ICNet: Incorporating Indicator Words and Contexts to Identify Functional Description Information

Qu Liu[†‡], Zhenyu Zhang[†‡], Yanzeng Li[†‡], Tingwen Liu[‡*], Diying Li[¶], Jinqiao Shi[§‡]
[†] *School of Cyber Security, University of Chinese Academy of Sciences.* Beijing, China
[‡] *Institute of Information Engineering, Chinese Academy of Sciences.* Beijing, China
[§] *Beijing University of Posts and Telecommunications.* Beijing, China
{liuqu, zhangzhenyu1996, liyanzeng, liutingwen, shijinqiao}@iie.ac.cn
[¶]*DiDi Chuxing.* Beijing, China
lidiying@didiglobal.com

***Abstract*—Functional description information refers to the texts that describe the functionality or performance characteristics of a certain object. This type of information is of great potential value for the field of intelligence discovery. Thus automatically and accurately identifying this information from large amounts of texts on the web is very important. In this paper we reduce the functional description problem to a binary classification task deciding whether the input sentence is a functional description sentence or not. However, there exist lots of comment texts in the web data, which are semantically very similar to description texts, making our task quite difficult. Also, existing methods only provide general sentence representation models, which can't lead to targeted ways to solve our problem. Therefore, to address the problem, we not only exploit contexts, like many other previous work did, but also introduce indicator word information to learn rich representations. And in order to incorporate them both, we propose two models, namely ICNet(multi-tasks) and ICNet(ensemble). ICNet(multi-tasks) exploits them jointly in a integrated process of learning representations, while ICNet(ensemble) exploits them by two respective but concatenated sub-models. Experimental results on the collected real-world dataset indicate that both ICNet(multi-tasks) and ICNet(ensemble) achieve higher F1 scores compared with FaxtText, CNN, RNN, LSTM and Bi-LSTM, QuickThought models on this task.**

I. Introduction

Functional description information refers to the texts that describe the functionality or performance characteristics of a certain object. For example, a description text containing the details about some technical parameters or functional configurations of an industrial product. This type of information is often considered to be of great potential value for the field of intelligence discovery, especially when it involves business intelligence or national secrets, like descriptions about the competitive products or weapons of hostile countries from undisclosed official files.

There are many sources on the web that may contain functional description information such as tweets from twitter, posts from forums and articles from blogs. Thus automatically and accurately identifying this type of information from paragraphs or long articles is very important, which not only helps to gather intelligence efficiently but also saves people from large amounts of tedious reading work.

* Corresponding author: Tingwen Liu.

In this paper, we take sentences as the basic units of functional description information, and reduce the problem of identifying functional description information with a given text to a binary classification task, which is to predict whether the input sentence is a functional description sentence or not. There are following reasons: first, larger units like functional description paragraphs are too rough to be put into use in practical application because it still requires lots of manual work to find truly useful information among them; second, smaller units like functional description phrases are hard also unnecessary to pinpoint their start points and end points, because in most cases people will still have to look up to the adjacent words to verify whether they are truly exploitable. third, sentences are the most applicable units, and also, dividing texts into sentences can simplify the problem in an effective way.

In terms of sentence classification tasks, that learning meaningful sentence representations and then using them for classification have become common practices. Based on that, a lot of previous work have proved that contexts are exploitable in learning sentences' semantic representations [1], [2], and meanwhile introducing external knowledge can lead to more targeted ways of solving specific problems [3], [4]. For our task, although we regard sentences as the basic units of functional description information, it doesn't mean that the sentences are isolated from their contexts. Thus the contexts of sentences can't be neglected in learning representations. Moreover, having discovered that in the web data there exist lots of comment texts which semantically are very similar to description texts, we put forward indicator word features inside sentences (details can be seen in section II-B) as the external knowledge to deal with this problem.

However, the existing sentence classification methods provide no framework for learning sentence representations incorporating prior knowledge both from context sentences and from inside words simultaneously. In that case, we come up with two ideas. The first idea is to learn two representations utilizing contexts and indicator word information respectively and then concatenate them as the final one. And the second is to learn one representation but exploiting these two kinds of prior knowledge jointly at the same time. According to these,

after having referred to the existing general sentence representation models utilizing contexts, we adopt quick-thoughts (QT) proposed by [1] and, on the basis of that, propose two models, namely ICNet(ensemble) and ICNet(muti-tasks), to solve our problem. ICNet(ensemble), aligning with the first idea, on the one hand applies QT to get a representation making use of context sentences, and on the other hand converts the input sentence to a vector sequence containing both the category and location information of indicator words so as to combine them with a CNN model for getting the other representation exploiting indicator words. As for ICNet(muti-tasks), guided by the second idea, it changes QT so as to utilize contexts along with indicator word information simultaneously. Specifically, drawing inspirations from fast-sent, we add an extra training objective of using sentence representations to predict inside indicator words to the original QT so as to make the change.

To sum up, main contributions of our paper are as follows:

- We propose the task of identifying functional description information in light of their high intelligence value, and reduce it to a binary sentence classification task. For this new task, we collect science and technology news corpus and manually annotate 6107 samples as our dataset.
- in order to get good results on this classification task, we propose two different models, ICNet(multi-tasks) and ICNet(ensemble), which incorporate prior knowledge from both contexts and indicator words in learning rich sentence representations.
- Experimental results on the collected real-world dataset indicate that our two models both achive higher F1 values compared with FaxtText, CNN, RNN, LSTM, Bi-LSTM and QT models, and additionally, they can be used for different application scenario respectively according to different needs.

The rest of this paper is organized as follows. Section II states the preliminaries about QT approach and indicator words. Section III explains our two models in detail. Section IV introduces the build-up process of dataset and indicator word dictionaries for experiments. Section V presents the details about the configurations and results of the experiments. Section VI introduces related works regarding text classification tasks. Section VII draws a conclusion.[1]

II. Preliminaries

As mentioned above, the problem of identifying functional description information can be resolved by regarding it as a sentence classification task. Formally described, let $S = \{s_1, s_2, s_3, ..., s_n\}, L = \{l^+, l^-\}$, where S represents a piece of text data and s_i represents the i-th sentence in S. Label l^+ represents the functional description information category while label l^- represents the opposite. And if s_i belongs to a functional description sentence, it will be tagged with l^+, otherwise l^-.

[1]We will release our code and datasets after the paper is accepted.

Before explaining our proposed models, it's important to introduce the QT approach and indicator words first because they both play important roles in learning sentence representations in our methods. Section II-A and Section II-B are arranged to introduce them in detail respectively.

A. Quick-thoughts Model

Similar to lots of unsupervised sentence representation approaches, QT is also constructed based on the encoder-decoder model where an encoding function computes a vector representation of an input sentence, and then a decoding function attempts to generate the words of a target sentence conditioned on this representation. But QT gives up attempts to generate a context sentence given an input sentence, but alternatively replaces the decoder with a classifier which chooses the target sentence from a set of candidate sentences as a discriminative approximation to the generation problem. In this way, it facilitates using the meaning of current input sentence to predict the meanings of context sentences. With no need to reconstruct the surface form of context sentences, it is able to ignore the irrelevant aspects in constructing a semantic embedding space.

The general architecture of QT can be seen in Figure 3(a). Let s be a given sentence and S_{ctxt} denotes the set of sentences appearing in the context of s (for a particular context size) in the training data. For a given context sentence $s_{ctxt} \in S_{ctxt}$, let S_{cand} be the set of candidate sentences which contains the only valid context sentence s_{ctxt} and many other non-context sentences. The probability that a candidate sentence $s_{cand} \in S_{cand}$ is the correct sentence (i.e., appearing in the context of s) is given by

$$p(s_{cand}|s, S_{cand}) = \frac{exp[c(f(s), g(s_{cand}))]}{\sum_{s' \in S_{cand}} exp[c(f(s), g(s'))]} \quad (1)$$

where f and g denote parametrized functions that take a sentence as input and encode it into a fixed length vector, and c is a scoring function/classifier.

The training objective maximizes the probability of identifying the correct context sentences for each sentence in the training data D, which is

$$J_{qt} = \sum_{s \in D} \sum_{s_{ctxt} \in S_{ctxt}} \log p(s_{ctxt}|s, S_{cand}) \quad (2)$$

In this method, $c(u, v) = u^T v$. Additionally, both f and g are RNNs using gated recurrent units (GRU) as the RNN cell, but they have different parameters. The words of the sentence are sequentially fed as input to the RNN and the final hidden state is interpreted as a representation of the sentence.

As you can see from above, QT do not take the form of a binary classifier which takes a sentence window as input and classifies them as plausible and implausible context windows. Instead, it only requires ground-truth contexts to be more plausible than contrastive contexts. Also, c is simply defined to be an inner product for the reason that minimizing the number of parameters in the classifier encourages the encoders

Type	Examples
Function Indicator Word	直径 (diameter)；最大射程 (maximum range)；速度 (speed)
Entity Indicator Word	导弹 (missile)；歼-10 (j-10)；中国人民解放军 (People's Liberation Army)
Quantifier Indicator Word	千米(kilometers)；千克 (kilograms)；马赫 (Mach)
Numeral Indicator Word	一 (one)；5；

Fig. 1. Examples of function indicator words, entity indicator words, quantifier indicator words and numeral indicator words.

Type	Samples
Positive Samples	"雷神"长约12米，翼展约10米，重量超过4吨，是世界上最大的无人机之一。 With a length of about **12 meters**, a **wingspan** of about **10 meters** and a **weight** of more than **4 tons**, **"Thor"** is one of the largest drones in the world.
Positive Samples	朝鲜炮兵第一波打击用的是122毫米火箭炮，122毫米火箭炮是40管"冰雹"无控火箭弹。 The first round of attack by the north Korean artillery used a **122mm bazooka**, which is **40** hailstones unguided **missile**.
Negative Samples	2015年，中巴签订了一份潜艇合同，巴方一次购买8艘常规动力潜艇，成为中国海军装备出口的最大一个订单。 In 2015, China and Pakistan signed a **submarine** contract, in which Pakistan purchased 8 conventional powered **submarines** at a time, making it the largest order of Chinese naval equipment export.
Negative Samples	根据《国家利益》杂志2015年评选的结果来看，排名第一是俄罗斯图-160轰炸机，第二则是俄罗斯RS-24"亚尔斯"洲际导弹。 According to a ranking from The National Interest, the **Tupolev Tu-160** was ranked **No. 1** and the **RS-24 Yars** was placed as **No. 2**.

Fig. 2. Two representative positive samples and two representative negative samples selected from our dataset.

to learn disentangled and useful representations which avoids an undesirable solution where the model learns poor sentence encoders and a rich classifier to compensate for it. In the end, the concatenation of the outputs of the two encoders $[f(s)\ g(s)]$ is used for the final representation given sentence s.

B. Indicator Words

we spot that the functional description sentences usually tend to contain certain kinds of words, which can be summarized into 4 categories, namely (1) function indicator words, which express certain functions or performances, (2) entity indicator words, which indicate the described objects, and usually are proper nouns of some specific domains, (3) quantifier indicator words, and (4) numeral indicator words. The examples are presented in Figure 1.

As shown in Figure 2, we present two representative positive samples and two representative negative samples of our dataset, and the bold-faced words are their indicator words. The first sentence overall gives a functional description about "Thor", and the second sentence partially contains functional description about "bazooka". The third and the fourth sentence give no functional descriptions. After taking a careful look at these samples, we draw the following conclusions. First, usually functional description sentences tend to contain more indicator words both in numbers and in categories, thus it's of good benefits to pay attention to those indicator words in our models. Second, Indicator word information alone is not sufficient to predict functional description sentences. As you can see, the fourth sentence also contains many indicator words but it's a negative sample. Third, compared with functional description sentences, commentary sentences may rarely involve quantifier indicator words like the fourth sentence. In that case, indicator word information helps to distinguish them.

III. Our Methods

As mentioned above, we regard sentences as the basic units of functional description information and have transformed the identification problem into a sentence classification task. However, existing methods of general sentence representation models are not sufficient for obtaining a good classification result in our case. And that is owning to their ignorance of the characteristics of functional description sentences and incompetence of telling the small differences of semantics between our targets and commentary sentences very well. Therefore, to cope with that challenge, we exploit prior knowledge from both the contexts and the indicator words of input sentences to learn rich representations. And in order to incorporating them both, we propose two models ICNet(ensembel) and ICNet(multi-tasks). And the following Section III-B and Section III-A will introduce them respectively.

A. ICNet(multi-tasks) Model

The architecture of the ICNet(multi-tasks) can be seen from Figure 3(b). As you can see, ICNet(multi-tasks) is constructed based on QT, but It is able to utilize not only context sentences but also indicator words simultaneously for learning sentence representations. Drawing inspiration from fast-sent, which learns sentence representation by predicting words of adjacent sentences, we construct ICNet(multi-tasks) that incorporates indicator word information by increasing a new training objective of predicting the indicator words of input sentences. In other words, it not only uses the meaning of current input sentence to predict the meanings of adjacent sentences, but also to predict the meanings of its indicator words. Consequently, it's equipped with two training objectives. The first objective remains the same as that of QT, and the second training objective is explained as the followings.

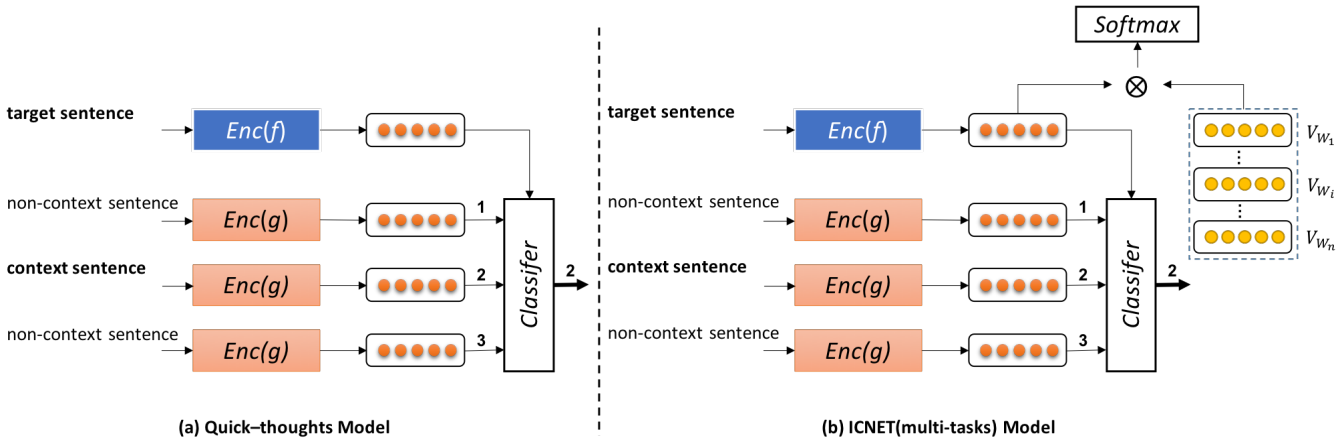

Fig. 3. (a) The Architecture of original QT, which aims to chooses the context sentence of target sentence from a set of candidate sentences. (b) The Architecture of ICNet(multi-tasks), which also needs to predicts the indicator words appearing in the target sentence.

For a given sentence s, let w be a word in c, then the probability of predicting w given s is:

$$p(w|s) = \frac{\exp(u_s^T \cdot u_w)}{\sum_{w' \in V_s} \exp(u_s^T \cdot u_{w'})} \tag{3}$$

where V_s is the set of words in s, u_w is the word embedding of w, and $u_s = f(s)$ is the output of encoder f as the sentence embedding of s. Thus, this training objective is to maximize the probability of predicting the correct indicator words for each sentence in the training data D, which is:

$$\sum_{s \in D} \sum_{w \in IV_s} \log p(w|s) \tag{4}$$

where IV_s is the set of indicator words in s.

It has to be mentioned that if we do not add constraints to this objective function, the final representations of sentence may degenerate into the representations of indicator words. Obviously this is not an ideal result, thus we modified the original objective function to alleviate this phenomenon:

$$J_{iw} = \max(\delta, \sum_{s \in D} \sum_{w \in IV_s} \log p(w|s)) \tag{5}$$

where δ is the upper threshold of J_{iw}. It means that when the value of J_{iw} exceeds δ, loss of current sample is no longer calculated.

In that case, the ultimate training objective of this model is to maximize:

$$J_{qt} + J_{iw} \tag{6}$$

In conclusion, the key thought of ICNet(multi-tasks) is to incorporate both contextual information and indicator word information at the same time. In that case, it learns sentence representation from those two aspects simultaneously. Given an input sentence s, first it is computed by an encoder function f as a vector representation $f(s)$, and then it is used as the inputs flowing into two parts for different uses. One part uses it to choose its adjacent sentence within the context window from a set of candidate sentences, while the other part is to predict the indicator words from all of its words. This way the representations are learned with rich semantics not only inferred from context sentences but also implied by important characteristics of sentences inside.

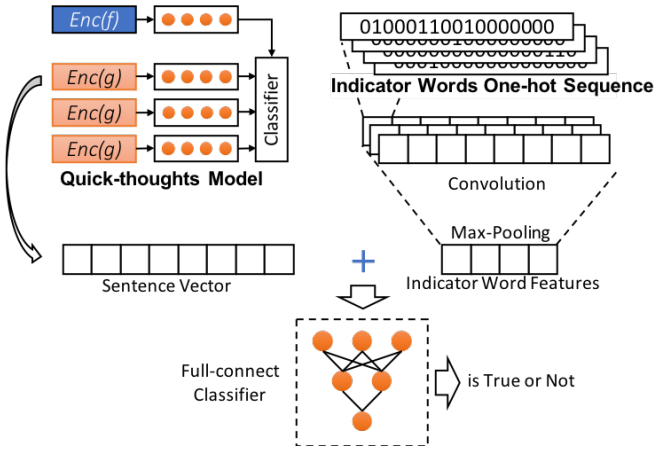

Fig. 4. The architecture of ICNet(ensemble) model. The left part is a original QT model and the right is a CNN model to encoder the sequence of indicator words in the target sentence. We concatenate the sentence vector and indicator word features to predict the target sentence is functional description or not.

B. ICNet(ensemble) Model

The architecture of the ICNet(ensemble) can be seen from Figure 4. As you can see, for a given sentence, its representation is the concatenation of two parts. One part is obtained from a pre-trained QT model, and the other is from a CNN model. These two parts learn different representations separately. Since the details of QT have been introduced in Section II-A, next we will focus on the CNN model to introduce how indicator words inside the input sentences are exploited here.

As illustrated in Section II-B, we have summarized four types of indicator words. For each type of them, we construct an independent dictionary. Note that the details about the construction of indicator dictionaries can be seen in Section IV. Formally described, let $D_1, D_2, ..., D_n$ be the n independent indicator dictionaries for n types of indicator words. In that case, given a word w, depending on whether it exists in some certain indicator dictionary, we can decide whether this word is an indicator word or not, and if yes then which category it belongs to. By that means, we can use a $(n+1)$-dimension word vector to denote this indicator word information. As shown in Table I, if w exists in D_i, the i-th entry of its vector will be set to 1 while the others are zeroes. Otherwise, the $(n+1)$-th entry will be set to 1 and the others will be zeroes at the same time.

For each input sentence, all of its words can be converted to this $(n+1)$-dimension vectors orderly. And then we use the sequence of these word vectors as inputs to CNN. In our case, this CNN is composed of one convolution layer and one max-pooling layer. By that means, CNN is able to further extract high-level statistical features regarding those indicator words inside the input sentences.

In conclusion, the key difference between ICNet(ensemble) and ICNet(multi-tasks) is that ICNet(ensemble) exploits contexts and indicator words separately by two sub-models while ICNet(multi-tasks) integrates them both in the same representation model. And that makes it more flexible on how we utilize these two kinds of prior knowledge. As you can see, ICNet(ensemble) is able to capture more information about the categories and the locations of indicator words inside the input sentences. The concatenation of two sub-models is very simple but also effective.

IV. Dataset

In this work, we collected science and technology news corpus of military category, which is obtained by crawling web-pages from two well-known Chinese forums: TieXue forum[2] and CJDBY forum[3], and both of them are the most popular online communities in China. A large number of people publish their posts and articles in these two web-forums every day, which includes large quantities of reposts of science and technology news. Compared with other social media like WeiBo and Twitter, the text contents published in these forums are more formal, and more suitable for experimentation.

We collected 34765 related posts in recent five years from the two target forums. 218513 sentences in total from those posts are utilized as the unlabelled datasets to train ICNet(multi-tasks) and QT of ICNet(ensemble). After the web-page parsing and filtering, we manually annotated 6107 samples for the sentence classification task, which includes 3073 positive samples and 3034 negative samples.This dataset is further divided into two parts: 80% samples are used for training the classifier and 20% for testing.

[2]http://bbs.tiexue.net

[3]https://lt.cjdby.net

TABLE I
Examples of Indicator Words

Word	Type	Word vector
altitude	Function Indicator Word	$(1, 0, 0, 0, 0)$
rifle	Entity Indicator Word	$(0, 1, 0, 0, 0)$
kilogram	Quantifier Indicator Word	$(0, 0, 1, 0, 0)$
three	Numeral Indicator Word	$(0, 0, 0, 1, 0)$
write	Other Word	$(0, 0, 0, 0, 1)$

In order to build the indicator word dictionaries, we crawled a lot of wikipedia entry pages under the military category [4]. and extracted entity words like weapons, function words, and quantifier words from those web-pages as indicator words. In total, we get 475 entity indicator words, 50 function indicator words, 44 quantifier indicator words.

A. Experimental Setup

V. Experiments

In this section, we first give a description of the experimental setup, including baseline models, evaluation metrics and experimental environment. Then we show our experimental results to demonstrate the effectiveness of our models compared with baseline models, the effectiveness of incorporating indicator word information and the effectiveness of incorporating contextual information. In the end, we give the comparison of our two models and state their different application scenarios

1) Baselines and Evaluation Metrics: In this paper, we introduce five baseline models, namely FaxtText, CNN, RNN, LSTM and Bi-LSTM, and compare them with our two models. Specially, FastText [5] is a library created by the Facebook Research Team for efficient learning of word representation and sentence classification. FastText combines some of the most successful concepts introduced by the natural language processing and machine learning communities in the last few decades. These include representing sentences with bag of words and bag of n-grams, as well as using sub-word information, and sharing information across classes through a hidden representation. FastText also employs a hierachical softmax that takes advantage of the unbalanced distribution of the classes to speed up computation. Many experiments show that FastText is often on par with deep learning classifiers in terms of accuracy, and many orders of magnitude faster for training and evaluation.

We evaluate our work with three metrics: precision, recall and F1-measure, which are the most widely used metrics in text classification tasks. The precision rate is defined as the proportion of predicted positive samples that are really functional description sentence, while the recall rate is the proportion of real functional description sentence that are correctly predicted. The F1-measure is the harmonic mean of the precision rate and the recall rate. These are also standard evaluation metrics in text classification tasks.

[4]https://www.wikipedia.org

TABLE II
COMPARISION WITH BASELINE METHODS

	Prec.	Rec.	F1
FastText	0.8059	0.7941	0.8000
CNN	0.8169	0.7483	0.7812
RNN	0.8158	0.8207	0.8182
LSTM	0.8088	0.8432	0.8256
Bi-LSTM	0.8197	0.8392	0.8293
CNN(iw)	0.7762	0.6970	0.7345
QT	0.7892	0.8639	0.8250
ICNet(ensemble)	**0.8460**	0.8501	0.8481
ICNet(multi-tasks)	0.8167	**0.8884**	**0.8510**

2) Parameter Settings: We implement our models with PyTorch v0.4.0, and optimize hyperparameters with annealing algorithm built in hyperopt library [6]. All the experiments in this paper are conducted on a physical server which is configured with two Nvidia Tesla P100 GPU cards and 24 GB memory. The physical server provides the ability to present high parallelism performance of CNNs due to cuDNN primitives [7]. In this paper, jieba segmentation [8] is employed to divide the sentences into words.

For the vector representation, we set the dimension of word embedding to 128, which is randomly initialized with uniform distribution between $[-1, 1]$ and learned by model training. The dimension of sentence embedding is set to 256. For the QuickThought in ICNet(ensemble) and ICNet(muti-tasks), the number of RNN units is 256, the batch size is 256, and the learning rate is $5e^{-4}$. For the CNN module in ICNet(ensemble), it has 128 filter kernels with a size of 5, and takes indicator feature vectors of 5 dimensions as inputs as mentioned above, and the learning rate is $1e^{-3}$. The upper threshold δ of J_{iw} is set to 0.5 heuristically. Following [9], the probability of dropout is set to 0.5 to prevent overfitting. All the input sequences are padded and truncated to a fixed words-length of 60, each experiment are enacted ten times and the average is taken as the final result.

A. Experiment Results

1) Comparison with Baseline Methods: The experimental results are shown in Table II. It is easy for us to draw following conclusions. First, the precision, recall and F1 score of our baseline methods are all close to 0.8000, which is very good in terms of real-world dataset. Second, our two models achieve the highest two F1 scores, 0.8510 and 0.8481, compared with other models, which validates the effectiveness of our methods.

By closely looking at CNN along with RNN models including RNN, LSTM, Bi-LSTM, we can find out that CNN obtains a higher precision value while RNN models achieve higher recall value. That is probably because that CNN is focused on extracting high-level statistical features of input sentences while RNN models pay much attention to the semantics. CNN tends to find the patterns of functional description sentences accurately so that it has a relative higher precision but lower recall. And RNN models are more good at learning sentences' semantics but without introducing indicator words they may mistakenly recall those commentary sentences, thus they get higher recalls but lower precisions. Also, no matter CNN or RNN models, they both learn representations without using contexts and indicator words thereby achieve lower F1 socres compared with our models.

TABLE III
EFFECTIVENESS OF CONTEXTUAL INFORMATION

	Prec.	Rec.	F1
RNN	0.8158	0.8207	0.8182
RNN-RNN	0.8216	0.8373	0.8294
RNN-QT	**0.8382**	**0.8756**	**0.8565**

2) Effectiveness of Indicator Words: In order to observe the effectiveness of indicator words, on one hand, we remove QT from ICNet(ensemble) and use the left CNN (referred as CNN(iw)) to test the classification result. It turns out that its F1 score reaches 0.7345. Although it's the lowest but it's much higher than the random guessing rate 0.5. This result validates the effectiveness of indicator words but also indicate that indicator word information alone is not sufficient for learning rich semantics. Also on the other hand, we compare the experimental results of our two models with that of QT respectively. As shown in Table II, ICNet(ensemble) improves 2.31%, and ICNet(multi-tasks) improves 2.60%, compared with original QT model. These manifest that despite the different ways of exploiting them, indicator words helps to improve the classification results significantly.

3) Effectiveness of Contextual Information: In order to observe the effectiveness of contextual information, we put a control experiment. The first thing to note is that we take the experimental result of RNN as the baseline. Then, in order to observe the classification result after incorporating contextual information, we test the experimental result on the RNN-QT model, which obtains sentence representations simply by concatenating the respective outputs of RNN and QT. In the mean time, in order to rule out the effects of dimension increase, we test the result of the RNN-RNN model for comparison, which is constructed simply by replacing QT with a same RNN. As shown in Table III, the RNN-QT model is 1.66% higher in precision, 3.83% higher in recall and 2.71% higher in F1. Apparently, contextual information contributes to learning better representations and promoting the experimental results, which is consistent with the results of previous works. Specifically in our case, we argue that it's possibly because in practice not all sentences are very clear in semantics when we look at them separately, and with contextual information we can estimate them more easily.

4) Comparison of our two models: Last but not least, we make a comparison between our two models, namely ICNet(ensemble) and ICNet(multi-tasks). It's easy to discover that ICNet(multi-tasks) is 0.29% higher in F1 score than ICNet(ensemble), which demonstrates that ICNet(multi-tasks) utilizes contexts and indicator words in a better way during the process of learning representations. That is possibly because

although ICNet(ensemble) is able to exploit more information about indicator words with CNN, it fails to integrate two representations very well with a simple concatenation operation. However, both two models can be of use in different application scenarios. As you can see, ICNet(multi-tasks) achieves a high recall with 0.8884. If we care less about precision and want to find functional description sentences as much as possible since we will manually verify the results later, ICNet(multi-tasks) is a better choice. But if we don't want the subsequent manual filtering procedure, normally we will expect a relative more balanced result about precision and recall, and in that case, ICNet(ensemble) is a better choice.

VI. Related Work

The related work can be roughly divided into two groups: text classification and sentence representation.

A. Text classification

The sentence classification problem in this paper is a typical text classification problem. Text classification refers to the activity of labelling natural language texts with thematic categories from a predefined set, and it is one kind of classic tasks in NLP field. The research process can be roughly divided into three phases with the advancement. The earliest methods were brought up based on pattern [10]–[12], which automatically classified the text with the help of rules. These methods are highly dependent on rules and usually can not deal with complicated problems.

Later, machine learning methods have been applied to deal with text classification problems. Those methods usually involve two procedures, namely feature engineering and classification algorithms. In terms of feature representation, the most commonly used method is the vector space model [13], also known as the bag-of-words model. It is an algebraic model that represents text as an index vector. Other features such as part-of-speech [14] and noun phrases [15] have also been introduced. The vector space model, despite its convenience for computer processing, brings about a problem of high dimension and sparsity concerning features. In that case, lots of feature selection methods like mutual information [16] and information gain [17] and feature extraction methods like principal component analysis [18], have been introduced to reduce feature dimension. As for classification algorithms, many of them have been successfully applied to the text classification problem, such as SVM [13], naive bayes [19], decision tree [20], random forest [21], Rocchio [22], etc.

In recent years, with the rise of various text representation models, deep learning models have made some progress on text classification tasks [23]. For examples, [24] proposed using CNN for sentence classification. [25] proposed using RNN for text classification with multi-tasks learning. Our baseline model FastText [5] extended word2vec [26] for representation learning and text classification. Furthermore, most deep learning methods learned text representations through neural language models [27]–[29].

B. Sentence Representation

Learning meaningful sentence representations is the first step towards the goal of language understanding, which has attracted a significant amount of research attention. Recently, many sentence representation approaches based on encoder-decoder models have been proposed. For examples, [2] proposed the skip-thought vectors model, which were composed of an encoder RNN mapping the input sentence into a vector representation and a decoder RNN that sequentially predicts the words of adjacent sentences. [1] proposed quick-thoughts which consisted of a RNN encoder and changes the decoder to a classifier choosing context sentence from a set of candidate sentences. [30] explored the use of convolutional neural network (CNN) encoders. Their base model used a CNN encoder and an RNN decoder reconstructing the input sentence as well as neighboring sentences. The hierarchical version of their model sequentially reconstructed sentences within a larger context.

Beyond that, an advantage of auto-encoders over context prediction models is that they do not require ordered sentences for learning. [31] bought up a de-noising auto-encoder model (SDAE) where noise was introduced in a sentence by deleting words and swapping bi-grams and the decoder is required to reconstruct the original sentence. [32] learn bag-of-words representations of sentences by considering a conceptually similar task of identifying context sentences from candidates and evaluate their representations on sentence similarity tasks. [31] introduced the FastSent model which uses a BoW representation of the input sentence and predicts the words appearing in context (and optionally, the source) sentences. The model is trained to predict whether a word appears in the target sentences. Meanwhile, [4] demonstrated that focusing on importance words can enhance semantic representations of sentences. [33] applied weighted summation of context words to predict the target word in learning word representations. Inspired by this work, we propose ICNet(multi-tasks) model, which predicts the indicator words appearing in the target sentences.

VII. Conclusion

In this work, we put forward a problem of identifying functional description information from given texts. Here this target information refers to texts describing the functionality or performance characteristics of a certain object. Note that sentences are the most applicable basic units of functional description information, we reduce the problem to a binary sentence classification task. In order to deal with the difficulty of distinguishing description texts from comment texts with similar semantics in our problem, we introduce indicator words as the external knowledge. In the mean time, context sentences are also exploited like many other previous work did. For the purpose of incorporating them both to learn rich representations, we propose two models, namely ICNet(multi-tasks) and ICNet(ensemble). The key difference between them is that ICNet(multi-tasks) exploits them jointly in a integrated process of learning representations, while ICNet(ensemble)

exploits them by two respective but concatenated sub-models. Experimental results on the collected real-world dataset indicate that both our ICNet(multi-tasks) and ICNet(ensemble) achieve higher F1 scores than baseline methods on this task. These results well demonstrate the effectiveness of our models. Additionally, in terms of other sentence classification tasks that also have contexts to use and pay attention to partially important words inside sentences as well, our models are of reference value.

ACKNOWLEDGMENTS

The authors would like to thank all the reviewers for their insightful and valuable suggestions, which significantly improve the quality of this paper. This work was supported in part by the National Key Research and Development Program of China under Grant No.2016YFB0801003.

REFERENCES

[1] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," *arXiv preprint arXiv:1803.02893*, 2018.

[2] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015, pp. 3294–3302.

[3] Q. Chen, X. Zhu, Z.-H. Ling, D. Inkpen, and S. Wei, "Neural natural language inference models enhanced with external knowledge," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 2406–2417.

[4] S. Wang, J. Zhang, and C. Zong, "Learning sentence representation with guidance of human attention," *arXiv preprint arXiv:1609.09189*, 2016.

[5] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, April 2017, pp. 427–431.

[6] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: a python library for model selection and hyperparameter optimization," *Computational Science & Discovery*, vol. 8, no. 1, p. 014008, 2015.

[7] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cudnn: Efficient primitives for deep learning," *Computer Science*, 2014.

[8] S. Junyi, "Jieba segmentation," https://github.com/fxsjy/jieba, accessed December 14, 2018.

[9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[10] T. Joachims, "Transductive inference for text classification using support vector machines," in *Sixteenth International Conference on Machine Learning*, 1999, pp. 200–209.

[11] E. Riloff, "Little words can make a big difference for text classification," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1995, pp. 130–136.

[12] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in Kernel Methods*, 1999.

[13] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[14] B. Rink and S. Harabagiu, "Utd: Classifying semantic relations by combining lexical and semantic resources," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2010, pp. 256–259.

[15] I. Heim, "The semantics of definite and indefinite noun phrases," 1982.

[16] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[17] Y.-C. Lu, M.-Y. Lu, F. Li, and L.-Z. Zhou, "Analysis and construction of word weighing function in vsm [j]," *Journal of Computer Research and Development*, vol. 10, p. 006, 2002.

[18] I. Jolliffe, "Principal component analysis," in *International encyclopedia of statistical science*. Springer, 2011, pp. 1094–1096.

[19] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1. Citeseer, 1998, pp. 41–48.

[20] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[21] Q. Wu, Y. Ye, H. Zhang, M. K. Ng, and S.-S. Ho, "Forestexter: an efficient random forest algorithm for imbalanced text categorization," *Knowledge-Based Systems*, vol. 67, pp. 105–116, 2014.

[22] T. Joachims, "A probabilistic analysis of the rocchio algorithm with tfidf for text categorization." Carnegie-mellon univ pittsburgh pa dept of computer science, Tech. Rep., 1996.

[23] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[24] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[25] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[27] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[28] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[29] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in neural information processing systems*, 2009, pp. 1081–1088.

[30] Z. Gan, Y. Pu, R. Henao, C. Li, X. He, and L. Carin, "Unsupervised learning of sentence representations using convolutional neural networks," *arXiv preprint arXiv:1611.07897*, 2016.

[31] F. Hill, K. Cho, and A. Korhonen, "Learning distributed representations of sentences from unlabelled data," in *Proceedings of NAACL-HLT*, 2016, pp. 1367–1377.

[32] T. Kenter, A. Borisov, and M. de Rijke, "Siamese cbow: Optimizing word embeddings for sentence representations," *arXiv preprint arXiv:1606.04640*, 2016.

[33] W. Ling, Y. Tsvetkov, S. Amir, R. Fermandez, C. Dyer, A. W. Black, I. Trancoso, and C.-C. Lin, "Not all contexts are created equal: Better word representations with variable attention," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1367–1372.