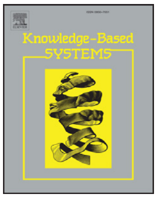
Knowledge-Based Systems 142 (2018) 149–159

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Response selection from unstructured documents for human-computer conversation systems

Zhao Yan[a,*], Nan Duan[b], Junwei Bao[c], Peng Chen[d], Ming Zhou[b], Zhoujun Li[a]

[a] *Beihang University, Beijing, China*
[b] *Microsoft Research Asia, Beijing, China*
[c] *Harbin Institute of Technology, Harbin, China*
[d] *Microsoft Xiaoice Team, Beijing, China*

ARTICLE INFO

Article history:
Received 19 June 2017
Revised 23 November 2017
Accepted 25 November 2017
Available online 2 December 2017

MSC:
00-01
99-00

Keywords:
Human-computer conversation system
Human-robot interaction
Text matching
Natural language processing
Artificial intelligence

ABSTRACT

This paper studies response selection for human-computer conversation systems. Existing retrieval-based human-computer conversation systems are intended to reply to user utterances based on existing utterance-response pairs. However, collecting sufficient utterance-response pairs is intractable in practical situations, especially for many specific domains. We introduce DocChat a novel information retrieval approach for human-computer conversation systems that can use unstructured documents rather than semi-structured utterance-response pairs, to react to user utterances. The key of DocChat is a learning to rank model with features designed at various levels of granularity which is proposed to quantify the relevance between utterances and responses directly. We conduct comprehensive experiments on both sentence selection and real human-computer conversation scenarios. Empirical studies of sentence selection datasets shows reasonable improvements and the strong adaptability of our model. We compare DocChat with Xiaoice, a famous open domain chitchat engine in China. Side-by-side evaluation shows that DocChat is a good complement for human-computer conversation systems using utterance-response pairs as the primary source of responses. Furthermore, we release a large scale open-domain dataset for sentence selection which contains 304,413 query-sentence pairs.

1. Introduction

Constructing human-computer conversation systems that can converse meaningfully with humans using natural language has been a long-standing goal of Artificial Intelligence. Human-computer conversation systems can be broadly divided into two categories: task-oriented systems which work in vertical domains to assist people to achieve specific goals, such as travel arrangements [1], restaurant recommendations [2], online shopping [3], and chat-oriented systems which focus on talking like a human and engaging in social conversation regarding a wide range of issues. Recent previous research has focused on task-oriented systems, with a large amount of conversation data being gained through social media websites (e.g., Twitter and Weibo) and community question answering (CQA) websites (e.g., Yahoo Answers and Baidu Zhidao), resulting in chat-oriented systems becoming a major focus of both academia and industry.

Existing approaches to designing a chat-oriented system fall into two categories. (i) Generation-based methods [4,5] usually leverage an encoder-decoder framework to generate a response $\hat{R}$ based on an utterance Q, where $\hat{R}$ can be a word sequence that has never been seen before. Lack of fluency and naturality is one drawback of generation-based methods. Another disadvantage is they tend to generate safe but boring responses such as “I don’t know” and “me too”. (ii) Retrieval-based methods [6,7] retrieve candidate utterance-response (or Q-R)[1] pairs from a pre-built index, rank the candidates, select the top ranked $\langle\hat{Q},\hat{R}\rangle$ pair, and then use $\hat{R}$ as Q’s response. The capability of retrieval-based methods depends heavily on existing Q-R pairs. However, collecting such Q-R pairs is intractable for many specific domains.

* Corresponding author.
E-mail addresses: yanzhao@buaa.edu.cn (Z. Yan), nanduan@microsoft.com (N. Duan), peche@microsoft.com (P. Chen), mingzhou@microsoft.com (M. Zhou), lizj@buaa.edu.cn (Z. Li).

[1] For convenience sake, we denote all utterance-response pairs (either QA pairs or conversational exchanges from social media websites like Twitter) as Q-R pairs in this paper.

https://doi.org/10.1016/j.knosys.2017.11.033
0950-7051/© 2017 Elsevier B.V. All rights reserved.

To overcome the aforementioned issues, we propose DocChat, a response selection method to retrieve responses from documents. It is easier to retrieve responses from unstructured documents than to collect semi-structured Q-R pairs. Using documents rather than Q-R pairs can greatly improve the adaptability of human-computer conversation systems. Conversely, responses retrieved from existing human generated documents are guaranteed to have their fluency and naturality. We separate the response selection approach into three cascaded steps to make a trade-off between efficiency and accuracy. In the first step (i.e., response retrieval), the proposed method finds a small set (e.g., 50 or 100) of candidate sentences based on a basic similarity measurement. In the second step (response ranking), a learning to rank model with features at four different levels of granularity is used to measure the semantic relevance between an utterance and a candidate response. The biggest challenge in utterance-response matching is that users often express similar meanings with different but semantically related expressions. Machine translation-based methods can help bridge such lexical gaps by extracting synonym words and phrases from parallel corpus such as bilingual sentence pairs. We present representation learning-based models, both sentence and semantic-levels. In the last step (i.e., response triggering), a triggering strategy determines whether the top-ranked sentence is a sufficient response.

We conduct comprehensive experiments on both real human-machine conversation scenarios and sentence selection benchmarks. Side-by-side evaluation between DocChat and a famous chatbot demonstrates that DocChat performs better on domain related queries. In addition, the performance of the proposed method is comparable to that of state-of-the-art methods on sentence selection task.

The primary contributions of this study are as follows.

- We propose DocChat, an information retrieval approach for human-computer conversation systems. DocChat can leverage unstructured documents to respond to utterances. The proposed method is based on learning to rank, that ranks candidate sentences using well-designed matching features at different levels.
- We evaluate DocChat on both answer sentence selection datasets and response sentence selection datasets. The results show that our method performs comparably with state-of-the-art methods. A comparison of DocChat and Xiaoice demonstrates that DocChat is a good complement to human-computer conversation systems using Q-R pairs as the source of responses.
- We create a large scale manually labeled corpus for sentence selection, which is released to the public domain.[2]

2. Related work

Our work is related to human-computer conversation systems and answer sentence selection.

2.1. Human-computer conversation system

Previous studies into human-computer conversation systems [8–11] rely on handcrafted patterns or learning-based approaches, which suffer from coverage issues and require vast resources devoted to in designing human selected rules. With the rapid development of social media, such as the microblog and CQA services, large-scale conversation data and data-driven approaches make it possible to solve the problem. Existing strategies to design a human-computer conversation system can be categorized as: (1) generation-based methods [12–14] that generate responses based on previous utterances; (2) retrieval-based methods [15–17] that select responses from a large corpus.

Typically, generation-based methods leverage an encoder-decoder framework to generate a response. Ritter et al. [4] train a statistical machine translation (SMT) model using large-scale human-human conversation data and use it as a response generator. Grammatical and fluency problems are the primary issue for such machine translation-based methods. Recently, neural network-based methods have been widely studied because of their capability to capture syntactic and semantic relations in an end-to-end manner. Vinyals and Le [18] propose a sequence-to-sequence framework where in the post utterance is encoded by a recurrent neural network (RNN) and its vector representation is used as raw input to another RNN to generate a response. Shang et al. [5] improve the Seq2Seq method using an attention mechanism to identify important parts within and among utterances. However, Seq2Seq based methods tend to generate safe but trivial responses, such as “I don’t know”, “I see” and “me too” [19].

Retrieval-based methods select the most suitable response to the current utterance from a large number of Q-R pairs. ARC-I and ARC-II [20] propose a convolutional neural network (CNN) based method to align sentences pairs word by word. Ji et al. [6] built a human-computer conversation system using learning to rank and semantic matching techniques. Wang et al. [21] present a deep neural network model using tree structures as the input. However, collecting enough Q-R pairs is often intractable in many domain specific tasks.

Compared to previous methods, DocChat learns the internal relationships between utterances and responses based on statistical models at different levels of granularity and relaxes the dependency on Q-R pairs as response sources. Such features make DocChat a generation solution for human-computer conversation systems with high adaptation capability.

2.2. Answer sentence selection

Our work also relates to a line of research works on answer sentence selection. Prior work in measuring the relevance between question and answer focuses mainly on the syntactic level by matching parse trees. Wang et al. [22] present a probabilistic model to learn tree-edit operations on a dependency parse tree. Tree kernel as a heuristic [23] and dynamic programming [24] are used to search for the minimal edit sequences between parse trees. Unlike previous work, LCLR [25] applies rich lexical semantic features that obtained from a wide range of linguistic resources including WordNet, the polarity-inducing latent semantic analysis (PILSA) model and different vector space models. Learning representation by neural network architecture has become a hot research topic beyond word-level or phrase-level methods. Convolutional neural networks (CNN) [26,27] and recurrent neural networks (RNN) [28] are used to encode questions and answer sentences into a semantic vector space. NASM [29] uses an enriching long short term memory (LSTM) with a latent stochastic attention mechanism to model similarity between Q-R pairs. AB-CNN [30] is an attention-based CNN which calculates a similarity matrix and takes it as a new channel of the CNN model. Tan et al. [31] introduce a two-way attention mechanism which can be used in both CNN and LSTM.

Compared to previous works, we find that, (i) large-scale existing resources with noise are more advantageous as training data; and (ii) knowledge-based semantic models can play an important role.

[2] https://github.com/nanduan/MSRA-Open-Domain-QA-Tasks.

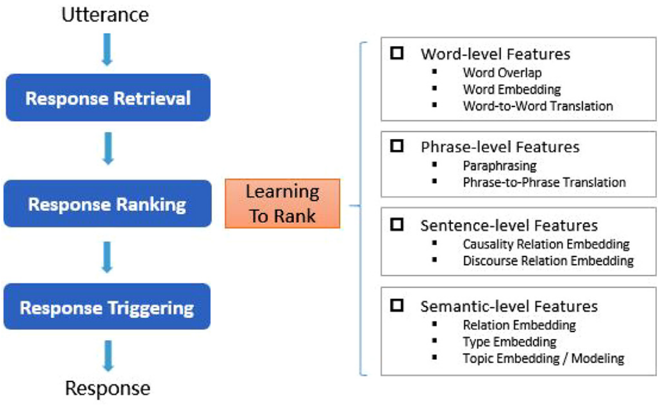


Fig. 1. A brief illustration of the DocChat system.

3. Approach overview

In this section, we give an overview of the proposed approach. Fig. 3 provides an illustration. To construct a highly efficient system, we split the task into three cascaded modules, i.e., **response retrieval, response ranking**, and **response triggering**. Given an utterance, our DocChat system retrieves related sentences from large unstructured documents, ranks the most appropriate sentence as the response candidate, and returns it if its relevance score is greater than a predefined threshold. The response retrieval component employs a fast matching model to find a small set of responses to reduce the candidate set. The response ranking component employs a learning to rank model with sophisticated features to measure the semantic relevance between an utterance and a response. Finally, the response triggering component determines whether the top-ranked candidate is a suitable response for the given utterance.

Formally, for a given user utterance Q and an indexed document set D, our approach retrieves response R based on the following three steps:

- *Response retrieval*, which retrieves response candidate set C from D based on Q:

  $$C = Retrieve(Q, D)$$

  Each $S \in C$ is a sentence existing in D.
- *Response ranking*, which ranks all response candidates in C and selects the most possible response candidate as $\hat{S}$:

  $$\hat{S} = \arg\max_{S \in C} Rank(S, Q)$$

- *Response triggering*, which decides whether the top-ranked sentence $\hat{S}$ is sufficient suitable to response Q:

  $$I = Trigger(\hat{S}, Q)$$

  where I is a binary value. When I is equal to *true*, we set the response $R = \hat{S}$ and output R; otherwise, there is no output.

We present the details of these three components in the following sections.

4. Response retrieval

For the response retrieval step, we are concerned with an efficiency of the system to find a small set of response candidates C from the large-scale index D. We apply Okapi BM25 term weighting formulas [32] to retrieve response candidates from an index D which is widely used in the information retrieval field. Given a query $Q = x_1, x_2, \ldots, x_n$ and a response sentence S, the BM25 score is calculated as:

$$BM25(Q, S) = \sum_{i=1}^{n} idf(x_i) \frac{f(x_i, S) \cdot (k_1 + 1)}{f(x_i, S) + k_1 \left(1 - b + b \frac{|S|}{avgtl}\right)},$$

where $f(x_i, S)$ and $idf(x_i)$ are word x_i's term frequency and inverse document frequency (idf), respectively. $|S|$ is the length of the response sentence, *avgtl* is the average length in index D, and k_1 and b are hyper-parameters.

5. Response ranking

The response ranking component is based on learning to rank, which ranks candidate responses with matching features. We adopt a point-wise learning to rank method [33] to re-rank the retrieved response candidate set. Given a user utterance Q and a response sentence S, the ranking function $Rank(S, Q)$ is designed as an ensemble of the individual matching features:

$$Rank(Q, S) = \sum_{k} \lambda_k \cdot f_k(S, Q),$$

where $f_k(\cdot)$ denotes the kth feature function and λ_k denotes $f_k(\cdot)$'s corresponding weight. The parameters in the ranking model are trained by stochastic gradient descent (SGD) based on a set of labeled $\langle Q, S, L\rangle$ triples. Here, L is a label ($+$ or $-$) that indicates whether S is a suitable response to Q ($+$) or not ($-$).

To measure the semantic relevance between an utterance and a response, we carefully designed a set of features at four levels of granularity including word-level, phrase-level, sentence-level, and semantic-level. The input of a feature function is two strings, i.e., an utterance string Q and a candidate sentence string S, and the output is a real number representing the relevance score.

5.1. Word-level feature

The intuition of word-level features is that an utterance is relevant to a response if a large amount of the same or similar words is used. We designed three word-level features, including a word matching feature f_{WM}, a word-level translation feature f_{W2W}, and a word embedding-based feature f_{W2V}.

f_{WM} is calculated based on the number of words shared by Q and S.

$$f_{WM}(Q, S) = \frac{\sum_{S_j \in S} \delta(S_j, Q) \cdot idf_{S_j}}{\sum_{S_j \in S} idf_{S_j}}$$

idf_{S_j} denotes the *idf* of the j^{th} word S_j in S. Note that $\delta(S_j, Q)$ equals 1 when S_j occurs in Q, otherwise $\delta(S_j, Q)$ is 0. A larger f_{WM} value indicates a larger amount of word overlap between S and Q.

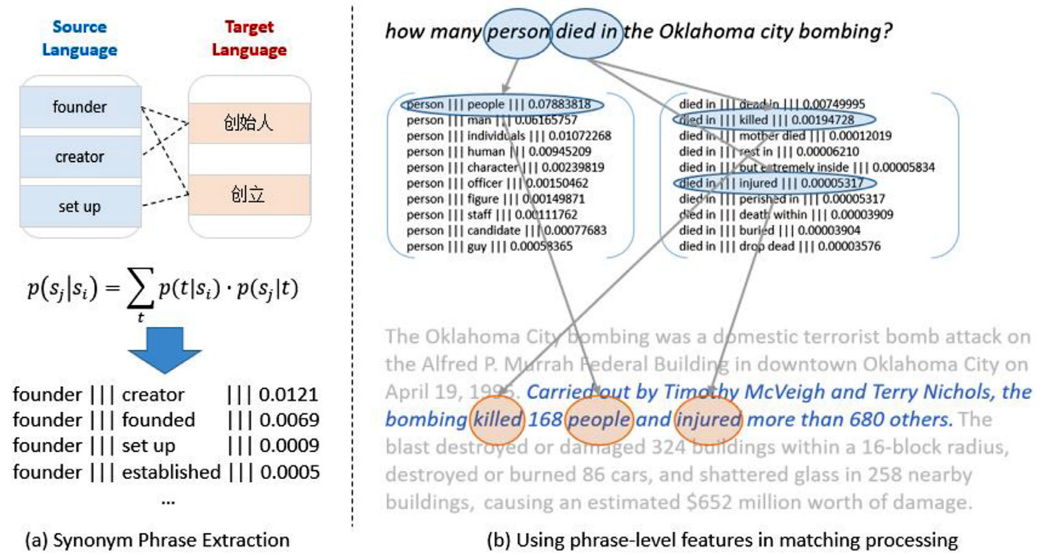
f_{W2W} is a translation-based feature that leverage word-to-word translation probabilities estimated from a sentence and its similar sentences to match a semantically similar Q-R pair. f_{W2W} calculates the relevance for an utterance and a sentence based on the IBM model 1 [34].

$$f_{W2W}(Q, S) = \frac{\sum_{i=1}^{|Q|} \ln \left(\frac{\sum_{j=1}^{|S|} p(S_j|Q_i)}{|S|} \right)}{|Q|}$$

where $p(S_j|Q_i)$ denotes the probability that word Q_i can be translated into word S_j trained on "sentence-similar sentence" pairs.

Feature f_{W2V} is based on word embedding which can be used to calculate the distance between sentences in the semantic space. Here, f_{W2V} calculates the average cosine distance between the word embeddings of all non-stopword pairs:

$$f_{W2V}(Q, S) = \frac{\sum_{i=1}^{|Q|} \ln \left(\frac{\sum_{j=1}^{|S|} cosine(v_{Q_i}, v_{S_j})}{|S|} \right)}{|Q|}$$


Source Language
Target Language
founder
creator
set up
创始人
创立
$p(s_j|s_i) = \sum_t p(t|s_i)\cdot p(s_j|t)$
founder ||| creator ||| 0.0121
founder ||| founded ||| 0.0069
founder ||| set up ||| 0.0009
founder ||| established ||| 0.0005
...
how many person died in the Oklahoma city bombing?
person ||| people ||| 0.07883818
person ||| man ||| 0.06165757
person ||| individuals ||| 0.01072268
person ||| human ||| 0.00945209
person ||| character ||| 0.00239819
person ||| officer ||| 0.00150462
person ||| figure ||| 0.00149871
person ||| staff ||| 0.00111762
person ||| candidate ||| 0.00077683
person ||| guy ||| 0.00058365
died in ||| dead in ||| 0.00749995
died in ||| killed ||| 0.00194728
died in ||| mother died ||| 0.00012019
died in ||| rest in ||| 0.00006210
died in ||| but extremely inside ||| 0.00005834
died in ||| injured ||| 0.00005317
died in ||| perished in ||| 0.00005317
died in ||| death within ||| 0.00003909
died in ||| buried ||| 0.00003904
died in ||| drop dead ||| 0.00003576
The Oklahoma City bombing was a domestic terrorist bomb attack on the Alfred P. Murrah Federal Building in downtown Oklahoma City on April 19, 1995. Carried out by Timothy McVeigh and Terry Nichols, the bombing killed 168 people and injured more than 680 others. The blast destroyed or damaged 324 buildings within a 16-block radius, destroyed or burned 86 cars, and shattered glass in 258 nearby buildings, causing an estimated $652 million worth of damage.

(a) Synonym Phrase Extraction

(b) Using phrase-level features in matching processing

Fig. 2. Figure (a) shows how to extract a synonym phrase based on “English-Chinese” bilingual pairs. Figure (b) illustrates how the extracted phrase table is helpful in bridging the lexical gaps in the matching procedure.

where v_{S_j} is the word vector of the j^{th} word in S and v_{Q_j} represents the word vector of the i^{th} word in Q.

5.2. Phrase-level features

We design two phrase-level features, including f_{PP} and f_{P2P}, to deal with cases in which an utterance and a sentence use different expressions to describe similar meanings. Both f_{PP} and f_{P2P} are based on extracted phrase tables (PT) by an existing statistical machine translation method [35]. A phrase table is defined as a quadruple, namely $PT = \{\langle src_i, trg_i, p(trg_i|src_i), p(src_i|trg_i)\rangle\}$, where src_i (or trg_i) denotes a phrase, in source (or target) language, $p(trg_i|src_i)$ (or $p(src_i|trg_i)$) denotes the translation probability from srg_i (or trg_i) to trg_i (or src_i). The difference between f_{PP} and f_{P2P} is that the PT of f_{PP} is extracted from “English-Chinese” bilingual pairs while the PT of f_{P2P} is extracted from “question-answer” pairs. Fig. 5.2 gives an example of synonym phrase extraction and their integration in the matching procedure.

The paraphrase-based feature f_{PP} is designed to deal with cases where an utterance and a response use different expressions to describe a similar meaning. The underlying hypothesis is that, two source phrases that are aligned to the same target phrase tend to be similar. The essence of this paraphrase-based model is to align phrases in a bilingual parallel corpus, and equate different phrases in source language that are aligned with the same phrase in the target language.

$$f_{PP}(Q,S) = \frac{1}{N}\sum_{n=1}^{N}\frac{\sum_{i,j} score_{PP}(src^S_{i,n}, src^Q_{j,n})}{|S|-N+1}$$

$$score_{PP}(src_x; src_y) = \sum_{PT} p(tgt_k|src_x)\cdot p(src_y|tgt_k)$$

The phrase-to-phrase translation feature f_{P2P} bridges the lexical gaps between question and answer. In this way, “question-answer” pairs are used in phrase table extraction which are crawled from community question answering sites. The f_{P2P} is defined as:

$$f_{P2P}(Q,S) = \frac{1}{N}\sum_{n=1}^{N}\frac{\sum_{i,n} score_{P2P}(S_{i,n}, Q_{j,n})}{|S|-n+1}$$

$$score_{P2P}(S_{i,n}; Q_{j,n}) = \sum_{PT} p(phr|S_{i,n})\cdot p(Q_{j,n}|phr)$$

Specifically, $score_{P2P}(S_{i,n}; Q_{j,n}) = 1$ if $S_{i,\ n}$ is equal to $Q_{j,\ n}$.

5.3. Sentence-level features

Due to the variety of lexical choices and inherent ambiguities in natural languages, surface-form matching approaches tend to produce brittle results. In this section, we introduce an attention-based convolution neural network (CNN) model to match an utterance and a sentence beyond word-level and phrase-level methods. Based on the proposed CNN model, we design two sentence-level features, including the sentence causality relationship feature f_{SCR} and the sentence discourse relationship feature f_{SDR}.

5.3.1. Attention-based CNN model

The backbone of our model is a distributional CNN model [36], which takes a sentence pair as the input and maps the inputs to two vector representations separately. Fig. 5.3.1 illustrates the architecture of our model.

In this work, we adopt an attention mechanism to update input vector representation S_X and S_Y to re-weighted representations $S^{'}_X$ and $S^{'}_Y$. The goal of updating the input sentence representation is to re-weight the importance of each word instead of treating them equally. The importance score is assigned to each piece of a word according to its semantic relatedness between Q-R pairs. The model first looks up an embedding table and represents the input sentence pair $\langle s_x,\ s_y\rangle$ as $SX = [e_{x,1}, \ldots e_{x,n_x}]$ and $SY = [e_{y,1}, \ldots e_{y,n_y}]$ respectively, where $e_{x,\ i}$, $e_{y,\ j}$ are the embedding of the i^{th} and j^{th} words in s_x and s_y. SX and SY are then used to compute a word-word similarity matrix M. Specifically, ∀ i,j, the $(i,\ j)^{th}$ element of M is defined as $M_{i,j} = cosine(e_{x,i}, e_{y,j})$. Then, column-wise attention vector V_c and row-wise attention vector V_r are computed based on M respectively, i.e., $V_c = \max_{1<i<|SX|}\{M(i,\cdot)\}$ and $V_r = \max_{1<j<|SY|}\{M(\cdot,j)\}$. Next, we obtain two attention score distributions α_x and α_y based on V_c and V_r as $\alpha_{x,k} = \frac{exp(V_c(k))}{\sum_{i=1}^{|V_c|} exp(V_c(i))}$ and $\alpha_{y,k} = \frac{exp(V_r(k))}{\sum_{j=1}^{|V_r|} exp(V_r(j))}$. Last, we update SX (or SY) to SX' (or SY'), by multiplying every value in $e_{x,\ k}$ (or $e_{y,\ k}$) with $\alpha_{x,\ k}$ (or $\alpha_{y,\ k}$).

At the convolutional layer, the parameters of convolution vector W_c project every trigram from SX' (or SY') into a feature vector h_t, computed as follows:

$$h_t = tanh(W_c\cdot\alpha_{t:t+2})$$

Since we intend to capture the most useful local features, we then aggregate the outputs of all features into a single vector H. In the

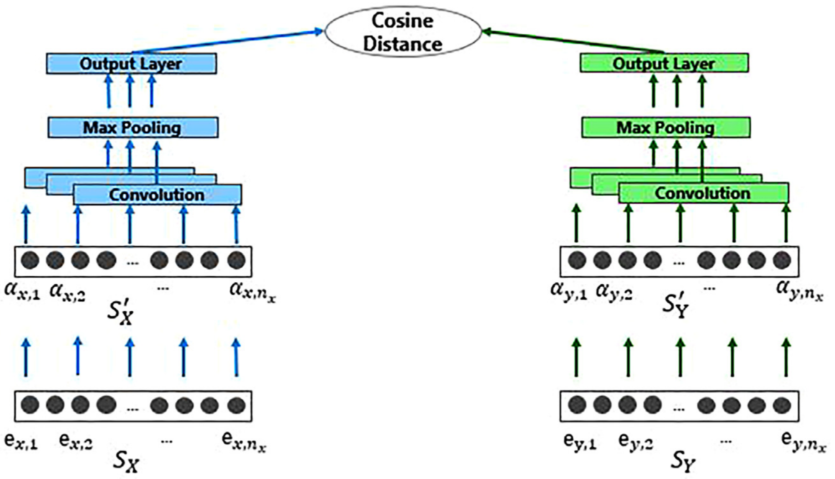

Cosine Distance
Output Layer
Max Pooling
Convolution
$\alpha_{x,1}$ $\alpha_{x,2}$ ... α_{x,n_x}
S'_X
$e_{x,1}$ $e_{x,2}$ ... e_{x,n_x}
S_X
Output Layer
Max Pooling
Convolution
$\alpha_{y,1}$ $\alpha_{y,2}$ ... α_{y,n_x}
S'_Y
$e_{y,1}$ $e_{y,2}$ ... e_{y,n_x}
S_Y

Fig. 3. The architecture of our attention-based CNN model.

output layer, a non-linear transformation is applied to H:

$$y(SX) = tanh(W_s \cdot H),$$

where W_s is the semantic projection matrix, $y(SX)$ is the final sentence embedding of SX.

We train model parameters W_c and W_s by minimizing the following ranking loss function:

$$L = \max\{0, M - cosine(y(SX), y(SY)) + cosine(y(SX), y(SY^-))\},$$

where SY^- is a negative instance and M is a constant.

5.3.2. Utterance-response pair modeling

We design the feature f_{SCR} to model causality relationships between utterance and indexed sentence. These features are calculated as:

$$f_{SCR}(Q, S) = cosine(CNN^Q_{SCR}(Q), CNN^S_{SCR}(S)),$$

where $CNN^Q_{SCR}(Q)$ and $CNN^S_{SCR}(S)$ are outputs of CNN model for utterance and sentence respectively. The sentence model CNN^Q_{SCR} and CNN^S_{SCR} are trained on a set of "question-answer" pairs. This feature plays an important role in question-like utterances.

We design discourse relationship feature f_{SDR} to model discourse relationships between utterance and indexed sentence. These features are calculated as:

$$f_{SDR}(Q, S) = cosine(CNN^Q_{SDR}(Q), CNN^S_{SDR}(S))$$

where $CNN^Q_{SDR}(Q)$ and $CNN^S_{SDR}(S)$ denote the sentence vector of utterance and indexed sentence respectively. The sentence model CNN^Q_{SDR} and CNN^S_{SDR} are trained on a large number of "sentence-next sentence" pairs which can be easily obtained from documents. This feature works on statement-like utterances.

We also feed previous and next sentence (S^{prev} and S^{next}) of a sentence into the two models to compute $f_{SCR}(Q, S^{prev})$, $f_{SCR}(Q, S^{next})$, $f_{SDR}(Q, S^{prev})$ and $f_{SDR}(Q, S^{next})$. The reason for computing each sentence together with its context sentences is intuitive: if a sentence within a document can respond to an utterance, then its context should be relevant to the utterance as well. Therefore, we collect a set of sentence triples $\langle S^{prev}, S, S^{next}\rangle$ when building the document index D, where S denotes a sentence in kth document D_k, S^{prev} and S^{next} denote S's previous sentence and next sentence respectively. Two special tags, $\langle BOD\rangle$ and $\langle EOD\rangle$, are added at the beginning and ending of each passage to make sure that such sentence triples can be extracted for every sentence in the document.

Table 1
Examples of questions and corresponding facts in knowledge base.

Example 1	
question	What does Jimmy Neutron do?
$\langle e_{sbj}, rel, e_{obj}\rangle$	⟨ jimmy_neutron, fictional_character_occupation, inventor⟩
$\langle e_{obj}, type\rangle$	⟨inventor, fictional_universe.character_occupation⟩
Example 2	
question	Which forest is Fires Creek in?
$\langle e_{sbj}, rel, e_{obj}\rangle$	⟨ fires_creek, containedby, nantahala_national_forest⟩
$\langle e_{obj}, type\rangle$	⟨nantahala_national_forest, location.location⟩

5.4. Semantic-level features

Considering semantic information is important to match a response with an utterance. When humans reply to an utterance, they may not only follow the literal content of the utterance, but also associate prior knowledge of the utterance. The prior knowledge could be topics, semantic tags, and entities related to the text pair. In this section, we present four semantic-level features based on knowledge, including relation embedding feature f_{RE}, type embedding feature f_{TE}, supervised topic model feature f_{STM}, and unsupervised topic model feature f_{UTM}.

We first describe f_{RE} and f_{TE} to model semantic relationships between utterance and response based on curated knowledge bases (KB), such as Freebase and DBpedia. A natural language question Q with its answer can be understood and mapped precisely to an assertion over a structured KB. The format of an assertion in a KB is $\langle e_{sbj}, rel, e_{obj}\rangle$, where e_{sbj} denotes a subject entity detected from the question, rel denotes the relationship expressed by the question, and e_{obj} denotes an object entity. In addition, we can extend each entity answer e_{obj} to $\langle e_{obj}, type\rangle$, where $type$ denotes the type name of e_{obj} over KB. From the two examples in Table 1, we can see that the semantic relation between question and answer can be identified with the assertion as a bridge.

Based on question-assertion pairs, we get two resources; $\langle question, rel\rangle$ pairs and $\langle question, type\rangle$ pairs. For a given question Q, the corresponding semantic tag^+ is treated as a positive example, and other randomly selects tags are used as negative examples tag^-. The posterior probability of tag^+ given Q is computed as:

$$P(tag^+|Q) = \frac{exp(cosine(y(tag^+), y(Q)))}{\sum_{tag^-} exp(cosine(y(tag^-), y(Q)))},$$

where $y(tag)$ and $y(Q)$ denote embeddings of tag and Q. We leverage the CDSSM model [37] to learn embedding functions $y(tag)$ and

$y(Q)$. We use $\langle Q, rel\rangle$ pairs as training data to train the *rel*-CDSSM. We train the *type*-CDSSM model on $\langle Q, rel\rangle$ pairs. CDSSM models are trained by maximizing the log-posterior. We calculate f_{RE} and f_{TE} as:

$$f_{RE}(Q,S) = cosine(y_{RE}(Q), y_{RE}(S))$$
$$f_{TE}(Q,S) = cosine(y_{TE}(Q), y_{TE}(S)),$$

where $y_{RE}(S)$, $y_{RE}(Q)$, $y_{TE}(S)$ and $y_{TE}(Q)$ denote semantic embeddings of S and Q respectively, coming from *rel*-CDSSM and *type*-CDSSM.

In addition to semantic tags from curated KB, the topic is another important information to measure the semantic relatedness between an utterance and a response. As the assumption that Q-R pairs should share a similar topic, we define another two semantic-level features including f_{STM} and f_{UTM}. We design an unsupervised topic modeling feature f_{UTM} as the average cosine distance between topic vectors of all non-stopword pairs $\langle v_{S_j}, v_{Q_i}\rangle$ where $v_w = [p(t_1|w), ..., p(t_N|w)]^T$ denotes the topic vector of a given word w.

$$f_{UTM}(Q,S) = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \ln\left(\frac{\sum_{j=1}^{|S|} cosine(v_{Q_i}, v_{S_j})}{|S|}\right)$$

Given a corpus, various topic modeling methods, such as pLSI (Probabilistic Latent Semantic Indexing) and LDA (Latent Dirichlet Allocation), can be used to estimate $p(t_i|w)$, which denotes the probability that w belongs to a topic t_i.

One shortcoming of the unsupervised topic modeling is that, the topic size is pre-defined, which might not reflect the truth on a specific corpus. In this paper, we explore a supervised topic model approach as well, based on "sentence-topic" pairs. Similar to *rel*-CDSSM and *type*-CDSSM, we use the ⟨*sentence, topic*⟩ pairs to train another CDSSM model. We crawl a large number of ⟨*sentence, topic*⟩ pairs from Wikipedia documents, in which *topic* is the section name that the sentence is extracted from. Such section names are labeled by Wikipedia article editors, and can be seen as the topic of the sentence. For example, "*There is evidence of human habitation in the Dunhuang area as early as 2,000 BC.*" is the first sentence in the "*History*" section.[3] Based on this, we define a supervised topic model-based feature as:

$$f_{STM}(Q,S) = cosine(y_{STM}(Q), y_{STM}(S)),$$

where $y_{STM}(Q)$ and $y_{STM}(S)$ denote topic embeddings of Q and S respectively, coming from *topic*-CDSSM. We will introduce the effectiveness of this feature in our experiments.

6. Response triggering

There are two types of utterances, i.e., chit-chat utterances and informative utterances. The former should be handled by chit-chat engines, and the latter is more suitable to our work because documents typically contain formal and informative contents. Thus, we respond only to informative utterances. Here, we define *response triggering* as a function that decides whether a response candidate S has sufficient confidence to be output:

$$\begin{aligned} I &= Trigger(Q,S) \\ &= I_U(Q) \wedge I_{Rank}(Q,S) \wedge I_R(S), \end{aligned}$$

where *Trigger*(Q, S) returns *true*, if and only if all of its three sub-functions return *true*. $I_U(Q)$ returns *true* if Q is an informative query. We collect and label chit-chat queries based on conversational exchanges from social media websites to train the classifier. $I_{Rank}(Q, S)$ returns *true*, if the score $s(Q, S)$ exceeds an empirical threshold τ:

$$s(Q,S) = \frac{1}{1+e^{-\alpha\cdot Rank(Q,S)}}$$

where α is a scaling factor that controls the distribution of $s(\cdot)$ smooth or sharp. Both α and τ are selected based on a separated development set. $I_R(S)$ returns *true* if (i) the length of S is less than a pre-defined threshold, and (ii) S does not start with a phrase that expresses a progressive relation, such as *but also, besides, moreover* and etc., as the contents of sentences starting with such phrases usually depend on their context sentences; thus, they are not suitable responses.

7. Experiments

In this section, we introduce the experimental setup and show empirical results on two tasks. Section 7.1 introduces the experiments on answer sentence selection. Section 7.2 reports the results on a real human-computer conversation.

7.1. Evaluation on answer sentence selection

Considering response ranking and answer selection are similar, we first evaluate DocChat in a question answering scenario as a simulation. Given a question, answer sentence selection chooses answer sentences from all candidate sentences in a corresponding document.

7.1.1. Data preparation

We use the WikiQA dataset [38] and the QASent dataset [39] in English. WikiQA is precisely constructed based on natural language questions and Wikipedia documents. QASent is another benchmark dataset for answer sentence selection, which is based on TREC-QA data.

Since there is no publicly available dataset for answer sentence selection in Chinese, we construct a dataset called DBQA through manual annotation. Here, we describe some important details during the data construction process. We randomly select 15,000 pages from 1,101,786 Chinese Wikipedia pages which cover a wide range of entities, including human, company, book, location, etc. For each page, we only keep the first 30 sentences as the candidate sentence list which are regularly from the abstract and the first section of the page. Afterwards, human annotators are asked to label the DBQA dataset by the following three steps: The first stage shows three sentences randomly selected from candidate sentences along with the associated entity name. An annotator is asked "Is one of the sentences suitable for asking a question for this entity?". If the annotator chooses "No", the system will switch to another three sentences. At the second stage, the system lets annotators edit a question based on the sentence they choose. At the last stage, the annotators are asked to check all candidate sentences to determine whether they can answer the question. Each question-sentence pair is checked by two other annotators to ensure quality of labeling. Our dataset ends up consisting of 14,769 questions and 304,413 sentences, where 15,305 sentences are labeled as "+" according to the corresponding questions.

As manually labeled data requires expensive human annotation, the size of this dataset is not particularly large. To get a larger dataset, we propose an automatic way to collect data from web. First, "question-answer" (or Q-A) pairs $\{Q_i, A_i\}_{i=1}^{M}$ are crawled from community QA websites, where Q_i denotes a question and A_i denotes Q_i's answer. Then, we index answer sentences of all questions and label the $\{Q_i, A_i\}$ as $+$. Finally, for each question Q_i, we retrieve 20 negative samples (labeled as $-$) by the response retrieval model which we present in Section 4.

[3] From https://en.wikipedia.org/wiki/Dunhuang.

Table 2
Evaluation of answer sentence selection on WikiQA.

#	Methods	MAP	MRR
(1)	LCLR [25]	59.93%	60.68%
(2)	bi-CNN [38]	65.20%	66.52%
(3)	NASM [29]	68.86%	70.69%
(4)	AB-CNN [30]	69.21%	71.08%
(5)	DocChat	68.25%	70.73%
(6)	DocChat+Bi-CNN	**70.08%**	**72.22%**

7.1.2. Experimental setup

We introduce our experimental setting of features at different levels of granularity.

In the English version,

- **Word-level features**: For word-to-word translation feature f_{W2W}, the word translation probability is trained on 11.6M "question-related question" pairs released by [40]. For f_{W2V}, we apply Word2Vec [41] to train word embedding on sentences from Wikipedia in English.
- **Phrase-level features**: For paraphrase feature f_{PP}, 0.5M Chinese-English bilingual sentences are used to extract a phrase-based translation table. For phrase-to-phrase feature f_{P2P}, we first construct 4M "question-answer" sentence pairs and then extract a phrase table from word alignments using the intersect-diag-grow refinement. The raw "question-answer" pairs are crawled from Yahoo Answers.
- **Sentence-level features**: For sentence causality relationship feature f_{SCR}, the attention-based CNN model is trained on 4M "question-answer" sentence pairs (also used for f_{P2P}). For sentence discourse relationship feature f_{SDR}, our model is trained on 0.5M "sentence-next sentence" pairs which is randomly selected from English Wikipedia.
- **Semantic-level features**: The "question-relation" pairs and "question-type" pairs based upon SimpleQuestions dataset [42] are used to train f_{RE} and f_{TE}. The SimpleQuestions dataset consists of 108,442 English questions written by human annotators based on a triple in Freebase. For f_{STM}, 4M "sentence-topic" pairs are extracted from Wikipedia, where the most frequent 25,000 content names are used as topics. For f_{UTM}, we run LightLDA [43] on sentences from English Wikipedia to estimate the topic distribution, where the number of topics is set to 1,000.

In the Chinese version, we first crawl 17M "question-related questions" pairs and 5M "question-answer" pairs from a Chinese community-based QA website (Baidu Zhidao). Pairs of "question-related questions" are used to train word alignments for f_{W2W}. Pairs of "question-answer" are used for f_{P2P} and f_{SCR}. Sentences from Chinese Wikipedia are used to train word embeddings for f_{W2V} and a topic model for f_{UTM}. The same bilingual phrase table is also used to extract a Chinese paraphrase table for f_{PP}. f_{SDR} is trained on 0.5M "sentence-next sentence" pairs from Chinese Wikipedia. 1.3M "sentence-topic" pairs crawled from Chinese Wikipedia are used to train *topic*−CDSSM for f_{STM}. Due to the lack of a curated knowledge base for Chinese, we ignore semantic-level features f_{RE} and f_{TE}.

7.1.3. Answer selection results

The performance of answer selection is evaluated with Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). Similar to previous work, questions without correct answers in the candidate sentences are not taken into account. We evaluate the quality of DocChat using the WikiQA (English) and the DBQA (Chinese) datasets.

Table 2 shows the performance of different approaches with on the WikiQA dataset where the first four rows are baseline methods, including (1) LCLR, which uses rich lexical semantic features; (2) bi-CNN, which uses a bi-gram CNN model with average pooling; (3) NASM, which uses an enriched LSTM with a latent stochastic attention mechanism to model similarity between Q-R pairs; and (4) AB-CNN, which adds an attention mechanism to the CNN architecture.

Table 3
A comparison between bi-CNN and our method on the same question.

Question	Top-ranked candidate
what religion is primary in Africa?	[bi-CNN]: religion in Africa is multifaceted and has been a major influence on art, culture and philosophy. [DocChat]: the continent's various populations and individuals are mostly adherents of Christianity or Islam.

Table 4
Impacts of features at different levels on WikiQA.

Level	Features	MAP	MRR
Word-Level	f_{WM}, f_{W2W}, f_{W2V}	60.25%	61.70%
Phrase-Level	f_{PP}, f_{P2P}	61.31%	62.61%
Sentence-Level	f_{SCR}, f_{SDR}	62.13%	64.79%
Semantic-Level	f_{TE}, f_{RE}, f_{STM}, f_{UTM}	59.47%	60.82%

Table 5
Impacts of different feature groups.

Models	MAP	Change	MRR	Change
DocChat	68.25%		70.73%	
DocChat w/o Word-Level	66.06%	−2.19	67.99%	−2.74
DocChat w/o Phrase-Level	66.80%	−1.45	68.66%	−2.07
DocChat w/o Sentence-Level	65.24%	**−3.01**	67.11%	**−3.62**
DocChat w/o Semantic-Level	66.08%	−2.17	67.90%	−2.83

Without using WikiQA's training set (only the development set is used to tune rank weights), DocChat performs comparably with state-of-the-art baselines on the WikiQA dataset. Furthermore, by combining the bi-CNN model trained on the WikiQA training set, we achieve the best results on both metrics. Our proposed method takes multiple granularity matching features into account, not just the sentence-level matching on which the baseline methods are focused. Table 3 gives a comparison between bi-CNN and our approach. The DocChat works well in this case because it captures the semantic relationships between the word "religion" (in the question) and the word "Christianity" and "Islam" (in the answer). This progress mainly comes from semantic-level features of our method.

Moreover, we evaluate the effectiveness of features at each level, and show results in Table 4 . The sentence-Level features performs best among all features as it captures lexical features. Table 5 shows the contributions of features at different levels of granularity. To highlight the differences, we report the percent deviation by removing different features at the same level from DocChat. From Table 5 we can see that, (1) each feature group is indispensable to DocChat; (2) features at the sentence-level are more important than other feature groups; (3) compared to the results in Table 2, combining all features can significantly improve performance.

To evaluate the performance of the proposed approach on the DBQA dataset, we re-implement three methods as baselines in this experiment. CDSSM is a CNN model taking a bag-of-words typed vector rather than pre-trained word embedding as input. Bi-CNN is the same to the model used in the WikiQA evaluation and LSTM denotes an LSTM model with max-pooling. Table 6 shows that DocChat outperforms the three baselines on the DBQA dataset.

Table 6
Evaluation of answer sentence selection on DBQA.

#	Methods	MAP	MRR
(1)	CDSSM [37]	68.53%	68.57%
(2)	bi-CNN [38]	69.31%	69.39%
(3)	LSTM [29]	71.36%	71.42%
(4)	DocChat	**72.70**%	**72.75**%

Table 7
Evaluation of answer sentence selection on QASent.

Methods	Training Set	MAP	MRR
Bi-CNN$_{WikiQA}$	WikiQA	65.75%	75.34%
Bi-CNN$_{QASent}$	QASent	**69.51%**	76.33%
DocChat	Auto-Generated data	68.96%	**76.88%**

Table 8
Evaluation of answer triggering task on WikiQA.

Methods	Precision	Recall	F1
Bi-CNN	28.34%	35.80%	31.64%
DocChat	**28.95%**	**44.44%**	**35.06%**

Compared to previous methods, DocChat has two advantages. First, our models based on existing resources are readily available (such as Q-Q pairs, Q-A pairs, “sentence-next sentence” pairs, etc.) rather than requiring manually annotated data (e.g., WikiQA and DBQA). Note that training the learning to rank model requires labeled data; however the size of the required data is acceptable. Second, as the training data used in our approach come from open-domain resources, we can expect high adaptation capability and comparable results for other answer sentence selection tasks because our models are task-independent.

To verify the second advantage, we evaluate DocChat on another answer sentence selection dataset, QASent, and the results are shown in Table 7. Bi-CNN$_{WikiQA}$ and Bi-CNN$_{QASent}$ refer to the results of Bi-CNN models which use WikiQA’s training set and QASent’s training set respectively. The *DocChat* in this evaluation is the same as in the WikiQA evaluation. According to the results in Table 7, DocChat outperforms Bi-CNN$_{WikiQA}$ in terms of MAP and MRR, and achieves comparable results compared to Bi-CNN$_{QASent}$. The experiment results show good adaptation capability of DocChat.

7.1.4. Answer triggering results

In both QA and human-computer conversation, response triggering is important. We evaluate answer triggering task by using Precision, Recall, and F1 score as metrics. We use the development set of WikiQA to tune the scaling factor α and trigger threshold τ that are described in Section 6, where α is set to 0.9 and τ is set to 0.5. Table 8 shows the performance compared to the Bi-CNN which is reported in [38]. The improvements lie in the fact that our response ranking model are more discriminative, as more semantic-level features are leveraged.

7.2. Evaluation on human-computer conversation system

In this section, we design two experiments to test DocChat on human-machine conversation. First, we design a side-by-side evaluation to compare DocChat with Xiaoice, a well-known and widely used open domain chitchat engine in China. Then, we evaluate our DocChat method on response sentence selection.

7.2.1. Experimental setup

We also perform experiments using WeChat, which is the most popular social media and instant messaging mobile application. Each official account in WeChat post several articles every day to their followers, usually focusing on one or a small number of topics. Users can converse with a followed official WeChat account. The owner of each official account can authorize other human-computer conversation systems, such as Xiaoice, to automatically respond to follower utterances.

We randomly select 10 official WeChat accounts, and index their articles separately as DocChat’s document set D. The average number of documents is 600. For each official account, human annotators are asked to freely issue 100 utterances to DocChat engines to get DocChat candidate lists with 50 candidate responses. Thus, we obtain 1,000 utterances and 50,000 ⟨utterance, DocChat candidate response⟩ pairs. Then, we ask annotators to label (0 or 1) each ⟨utterance, DocChat candidate response⟩ pair to determine whether the response is suitable for the given query. We also send the same utterance to Xiaoice and obtain 1,000 ⟨utterance, Xiaoice response⟩ pairs.

Table 9
Result of side-by-side evaluation.

	Better	Worse	Tie
Compare to Xiaoice	58	47	51

7.2.2. Experiment on side-by-side evaluation

Due to a response triggering mechanism, only 156 utterances out of 1,000 are triggered by the DocChat engine. Given those 156 ⟨query, XiaoIce response, DocChat response⟩ triples, we let human annotators do a side-by-side evaluation, asking them which response is better for each query. Note that, the source of each response is masked during evaluation.

Table 9 shows the side-by-side results. **Better** (or **Worse**) denotes a DocChat response is better (or worse) than a XiaoIce response, **Tie** denotes a DocChat response and a XiaoIce response are equally good or bad. From Table 9, we observe the following: (1) There are more better cases than worse cases. Most queries in better cases are non-chitchat queries, and their content is strongly related to the domain of their corresponding official WeChat accounts. (2) Our proposed method is a good complement for chitchat engines on informative utterances. (3) We check un-triggered queries, and find most of them are chitchat, such as “hi”, “hello” and “who are you?”. (4) We wrongly send chitchat queries to DocChat, as currently, we only use a white list of chitchat queries for chitchat/non-chitchat classification.

The example in Table 10 shows that Xiaoice response is more colloquial, as it comes from Q-R pairs; DocChat response is more formal, as it derived from documents.

7.2.3. Error analysis

We conduct error analysis on the triggered results and Answer Sentence Selection on QASent. summarize the errors into three categories. The first category is question type mismatch. Taking the first row in Table 11 as an example. Here, the response is a related sentence does not yet contain any frequency information. The second category is user intention mismatch. An example utterance is “*Recommend some Shanghai cuisines to me*”. Our current model does not have the ability to deal with this. Handling the second type of utterances need more external knowledge. The third category is that the top-ranked response is too similar to the utterance. In this situation, a DocChat response overlaps with an utterance, but cannot actually respond to it. For this issue, we need to refine the capability of our response ranking model.

Table 10
Example of side-by-side evaluation between DocChat and Xiaoice.

Utterance	Response
你知道北京的历史么？ (Do you know the history of Beijing?)	[XiaoIce Response]: 我的历史课学的不太好。(I am not good at history) [DocChat Response]: 北京历史悠久， 可以追溯到3000年前。 (Beijing is an historical city that can be traced back to 3,000 years ago.)

Table 11
Three typical Errors including question type mismatch, user intention mismatch and repeated response.

Utterance	**DocChat response**
上海车展几年举办一次? (How often will Shanghai automobile exhibition be held?)	2005 年成为国内最具影响力车展的这一年，上海车展办了整整20年。(By 2005, the year it became the most influential auto show in China, the Shanghai automobile exhibition had successfully been held for 20 years.)
上海美食推荐 (Recommend some Shanghai cuisines to me)	上海人喜欢吃，喜欢美食。(Shanghai people like to eat, like food.)
怎么变漂亮? (How to become beautiful?)	让你轻松变漂亮. (Let you easily become beautiful.)

Table 12
Evaluation of DocChat with different development set on response sentence selection task.

Dev Set	# of ques. in dev set	MAP	MRR
DBQA	14,765	44.85%	60.18%
Auto-Small	15,516	41.82%	54.57%
Auto-Large	90,321	44.91%	59.86%

7.2.4. Experiment on response sentence selection

We then train three DocChat models based on three datasets, i.e., the DBQA, Auto-Small, and Auto-Large datasets. Auto-Small and Auto-Large are pseudo-annotated datasets generated automatically based on question-answer pairs crawled from Baidu Zhidao. Auto-Small and Auto-Large contain 15,516 and 90,321 ⟨query, answer sentence candidate⟩ pairs, respectively. The result of response sentence selection is shown in Table 12.

The performance of DocChat trained on Auto-Small is worse than DBQA when the amount of data in the development set is slightly different. This makes sense in that an auto-generated dataset usually contains more noise than a human labeled dataset. With nearly 6 times the amount of data being expanded, the performance of DocChat on Auto-Large is significantly better than DocChat on Auto-Small, and very close the results of DocChat on DBQA. This proves the efficiency of our pseudo-annotated data collection method.

8. Conclusion

In this paper, we propose DocChat, a response retrieval method for human-computer conversation systems. Unlike existing retrieval-based methods using utterance-response pairs as candidates, our method finds responses based on unstructured documents. The core of our method is a learning to rank model, which ranks candidate responses with a set of well-designed matching features at different levels. We evaluate our method on both answer sentence selection and response sentence selection, and find our method achieving promising results. A side-by-side evaluation with Xiaoice shows that DocChat is a good complement for human-computer conversation systems. We also release our answer sentence selection dataset to research communities to enable researchers to use it for more natural language proceeding tasks. We leave better triggering components and multiple rounds of conversation handling to be addressed in our future work.

Acknowledgment

We wish to express great thanks Yeyun Gong, Yu Wu, Duyu Tang and Yang Yang for their tremendously helpful discussions. We also want thank the anonymous reviews for their valuable comments. This work is supported by the National Natural Science Foundation of China (Grand Nos. 61672081, U1636211, 61370126), the Beijing Advanced Innovation Center for Imaging Technology (No.BAICIT-2016001), and the National High Technology Research and Development Program of China under grant (No.2015AA016004).

References

[1] A.I. Rudnicky, E.H. Thayer, P.C. Constantinides, C. Tchou, R. Shern, K.A. Lenzo, W. Xu, A. Oh, Creating natural dialogs in the carnegie mellon communicator system., Eurospeech, 1999.

[2] M. Henderson, M. Gašić, B. Thomson, P. Tsiakoulis, K. Yu, S. Young, Discriminative spoken language understanding using word confusion networks, in: Spoken Language Technology Workshop (SLT), 2012 IEEE, IEEE, 2012, pp. 176–181.

[3] Z. Yan, N. Duan, P. Chen, M. Zhou, J. Zhou, Z. Li, Building task-oriented dialogue systems for online shopping, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 4618–4626.

[4] A. Ritter, C. Cherry, W.B. Dolan, Data-driven response generation in social media, in: Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), 2011, pp. 583–593.

[5] L. Shang, Z. Lu, H. Li, Neural responding machine for short-text conversation, in: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2015, pp. 1577–1586.

[6] Z. Ji, Z. Lu, H. Li, An information retrieval approach to short text conversation, arXiv:1408.6988 (2014).

[7] R. Yan, Y. Song, H. Wu, Learning to respond with deep neural networks for retrieval-based human-computer conversation system, in: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2016, pp. 55–64.

[8] J. Weizenbaum, Eliza—a computer program for the study of natural language communication between man and machine, Commun. ACM 9 (1) (1966) 36–45.

[9] D. Litman, S. Singh, M. Kearns, M. Walker, Njfun: a reinforcement learning spoken dialogue system, in: Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems, 2000, pp. 17–20.

[10] J. Schatzmann, K. Weilhammer, M. Stuttle, S. Young, A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies, Knowl. Eng. Rev. 21 (02) (2006) 97–126.

[11] J.D. Williams, S. Young, Partially observable markov decision processes for spoken dialog systems, Comput. Speech Lang. 21 (2) (2007) 393–422.

[12] I.V. Serban, A. Sordoni, Y. Bengio, A. Courville, J. Pineau, Building end-to-end dialogue systems using generative hierarchical neural network models (2016) 3776–3783.

[13] J. Li, M. Galley, C. Brockett, G.P. Spithourakis, J. Gao, W.B. Dolan, A persona-based neural conversation model, in: Proceedings of the 54th Annual Meeting of the Association for Computational, 2016.

[14] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, W. Ma, Topic aware neural response generation, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 3351–3357.

[15] X. Li, L. Mou, R. Yan, M. Zhang, Stalematebreaker: a proactive content-introducing approach to automatic human-computer conversation, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 2845–2851.

[16] R. Yan, Y. Song, H. Wu, Learning to rank short text pairs with convolutional deep neural networks, in: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, 2016, pp. 55–64.

[17] Y. Wu, W. Wu, C. Xing, M. Zhou, Z. Li, Sequential matching network: a new architecture for multi-turn response selection in retrieval-based chatbots, in: Proceedings of the 55th Annual Meeting of the Association for Computational

Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 496–505.

[18] O. Vinyals, Q. Le, A neural conversational model, in: Proceedings of Conference of the 32nd International Conference on Machine Learning (ICML 2015) Deep Learning Workshop, 2015.

[19] J. Li, M. Galley, C. Brockett, J. Gao, B. Dolan, A diversity-promoting objective function for neural conversation models, in: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2016, pp. 110–119.

[20] B. Hu, Z. Lu, H. Li, Q. Chen, Convolutional neural network architectures for matching natural language sentences, in: Annual Conference on Neural Information Processing Systems, NIPS, 2014, pp. 2042–2050.

[21] M. Wang, Z. Lu, H. Li, Q. Liu, Syntax-based deep matching of short texts, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, 2015, pp. 1354–1361.

[22] M. Wang, C.D. Manning, Probabilistic tree-edit models with structured latent variables for textual entailment and question answering, in: Proceedings of the International Conference on Computational Linguistics (COLING), 2010, pp. 1164–1172.

[23] M. Heilman, N.A. Smith, Tree edit models for recognizing textual entailments, paraphrases, and answers to questions, in: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2010, pp. 1011–1019.

[24] X. Yao, B. Van Durme, C. Callison-Burch, P. Clark, Answer extraction as sequence tagging with tree edit distance., in: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2013, pp. 858–867.

[25] W.-t. Yih, M.-W. Chang, C. Meek, A. Pastusiak, Question answering using enhanced lexical semantic models, in: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2013, pp. 1744–1753.

[26] L. Yu, K.M. Hermann, P. Blunsom, S. Pulman, Deep learning for answer sentence selection, NIPS Deep Learning and Representation Learning Workshop, 2014.

[27] A. Severyn, A. Moschitti, Learning to rank short text pairs with convolutional deep neural networks, in: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 373–382.

[28] D. Wang, E. Nyberg, A long short-term memory model for answer sentence selection in question answering, in: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2015, pp. 707–712.

[29] Y. Miao, L. Yu, P. Blunsom, Neural variational inference for text processing, in: Proceedings of Conference of the 33rd International Conference on Machine Learning (ICML 2016), 2016.

[30] W. Yin, H. Schtze, B. Xiang, B. Zhou, Abcnn: attention-based convolutional neural network for modeling sentence pairs, Trans. Assoc. Comput.Ling. 4 (2016) 259–272.

[31] M. Tan, C.N. dos Santos, B. Xiang, B. Zhou, Improved representation learning for question answer matching., in: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2016.

[32] K.S. Jones, S. Walker, S.E. Robertson, A probabilistic model of information retrieval: development and comparative experiments: part 2, Inf. Process. Manage. (2000) 809–840.

[33] R. Nallapati, Discriminative models for information retrieval, in: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2004, pp. 64–71.

[34] P.F. Brown, V.J.D. Pietra, S.A.D. Pietra, R.L. Mercer, The mathematics of statistical machine translation: parameter estimation, Comput.Ling. 19 (2) (1993) 263–311.

[35] P. Koehn, F.J. Och, D. Marcu, Statistical phrase-based translation, in: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 1, 2003, pp. 48–54.

[36] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.

[37] Y. Shen, X. He, J. Gao, L. Deng, G. Mesnil, A latent semantic model with convolutional-pooling structure for information retrieval, in: Proceedings of the Conference on Information and Knowledge Management (CIKM), 2014, pp. 101–110.

[38] Y. Yang, W.-t. Yih, C. Meek, Wikiqa: a challenge dataset for open-domain question answering, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, pp. 2013–2018.

[39] M. Wang, N.A. Smith, T. Mitamura, What is the jeopardy model? A quasi-synchronous grammar for qa., in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 7, 2007, pp. 22–32.

[40] A. Fader, L. Zettlemoyer, O. Etzioni, Paraphrase-driven learning for open question answering, in: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), 2013.

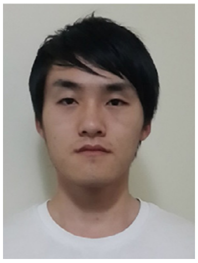
[41] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems (NIPS), 2013, pp. 3111–3119.

[42] A. Bordes, N. Usunier, S. Chopra, J. Weston, Large-scale simple question answering with memory networks, arXiv:1506.02075v1(2015).

[43] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E.P. Xing, T.-Y. Liu, W.-Y. Ma, Lightlda: big topic models on modest computer clusters, in: Proceedings of the Annual International Conference on World Wide Web (WWW), 2015, pp. 1351–1361.

Zhao Yan is currently a Ph.D. candidate at Beihang University, Beijing, China. He received his B.S. degree in school of computer science and engineering from Beihang University in 2011. His research interests include open domain question answering, dialogue system and information retrieval.

Nan Duan is a lead researcher in Natural Language Computing (NLC) group at Microsoft Research Asia. He received his Ph.D. degree from Tianjin University of China in 2011. His research interests include open domain question answering, semantic parsing, dialogue system, paraphrasing, and etc.



Junwei Bao received the Master's Degree in Jul. 2014 from the Department of Computer Science, Harbin Institute of Technology, Harbin, China. Since 2014, he is a Ph.D. candidate at the Department of Computer Science, Harbin Institute of Technology. His current research interests include natural language processing, knowledge-based question answering, semantic parsing and natural language generation.

Peng Chen is a principal software engineer in Xiaoice team at Microsoft. He graduated with M.S. degree in Computer Science from Peking University in 2010. He participated in the design and implement a lot of chatbot systems. He has a wealth of experience in building practical intelligent systems.

Ming Zhou is a Principal Researcher in Natural Language Computing group at Microsoft Research Asia. He works in the areas of machine translation and natural language processing, who has led various projects on NLP, machine translation, text mining and social network. He has authored or co-authored about 100 papers published at top conferences, and has served as area chairs of ACL, UCAI, AAAI, EMNLP, COLING, SIGIR, UCNLP for many times.

Zhoujun Li is working as a professor of Beihang University. He graduated with M.S. degree in Computer Science from Wuhan University in 1984. He received his M.S. and Ph.D. degrees in Computer Science from National University of Defense Technology in 1986 and 1999 respectively. His research interests include data mining, information retrieval, and information security. He is a member of the IEEE and ACM.