Mining User Intentions from Medical Queries: A Neural Network Based Heterogeneous Jointly Modeling Approach

Chenwei Zhang[†*] Wei Fan[‡] Nan Du[‡] Philip S. Yu[†§]
[†]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[‡]Baidu Research Big Data Lab, Sunnyvale, CA, USA
[§]Institute for Data Science, Tsinghua University, Beijing, China
[†]{czhang99,psyu}@uic.edu, [‡]{fanwei03,nandu}@baidu.com

ABSTRACT

Text queries are naturally encoded with user intentions. An intention detection task tries to model and discover intentions that user encoded in text queries. Unlike conventional text classification tasks where the label of text is highly correlated with some topic-specific words, words from different topic categories tend to co-occur in medical related queries. Besides the existence of topic-specific words and word order, word correlations and the way words organized into sentence are crucial to intention detection tasks.

In this paper, we present a neural network based jointly modeling approach to model and capture user intentions in medical related text queries. Regardless of the exact words in text queries, the proposed method incorporates two types of heterogeneous information: 1) **pairwise word feature correlations** and 2) **part-of-speech tags of a sentence** to jointly model user intentions. Variable-length text queries are first inherently taken care of by a fixed-size pairwise feature correlation matrix. Moreover, convolution and pooling operations are applied on feature correlations to fully exploit latent semantic structure within the query. Sentence rephrasing is finally introduced as a data augmentation technique to improve model generalization ability during model training. Experiment results on real world medical queries have shown that the proposed method is able to extract complete and precise user intentions from text queries.

Keywords

Text Query; Intention Detection; Neural Network; Jointly Modeling

1. INTRODUCTION

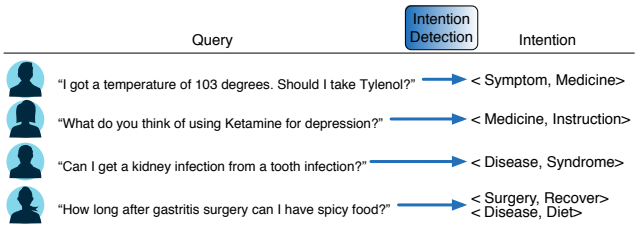

Figure 1: Common queries on medical QA websites. Intention is defined as a tuple pair in this paper. Intuitively, the intention $< Symptom, Medicine >$ in the first medical query means "by describing related information about *symptoms*, the user is looking for corresponding information about *medicine*".

ing intentions. As we have observed from medical queries, topic-specific words with concepts of disease, symptom and medicine could co-exist in one text query. Hence the existence of topic-specified words from neither disease category nor symptom category could easily dominate our prediction in intention detection tasks for medical queries. In order to achieve a better intention detection result, we need to look beyond exact words and have a deeper understanding of semantics inside query.

To sum up, intention detection task in this paper involves the following three challenges:

Intention modeling challenge: how to define discriminative representation for intentions so that they can be distinguished accurately and understood precisely from a text query. Previous works such as [7, 8] assign labels to sentence by first stacking vector representation of each word as a sentence matrix and then feeding sentence matrix into a convolutional neural network(CNN) model. However, those models fail to explicitly characterize correlations of two words that don't fit in one single convolutional region. Traditional CNN based methods often apply an average (possibly weighted) or a max operation over the whole sequence [9]. Those operations dealt with unfixed-length sequence by keeping the most salient information while neglecting other information.

Domain coverage challenge: how to deal with the situation where labeled queries only cover a small domain of all possible queries that users might provide. In medical queries, identical sentences are observed rarely or even not observed at all. For example, two patients get different diseases and their symptoms are not overlapping at all. But they may both look for drugs that help them relieve their symptoms thus may share the same intention. Even if for the same intention with an identical disease, people with different background and linguistic preferences tend to give different expressions. Due to high cost in human labeling, we are not able to label queries that cover all concepts, e.g. a set of queries talking about all kinds of diseases. However, correctly recognizing unseen queries are expected. [10] addresses a similar problem and proposed some techniques to automatically label more queries by a semi-supervised approach, which still required extra human efforts.

Diverse expressions challenge: how should we detect intentions from queries where the intentions are expressed partially, implicitly or diversely. Also, background information and other related descriptions bring diversity as well as noise into user's expression, which can't be avoided and may misguide the intention detection task. Limited by bag size and regardless of word order, bag-of-words and bag-of-N-grams neural network models are not designed to infer intentions from text queries with diverse expressions.

This work meets the challenges above by proposing a neural network based jointly modeling approach that incorporates heterogeneous information into intention modeling. Overall, this paper makes the following contributions:

- Pairwise feature correlation is designed to model intentions encoded in text queries regardless of the words a query contains. The proposed method can adapt to variable-length of text queries as inputs. An interleaving of convolution and pooling operation is then able to harness pairwise feature correlations among any two dimensions in word vector representations to help modeling the semantic transitions in text queries.
- A heterogeneous joint model is proposed, in which part-of-speech(POS) tags are utilized as the other heterogeneous information in addition to pairwise feature correlations for jointly modeling intentions. With extra medical related word tags attached to a bag of POS tags, the model is able to utilize previously labeled queries to a deeper extent.
- Sentence Rephrasing technique is introduced in the neural network training process as a data augmentation technique to automatically rephrase the text query in purpose of expression diversity. With sentence rephrasing, the proposed method is able to achieve a higher convergence rate and a lower error rate.

We demonstrate the effectiveness of the proposed method by applying it on a real world medical QA data set. Experimental results show that the proposed method outperforms other baseline models. Note that, although the proposed model is applied on text queries in medical domain, we expect the model can be easily integrated into applications in various domains, such as recommendation system for search engines, phone call routing for public services and so on.

2. PRELIMINARIES AND RELATED WORKS

2.1 Preliminaries

2.1.1 Definition of intention

The intentions in this work characterize semantic transitions from declarative statements to user's information needs. Semantics in both declarative statements and user's information needs are embodied in words, especially words belong to certain concepts, that constitute text queries. Formally, the intention is defined as follows:

Definition 1. Assume that each text query consists of some declarative sentences followed by a few questions. Thus for each query Q encoded with intention(s), the resulting intention(s) is (are) denoted as a set of tuple pairs:

$$I = \{< s, n >\}$$

where in each tuple pair $< s, n >$, s is the concept mentioned in declarative sentences of the text query as the information a user already knows, while n involves a concept mentioned in questions indicating user's information needs.

Based on the the medical diagnosis study in [11], as well as the real-world QA pairs we crawl from online medical QA websites, majority concepts in medical domain lie in the following five main categories: disease, symptom, medicine, examine and surgery. As shown in Figure 2, we use directed edges to show semantic transitions from one of concepts in five main categories to other concepts correspondingly. Note that for all combinations of semantic transition $< s, n >$, only those with high support in real-world data form directed edges.

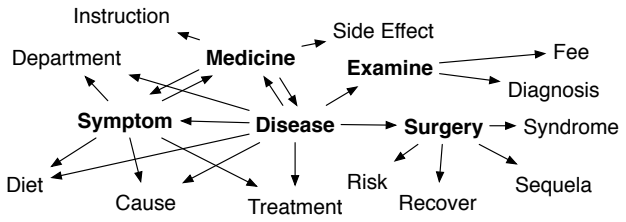

Figure 2: Semantic transitions among concepts. Main concepts are in bold font and other concepts are in normal font. The intention can be seen as a directed edge between two concepts, indicating the semantic transition from the concept that is mentioned in a declarative statement to concept mentioned in a question within a text query.

Intuitively, intention $< s, n >$ in medical queries can be explained as **"By describing related information in concept *s*, the user is looking for corresponding information about concept *n*"**. For example, we detected $< Symptom, Medicine >$ from query: "I got a temperature of 103 degrees. Should I take Tylenol?" as shown in Figure 1. In this query, by describing symptoms ("got a temperature of 103 degrees"), the user is looking for related information about medicine correspondingly("take Tylenol"). Since concepts in declarative sentences or questions can be mentioned either explicitly or implicitly(e.g. medicine related concepts can be explicitly expressed as "should I take Tylenol" or implicitly expressed as "what to take"), semantic transition is not only about co-occurrence of topic-specific words. Also, people usually tend to encode more than one intention in their queries at the same time (e.g. last example in Figure 1), therefore set I may have more than one element. Our intention detection task aims to associate an accurate and complete set I to every text query Q.

2.1.2 Word Embedding

The most straightforward way to represent a word is to use one-hot representation, also known as bag-of-word representation. One-hot representation describes each word by a sparse binary vector, whose size is equal to the vocabulary size of the text corpus. Each word only activates one entry in the vector by setting the value of corresponding entry as one, and all the other entries as zeros. Due to high vocabulary size in real life and low data sparsity by considering all words into a bag, one-hot representation always fails to incorporate rich word semantics into the representation.

Neural word embedding methods[12, 13, 14] represent each word by a real-valued dense word vector. From a statistical language modeling perspective, meaning of a word can be characterized by its context words. Therefore, neural word embedding method such as [12] aims to predict context words by the given input word while at the same time, learning a real-valued vector representation for each word. Since neural word embedding can be trained in a totally unsupervised fashion, word vectors can be obtained from a large corpus without syntax analysis or any manual labeling beforehand. Therefore in this paper, we are able to learn a high quality, fixed-dimension(usually far less than the vocabulary size) discrete representation for each word efficiently.

2.2 Related works

Previously, most intention detection(usually referred to as query classification) tasks [15, 16, 17, 10] have been studied in a search engine setting, where the task is to classify a query submitted to a search engine so as to determine what the user is searching for. Especially, some of them utilize user's browsing history or clicking logs to predict user's actions in the future.

For texts classification, since the work on token-level application in [9] by Collobert et al., many recent works [18, 19, 20, 21, 22, 23, 7] incorporate incredible learning ability of neural networks into text classification tasks. Those methods range from traditional bag-of-words and bag-of-N-gram neural network models where word order is totally ignored, to CNN based models that are able to keep word order and recurrent neural network based models that treat sentence as a temporal sequence inherently. Besides the consistent superior results that CNN achieved in various visual classification tasks in [24], CNN based methods also show their superiority in various natural language related tasks[25] compared with other highly tuned state-of-the-art systems.

Similar with LeNet[26] for image recognition tasks where convolutional layer and pooling layer are applied consecutively to extract features, CNN for text classification tasks tries to regard sentence as an image, usually by stacking vector representation of each word into a sentence matrix. For example, a sentence $Q = (w_1, w_2, w_3, ..., w_n)$ will be transformed into a sentence matrix $M \in \mathbb{R}^{n \times m}$ where n is the length of sentence and m is the dimensionality of vector representation.

In many CNN based text classification models, the first step is to convert word from one-hot sparse representation to a distributed dense representation using Word Embedding. This objective is fulfilled by either having a layer to perform the transformation or looking up word vectors from a table which is filled by word vectors that are trained separately using additional large corpus. In order to utilize the effectiveness of CNN without additional resources, Johnson et al. directly apply CNN with high dimensional one-hot vectors as input in [8]. The embedding of words are learned simultaneously as we feed all labeled sentences to the model. Computational infeasibility caused by using one-hot representation is alleviated by handling data on GPU efficiently. Methods like this rely on large labeled training set to cover as much words as possible, so that we can take advantage of word embedding to get high quality word vectors. In [18], convolutional layers are employed directly from the embedded word sequence, where embedded words are pre-trained separately.

Different from fixed-size input of image in image recognition tasks, texts are naturally variable-sized. Existing methods use different ways to deal with variable-size texts: either

adjust data to a unified size or design a resizable neural network model that adapts to varying input size. Zero-fill would be one of the simplest ways to adjust the data to get a fixed-size input. It is done by setting a maximum length constrain and appending shorter texts with zero vectors. On the other hand, in order to modify the neural network structure to achieve the same goal, Kalchbrenner et al. introduced a Dynamic Convolutional Neural Network (DCNN) structure to model text with unfixed length in [7]. Their method applies global pooling operations over varying-length texts to provide a fixed-length hidden layer. Note that without dynamic k-max pooling layer mentioned in [7], traditional max pooling layer for image will provide an unfixed-sized output, which later feeds forward to a convolution layer, resulting in an unfixed hidden unit size.

In order to keep the word order, most existing CNN based methods focus on word-level modeling by using various kinds of neural network structures with a stack of word vectors as inputs. However, word-level modeling doesn't explicitly deal with the correlation between two words that can't fit in a single convolution/pooling region size. To overcome the limitations of existing word-level modeling approaches in intention detection, we bring a new perspective to intention modeling for text queries: a neural network model that deals with arbitrary text length and can fully utilize feature-level correlations to model semantic transitions in text queries.

3. METHODOLOGY

3.1 Overview

An overview of the proposed method is given in Figure 3. The proposed method consists of modeling from two perspectives. The first perspective, upper branch in Figure 3, first transforms each word in a query into a vector representation. After stacking vectors to a sentence matrix, this module then uses pairwise feature correlations of the sentence matrix to model semantic interactions among different dimensions. An interleaving of convolution and pooling operations are applied to summarize interactions of semantic contents from raw feature correlations and extract the most salient semantic transition through those layers to help predict the intention of a text query.

The second perspective, shown as the lower branch, uses the same input as the upper branch. Instead, we tag each word with a POS tag to generate a bag of POS tags. Then, a fully connected layer is applied on the bag of POS tags followed by a merging layer which merges outputs from two branches of the neural network. The jointly modeling results are fed into following layers to get a probability distribution of intentions.

3.2 Feature-level modeling

Feature-level modeling gives a new perspective for intention modeling. Traditional neural network models for text classification are word-level models, which usually keep word orders and consider each query as a sequence of words. Word-level modeling utilizes the knowledge from previous labeled text queries when they are also similar on the word-level, e.g. two queries having similar word order and sharing almost all words in common. However, with the observation that identical text queries are rarely obtained in the medical domain, word-level modeling limits the generalization ability of intention detection since we are only able to learn from very few queries that are literally identical.

In this work, the intention detection task focuses on queries that are semantically similar. More specifically, we are interested in learning from labeled queries that have similar semantic transitions within them and do not necessarily have many words in common. Even though all natural languages rely to some extent on word order to signal relational information [27], the word order information do not always match the directions of semantic transitions in queries. Therefore, we harness feature-level correlations among each dimension of word vector to help modeling intentions. In [28], authors use feature-level correlations to learn a dimension-wise alignment between word vectors and designated linguistic property vectors so that higher dimension-wise correlation corresponds to more salient semantic content of that word vector dimension. Based on the hypothesis that dimensions in word vectors correspond to semantic contents, feature-level correlations of word vectors with itself are used to measure how semantic contents interact with each other within text queries.

3.2.1 Pairwise feature correlation matrix

For any word w_i in text query $Q = (w_1, w_2, w_3, .., w_n)$, we first transform each word w_i to a m-dimensional vector representation $vec(w_i) \in \mathbb{R}^m$. The transformation is done in Word Representation Layer by a table look-up. The word and its corresponding word vector are stored in a table, which is obtained by a separate word embedding training process. We derive a sentence matrix: $M \in \mathbb{R}^{n \times m}$ by stacking n word vectors together. Each dimension of the sentence matrix is considered to be filled with semantic content as features, thus pairwise feature correlation matrix $S \in \mathbb{R}^{m \times m}$ is computed by M as:

$$S = \begin{bmatrix} sim(M_1, M_1) & sim(M_1, M_2) & \cdots & sim(M_1, M_m) \\ sim(M_2, M_1) & sim(M_2, M_2) & & sim(M_2, M_m) \\ \vdots & \vdots & & \vdots \\ sim(M_m, M_1) & sim(M_m, M_2) & \cdots & sim(M_m, M_m) \end{bmatrix}, \quad (1)$$

where M_j is the j-th column in sentence matrix M. $sim(M_i, M_j)$ is the function measuring similarity between features M_i and M_j. In this paper, we use the cosine similarity as the similarity measurement. Pairwise feature correlation matrix S is computed in Feature Correlation Layer.

By using feature-level correlation of a query rather than exact words and its corresponding representations, the proposed approach provides a new perspective to model intentions, which differentiates itself from previous text classification tasks in essence. Moreover, since dimensionality of word vector is fixed during word embedding training, feature-level modeling also perfectly deals with unfixed length of queries.

3.2.2 Convolution operation

Convolution operation usually involves multiple filters to extract features. Each filter generates a feature map by moving a fixed-size convolution region all over the input space. In this paper, we use 2D convolution region of size $\mathbb{R}^{c \times c}$ to generate feature maps. For a convolution operation involving k filters, each filter k is associated with its corresponding weight matrix t_k of size $\mathbb{R}^{c \times c}$.

For each convolution region $\mathbf{x}$ of size $\mathbb{R}^{c \times c}$, we calculate feature c extracted from this region by:

$$c = f(\mathbf{t}_k \cdot \mathbf{x} + b_k), \quad (2)$$

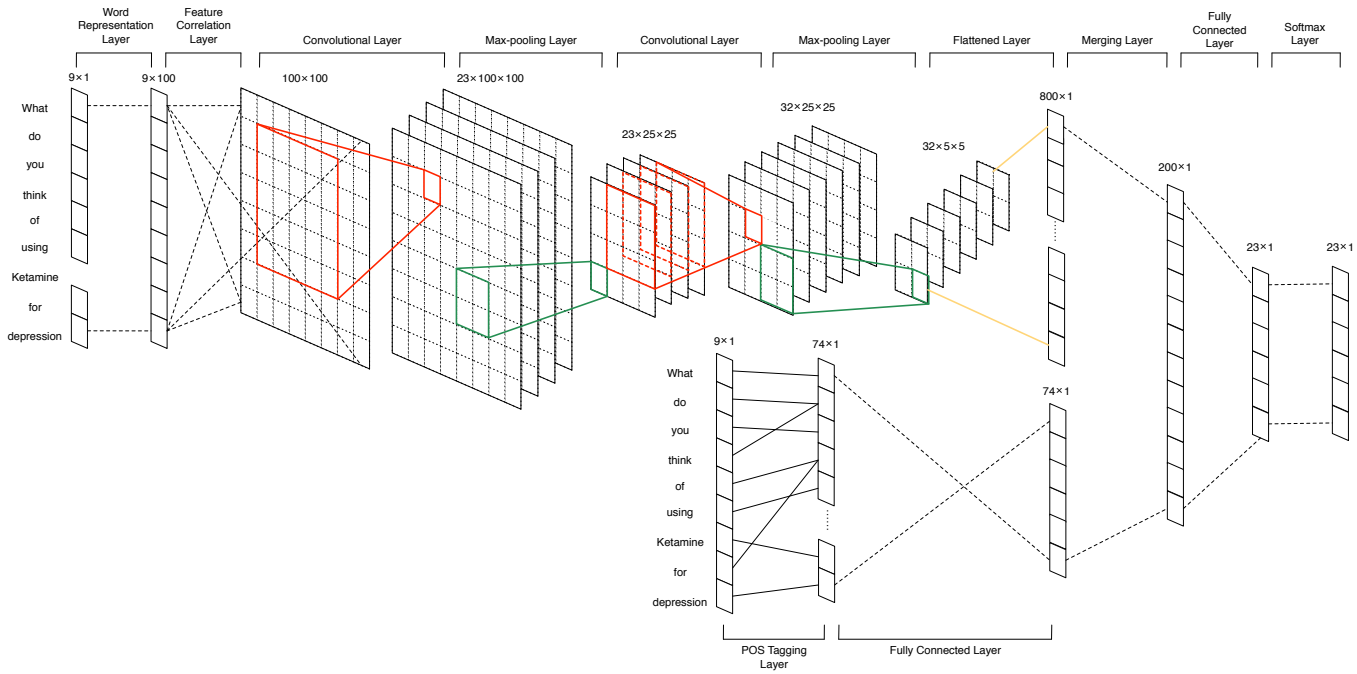

Figure 3: Architecture for the proposed method. Red blocks stand for convolution operations and green blocks stand for pooling operations. Yellow lines stand for flattening operations. The upper branch is for feature correlation modeling, while the lower branch is for POS tagging. Two perspectives take the same input but for clarity we duplicate the input in the figure above.

where b_k is the bias term of filter k and f is a non-linear transformation such as rectified linear unit(ReLU) $f(x) = \text{ReLU(x)} = \max(0, x)$. Convolution region moves a certain stride at each time, either vertically or horizontally. The stride is defined as the sliding step size of the convolution. Moving convolution region over all input space with a certain stride, we obtain a feature map

$$C_k = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1h} \\ c_{21} & \ddots & & c_{2h} \\ \vdots & & \ddots & \vdots \\ c_{h1} & c_{h2} & \cdots & c_{hh} \end{bmatrix}, \tag{3}$$

where h, the size of feature map, depends on the stride size as well as convolution region size. By applying convolution operation with each filter, we derive multiple feature maps for the input space. Note that in this paper, when we move convolution region partially out of input space, we pad the outside part with zeroes.

3.2.3 Pooling operation

The max pooling operation is a subsampling function that returns the maximum of a set of values [26]. In this paper, pooling layer is applied after each convolutional layer. We set the pooling region as a 2D square. For all feature maps, max pooling operation divides each feature map into non-overlapping sub-regions and selects the max value from each sub-region to only keep the most salient local feature within the region.

Generally, by an interleaving of convolution layer and pooling layer, the model summaries interactions of semantic contents from raw feature correlations and extracts the most salient semantic transition through several consecutive lay-

types. Each entry p_k has an integer value equal to the number of occurrence of tag p_k in a text query. Note that the POS tagging layer uses existing POS tagging algorithms such as [29] to come up with a tag for each word. Especially, words with existing special tags are collected and incorporated into the tagging algorithm. The fully connected layer after the POS tagging layer further estimates the contribution of different POS tags to the determination of intentions.

3.4 Jointly modeling

Feature correlations give us a feature-level modeling while POS tags give us word-level information about word-categories. By incorporating two heterogeneous information in a joint model, the proposed method is aimed to overcome the domain coverage challenge. Intuitively, in the partial query "I have been taking Tylenol", the existence of a drug name may probably indicate medicine related intentions. What we want to learn from this drug name is the semantic impact of mentioning this medicine related concept to modeling semantic transitions, rather than the word occurrence itself. By jointly modeling, the word-category of "Tylenol", which is "n_medicine" in our POS tagging, helps the model by giving explicit guidance that some medicine related words are probably involved in semantic transitions. When we come across a new query "I have been taking aspirin" in a joint model, we get extra side information for word "aspirin" as well. Note that, a word vector itself is embedded with word-category information since words in the same category are located near each other. With jointly modeling, we don't have to totally rely on the quality or the vocabulary size of word vectors to let the model recognize "Tylenol" and "aspirin" as two words under the same word-category. Also, POS tagging in jointly modeling alleviates a considerably number of queries that need to be labeled and achieves a wider domain coverage.

In the merging layer, we simply concatenate results from two incoming layers together. However, since the number of units in the neural network is still large, we therefore fully connect the concatenated result to subsequent layers. The merging layer and the following fully connected layer thus perform the dimension reduction.

3.5 Increasing model generalization ability

3.5.1 Data augmentation

Data augmentation methods have been used in image classification tasks prevalently to reduce over-fitting. The idea of data augmentation is to artificially enlarge number of labeled data using label-preserving transformations [30]. Replacing words in text with their similar words is considered as a typical label-preserving transformation. In intention detection tasks, human rephrasing for sentences would be the best choice in term of augmentation quality. However, it is unfeasible and not practical to do so. As a result, in this paper Sentence Rephrasing is introduced as an augmentation method for the proposed method. In well-trained word vectors, words that are semantically similar are close to each other on vector space. As we observe from pre-trained vector representations, nearest neighbors of a medicine are also names of similar medicines. We utilize word vectors trained on large corpus to rephrase the sentence automatically. Theoretically, word embedding model is aiming to produce similar vector representation to words that are likely to occur in the same context. Therefore, it is also reasonable to do sentence rephrasing by using nearest neighbors of pre-trained word vectors. As long as we use the nearest neighbors of a word in a vector space to generate candidate rephrasing words, we just pick up new words that always occur in the same context with the original word. Replacing each word in a sentence to generate more sentences is neither efficient nor necessary, so we constrain original word and candidate words with a equality constraint on POS type as well as similarity constraints. Sentence rephrasing tries to add minor perturbations to provide some extra diversity in expressing specific words to generate new data. As for diversities from sentence structure aspect, we assume they are already observed from original training data before Sentence Rephrasing.

Algorithm 1 Sentence Rephrasing Algorithm

Input:
$Q = (w_1, w_2, w_3, ..., w_n)$: a query with n words.
candidate_num: number of candidate generated by nearest neighbors.
δ_I/δ_O: thresholds as the lower bound similarity constraint for words with /without special tags.

Output:
A set of m queries $Q' = \{Q'_1, Q'_2, Q'_3, ..., Q'_m\}$ generated by augmenting the input query Q.

1: **function** SENREPHRASING(Q,*candidate_num*,δ_I,δ_O)
2: **for** $i = 1$ to n **do**
3: **if** w_i in D and $vec(w_i) \neq Null$ **then**
4: $W = nearest\_neighbor(vec(w_i), candidate\_num)$
5: **if** ($w_{ni} \in W$ has $pos(w_{ni}) = pos(w_i)$ **and** $sim(w_{ni}, w_i) \geq \delta_I$) **or** ($w_{ni} \in W$ has $pos(w_{ni}) \neq pos(w_i)$ **and** $sim(w_{ni}, w_i) \geq \delta_O$) **then**
6: Generate a new query $Q_j{}'$ by replacing w_i with w_{ni} in Q
7: Add the new augmented query $Q_j{}'$ to Q'
8: **end if**
9: **end if**
10: **end for**
11: **return** Q'
12: **end function**

The detailed algorithm for Sentence Rephrasing is described in Algorithm 1. The algorithm takes four input arguments, the original query Q, the number of candidate(s) *candidate_num* from the nearest neighbor and two thresholds δ_I and δ_O to constrain the similarities. We iterative through all words of Q in Line 2. Line 3 determines whether each word w_i is associated with a special POS tag in a dictionary D, as we indicated in Table 1. $vec(w_i)$ function mentioned in Lines 3 and 4 returns the vector representation of word w_i. W is the set of top-*candidate_num* nearest neighbors of w_i. Line 5 further checks each candidate word w_{ni} that whether it has the same tag with w_i. If so, we use a relatively lower threshold δ_I for the similarity measurement. Otherwise δ_O is used as the threshold for the similarity measurement of the new word. If the new word meets those constrains we replace w_i by the candidate w_{ni} in Line 6 to generate an augmented query and add it to Q'.

3.5.2 Dropout

Dropout[31] is a regularization method designed to overcome co-adapting of feature detectors in deep neural net-

works. It is easy to implement and can be considered as an effective way to reduce test error. The dropout method randomly sets value zero to output of each hidden neuron with a certain probability. The "dropout" neurons do not contribute to the feed-forward process as well as the back propagation process. In this paper, dropout layer is applied after each pooling layer, with 0.5 probability that each output of hidden neuron is "dropped out" and set to zeros.

4. EXPERIMENTS

4.1 Data description

We collect large-scale corpus from a popular medical QA website http://club.xywy.com/. The website is claimed to be the top online health care service platform in China, with over 200 thousand registered doctors and over 80 million registered users by the end of 2014. The corpus contains 64 million records, covering over 10 thousand diseases. Each record consists of a department indicator, a text query as question and an answer from one doctor. In this paper we only study questions as the text queries since we focus on the intentions users encoded in their questions. We clean the data in advance by removing meaningless characters such as repeated characters in text queries.

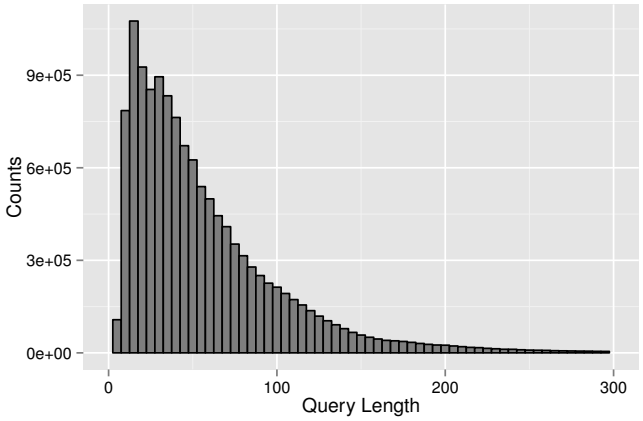

Figure 4: Length distribution of all the queries.

Figure 4 shows the length distribution of text queries in the raw data. In this paper, we only consider text queries which are shorter than 150 words. After random sampling and human labeling, we have 21529 labeled text queries. Since a query can encode more than one intention, we label each query with multiple intentions if they exist in the query. Figure 5 shows the intention distribution in the labeled text queries. Among all labeled text queries, 73% text queries has only one intention while other 24% text queries are encoded with two intentions. The remaining 3% text queries have three or more intentions. Text queries with more than four intentions are not observed in our data set.

Words that have existing special POS tags in the medical domain are also crawled separately from the medical knowledge base on http://club.xywy.com/. Among 80726 words with special tags in the medical domain, the top three frequently tagged word-categories are "n_disease"(25259 words), "n_medicine"(22689 words) and "n_symptom"(14726 words).

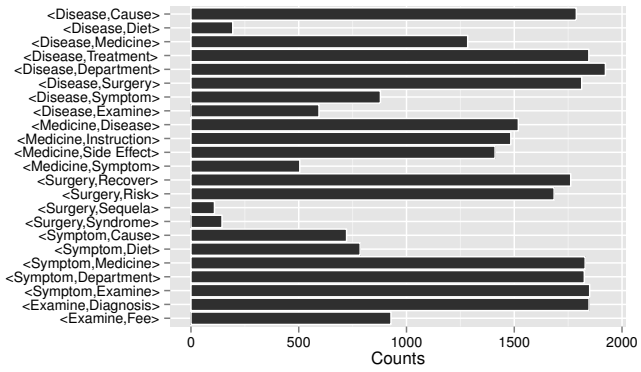

Figure 5: Intention distribution.

4.2 Pre-trained word vectors

The language of the obtained corpus is in Chinese. Since Chinese text queries are not naturally split by spaces, first we apply word segmentation on the whole corpus. Word segmentation for Chinese doesn't simply segment query by each Chinese character. Instead, it tries to combine several consecutive strongly correlated characters into words, thus "word" can contain more than one Chinese character. After word segmentation we get a sequence of meaningful words from each text query. By a separately trained word embedding model using large corpus in a totally unsupervised fashion, we can alleviate the negative impact from limited word embedding training corpus from only labeled queries. In this paper, we use word2vec [12] to train vector representation of words on 64 million text queries. The vectors have dimensionality of 100 and were trained using the Skip-gram [12] architecture. Context window size is set to 8 and we specify a minimum occurrence count of 5. The resulting vocabulary contains 382216 words. Words not presented in the set of pre-trained words are initialized as random vectors.

4.3 Baseline methods

To demonstrate the effectiveness of our method, we compare the performance of methods described below in our experiment.

- SVM-FC: Support Vector Machine model with RBF kernel that uses feature-level correlations on a stack of word vectors to represent each text query.
- LR-FC: Logistic Regression model that also uses feature-level correlations on a stack of word vectors to represent each text query.
- NNID-ZP: neural network based intention detection model that uses a stack of word vectors to represent each text query. Zero-padding is performed for queries shorter than 150 words.
- NNID-FC: neural network based intention detection model that uses feature-level correlations on a stack of word vectors to represent each text query.
- NNID-JM. The proposed method which incorporates heterogeneous information to jointly model feature-level correlations and word-level POS tags.
- NNID-JMSR. The proposed method that takes advantage of Sentence Rephrasing.

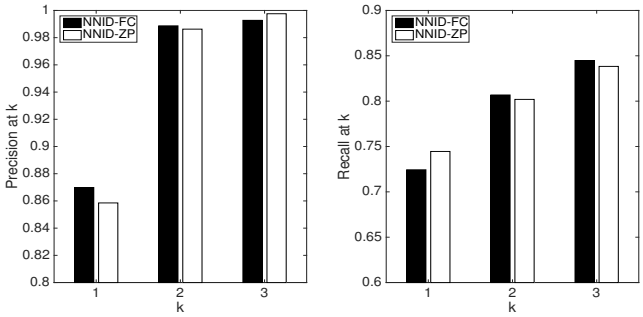

Figure 6: NNID-FC vs NNID-ZP.

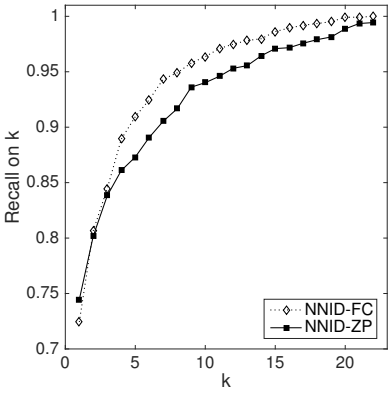

Figure 7: Recall of NNID-FC vs NNID-ZP.

4.4 Performance study

60% of the data are used as training set and 20% as validation set for parameter tuning. The remaining 20% data are used for testing purpose. Performance study of the proposed method is based on the results of 5-fold cross validation. The proposed model is implemented on Minerva [32] and trained using stochastic gradient descent with momentum [33].

4.4.1 Evaluation metrics

Standard evaluation metrics such as precision and recall are used to evaluate the performance. Especially, we are interested in whether the model is able to hit true intentions encoded in test queries within its top-k predictions. In medical-related applications, incorrect detection is much worse than missing an intention. Therefore, improvement on precision within the first few predictions also attracts our interests. Precision and recall at the top-k predictions are evaluated. After that, some illustrative cases are presented.

4.4.2 Effectiveness of feature correlation modeling

To show the effectiveness of using feature-level modeling methods for intention detection, we compare NNID-ZP with NNID-FC. NNID-ZP simply treats input as a stack of word vectors. When input query is shorter than an upper bound, which is 150 words in our case, we pad remaining entries as zeros. Figure 6 shows the precision and recall from top-1 to top-3 predictions using NNID-ZP and NNID-FC. Two methods have tied performance in precision. By ignoring exact words of the query and using feature-level modeling, NNID-FC is able to keep the semantic structure and achieve competitive results.

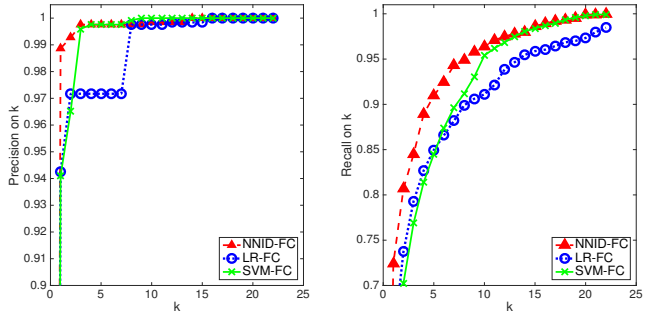

Figure 8: NNID-FC vs LR-FC vs SVM-FC.

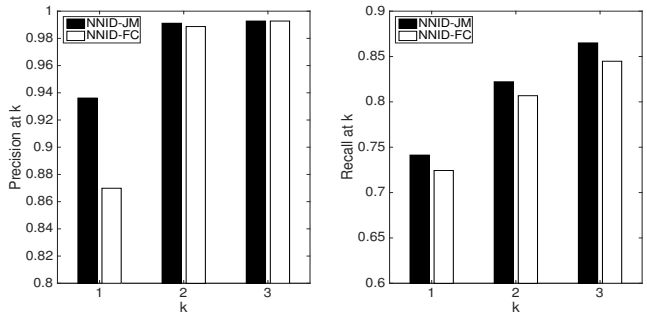

Figure 9: NNID-JM vs NNID-FC.

We witness the counterattack of feature-level modeling from the performance in recall. As shown in Figure 7, except minor lost in top-1 prediction, we observe consistent improvements for recall at k when k>1 by NNID-FC. In our application that detects intentions from medical related queries, this can be a trivial issue because once the algorithm failed to understand user intention in the first few prediction, it should not expect to be rewarded for recalling the correct intention in its, for example, 10th predictions. However, such observation in turn validates for our intuition to use feature-level correlations to model intentions, rather than using exact words. For word-level models like NNID-ZP, their generalization abilities are limited by using exact words in training. Therefore even given more chances to predict, they are still limited by the knowledge learned on exact words and failed to recall queries that are semantically related but literately different, as shown in Figure 7.

Furthermore, we compare the neural network approach(NNID-FC) with others baseline methods such as support vector machine(SVM-FC) as well as logistic regression(LR-FC). As shown in Figure 8, we can see NNID-FC outperforms the other two baseline methods in recall when precision are tied after top-5 predictions.

4.4.3 Effectiveness of jointly modeling

As shown in Figure 9, we have compared two models: the proposed NNID-JM model which considers two heterogeneous information and NNID-FC which only considers feature-level correlations. For each model, we measure the precision and recall from top-1 to top-3 predictions. Note that by incorporating POS information into modeling, we observe a significant improvement (7%) of NNID-JM in its top-1 precision without losing any performance in all the recalls.

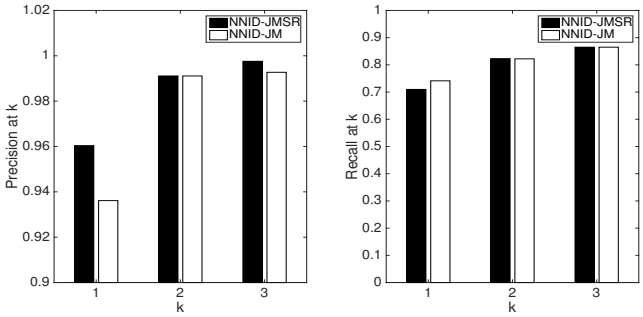

Figure 10: NNID-JMSR vs NNID-JM.

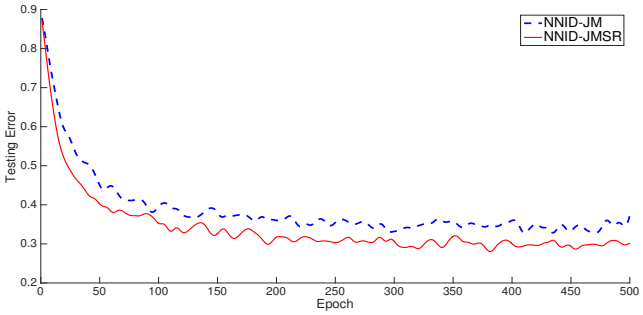

Figure 11: Testing error on top-1 prediction for NNID-JMSR and NNID-JM.

4.4.4 *Effectiveness of Sentence Rephrasing*

Sentence rephrasing is introduced in our model to further improve the performance. We perform the sentence rephrasing on CPU to generate more labeled data. While the model is loading data into GPU, sentence rephrasing generates new queries and loads them into GPU memory. Therefore rephrased text queries are not necessary to be stored on disk. For parameters in Sentence Rephrasing, we pick in-dict threshold $\delta_I = 0.85$ and out-of-dict threshold $\delta_O = 0.95$ from a validation set. Such strict thresholds may limit the number of augmented queries generated by sentence rephrasing, but will guarantee that the generated queries are in natural language and understandable.

Before Sentence Rephrasing, 12917 text queries are used for training. Sentence Rephrasing generates 5382 new queries, resulting in 18299 text queries for training. Figure 10 shows the performance of NNID-JMSR which takes advantage of sentence rephrasing and NNID-JM. NNID-JMSR achieves a 3% improvement on precision of top-1 prediction by trading off 3% drops in top-1 recall. Further top-k predictions show consistent improvement on both precision and recall. Moreover, sentence rephrasing contributes not only in model performance, but also in model training. Higher convergence rate (especially in first 50 epochs) as well as lower error rate achieved by NNID-JMSR model in Figure 11.

4.5 Case study

To have a better understanding of intention detection in real-world medical queries, we present some illustrative cases in this section. Due to limited space, we present one query for each main category.

•**Query**: I got fever with a temperature of 103 degrees. Should I take Tylenol? (发烧发到40度可以吃泰诺吗?)
Prediction:

Rank	Intention	Probability (NNID-FC)	Probability (NNID-JM)
1	<symptom,medicine>	0.442620	0.782177
2	<disease,medicine>	0.414766	0.163278
3	<drug,symptom>	0.054226	0.040807
4	<symptom,cause>	0.019793	0.004721
5	<symptom,treatment>	0.021714	0.003180

"Fever" mentioned in the query can be seen as a noun (disease name) or a verb (describe a symptom), therefore the prediction may fall into either symptom related intentions. Since extra information about temperature is described in the query, the model may prefer treating "Fever" as a verb thus $< symptom, medicine >$ ranks higher than $< disease, medicine >$ by NNID-FC. In jointly modeling, since "Fever" is tagged with "n_symptom" rather than "n_disease", NNID-JM model shown a stronger preference in supporting $< symptom, medicine >$

•**Query**: How should I give medicine for my boy who got pneumonia and keep coughing during night? If using amoxicillin pills, how many tablets should I give and how many times the medicine should be taken by a 44lbs boy every day? (孩子肺炎该怎么吃药? 孩子四十斤。吃阿莫西林一次吃几袋，一天吃几次呢?)
Prediction:

Rank	Intention	Probability
1	<medicine,instruction>	0.965866
2	<disease,medicine>	0.012223
3	<medicine,symptom>	0.007534
4	<symptom,medicine>	0.005455
5	<medicine,side effect>	0.005212

As we can see from this query, many noisy information are involved in the text query. Even if the user wants to acquire information about medicine, he/she sometimes starts with corresponding diseases and symptoms as the background information. However, those factors are not the key idea of user's query. From the result we can see that our model is able to specify major intention from the question, rather than other background information.

•**Query**: How long after gastritis surgery can I have spicy food? (请问胃炎手术后多久可以吃辛辣食物?)
Prediction:

Rank	Intention	Probability
1	<surgery,recover>	0.640065
2	<disease,diet>	0.205127
3	<disease,surgery>	0.057054
4	<surgery,sequela>	0.053530
5	<surgery,syndrome>	0.042654

In this query, multiple intentions co-exist in one query. User would like to know 1) Whether he/she can have spicy food with stomach diseases, and 2) Is it okay to have spicy food after certain surgery. Our model handles this situation

well probably due to multiple feature maps within each convolutional layer, where each feature map has the ability to capture certain semantic transition independently.

•**Query**: I was diagnosed with high blood pressure in my annual physical examination last month. What foods should I be eating on a regular basis? (上个月组织体检检查出了高血压，平时注意些啥该多吃什么?)
Prediction:

Rank	Intention	Probability
1	<disease,diet>	0.469885
2	<symptom,diet>	0.305881
3	<symptom,drug>	0.197734
4	<symptom,examine>	0.007157
5	<treatment,diagnose>	0.007089

In the query a user expresses his/her intention for seeking diet related information very implicitly in Chinese. Our model is able to put diet related intention at the first two places. Especially, the proposed model doesn't let "examine" related queries dominate its prediction simply due to the occurrence of word "examination" in the query. Actually "examine" in "physical examination" is mentioned as part of the background, which doesn't indicate "examine" related intentions.

•**Query**: How much does it costs for a Lumbar CT? Recently my lumbar always hurts. (腰椎CT检查大概需要多少费用? 最近后腰老是酸疼。)
Prediction:

Rank	Intention	Probability
1	<examine,fee>	0.986955
2	<symptom,examine>	0.012433
3	<symptom,department>	0.000475
4	<disease,department>	8.50e-05
5	<examine,diagnose>	3.51e-05

Difficulty in detecting indention in this query is that there are tons of examinations one can receive in hospitals while we only have limited labeled queries which only cover a small portion of the examination terms. The proposed jointly modeling approach is feasible to solve this problem by integrating POS tag into the model. For example, if some other patient posts a query regarding the cost of cervical CT, not the lumbar in our case, then singular modeling methods such as NNID-FC only capture the fluctuation value in feature correlation matrix while the jointly modeling approach like NNID-JM can also utilize POS tags as long as the model is able to tag both "cervical CT" and "lumbar CT" with "n_examine". In that case, this special POS tag "n_examine" gives us extra information that some examination terms exist in the new query. The jointly modeling approach can take advantage of this extra clue to learn from previous knowledge more efficiently.

5. CONCLUSIONS

Intention detection for medical query will provide a new opportunity to connect patients with medical resources more seamlessly both in physical world and on the World Wide Web. Knowing what a user is looking for (e.g. a specific medicine that relieves headache, or a schedule of a doctor with high expertise in stomach diseases, or the average cost for lung surgery), health care resources can be made more accessible to the general public.

In this paper we present a jointly modeling approach to model intentions that users encoded in medical related text queries. By using feature-level modeling as one perspective to model intention, the proposed method is able to take variable-size text query as input. The resulting feature-level correlations are fed forward through an interleaving of convolution and pooling layer to extract semantic transitions from text queries. To aid the modeling, a bag-of-POS tags are used as the other perspective of modeling to indicate the existence of certain topic-specific words inside a query. Taking advantage of two heterogeneous perspectives in a joint model, the proposed approach has the capability to adapt to diverse user expressions covering a wide range of intentions. Performance evaluation and case studies have shown the effectiveness of the proposed method in discovering a complete and accurate intentions from text queries. Note that although in this work we focus on a medical related application, the proposed intention detection methods can be generalized and integrated into other existing applications such as recommendation system in search engines or phone call routing system for public services as well.

6. ACKNOWLEDGMENTS

This work is supported in part by NSF through grants III-1526499, CNS-1115234, and OISE-1129076. We thank Chaochun Liu, Tao Yang, Hao Wu and Yaliang Li who provided insights and expertise that assisted the research.

7. REFERENCES

[1] James F Allen and C Raymond Perrault. Analyzing intention in utterances. *Artificial intelligence*, 15(3):143-178, 1980.

[2] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *SIGIR*, 1999.

[3] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[4] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1-135, 2008.

[5] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415-463. Springer, 2012.

[6] Yanshen Yin, Yong Zhang, Xiao Liu, Yan Zhang, Chunxiao Xing, and Hsinchun Chen. Healthqa: A chinese qa summary system for smart health. In *Smart Health*, pages 51-62. Springer, 2014.

[7] Edward Grefenstette and Phil Blunsom. A Convolutional Neural Network for Modelling Sentences. In *ACL*, 2014.

[8] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *NAACL-HLT*, 2015.

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

[10] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. Understanding user's query intent with wikipedia. In *WWW*, 2009.

[11] Klaus-Peter Adlassnig. Fuzzy set theory in medical diagnosis. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(2):260–265, 1986.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[13] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *ICML*, 2007.

[14] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008.

[15] Zheng Chen, Fan Lin, Huan Liu, Yin Liu, Wei-Ying Ma, and Liu Wenyin. User Intention Modeling in Web Applications Using Data Mining. In *WWW*, 2002.

[16] Botao Hu, Yuchen Zhang, Weizhu Chen, Gang Wang, and Qiang Yang. Characterizing search intent diversity into click models. In *WWW*, 2011.

[17] Honghua (Kathy) Dai, Lingzhi Zhao, Zaiqing Nie, Ji-Rong Wen, Lee Wang, and Ying Li. Detecting online commercial intention (OCI). In *WWW*, 2006.

[18] Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings*, 2013.

[19] Long Chen, Dell Zhang, and Mark Levene. Question retrieval with user intent. In *SIGIR*, 2013.

[20] Xiao Ding, Ting Liu, Junwen Duan, and Jian-yun Nie. Mining User Consumption Intention from Social Media Using Domain Adaptive Convolutional Neural Network. In *AAAI*, 2015.

[21] Puyang Xu and Ruhi Sarikaya. Contextual Domain Classification in Spoken Language Understanding Systems Using Recurrent Neural Network. In *ICASSP*, 2014.

[22] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[23] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent Convolutional Neural Networks for Text Classification. In *AAAI*, 2015.

[24] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPRW*, 2014.

[25] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *EMNLP*, 2014.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

[27] L Gershkoff-Stowe and S Goldin-Medow. Is there a natural order for expressing semantic relations? *Cognitive Psychology*, 45(3):375, 2002.

[28] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. In *EMNLP*, 2015.

[29] Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. HHMM-based chinese lexical analyzer ICTCLAS. In *SIGHAN*, 2003.

[30] Patrice Y Simard, Dave Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, 2003.

[31] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.

[32] Minjie Wang, Tianjun Xiao, Jianpeng Li, Jiaxing Zhang, Chuntao Hong, and Zheng Zhang. Minerva: A scalable and highly efficient training platform for deep learning. In *NIPS Workshop, Distributed Machine Learning and Matrix Computations*, 2014.

[33] Ilya Sutskever, James Martens, George Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013