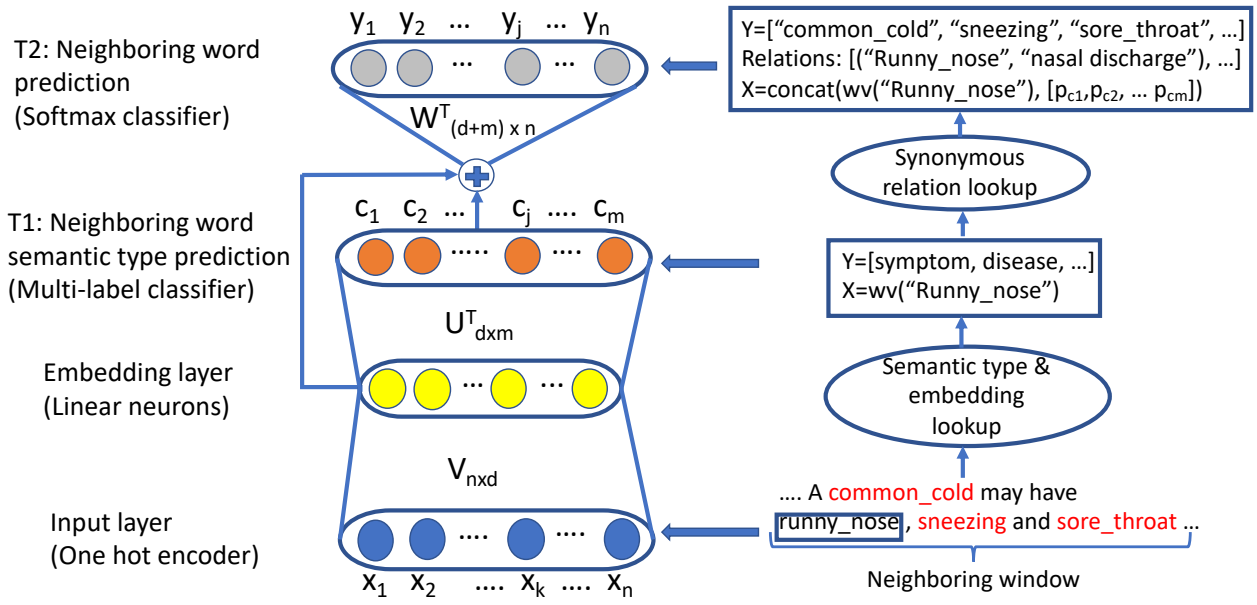

Figure 2: The hierarchical multi-task word embedding model architecture. Note that both tasks have access to the embedding layer. The task T1 will use semantic type information, and T2 will use existing synonymous relation knowledge. Right side shows an example of how data is fed into the model. Here "Runny_nose" is the target word, and the context window size is 7.

the positive sample weight for class c_j that can be set as the inverse of positive/negative samples ratio. The conditional probability $p(y_{tj}|x_t)$ is defined as $p(y_{tj}|x_t) = \sigma(U_j^T V_{x_t})$. V_0 is the word embedding after training the task T2 at the previous epoch and λ is a regularization parameter. For simplicity, we omit the bias term in Eq. (3), although we include bias terms in our implementation.

The term $\|V - V_0\|_F^2$ in Eq. (3) is called a successive regularization term [12], which penalizes the deviation of the current embedding parameters and those learned from the other task. Such a regularization term helps prevent parameters from varying too much when switching tasks hence can stabilize the training process.

Note that we assume each medical term has only one semantic type, which is valid in the medical domain as it is rare for a medical entity to have two or more semantic types. For example, "aspirin" is a drug entity and it cannot have semantic types of disease. When extending the task T1 to other areas where a term may have multiple semantic types, one can utilize context-aware models such as contextual dependency networks [26]. Since our focus is in the medical domain, we do not discuss general cases in this paper.

3.2.2 T2: Neighboring Word Prediction. Our approach to neighboring word prediction task is similar to recent works on improving word embeddings using prior knowledge (e.g., paraphrase, synonyms) [7, 34, 37]. Those methods modify the original word embedding objective with a regularization term that encourages semantically related words to share similar word embeddings. The difference is that we approach the problem in a multi-task setting, whereas their methods are all single task learning.

In particular, we augment the input to task T2 with outcomes from the semantic type prediction task T1 and also utilize the successive regularization term to encourage "a certain level" of consensus between parameters of the two tasks.

Let $\boldsymbol{\theta} = [V, U]$ denote the model parameters associated with task T1. The objective of task T2 to be minimized is as follows:

$$L_{T2} = \frac{1}{n}\sum_{t=1}^{n}\{\sum_{-c\le j\le c, j\neq 0} -\log p(x_{j+t}|x_t, f_{T1}(x_t)) + \lambda_1 \sum_{x\in\mathcal{S}(x_t)} \|V_x - V_{x_t}\|_2^2\} + \lambda\|\boldsymbol{\theta} - \boldsymbol{\theta_0}\|_F^2 \quad (4)$$

where $\mathcal{S}(x_t)$ is the synonym/paraphrase set of x_t from the external knowledge, $f_{T1}(x_t)$ is the neighborhood semantic type prediction result of x_t, λ_1 is the regularization parameter for synonym priors, $\boldsymbol{\theta_0}$ are the task T1's parameters after training T1 at the current training epoch. The second regularization term is enforcing the word embedding similarity between the target word x_t and its known synonyms, while the third term is the successive regularization term to stabilize the training process.

Let $\boldsymbol{\phi}_I = [V_{x_I}, f_{T1}(x_I)]$. The conditional probability of observing word x_O given x_I and $f_{T1}(x_I)$ is defined as:

$$p(x_O|x_I, f_{T1}(x_I)) = \frac{\exp(\boldsymbol{\phi}_I^T \boldsymbol{W}_{x_O})}{\sum_{x'\in\mathbb{X}} \exp(\boldsymbol{\phi}_I^T \boldsymbol{W}_{x'})} \quad (5)$$

One problem in Eq. (5) is the high complexity to compute the normalization factor as it involves summation over all words in the vocabulary. To address the problem, we use *negative sampling* (NEG) [24] to convert the original one-vs-all multi-class objective into a binary classification objective. With negative sampling, the negative logarithm of Eq. (5) can be rewritten as:

$$J(x_O, x_I) = -\log\sigma(\boldsymbol{\phi}_I^T \boldsymbol{W}_{x_O}) - \sum_{x_k\in\mathbb{P}_{neg}(x_O)} \log\sigma(-\boldsymbol{\phi}_I^T \boldsymbol{W}_{x_k}) \quad (6)$$

where $\mathbb{P}_{neg}(x_j)$ is the set of negative samples for x_j. Plugging Eq. (6) into Eq. (4), we have a simplified objective of the task T2:

$$L_{T2} = \frac{1}{n}\sum_{t=1}^{n}\{\sum_{-c\le j\le c, j\neq 0} J(x_{j+t}, x_t) + \lambda_1 \sum_{x\in S(x_t)} \|V_x - V_{x_t}\|_2^2\} + \lambda\|\theta - \theta_0\|_F^2 \quad (7)$$

3.3 Training

The model is trained over a large text corpus with an external knowledge base support, in which semantic types and term-term synonymous relation are available. We use mini-batch stochastic gradient descent (SGD) with a schedule to decay the learning rate by half after certain global steps.

During each epoch, the optimization iterates from the lower task to the higher task as described in Figure 2. In particular, we first minimize L_{T1} in Eq. (3) to update $\boldsymbol{V}$ and $\boldsymbol{U}$ over the full training set, then pass the optimized $\boldsymbol{V}$ and $\boldsymbol{U}$ to upper. By minimizing L_{T2} in Eq. (7) over the full training set, we update $\boldsymbol{W}$, $\boldsymbol{V}$ and $\boldsymbol{U}$ and pass $\boldsymbol{V}$ to the lower level task at the beginning of the next epoch. We repeat the above process until reaching the predefined number of epochs and output $\boldsymbol{V}$ as the final word embeddings.

The reason of choosing $\boldsymbol{V}$ instead of $\boldsymbol{W}$ as the final embedding is that $\boldsymbol{V}$ is shared between two tasks and will be updated for both tasks, while $\boldsymbol{W}$ is only updated when training the neighboring word prediction task. Therefore $\boldsymbol{V}$ carries more semantic information regarding the entity types. We also tried to use $\boldsymbol{W}$ as the final embedding, but it did not show any improvement.

3.4 Application to Synonym Prediction

Although synonymous relation is utilized during word embedding learning, the available synonyms have a limited coverage. To extract more synonymous pairs, one can either train more complex models, or use a simple model (e.g. linear SVM [6]) but include more informative features. In this paper, we adopt the latter one since our focus is to learn more representative embeddings.

To capture more useful information for synonym extraction, we follow Wang et al. [32] to construct features for pairs of terms, including expanded embeddings and lexical matching features. Furthermore, we add two more features, 1) cosine similarity between a pair of word vectors, 2) Jaro–Winkler similarity [33] between two terms at string level, which achieved the best performance in entity name-matching tasks [2].

4 EXPERIMENT

We have collected a Chinese medical corpus from 9 textbooks, medical wiki A+ hospital [1] and medical QA forums[2]. In total, the corpus contains around 10M sentences. We follow the UMLS entity type taxonomy[3], but merge low-level semantic types to its upper-level concepts (e.g., detailed drug components to drugs) and rename several semantic types to make crowd-sourcing validation easier. In total, there are 18 types: symptom, disease, drug, food, therapy, surgery, prevention, medical device, department, cause, body part, external injury, biochemistry, examination and medical index, physiology, psychology, medical regulation, microbiology.

4.1 Medical Entity and Synonym Collection

From the medical wiki website, we collect 70K professional entities. To identify informal medical terms, we use crowd-sourcing to collect 30K informal medical descriptions. We train the well-known named entity recognition model "CNN-BiLSTM-CRF" [21] implemented by [36] on 200K sentences, in which the initial 100K medical terms were annotated under the "BIOES" scheme [3]. Since there are 18 semantic types, we have 73 NER tags in total. We obtain 90.7% F1 score on another 20k labeled test sentences.

With the trained NER model, we find 58K new entities and phrases from the large medical corpus with 10M sentences. After crowd-sourcing validation, we keep 51K and combined them with the initial 100K to build a medical dictionary of 151K entities belonging to 18 semantic types. In Figure 3, we provide the summary statistics of our medical dictionary.

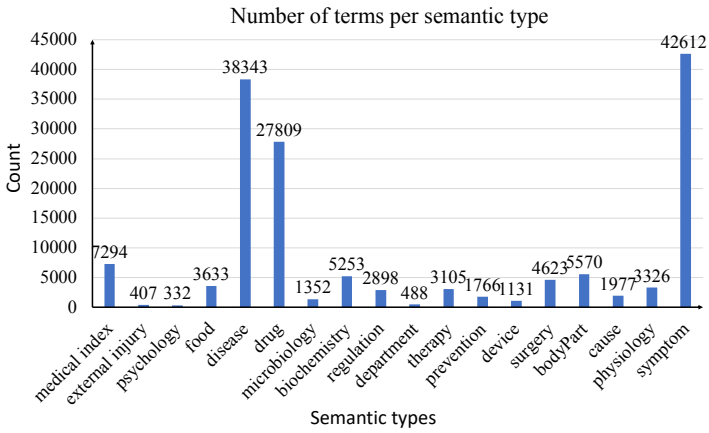

Figure 3: Summary statistics of the medical dictionary

To collect initial synonymous pairs for word embedding and synonym model training, we first utilize rules (e.g., A a.k.a. B) and regular expression on the wiki text to identify the synonyms following certain patterns. Since rules have limited coverage, we also use unsupervised methods to collect more synonyms. In particular, we train the embeddings of 151K entities on the text corpus using word2vec model [24], then apply density-based spatial clustering (dbscan) [4] to find compact clusters. The reason of using dbscan is that it does not require to specify the number of clusters and can find clusters with any shapes. We set a smaller distance threshold $\epsilon = 2$ for two samples to be considered in the same neighborhood and $minPoint = 3$ for the minimum number of samples for one sample to be a core point. A smaller distance threshold will help reduce false positives and achieve a higher precision.

After obtaining synonymous clusters (30K), we use crowd-sourcing to guarantee that each cluster contains only high-quality synonyms. We divide all annotators into several groups and let two groups of people label the same batch of data. For disagreements, a third group make a choice. The average annotator agreement is 0.80 ± 0.09. In total, we obtain 185K synonymous pairs.

[2] www.xywy.com
[3] https://metamap.nlm.nih.gov/SemanticTypesAndGroups.shtml

4.2 Experimental Data Preprocessing

To prepare the training data for word embedding, we use jieba[4], a well-known Chinese word segmentation tool, customized with our medical dictionary, to cut sentences in the medical text corpus into sequences of words and entities/phrases. Such a procedure will ensure word embedding algorithms to treat medical terms as a whole and learn their representations. By filtering out rare words that appear less than five times and removing punctuation characters, we obtain 411,256 unique words and phrases. We split the segmented corpus into 3 parts: 80% training, 10% validation and 10% testing for neighboring semantic type prediction experiment.

Among all the synonymous pairs, we first sample 25k pairs containing 3586 unique entities for entity semantic relatedness evaluation in subsection 4.4. The rest 160k pairs are further split by 80%, 10%, 10% for training, validation and testing for synonym prediction experiment in subsection 4.6. The 80% split of synonymous pairs is also used as our term-term knowledge for word embedding training. In Table 3, we summarize characteristics of the datasets.

Table 3: Characteristics of the datasets. "-" indicates no splitting. Semantic relatedness eval pairs data is sampled from the overall 185K synonymous pairs and not used in word embedding training.

Dataset	Total	Train	Dev	Test
Medical corpus	10M	8M	1M	1M
Medical dictionary	151K	-	-	-
Synonymous pairs	160K	128K	16K	16K
Semantic relatedness eval pairs	25K	-	-	-

4.3 Experiment Setup

We set word vector length d to 200, initial learning rate to 0.001, neighboring window size to 5, mini-batch size to 400, number of epochs to 20, and number of negative samples to 20.

To find the best hyper-parameters for our model, we run a parameter search on a combination of the successive regularization parameter $\lambda = \{0.1, 0.5, 1, 2, 8\}$ and synonym prior regularization $\lambda_1 = \{0.01, 0.05, 0.1, 0.5, 1\}$, and computed the average pair-wise cosine similarity on the synonymous pair dev data. We find that the parameters did not significantly change the performance (1.0% at most). We set $\lambda = 0.5$ and $\lambda_1 = 0.05$ that yields the best result.

To have a fair comparison, we train each method (ours and competing methods) on the 80% split of corpus data (8M sentences) and the term-term synonymous relation data. Also, each method shares the same setup for the word vector length, the mini-batch size, the number of negative samples, and the number of epochs.

We compare our method with several state-of-the-art word embedding approaches.

Mikolov et al. [23]. We use the gensim package[5] to train a skip-gram model with the same configuration as our method.

Yu and Dredze [37]. We train word vectors using their joint model training code[6] using the same settings as above. The 80% split of "golden" synonyms are used as the paraphrase DB input. C is set by default.

Wang et al. [32]. The method only utilized semantic type information during training, and there are no other hyper-parameters to tune. Since there is no open source implementation for this method, we carefully implement it in Tensorflow for comparison.

Faruqui et al. [7]. The "retrofitting" algorithm in this paper is a post-processing method to make word vectors more similar for synonym pairs. We use the source code[7] and apply it to the word vectors from Mikolov et al. [23]. The semantic graph is constructed from the 80% split (128K) of the"gold" synonyms.

w2vRegSTL. A single-task version of our method, which only keeps the neighboring word prediction task at the upper level.

4.4 Entity Semantic Relatedness Evaluation

This evaluation is to test the quality of learned word/phrase representations in a direct way without training any supervised models. Among all metrics, the cosine similarity between a pair of word vectors is commonly used to quantify how similar two words are. However, since each method learns word embeddings in different embedding space, directly comparing cosine values across all methods is inappropriate. Instead, we compare the precision of its top k ranked entities based on the cosine similarity for each entity.

In particular, given an entity for each method, we first compute the cosine similarity between the input entity and the rest entities in the semantic relatedness evaluation pairs data, then sort them in descending order. Since the true synonyms within the evaluation data for the input are known, we can compute precision@$k = tp/k$, where tp is the number of true synonyms in the input entity's top k ranked entities.

Table 4: Average precision@k for $k = 1, 3, 5$. Bold font indicates the best performance. Cells marked with * designates that our method significantly outperforms (p < 0.05) all the compared methods.

Model	Precision@k		
	$k = 1$	$k = 3$	$k = 5$
Our method	**0.654***	**0.603***	**0.571***
Mikolov et al. [23]	0.538	0.507	0.467
Yu and Dredze [37]	0.619	0.572	0.547
Wang et al. [32]	0.579	0.532	0.487
Faruqui et al. [7]	0.588	0.558	0.513
w2vRegSTL	0.622	0.579	0.545

In Table 4, we report the average precision@k for the unique 3586 entities in the semantic relatedness evaluation data. From the table, we observe that the original skip-gram model performs the worst, which is reasonable as it does not utilize any semantic knowledge. Although Wang et al. [32] leverages the semantic type information, its performance is slightly better than Mikolov et al. [23], but still inferior to those methods using synonymous relations. Faruqui et al. [7], the post-processing method after embedding training, performs worse than Yu and Dredze [37] and w2vRegSTL, which leverages the same term-term synonymy relations but uses them during training.

We suspect one possible reason is that Faruqui et al. [7] only utilized the training synonym pairs, which may have little overlaps

[4]https://github.com/fxsjy/jieba
[5]https://radimrehurek.com/gensim/models/word2vec.html
[6]https://github.com/Gorov/JointRCM
[7]https://github.com/mfaruqui/retrofitting

with the test synonym data. In that case, even though Faruqui et al. [7] enforces smoothness of synonym pairs in the training data, it makes no difference for the terms in the leave-out data. To the contrary, Yu and Dredze [37], w2vRegSTL and our proposed method iteratively learn embeddings not only from synonymous relation but also from texts, which will allow the similarity to propagate between two isolated terms via some intermediate terms.

Finally, our proposed multi-task method outperforms all baselines with statistical significance under t-test (p<0.05), which demonstrates the benefit of utilizing both semantic type and synonymous knowledge and hierarchically arranging the two tasks.

4.5 Semantic Type Prediction Evaluation

Since we introduce the auxiliary task "neighboring word semantic type prediction" to skip-gram model, it is worthwhile to conduct a study on the effectiveness of our framework on this task.

For comparison, we fix all the word vectors from competing methods and train the same binary relevance model as described in Eq. (3) except for replacing the successive regularization term with an L_2 norm penalty on parameters $\boldsymbol{U}$.

Table 5: AUC scores for "neighboring word semantic type prediction" task. MacroAUC means macro-averaged AUC and MicroAUC means Micro-averaged AUC.

Method	MacroAUC	MicroAUC
Our method	**79.92%***	**80.03%***
Mikolov et al. [23]	76.06%	76.90%
Yu and Dredze [37]	76.21%	76.71%
Wang et al. [32]	63.09%	65.27%
Faruqui et al. [7]	76.27%	76.84%
w2vRegSTL	77.58%	77.88%

Table 5 shows the micro-average and macro-average AUC scores for the 18 semantic types. We observe that Wang et al. [32] performs much worse than any other methods. The rest baselines behave similarly to each other. Again our method achieves the best result of around 80% AUC, which demonstrates the importance of jointly learning related tasks.

4.6 Synonym Prediction Evaluation

Since our focus of this paper is to learn better medical entity/description representations for synonym prediction, we utilize a linear classifier [6] rather than complicated ones to demonstrate the utility of learned embeddings. As discussed in subsection 3.4, we extract both expanded embedding features and syntax similarity features, leading to 1406 features in total for each pair of terms. To have a fair comparison, we use the same feature construction procedures and run the same classifier for all competing methods.

To construct negative samples, we randomly sampled 1.4M pairs of medical terms from our dictionary. Such a procedure may introduce false negatives, but the chance is low given a relatively large number of terms. We split the 1.4M negative samples by 80%, 10%, 10% as well and combine with the true synonymous pairs shown in Table 3 to make training, validation and testing data. We use the L_2 regularized logistic regression in LIBLINEAR package [6] and tune the hyper-parameter in $\{0.01, 0.1, 0.5, 1, 4, 16, 64, 256\}$ over the validation data on F_1 metric. The positive sample weight is set to 8.75 according to the inverse of positive and negative samples ratio in the training data (1.4M/160K).

Table 6: Precision, recall and F_1 score of all methods on the test data. Cells marked with * designates that our method significantly outperforms (p < 0.05) all baselines.

Method	Precision	Recall	F_1 score
Our method	**82.34%***	**93.07%***	**87.37%***
Mikolov et al. [23]	75.39%	85.53%	80.14%
Yu and Dredze [37]	80.23%	92.03%	85.73%
Wang et al. [32]	81.36%	85.86%	83.55%
Faruqui et al. [7]	80.09%	88.08%	83.89%
w2vRegSTL	79.87%	91.48%	85.28%

Table 6 shows precision, recall and F_1 score on the test data. We first observe that all methods have a relatively higher recall than precision, which is resulted from the positive class weight. Actually in real-world applications, one can tune different sample weights and prediction threshold to trade off between precision and recall. The fact that Wang et al. [32] obtains the second best precision but has a lower recall reveals that the term-term synonymous relation is more important than the semantic type knowledge for synonym prediction task. Our method leverages both semantic type information and term-term synonymy knowledge and achieves the best performance on all the three metrics with statistical significance under proportion test (p-value<0.05).

To understand how much contribution each component of our full model makes to the synonym prediction, we did an ablation study and reported the F_1 score when each component was disabled, as illustrated in Table 7.

Table 7: Ablation study on synonym prediction task: F_1 score when each component was removed from the full model, while the rest components are unchanged.

Model	F_1 score
Our full model	**87.37%**
w/o the auxiliary task	85.28%
w/o the synonymous regularization	86.23%
w/o the pairwise lexical matching features	86.93%

Removing the auxiliary task of neighboring word semantic type prediction and synonymous regularization significantly deteriorates the model performance by 2.09% and 1.14% respectively (significant statistical t-test with $p < 0.01$). Such a huge performance drop demonstrates the importance of introducing the auxiliary task and incorporating synonymy knowledge. Furthermore, disabling the pairwise lexical matching features will slightly reduce the prediction performance, which is consistent with Wang et al. [32].

4.7 Application to unlabeled symptom pairs

In medical domain, the high language use variability usually originates from symptom terms since users have different expressions to describe the same concept. To generate more synonymous pairs, we apply our trained synonym model to 400M symptom pairs that

Table 8: Example: 8 symptom terms and their top 5 synonymous terms with translation. Bold font indicates false positives.

Query term	Top 5 most synonymous terms and their probability scores
头皮屑好多 (lots of dandruff)	头皮多 (lots of dandruff),0.999: 起皮屑 (having dandruff),0.995: 头屑多 (lots of dandruff),0.949: **头皮好痒 (Very Itchy scalp)**,0.895: 头皮藓 (Scalp Ringworm),0.840
例假特别少 (very low menstrual flow)	例假少 (low menstrual flow),1.000: 经量很小 (less bleeding during periods),0.990: 例假很少 (very low menstrual flow),0.956: **尿量特别少 (very low urine flow)**,0.766: **尿特别少 (low urine flow)**,0.659
小肚右侧疼 (abdominal pain on right side)	小肚疼 (abdominal pain),1.000: 小肚痛 (abdominal pain),0.998: 小肚有点疼 (little abdominal pain),0.961: **小腿肚疼 (calf pain)**,0.958
胸部痛 (chest pain)	胸部都会胀痛 (chest swelling and pain),1.000: 胸部有点痛 (a little pain in the chest),1.000: **胸部胀 (chest swelling)**,1.000: 胸部疼 (chest pain),0.999: 胸部刺痛 (stabbing pain in the chest),0.999
口干 (dry mouth)	口干苦 (dry and bitter mouth),1.000: 口很干 (very dry mouth),1.000: 口干燥 (dry mouth),0.998: 口易干 (mouth gets dry easily),0.995: 口会干 (mouth gets dry),0.995
老是尿尿 (very frequent urination)	小便有点勤 (a little frequent urination),1.000: 小便很频 (very frequent urination),1.000: 尿比较频 (very frequent urination),1.000: 小便过多 (too much urination),1.000: **尿老是黄 (yellow urine)**,0.988
面色无华 (pale-faced)	面色发黄 (pale-faced),1.000: 面色晦暗 (pale-faced),1.000: 面色淡白 (pale-faced),1.000: 面色暗黄 (pale faced),1.000: **面色红润 (complexions rosy)**,1.000
肚脐周围疼痛 (pain around the navel area)	肚脐上痛 (navel pain),1.000: **肚脐右边疼 (navel pain on right side)**,1.000: **肚脐上边疼 (navel pain on upper side)**,1.000: 肚脐痛 (navel pain),1.000: **肚脐左下隐痛 (novel pain on lower-left side)**,1.000

never occur in our collected synonym data and obtained around 1M new synonymous pairs. Although there is no way to thoroughly validate the accuracy of the newly generated pairs, we perform an manual validation by following a similar procedure for entity semantic relatedness evaluation.

First, we randomly select 200 symptoms as queries and collect each symptom's top 5 most synonymous terms based on the probability score, then manually labeled each term whether it is a true synonym to the query entity and compute the metric of precision@k. Finally, we calculate the average precision@k and report the result in Figure 4. Compared with Table 6, the precision is decreased. The possible reason is that we only sample the symptom pairs that are very similar to each other from the unlabeled data, which is more challenging than random sampling regardless of semantic types. Nevertheless, our model still achieves 73% precision up to k=3.

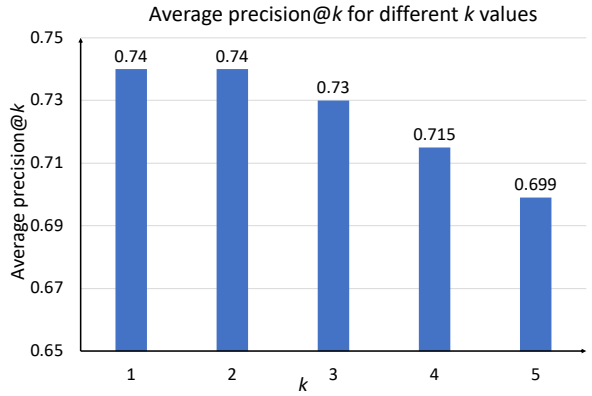

Figure 4: The average precision@k for 200 randomly sampled symptom terms

4.8 Error analysis

We also carefully analyze a few typical errors found during our manual validation to guide future research. In Table 8, we list 8 symptom terms and their top 5 most synonymous terms, wherein the false positives are highlighted in bold font.

From the table, we observe that although our method can successfully link a few semantically equal but lexically different descriptions, for example, 例假特别少 (lots of dandruff) v.s. 经量很小 (less bleeding during periods) and 老是尿尿 (very frequent urination) v.s. 小便很频 (very frequent urination), there are several limitations to prevent the proposed method working flawlessly:

- Fail to distinguish the body parts that share very similar lexical patterns. For example, 小肚 (abdomen) and 小腿肚 (calf) have only one character difference, but they refer to different body parts. To reduce such errors, a subject matching modular could be developed to detect if two phrases share the same subject before applying synonym predictive model.
- Fail to differentiate synonymy from semantic relatedness. Although word embedding has captured a certain level of semantic relatedness, it is not always reliable to tell the difference between synonymy and semantic relatedness, especially for pairs of terms that are both lexically and semantically related. For example, 胸部痛 (chest pain) and 胸部胀 (chest swelling) often co-occur with each other, and their embeddings are quite similar to each other, hence are predicted to be synonymous. To minimize such errors, more high-quality negative samples covering such cases are needed to guide classifiers to learn the subtle difference.
- Fail to sense the position difference. For example, 肚脐周围疼痛 (pain around the navel area) and 肚脐右边疼 (navel pain on the right side) belong to the same concept of 肚脐疼 (novel pain) but have different locations. Strictly speaking, they are not synonymous pairs. To alleviate such problems, more such negative samples are needed, and adverbs of location features can be extracted to learn the position difference.

5 CONCLUSION

We propose a hierarchical multi-task word embedding model to learn more representative medical entity embeddings and apply

them to medical synonym prediction. By introducing an auxiliary task of neighboring word semantic type prediction and fully utilizing medical domain knowledge, our model yields more semantically meaningful word representations as evaluated by entity semantic relatedness, neighboring word semantic type prediction and synonym prediction. Although our model is developed for the medical domain, it can be applied to other domains where external knowledge is tremendous, and language use variability is very high. Furthermore, we create a large medical text corpus in Chinese that includes annotations for entities, descriptions and synonymous pairs for future research in this direction.

Future work includes applying the model to medical domains in other languages and exploring an end-to-end framework to integrate word representation learning and synonym prediction.

REFERENCES

[1] A+ 医学百科. 2017. A+ 医学百科, 在线医学百科全书. http://www.a-hospital.com/ Accessed: 2017-09-30.

[2] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A Comparison of String Distance Metrics for Name-matching Tasks. In *Proceedings of the 2003 International Conference on Information Integration on the Web (IIWB)*. Acapulco, Mexico, 73–78.

[3] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* 12 (2011), 2493–2537.

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 226–231.

[5] Antoinette M. Fage-Butler and Matilde Nisbeth Jensen. 2016. Medical terminology in online patient–patient communication: evidence of high health literacy? *Health Expectations* 19, 3 (2016), 643–653.

[6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9 (2008), 1871–1874.

[7] Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. 1606–1615.

[8] Hongliang Fei and Jun Huan. 2011. Structured Feature Selection and Task Relationship Inference for Multi-task Learning. In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM)*. Vancouver, BC, Canada, 171–180.

[9] D. A. Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development* 56, 3 (May 2012), 235–249.

[10] Masato Hagiwara. 2008. A Supervised Learning Approach to Automatic Synonym Identification Based on Distributional Features. In *ACL Student Research Workshop*. 1–6.

[11] Sadid A. Hasan, Bo Liu, Joey Liu, Ashequl Qadir, Kathy Lee, Vivek Datla, Aaditya Prakash, and Oladimeji Farri. 2016. Neural Clinical Paraphrase Generation with Attention. In *Proceedings of the Clinical Natural Language Processing Workshop*. 42–53.

[12] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1923–1933.

[13] Aron Henriksson, Hans Moen, Maria Skeppstedt, Vidas Daudaravičius, and Martin Duneld. 2014. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of Biomedical Semantics* 5, 1 (2014), 6.

[14] Aron Henriksson, Hans Moen, Maria Skeppstedt, Vidas Daudaravicius, and Martin Duneld. 2014. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of Biomedical Semantics* 5 (2014), 6.

[15] Artuur Leeuwenberg, Mihaela Vela, Jon Dehdari, and Josef van Genabith. 2016. A Minimally Supervised Approach for Synonym Extraction with Word Embeddings. *Methods of Information in Medicine* 105 (2016), 111–142.

[16] Dingcheng Li, Siamak Zamani Dadaneh, Jingyuan Zhang, and Ping Li. 2019. Integration of Knowledge Graph Embedding into Topic Modeling with Hierarchical Dirichlet Process. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. Minneapolis, USA.

[17] Dingcheng Li, Jingyuan Zhang, and Ping Li. 2019. TMSA: A Mutual Learning Model for Topic Discovery and Word Embedding. In *Proceedings of the SIAM Data Mining Conference (SDM)*. Calgary, Alberta, Canada.

[18] D. Linderg, B. Humphreys, and A. McCray. 1993. The Unified Medical Language System. *The Prague Bulletin of Mathematical Linguistics* 4 (1993), 281–91.

[19] Oscar Luaces, Jorge Díez, José Barranquero, Juan José del Coz, and Antonio Bahamonde. 2012. Binary relevance efficacy for multilabel classification. *Progress in AI* 1, 4 (2012), 303–313.

[20] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis Prediction in Healthcare via Attention-based Bidirectional Recurrent Neural Networks. In *Proceedings of the 23rd ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. Halifax, Canada, 1903–1911.

[21] Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 1.

[22] S M Meystre, G K Savova, K C Kipper-Schuler, and J F Hurdle. 2008. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearbook of medical informatics* (2008), 128–44.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR)*. 1–12.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems (NIPS)*. 3111–3119.

[25] Einat Minkov and William W. Cohen. 2008. Learning Graph Walk Based Similarity Measures for Parsed Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 907–916.

[26] Thien Nguyen and Ralph Grishman. 2016. Modeling Skip-Grams for Event Detection with Convolutional Neural Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 886–891.

[27] James Petterson and Tibério S. Caetano. 2010. Reverse Multi-Label Learning. In *Advances in Neural Information Processing Systems (NIPS)*. 1912–1920.

[28] Meng Qu, Xiang Ren, and Jiawei Han. 2017. Automatic Synonym Discovery with Knowledge Bases. In *Proceedings of the 23rd ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. Halifax, Canada, 997–1005.

[29] Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 2.

[30] Abhinav Thanda and Shankar M. Venkatesan. 2017. Multi-task Learning Of Deep Neural Networks For Audio Visual Automatic Speech Recognition. *CoRR* abs/1701.02477 (2017). arXiv:1701.02477

[31] Lonneke van der Plas and Jörg Tiedemann. 2006. Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity. In *Proceedings of the COLING 2006 Main Conference Poster Sessions*. Association for Computational Linguistics, Sydney, Australia, 866–873.

[32] Chang Wang, Liangliang Cao, and Bowen Zhou. 2015. Medical Synonym Extraction with Concept Space Models. In *Proceedings of the 24th International Conference on Artificial Intelligence (Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI))*. 989–995.

[33] W. E. Winkler. 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods*. Wiley, 354–359.

[34] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. RC-NET: A General Framework for Incorporating Knowledge into Word Representations. In *Proceedings of the 23rd International Conference on Conference on Information and Knowledge Management (CIKM)*. 1219–1228.

[35] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*. Berlin, Germany, 250–259.

[36] Jie Yang and Yue Zhang. 2018. NCRF++: An Open-source Neural Sequence Labeling Toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. Austin, TX, USA, 74–79.

[37] Mo Yu and Mark Dredze. 2014. Improving Lexical Embeddings with Semantic Knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. 545–550.

[38] Wang Zhan and Min-Ling Zhang. 2017. Inductive Semi-supervised Multi-Label Learning with Co-Training. In *Proceedings of the 23rd ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. Halifax, Canada, 1305–1314.

[39] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S. Yu. 2018. On the Generative Discovery of Structured Medical Knowledge. In *Proceedings of the 24th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. London, UK, 2720–2728.

[40] Wen-Ji Zhou, Yang Yu, and Min-Ling Zhang. 2017. Binary Linear Compression for Multi-label Classification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 3546–3552.