2017 IEEE International Conference on Data Mining

Knowledge Guided Short-Text Classification For Healthcare Applications

Shilei Cao*, Buyue Qian*, Changchang Yin* Xiaoyu Li* Jishang Wei† Qinghua Zheng* Ian Davidson‡
*Xi'an Jiaotong University, Xi'an, Shaanxi, China 710049
Email: shileicao@stu.xjtu.edu.cn, qianbuyue@xjtu.edu.cn, {lentery,wemakefocus}@stu.xjtu.edu.cn, qhzheng@xjtu.edu.cn
†HP Labs, 1501 Page Mill Rd, Palo Alto, CA 94304, USA
Email: jishang.wei@hp.com
‡Department of Computer Science, University of California, Davis, CA 95616, USA
Email: indavidson@ucdavis.edu

***Abstract*—The need for short-text classification arises in many text mining applications particularly health care applications. In such applications shorter texts mean linguistic ambiguity limits the semantic expression, which in turns would make typical methods fail to capture the exact semantics of the scarce words. This is particularly true in health care domains when the text contains domain-specific or infrequently appearing words, whose embedding can not be easily learned due to the lack of training data. Deep neural network has shown great potentials in boost the performance of such problems according to its strength on representation capacity. In this paper, we propose a bidirectional long short-term memory (BI-LSTM) recurrent network to address the short-text classification problem that can be used in two settings. Firstly when a knowledge dictionary is available we adopt the well-known attention mechanism to guide the training of network using the domain knowledge in the dictionary. Secondly, to address the cases when domain knowledge dictionary is not available, we present a multi-task model to jointly learn the domain knowledge dictionary and do the text classification task simultaneously. We apply our method to a real-world interactive healthcare system and an extensively public available ATIS dataset. The results show that our model can positively grasp the key point of the text and significantly outperforms many state-of-the-art baselines.**

I. Introduction

Short-Text classification [1]–[5] is a challenge task in the text mining domain. Traditional methods used to tackle this problem need well designed hand-crafted features, which require much time to find the optimal text representation. Recently, Word2vec [6] was proposed to learn high-quality embedding for words. Using this as a building block, a substantial number of deep neural network models were proposed, which have shown improvement for this problem. However, there are still lots of difficulties when the text is short, since there may not be enough information that we can extract from the individual meaning of words.

Short text classification in healthcare applications is even more challenging. Not only are there much precise language but the underlying concepts are inherently difficult. In a typical Electronic Health Record (EHR) [7] system, the number of diseases, medicines, laboratories, operations typically number in the hundreds or even thousands. But these medical concepts are extremely unevenly distributed—some common disease may be seen often, whereas others either only appear in several records or not at all. This makes learning an unbiased embedding challenge. To overcome these challenges we explore how domain knowledge can be used.

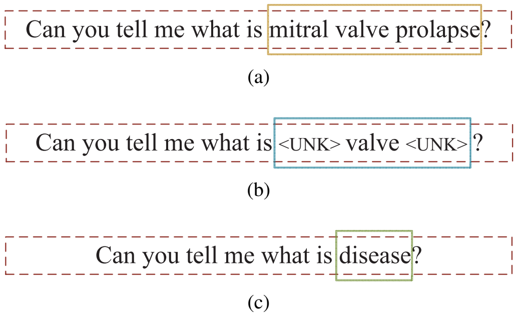

Can you tell me what is mitral valve prolapse?
(a)
Can you tell me what is <UNK> valve <UNK> ?
(b)
Can you tell me what is disease?
(c)

Fig. 1. Illustration of challenges that chatbot confronts in a typical dialog system. (a) shows what the chatbot originally received. (b) shows what the chatbot actually process since "mitral" and "prolapse" are not included in the pretrain vocabulary. (c) shows using entity type to replace the entity words can achieve a better classification

Consider identifying the intent of an utterance in an interactive healthcare system as shown in Figure 1. The chatbot was given an utterance with several unknown medical entities (such as uncommon medicines or diseases) within it. It is not easy for a chatbot to identify the user's intents, since the unknown medical entities provide no information for the chatbot at all. The utterance in Fig. 1a is an example what the chatbot received. Since *mitral valve prolapse*[1] is not a common disease and the chatbot may not have seen it in the training data. Since "mitral" and "prolapse" are not commonly used words, they will be identified as <UNK>. This poses the challenge to determine the intent of a user, since the only useful word "valve" provides little information about what entity type that "<UNK> valve <UNK>" belongs to. Given that, the chatbot doesn't know whether "<UNK> valve <UNK>" is a disease name or not, thus it would not get the user's intent (asking for a disease). However, if the chatbot was told that "mitral valve prolapse" is the name of a disease, it would easily determine

[1] *mitral valve prolapse* is a valvular heart disease characterized by the displacement of an abnormally thickened mitral valve leaflet into the left atrium during systole

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.12

the intent, as the medical type of "mitral valve prolapse" is a key indicator to the dialog system. If this is the case, the utterance the chatbot received turns into Fig. 1c.

As Fig. 1 shows, we refer to the entity types that the corresponding entities belong to as our domain knowledge[2]. This information can not be easily learned from the existing models since they are always latent in the text. We will show that the entity types play a key role in short text classification. In this paper, we develop two strategies to further explore the value of domain knowledge in our short text classification task. The first one we call it Entity Words Replacing mechanism. It works as follows: (1) identifying the name entities appearing in the texts, which can be done using a dictionary lookup approach; (2) We use the entity types to replace the corresponding entities. This makes the information being utilized more directly. We will show that this mechanism can be used as an effective preprocessing step for a number of different models which not only improves the classification performance but also can speed up the model convergence, as well as significantly reducing the vocabulary size.

The second strategy we explore is to utilize the domain knowledge with an attention mechanism [8]–[11]. Specifically, we employ a bidirectional Long-Short Term Memory (BI-LSTM) [8] network as the underlying architecture to capture the semantic relation amongst words in the texts. After that, we aggregate the entity types and then use the extracted information to attend the key point distributed among word representations learned by BI-LSTM. In this paper, we name it as Domain Knowledge Guided Attention Model (DKGAM).

The domain knowledge dictionary can be easily accessed. For example, in the domain of healthcare analytics, we can easily collect the medical entity dictionary (such as medicine, disease) from online resources, such as Medpedia or public medical knowledge graphs. In real world cases, the domain knowledge dictionary collected may not be complete, so that we can not rely on the simple dictionary lookup algorithm. To address this, we propose a Multi-Task Domain Knowledge Guided Attention Model (MT-DKGAM) to learn domain entities and perform Short-Text classification jointly.

To evaluate our method, we conduct a number of experiments on a real-world self-crawled medical dataset and a public accessible dataset ATIS. The results show that our method does capture the keypoint for short-text classification and outperform the state-of-the-art result on both datasets.

The technical contributions of our paper are as follows:

- We propose an Entity Words Replacing mechanism which can be readily used with other text classification models as a preprocessing step. With this mechanism, vocabulary size can be significantly reduced, and then the learning models can utilize the key information in a more direct fashion, thus speed up the model convergence rate. In addition, it also addresses the problem of missing embeddings of the infrequent entity words.
- We propose a novel knowledge guided attention model to maximally take advantages of the most discriminating information—entity types—in short-text classification problems. It is, to the best of our knowledge, the very first paper to explore the value of entities with attention mechanism in text classification problems.
- We present a multi-task model to jointly learn the domain entities and perform the classification. This is particularly useful when the collected entity dictionary is incomplete.
- The proposed method is particularly useful in healthcare applications, since there commonly exists lots of medical concepts which are extremely unevenly distributed.

The rest of the paper is organized as follows. Section II provides some relevant background regarding Text Classification, Name Entity Recognition, Intent Determination and Slot Filling. The details of our proposed method, which utilize the domain knowledge to improve text classification accuracy, are presented in Section III. To make the training more smoothly, a multi-task model which can discovery domain knowledge itself is proposed at Section IV. We then give an experimental analysis of our proposed method in Section V. Finally we conclude our work in Section VI.

II. Related work

Our work involves multiple areas. In this section, we present three most related works, including Text Classification, Name Entity Recognition, Intent Determination and Slot Filling. we will also highlight the correlations and primary distinctions of our proposed method compared to these existing methods appearing in different domains that are available in the literature.

Text Classification: Traditional text classification [4] mainly focus on the feature engineering and designing algorithms suited for the proposed feature representation. These two aspects are developed separately though with small alternate effect. The former growth line usually needs well designed hand-crafted features according to the characteristic of the problem. Some common used features can be obtained from Bag of Words model [4], [12], [13] (which is constructed by selecting most frequent words from the dataset) or Bag of n-gram [3], [4], [13] models (which are constructed by selecting the most frequent n-grams [3], [13] (n can be 2, 3 or both) from the dataset). Features can be binary [12] which denotes the presence/absence of a word/n-gram, multinomial which denotes the count of a word/n-gram or continuous which can be term-frequency inverse-document-frequency [12] of a word/n-gram. The latter growth line always absorbs mature algorithms from the machine learning research community. These algorithms include but not limited to Naïve Bayes, Logistic Regression [3], Support Vector Machines (SVM) [3], [14], Random Forest. With these hand-crafted features and algorithms at hand, people always need a lot of tedious attempts to get a satisfactory result.

Deep neural network shows great potential for this problem under the shine of its big success on image. Since the Word2vec [6] was proposed, more and more text mining tasks turn to deep neural network for better performance. Socher

[2] In the rest of paper, we shall consider these two concepts interchangeable.

et al. [12] learn vector space representations for multi-word phrases using recursive auto-encoders. With these representations, they perform state-of-the-art result for sentence-level prediction of sentiment label distributions. To capture the compositional meaning of longer phrases, Socher et al. [15] incorporate a vector and a matrix representation into a recursive neural network model for learning the compositional vector representations of phrases and sentences. Kim [1] proposed a convolutional neural networks model for sentence-level classification, he showed with the help of pre-trained word vectors, a simple model architecture can get a comparable result with little hyperparameter tuning. After that, Lai et al. [3] introduce a recurrent convolutional neural network for text classification to tackle the bias problem of Recurrent Neural Network. Unlike model using word embedding, [4] give an empirical exploration on using character-level convolutional networks for text classification. Miyato et al. [16] extend adversarial and virtual adversarial training to the text domain by applying perturbations to the word embeddings in recurrent neural network with a semi-supervised setting.

These models either focus on the structure of text or the architecture of neural network. They neglect the domain knowledge behind the text, which is especially important when facing lots of infrequently used words in short text. It is noted that [17] also noticed the importantness of concepts in text classification task, the difference is they focus on topic drifting detection method for concept clustering.

Name Entity Recognition: Our work is partially relevant to Name Entity Recognition (NER). NER is a fundamental task in text mining domain. It is also an application of sequence labeling/tagging [18]. Traditional NER methods use dictionary lookup method to fetch the entities, which may not applicable since the dictionary may not be available in most case. State-of-the-art models either model a chain-structured graphical model that contains an inference step to search the space of possible output sequences, or use beam search to approximate [19]–[21].

Huang [19] propose a series of models for sequence tagging, including LSTM networks, bidirectional LSTM (BI-LSTM) networks, LSTM with a Conditional Random Field (CRF) layer (LSTM-CRF) and bidirectional LSTM with a CRF layer (BI-LSTM-CRF). They show comparative accuracy on POS, chunking and NER data sets. Lample et al. [20] proposed two models both contain a CNN module to encode the character-level information—one is similar to [19] and another uses a transition-based approach to construct and label segments which are inspired by shift-reduce parsers. Strubell et al. [21] propose an alternative to BI-LSTMs, they use iterated dilated CNN and get a competitive result with speed greatly accelerated.

In this work, we absorb the BI-LSTM-CRF model used in [20] to discover the domain knowledge in text.

Intent Determination and Slot Filling: In Spoken Language Understanding (SLU) domain, Intent Determination and Slot Filling are two major tasks which are similar to the task of text classification and name entity recognition separately. The research of SLU can be traced back to 1990 [22], which emerged from identifying what people actually say and the Air Travel Information Services (ATIS) project. The development of these two tasks is also accompanied with hand-crafted features and classification algorithms. Our work is related to joint training of Intent Determination and Slot Filling, the difference is our work focus on using the entity type information to refine our classification accuracy. So we briefly review joint model of Intent Determination and Slot Filling.

Mairesse et al. [23] present a semantic parsing based model using support vector machine, resulting in both slot and intent labels. Xu et al. [24] use CNN to extract features, and then a CRF layer and softmax are separately serving for two tasks. Along with this line, Guo et al. [25] use recursive neural networks (RecNNs) to model the semantic information with semantic parse tree support. Zhang et al. [26] firstly use Gated recurrent unit (GRU) as sharing layers to learn the representation of each word for these two tasks. Liu et al. [27] propose a series of encoder-decoder neural network models, which incorporate attention model to align the sequence using different policy for different tasks. The same authors also explore adding a new task (Language Modeling) for joint training in [28] using RNN.

We claim that the task of name entity recognition is different from the task of slot filling. For the latter, we can see slot filling as a fine-grained version of name entity recognition.

III. Domain Knowledge Guided Attention Model

In this section, we present our proposed models to better utilize the domain knowledge. More specifically, we propose an Entity Words Replacing mechanism applied to the word representation layer. We will show this simple replacing mechanism can speed up model convergence and ease the impact of lacking embedding of infrequently used words, which will be identified as unknown words in the most case. Finally, we give our attention model which uses the domain knowledge to guide the network where to attend for the classification task.

A. Long Short-term Memory Recurrent Network

Recurrent Neural Network (RNN) can be seen as a natural generalization of feed-forward neural networks to sequences. It takes sequence of vectors $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$ as input, and output another sequence of vectors $(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T)$ by iterating the following equation:

$$\mathbf{h}_t = f(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}) \tag{1}$$

$$\mathbf{y}_t = g(\mathbf{V}\mathbf{h}_t) \tag{2}$$

where $\mathbf{U}, \mathbf{W}, \mathbf{V}$ are parameters which share across the sequence. f and g are two nonlinear functions. $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$ are word embeddings in the text classification task which can be learned from Word2vec [6] model or using the pre-train embeddings provided by well-known organizations.

Using RNN to process text arises naturally due to its sequential characteristic. It has been widely applied to many text mining related tasks, such as machine translation [29], [30], image captioning [9], [29], and sequence tagging [19], [20].

Though with successful applications, it suffers the well-known problem, i.e. gradient vanishing and gradient exploding. They arise when back-propagates the derivative of the shared parameters through time. This makes the traditional RNN not capable of handling long-term dependencies appearing in text.

Long Short-term Memory network [31] is a variety of RNN. It was proposed to tackle above issues and extend the memory power of RNN. The ability of LSTM to block or pass on information owes to structures called gates, which are a way to optionally let information through. Gates can be implemented with element-wise multiplication by sigmoids, which are all in the range of $(0, 1)$. Specifically, LSTM uses three gates—input gate, forget gate and output gate—to protect and control the cell state. An input gate is used to protect the memory state from perturbation by irrelevant inputs. A forget gate is used to block the information stored in the cell state through the network which will make no contribution to following inputs. Likewise, an output gate is used to protect other units from perturbation by currently irrelevant memory state.

Each LSTM cell can be computed as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{3}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{4}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{5}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot tanh(\mathbf{W}_c \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \tag{6}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t) \tag{7}$$

where $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t$ are input gate, forget gate and output gate respectively. $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o$ and $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o$ are shared parameters to formulate the gates. σ is the sigmoid function to make the gates in the range of $(0, 1)$. $\mathbf{c}_t$ is cell state which store information updated along with the sequence. $\mathbf{W}_c, \mathbf{b}_c$ are weighted and biases parameters used for the cell state. $\odot$ is element-wise multiplication operator to control the information flow. $tanh$ is used to project the output in the range of $(-1, 1)$. $\mathbf{h}_t$ is hidden state which can be seen as the LSTM output of each time step.

In practical application, we can absorb bidirectional LSTM to learn the sequential information in the presence of whole text. In this scene, we can better utilize the past and the future information to balance the different impacts due to the location deviation. At each time step, we can get the final word representation $\mathbf{w}_t$ by concatenating [20] its left context representation $\mathbf{l}_t$ and right context representation $\mathbf{r}_t$, $\mathbf{w}_t = [\mathbf{l}_t; \mathbf{r}_t]$. To get the final text representation $\mathbf{t}$, we can use an aggregation function f_w to act on the final word representations. This aggregation function f_w can be concatenation, max pooling, and average pooling. A graphical illustration of BI-LSTM is shown in Fig. 4. Note the aggregation function is not shown in this graph.

$$\mathbf{s} = f_w(\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T) \tag{8}$$

B. Entity Words Replacing

In the Short-text classification task, the number of words is short, which means the information we can collect is scarce. In this scene, we face lots of difficulties to make machines comprehend natural language at a human level. These difficulties can be seen more clearly if there are lots of domain knowledge behind text. Furthermore, infrequently used words in text aggravate these difficulties since they would be identified as <UNK> in a typical dialog system. As Fig. 1 shows, the chatbot faces a big challenge to determine the user's intent in a typical dialog system if the utterance contains unrecognized entity words. However, we can use the entity type embeddings to replace the entity words embeddings. we claim that what contributes to classification is not the shallow entity words, but types the entities belong to in most text classification task.

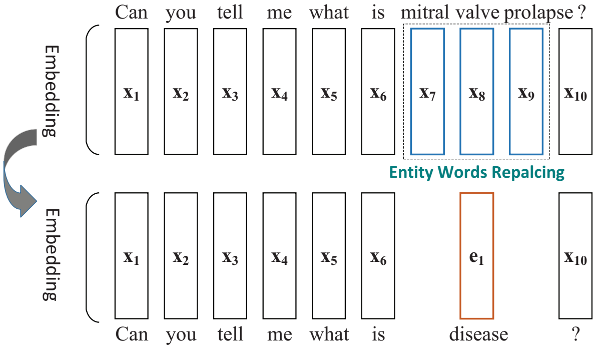


Fig. 2. Illustration of entity words replacing using corresponding entity type. In this figure, "mitral valve prolapse" is replaced with "disease"

Actually, when do domain-based classification task, we can easily collect the entity vocabulary thanks to the openness of Internet. In a sports types classification task, a player database can provide key information to identify which sports domain he/she belongs to. In a sentiment analysis task, the positive/negative vocabulary would guide the classifier to fastly attend the correct direction. Without this information, the classifier may consider the entity words in text as plain text and then misses the important indicators.

Fig. 2 is a graphical illustration of our entity words replacing policy. Given a sequence of word embeddings $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$, we replace the word embeddings corresponding to entity words with their entity type $(\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_m)$. Then the sequence turns into another sequence like this: $(\mathbf{x}_1, \cdots, \mathbf{e}_1, \cdots, \mathbf{x}_i, \cdots, \mathbf{e}_2, \cdots, \mathbf{x}_j, \cdots, \mathbf{e}_m, \cdots, \mathbf{x}_T)$. In the following, we will call this as Entity Words Replacing mechanism.

C. Attention Model

Though with some raised problem solved, we still can not take advantage of the domain knowledge behind the entity words from simple entity words replacing. To remedy this deficiency, we absorb attention mechanism into our framework to guide the network learning.

We denote another entity type embeddings in text as $(\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_m)$ which will be used for our attention mechanism, we firstly claim that these entity type embeddings

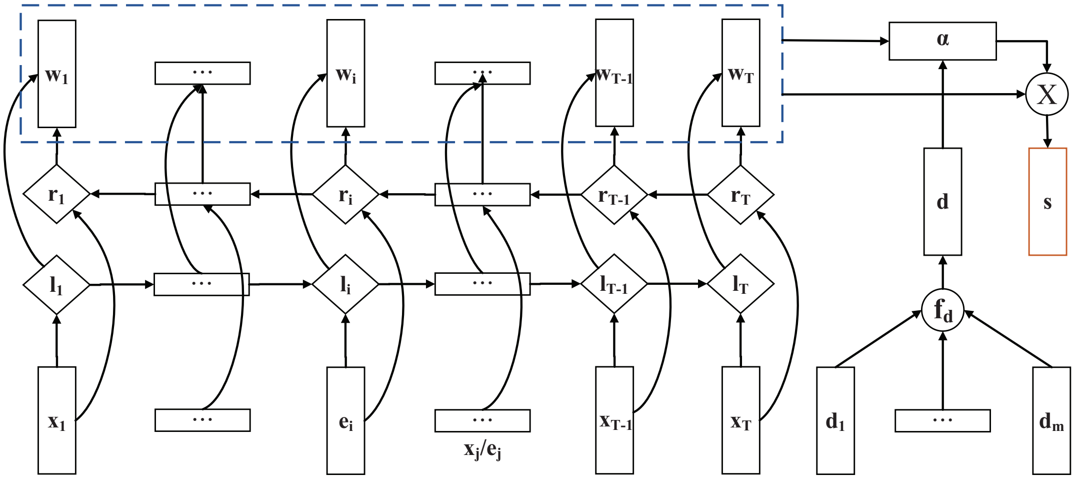


Fig. 3. Illustration of domain knowledge guided attention model (DKGAM). The entity type embeddings $\mathbf{e}_i, i \in (1, m)$ are used for entity words replacing, whereas the entity type embeddings $\mathbf{d}_i, i \in (1, m)$ are used for attention mechanism.

are different from the entity type embeddings used in Entity Words Replacing mechanism, they are two type embedding modalities of the same entities. The entity type embeddings $\mathbf{e}_i, i \in (1, m)$ for Entity Words Replacing are used to capture the semantic relation among text, whereas the entity type embeddings $\mathbf{d}_i, i \in (1, m)$ for attention mechanism are used to guide the network to attend key points. We apply a function f_d to the entity type embeddings $\mathbf{d}_i, i \in (1, m)$ to get the entity types representation $\mathbf{d}$ of the whole text.

$$\mathbf{d} = f_d(\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_m) \tag{9}$$

This function can be a feed-forward network, another LSTM, or element-wise addition. In application, we find that the element-wise addition is sufficient.

With the concatenation $\mathbf{w}_t$ of the output of BI-LSTM at hand, we explore the following attention mechanism to combine the entities representation with the context $\mathbf{w}_t$.

$$\mathrm{u}_t = \mathbf{v}^T tanh(\mathbf{W}_w \mathbf{w}_t + \mathbf{W}_d \mathbf{d}) \tag{10}$$

$$\alpha_t = \frac{\exp(\mathrm{u}_t)}{\sum_{i=1}^{T} \exp(\mathrm{u}_i)} \tag{11}$$

$$\mathbf{s} = \sum_{t=1}^{T} \alpha_t \mathbf{w}_t \tag{12}$$

where vector $\mathbf{v}$ and matrices $\mathbf{W}_w, \mathbf{W}_d$ are learnable parameters of the attention model. The length of vector $\mathbf{u}_t$ is the same as the input sequence, which denotes how important each word contributes to the task. The softmax function projects its value to the probability space and then enters into a weighted sum to get the final text representation $\mathbf{s}$.

With the text representation at hand, we can add a softmax layer to transform $\mathbf{s}$ to conditional probability distribution.

$$\mathbf{y} = \frac{\exp(\mathbf{W}_s \mathbf{s} + \mathbf{b}_s)}{sum(\exp(\mathbf{W}_s \mathbf{s} + \mathbf{b}_s))} \tag{13}$$

where $\mathbf{y}$ is the classification label, $\mathbf{W}_s, \mathbf{b}_s$ are the parameters for softmax layer.

Then cross-entropy loss can be employed as follows:

$$\mathcal{L}_{cross-entropy} = -\mathbf{t}^T \log(\mathbf{y}) \tag{14}$$

where $\mathbf{t}$ is one-hot encoding ground-truth labels.

D. Regularization of Entity Type Embeddings

Two entity type embeddings served as different role in the network framework. The entity type embeddings $\mathbf{e}_i, i \in (1, m)$ used for entity word replacing is adapted to remedy the impact of lacking embeddings of unrecognized entity words to utilize the available information more efficient and reduce the vocabulary size; Whereas the entity type embeddings $\mathbf{e}_i, i \in (1, m)$ used for attention mechanism is adapted to guide the network to attend most important point. Both entity type embeddings are learnable parameters. The former will be suited for context, so regularization is not needed. Since we expect the latter embeddings to be distinguishable, we add a cosine similarity regularization term on it:

$$\mathcal{L}_d = \sum_{i,j \in (0,m), i \neq j} \frac{\mathbf{d}_i^T \mathbf{d}_j}{\|\mathbf{d}_i\| \|\mathbf{d}_j\|} \tag{15}$$

This cosine similarity term measures the closeness between the entity word embeddings. Finally, we get the DKGAM model loss as follows:

$$\mathcal{L}_{DKGAM} = \mathcal{L}_{cross-entropy} + \lambda_1 \mathcal{L}_d \tag{16}$$

where λ_1 balances the importances of the regularization item.

IV. Multi-Task Domain knowledge Guided Attention Model

In real world application, there may not be domain entity dictionary at hand. In this section, we incorporate a NER model into our proposed framework for jointly training. And we give a comparison with series models of R-CNN which inspire this work.

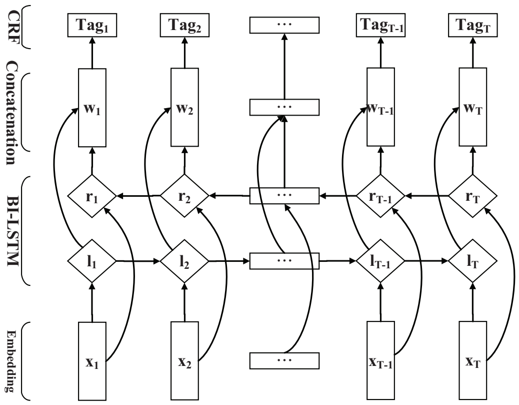


Fig. 4. Illustration of BI-LSTM-CRF model for NER. This model is incorporated into domain knowledge guided attention model Fig. 3 for multi-task training. The embedding layer and BI-LSTM layer are shareable for these two tasks.

A. Name Entity Recognition

Name entity recognition [19]–[21] is a fundamental task in text mining domain. It aims to locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, locations, etc. There are a number of neural network models proposed to address this problem. Recent success models always contain a conditional random field (CRF) [32] layer to account for both the joint probability of the entire sequence of labels given the observation sequence and transition scores between possible next states and the given current state. In name entity recognition task, the former consideration ensures the "global" consistency while the latter ensure the "local" correctness of consecutive transition which is urgent due to the tag scheme (e.g. B-ORG can not follow I-PER). We absorb BI-LSTM-CRF model proposed in [20] to identify entities in text.

Given the output $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T$ of BI-LSTM described above and the sequence tags $\mathrm{g}_1, \mathrm{g}_2, \cdots, \mathrm{g}_T$ of the text. we formulate a linear-chain CRF as follows:

$$P(\mathbf{g}|\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T) = \frac{1}{Z_w} \prod_{t=1}^{T} \psi_t(\mathrm{g}_t|F(\mathbf{w}_t))\psi_g(\mathrm{g}_{t-1}, \mathrm{g}_t) \tag{17}$$

where F is linear projection function which transforms the $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T$ into tag score space. ψ_t is a local function for the current word amounted to account for "global" consistency, whereas ψ_g is the transition function to account for "local" correctness. Z_w is a normalization term accounted for all the possible sequence tags.

We minimize following loss to encourage the correct sequence tags to have a high probability as follows:

$$\mathcal{L}_{ner} = -log(P(\mathbf{g}|\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T)) \tag{18}$$

When decoding, we can employ a Viterbi [20], [21] algorithm to determine sequence tags with maximum probability by maximizing Eq. 17 in this model.

B. Multi-task Model and Training Strategy

Next, we describe how to incorporate entity discovery into our short-text classification framework. We make the word embedding layer and BI-LSTM layer shareable for these two task and fine-tune task-specific parameters above the BI-LSTM layer. Since the performance of text classification rely on the correctness of the "entity proposals". We add a trade-off hyperparameter μ on the first equation in Eq. 10 to balance the importances of the "entity proposals" and hidden vectors.

$$\mathrm{u}_t = \mathbf{v}^T tanh(\mathbf{W}_w \mathbf{w}_t + \mu \mathbf{W}_d \mathbf{d}) \tag{19}$$

The final synthesis loss function of MT-DKGAM is a weighted sum of all the losses defined above.

$$\mathcal{L}_{MT-DKGAM} = \mathcal{L}_{cross-entropy} + \lambda_1 \mathcal{L}_d + \lambda_2 \mathcal{L}_{ner} \tag{20}$$

where λ_1 and λ_2 are weighting parameters.

Since entities discovery task is served for the text classification task, we need to notice the difference between training and testing. When training, our entities is at hand, so these two tasks can be jointly trained. However, in test time, we need firstly do the entity discovery task, so the result can be utilized by the classification task. This makes our work be similar to series models of R-CNN in object detection domain. In fact, they inspired us for this work, correlation with R-CNN will be discussed in Section IV-C.

We modified 4-Step Alternating Training strategy used in [33] to a more pragmatic 2-Step Alternating Training strategy to learn shared features and task-specific parameters via alternating optimization. In the first step, we train the entity discovery model as described in Section IV-A and classification network jointly. They use pre-train or random initialized word-embedding as input and fine-tuned random initialized parameters end-to-end for both tasks. In the second step, we fix the shared word embedding layer and BI-LSTM layer and only fine-tune the layers unique to respective tasks. When conduct experiments, we find this setting is especially suited for training with our method.

C. Correlation with Region-based Convolution Neural Network

Our work was inspired by region-based convolution neural network [18], [33] in object detection domain. Find the entities using fixed domain knowledge database in text can be seen as "entity proposals" corresponding to "region proposals" in R-CNN [18]. These "proposals" both appear out of the network training—one is proposed by dictionary lookup, another is proposed by selective search method. Training NER and Short-Text classification in a multi-task setting can be seen as the Multi-task training in Faster R-CNN [33], since the latter introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network which is the same as ours' "entity proposals" using name entity recognition method. The difference is they conduct image classification on the "region proposals", whereas we combine the "entity proposals" with the whole text to refine the

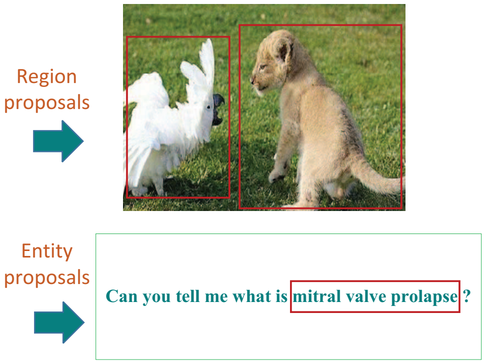


Fig. 5. Illustration of correlation with R-CNN. “entity proposals” is corresponding to “region proposals” in object detection domain.

classification performance. A graphical illustration is shown Fig. 5.

V. Experimental Setup

In this section, we present several experiments on two different datasets. The first dataset consists of questions collected from Chunyuyisheng[3], Xunyiwenyao[4] and Muzhiyisheng[5], which are three popular Chinese online healthcare consulting platforms (we refer it as COHCP in this paper). The second is the public available Air Travel Information Services (ATIS) dataset, which is widely used in Spoken Language Understanding. When conduct experiment for DKGAM method, we assume domain entity dictionary was at hand. While for MT-DKGAM method, we firstly recognize the name entities in text and then feed them into the attention mechanism for short-text classification. The results show that our methods outperform other existing off-the-shelf methods with many other performance gains.

A. Dataset

The detailed descriptions of the datasets are presented as follows:

COHCP: This dataset were collected from three popular Chinese online healthcare consulting platforms—Chunyuyisheng, Xunyiwenyao, and Muzhiyisheng. We totally crawled 100 million+ questions, which contains patient’s condition descriptions, drug counseling, the price of a clinical laboratory test, etc. We summarized the intent of these questions into 30 categories. With these categories, we random selected 20000 questions to assign each question a category. For questions whose intent not concluded in these 33 categories were skipped. After this, We got 1728 labeled questions. For some categories, there exist a few questions corresponding to, which means these question categories are not frequently asked. So we dropped the categories whose number of collected questions falls below a threshold. Finally, We got 1298 labeled questions belong to 7 categories which are also the patient most cared about. The details of question are summarized at Table I. To formulate the training set and test set, we split the whole questions at a ratio of 80/20.

TABLE I
The details of COHCP dataset.

Question Intents	Cardinality
Inquiry drug according to the disease	388
Symptoms of the disease	81
Consult department according to symptoms	78
The introductions of symptoms	150
The introductions of diseases	162
The price of a laboratory test item or examination item	113
Inquiry based on symptoms	326

TABLE II
The details of medical domain entity dictionary.

Entity Types	Cardinality
Disease	10303
Medicine	9197
Symptom	9429
Operation	6488
Laboratory test item and examination item	3148
Body parts	292
Departments	193

We collected 7 medical domain entity types from 39Jiankangwang[6] which is a China health portal. The details of the domain entity dictionary are summarized at Table II.

ATIS: This dataset is commonly used in Spoken Language Understanding, which contains audio recordings of people making flight reservations. We absorb it here to make sure our proposed method is scalable for other domains. There are some variants of this dataset, we follow the ATIS corpus setup used in [26]–[28]. The training set contains 4978 utterances from the ATIS-2 and ATIS-3 corpora, and the test set contains 893 utterances from the ATIS-3 NOV93 and DEC94 data sets. There are in total 127 distinct slot types, 44 distinct entity types and 18 different intent types.

In this work, we focus on using the domain entity information to refine the performance of short-text classification. To this end, we use the entity types in text rather than slot types as our domain entity knowledge. We observed that the cardinality of some entity types/intent types is small, we filters them if they are lower than a threshold number (25/15 in this work). We also observed that there exist 41 multi-label utterances, we filter them since this is not our focus in this paper. After that, there are 38 distinct entity types and 14 different intent types left. Besides, there exists discrepancies between the training set and test set, i.e. some entity types can only be found in test set. To address this problem, we rearrange the training set and test at a ratio of 80/20.

B. Implementation Detail

1) Pre-train Word Embedding: For COHCP dataset, we random sample 9,316,162 questions for training word em-

[3] http://www.chunyuyisheng.com/
[4] http://www.xywy.com/
[5] http://muzhi.baidu.com/
[6] http://www.39.net/

bedding. We absorb skip-gram architectures of Word2vec[7] proposed in [6] to learn vector representation of words. Since COHCP is a Chinese dataset, we conduct experiments on both character embedding and word embedding to determine which perform better. The result shows character embedding is better. Therefore, we use character embedding as input to all methods on COHCP dataset.

For ATIS dataset, word embedding is randomly initialized and fine tuned when the network training following [26], [27].

2) Model Parameters and Optimization: We use 100 as embedding size for word/character embedding, 128 hidden units for LSTM cell. Besides, we absorb Adam [34] optimization algorithm and fix the learning rate at 0.001. All models are trained on a single Nvidia M40 GPU.

3) Tagging Scheme for Name Entity Recognition: For the task of name entity recognition, text is usually labeled with the IOB format (Inside, Outside, Beginning). B-*label* denotes the token is the beginning of a name entity. I-*label* denotes the token is the inside of a name entity but not at beginning. O-*label* denotes the token is not a name entity. We absorb this tagging scheme for ATIS data set which is the same as [25], [27]. Since the length of a name entity in COHCP data set is longer than name entity in ATIS data set, we absorb IOBES tagging scheme for this data set, where the additional E-*label* denotes the token is the end of a name entity and S-*label* denotes the token is a singleton entity. [20] showed that the IOBES tagging scheme improves model performance marginally due to its expressive power.

C. Baseline Methods

We compare our proposed methods with many state-of-the-art text classification algorithms including:

Bag of Words/Bigrams with TFIDF [13]: Bag of Words/Bigrams are common baselines used in text classification problem. TFIDF (term-frequency inverse-document-frequency) take the length of text into consideration to get a better text representation. We report the best result among with/without Bag of Words/Bigrams and TFIDF using grid search provided by [35].

CNN [1]: A simple CNN-based model for sentence classification. It firstly embeds words into low-dimensional vectors and performs convolutions over them using multiple filter sizes. Next, a max-pooling layer is followed to get the vector representation with dropout regularization, and finally, classify the result using a softmax layer.

BI-LSTM: It employs a bidirectional LSTM on the word embeddings, and then conduct average pooling across the time step. Finally, feed the output of average pooling to a softmax layer.

Entity Words Replacing mechanism: We examine the effectiveness of Entity Words Replacing mechanism on the above CNN and BI-LSTM model.

Fasttext [2]: a simple and efficient baseline for text classification. It employs the n-gram features which are embedded and averaged into a text representation. The representation is in turn fed to a linear classifier.

TABLE III
ERROR RATE OF CLASSIFICATION RESULTS.

Model	COHCP	ATIS
Bag of Words/Bigrams with TFIDF	23.95	5.49
CNN [1]	15.92	2.32
BI-LSTM	19.78	2.58
Fasttext [2]	20.15	19.06
RCNN [3]	19.58	3.18
Attention Encoder-Decoder [27]	18.63	1.33
Entity Words Replacing mechanism (CNN)	14.45	1.46
Entity Words Replacing mechanism (BI-LSTM)	15.97	1.89
DKGAM	**13.57**	**1.20**
MT-DKGAM	18.25	1.80

TABLE IV
VOCABULARY SIZE.

Dataset	Normal	Entity Words Replacing	ratio
COHCP	1323	738	**1.79/1**
ATIS	714	472	**1.51/1**

RCNN [3]: It applies a recurrent structure to capture contextual information on word representations. The output concatenated with the word representations are then fed into a max-pooling layer.

Attention Encoder-Decoder [27]: It employs an encoder-decoder framework to joint learn the slot filling task and intent detection task. Attention mechanism was absorbed to focus on saliency points in encoder sequence when decoding.

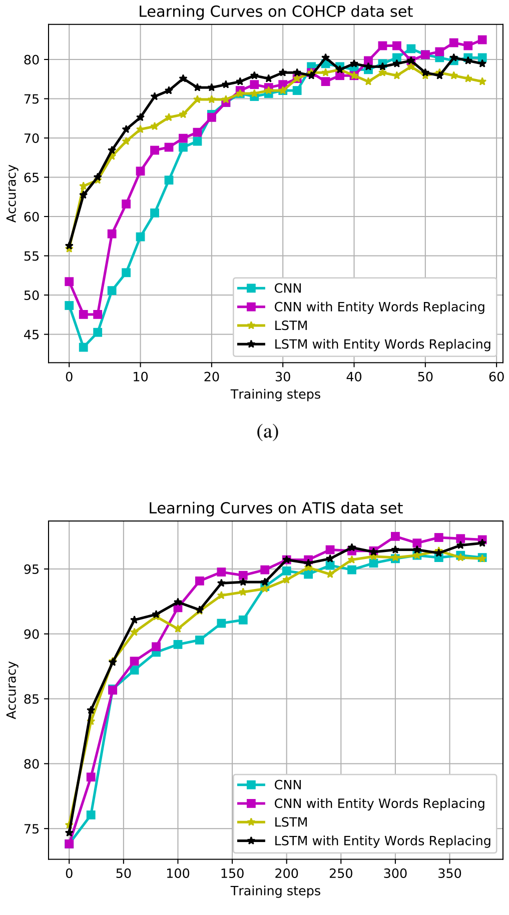
D. Results

Next, we will give a detailed result analysis to demonstrate the pros and cons of our method compared with others.

1) Classification Result Analysis:

- The result shows our method can outperform other methods on both datasets. Fasttext [2] can give an acceptable result which is a little bit worse than other neural network models on COHCP data, but the error rate on ATIS dataset is obviously higher than other neural network methods. It may be the bottleneck of the method though it can be efficiently trained. Attention Encoder-Decoder present a good performance which is on par with ours, we think it may benefit from the multi-task setting. In contrast, our method can better utilize the information.
- The result shows that our proposed Entity Words Replacing mechanism acted on CNN/LSTM can have a significant improvement than those models using usual embeddings. This verifies our motivation that the domain entity words in text can provide key information for short text classification problem. Since medical domain has lots of unusual concepts, which make the network hard to learn an acceptable embedding from the scarce data especially in short text context. Our further explanation will be presented in the following sections.

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



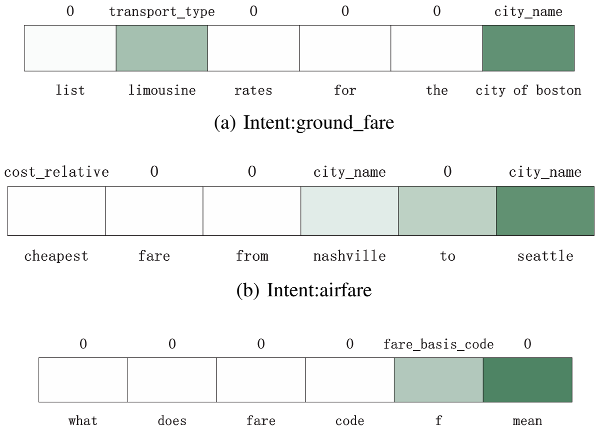
(a)

(b)

Fig. 6. Illustration of learning curves on two normal model (CNN/LSTM) with/without our proposed Entity Words Replacing mechanism. The result shows with the help of Entity Words Replacing mechanism, the convergence is accelerated compared orignal methods without Entity Words Replacing.

- MT-DKGAM model can not get a better result than DKGAM model. Since DKGAM can be seen as “hard” attention whose “entity proposals” is better than “entity proposals” of MT-DKGAM model which is learned from the data. But from another point of view, the MT-DKGAM model can provide by-product (name entities) which can be utilized by other tasks.

2) Effectiveness of Entity Words Replacing mechanism: Table IV shows the vocabulary size with/without using Entity Words Replacing mechanism on both data sets. We can see our proposed method can significantly reduce the vocabulary size, which can give a more reasonable memory usage when the corpus gets bigger and eliminate the meaningless words. To further explore the effectiveness of the proposed Entity Words Replacing mechanism and verify it can be effortlessly add to other models. We conduct experiments on two normal neural network models—CNN/LSTM—with Entity Words Replacing mechanism compared with those without. Fig. 6 shows the learning curve of these two model with/without Entity Words Replacing mechanism on both datasets. We can see Entity Words Replacing mechanism can speed up the model convergence, this may imply the entity type embedding after replacing provide a more valuable extracted information for learning.



(a) Intent:ground_fare

(b) Intent:airfare

(c) Intent:abbreviation

Fig. 7. Illustration of attention effects of DKGAM model. It shows our model is more inclined to attend to the entities in text which have been replaced by their type embeddings when learning.

3) Attention Effects: Fig. 7 shows some example text on ATIS dataset our DKGAM model attends. The result shows DKGAM model more inclined to attend to the entities in text which have been replaced by their type embedding when learning. This give another angle of view to see the effectiveness of Entity Words Replacing mechanism and our attention mechanism.

VI. Conclusion

Traditional classification methods can not utilize the domain knowledge behind text adequately. This problem can be seen more clearly when processing medical data since it is flooded with lots of difficult concepts. In this paper, we incorporate an Entity Words Replacing mechanism to remedy the impact of lacking embeddings of unrecognized entity words so as to utilize the available information more efficiently. And then proposed a domain knowledge guided attention model which aims to utilize the domain knowledge dictionary at hand to refine the classification performance. In real application, the domain knowledge dictionary may collect incompletely. In this scene, we develop a multi-task model to jointly learn the domain knowledge dictionary and do the classification task. The results show our methods can not only improve the classification accuracy but also provide useful embeddings for domain-based infrequently used words.

Acknowledgment

This work is sponsored by ”The Fundamental Theory and Applications of Big Data with Knowledge Engineering” under the National Key Research and Development Program of

China with grant number 2016YFB1000903; Project of China Knowledge Centre for Engineering Science and Technology; National Natural Science Foundation of China Innovation Research Team No. 61721002; Ministry of Education Innovation Research Team No. IRT_17R86; Project of China Knowledge Centre for Engineering Science and Technology; ”Multi-model based Patient similarity learning for medical data modelling and learning” under National Natural Science Foundation of China General Program; ”Reliability and Persistence of Big Data Hybrid Storage” under the National Key Research and Development Program of China with grant number 2016YFB1000303.

References

[1] Y. Kim, “Convolutional neural networks for sentence classification,” *CoRR*, vol. abs/1408.5882, 2014.

[2] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” *CoRR*, vol. abs/1607.01759, 2016.

[3] S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent convolutional neural networks for text classification,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 2015, pp. 2267–2273.

[4] X. Zhang, J. J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” 2015, pp. 649–657.

[5] J. Y. Lee and F. Dernoncourt, “Sequential short-text classification with recurrent and convolutional neural networks,” *CoRR*, vol. abs/1603.03827, 2016.

[6] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *CoRR*, vol. abs/1310.4546, 2013.

[7] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, and F. Wang, “Measuring patient similarities via a deep architecture with medical concept embedding,” in *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, 2016, pp. 749–758.

[8] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 2204–2212.

[9] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” *CoRR*, vol. abs/1502.03044, 2015.

[10] Y. Wang, M. Huang, X. Zhu, and L. Zhao, “Attention-based LSTM for aspect-level sentiment classification,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016, pp. 606–615.

[11] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. E. Hinton, “Grammar as a foreign language,” in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 2773–2781.

[12] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, “Semi-supervised recursive autoencoders for predicting sentiment distributions,” in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2011, pp. 151–161.

[13] S. I. Wang and C. D. Manning, “Baselines and bigrams: Simple, good sentiment and topic classification,” in *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, 2012, pp. 90–94.

[14] J. P. C. G. da Silva, L. Coheur, A. C. Mendes, and A. Wichert, “From symbolic to sub-symbolic information in question classification,” *Artif. Intell. Rev.*, vol. 35, no. 2, pp. 137–154, 2011.

[15] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, “Semantic compositionality through recursive matrix-vector spaces,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, 2012, pp. 1201–1211.

[16] T. Miyato, A. M. Dai, and I. Goodfellow, “Adversarial training methods for semi-supervised text classification,” *ICLR*, 2017.

[17] P. Li, L. He, X. Hu, Y. Zhang, L. Li, and X. Wu, “Concept based short text stream classification with topic drifting detection,” in *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, 2016, pp. 1009–1014.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Computer Vision and Pattern Recognition*, 2014.

[19] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” *CoRR*, vol. abs/1508.01991, 2015.

[20] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” *CoRR*, vol. abs/1603.01360, 2016.

[21] E. Strubell, P. Verga, D. Belanger, and A. McCallum, “Fast and accurate sequence labeling with iterated dilated convolutions,” *CoRR*, vol. abs/1702.02098, 2017.

[22] A. L. Gorin, G. Riccardi, and J. H. Wright, “How may I help you?” *Speech Communication*, vol. 23, no. 1-2, pp. 113–127, 1997.

[23] F. Mairesse, M. Gasic, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, and S. Young, “Spoken language understanding from unaligned data using discriminative classification models,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 4749–4752.

[24] P. Xu and R. Sarikaya, “Convolutional neural network based triangular crf for joint intent detection and slot filling,” in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec 2013, pp. 78–83.

[25] D. Guo, G. Tür, W. Yih, and G. Zweig, “Joint semantic utterance classification and slot filling with recursive neural networks,” in *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014*, 2014, pp. 554–559.

[26] X. Zhang and H. Wang, “A joint model of intent determination and slot filling for spoken language understanding,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 2016, pp. 2993–2999.

[27] B. Liu and I. Lane, “Attention-based recurrent neural network models for joint intent detection and slot filling,” in *Interspeech 2016*, 2016, pp. 685–689.

[28] ——, “Joint online spoken language understanding and language modeling with recurrent neural networks,” in *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, 2016, pp. 22–30.

[29] Z. C. Lipton, “A critical review of recurrent neural networks for sequence learning,” *CoRR*, vol. abs/1506.00019, 2015.

[30] M. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, “Multi-task sequence to sequence learning,” *CoRR*, vol. abs/1511.06114, 2015.

[31] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[32] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, 2001, pp. 282–289.

[33] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *CoRR*, vol. abs/1506.01497, 2015.

[34] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.