Expert Systems with Applications 42 (2015) 8146–8155

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Query-oriented unsupervised multi-document summarization via deep learning model

Sheng-hua Zhong [a,b], Yan Liu [b,*], Bin Li [c], Jing Long [d]

[a] College of Computer Science & Software Engineering, Shen Zhen University, Shen Zhen, Guang Dong 518060, China
[b] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong 999077, China
[c] Department of Linguistics and Translation, City University of Hong Kong, Kowloon, Hong Kong 999077, China
[d] Business School, Nanjing University, Nan Jing, Jiang Su 210093, China

ARTICLE INFO

Article history:
Available online 15 June 2015

Keywords:
Deep learning
Query-oriented summarization
Multi-document
Neocortex simulation

ABSTRACT

Capturing the compositional process from words to documents is a key challenge in natural language processing and information retrieval. Extractive style query-oriented multi-document summarization generates a summary by extracting a proper set of sentences from multiple documents based on pre-given query. This paper proposes a novel document summarization framework based on deep learning model, which has been shown outstanding extraction ability in many real-world applications. The framework consists of three parts: concepts extraction, summary generation, and reconstruction validation. A new query-oriented extraction technique is proposed to extract information distributed in multiple documents. Then, the whole deep architecture is fine-tuned by minimizing the information loss in reconstruction validation. According to the concepts extracted from deep architecture layer by layer, dynamic programming is used to seek most informative set of sentences for the summary. Experiment on three benchmark datasets (DUC 2005, 2006, and 2007) assess and confirm the effectiveness of the proposed framework and algorithms. Experiment results show that the proposed method outperforms state-of-the-art extractive summarization approaches. Moreover, we also provide the statistical analysis of query words based on Amazon's Mechanical Turk (MTurk) crowdsourcing platform. There exists underlying relationships from topic words to the content which can contribute to summarization task.

1. Introduction

Automatically generating summaries from large text corpora has long been attracting research attention from both information retrieval and natural language processing, the earlier studies of which could be dated back to the 1950s and 1960s (Baxendale, 1958; Edmundson, 1969; Luhn, 1958). Automatic generation of summaries creates shortened versions of texts to help users catch important information in the original text with bearable time costs (Khanpour, 2009). Currently, the creation of summaries is a task best handled by humans. However, with the explosion of textual data, especially in big data era, it is no longer financially possible, or feasible, to produce all types of summaries by hand. Earlier studies on text summarization aimed at summarizing from pre-given documents without requirements, which is usually referred to as generic summarization (Berger & Mittal, 2000). With the development of information retrieval, query-oriented summarization task, which requires summarizing from a set of document to answer a pre-given query, has started attracting more and more attention (Tang, Yao, & Chen, 2009). According to the size of the input, text summarization tasks can be grouped into single-document and multi-document summarization tasks (Shen, Sun, Li, Yang, & Chen, 2007). Based on the writing style of the output summary, text summarization techniques can be divided into extractive approaches and abstractive approaches (Song, Choi, Park, & Ding, 2011; Wong, Wu, & Li, 2008). Due to the limitation of current natural language generation techniques, extractive approaches are the mainstream in the field. An extractive approach selects a number of indicative text fragments from the input documents to form a summary instead of re-writing an abstract (Chen, Yang, Zha, Zhang, & Zhang, 2008) under a budget constraint. A budget constraint is natural in summarization task as the length of the summary is often restricted (Lin & Bilmes, 2010). In the paper, we adopt the extractive style to develop techniques for query-oriented multi-document summarization.

Almost all extractive summarization methods are faced with two key problems: how to rank textual units, and how to select a subset of those ranked units (Jin, Huang, & Zhu, 2010). The first

* Corresponding author.
E-mail addresses: csshzhong@szu.edu.cn (S.-h. Zhong), csyliu@comp.polyu.edu.hk (Y. Liu), binli2@cityu.edu.hk (B. Li), longjing@nju.edu.cn (J. Long).

http://dx.doi.org/10.1016/j.eswa.2015.05.034
0957-4174/

one on ranking requires systems to model the relevance of a textual unit to a topic or a query. The second one on selection requires systems to improve diversity or to remove redundancy so that more relevant information can be covered by the summary within a limited length.

Attempts to solutions of sentence ranking are varied. Some of solutions are based on surface features (Luhn, 1958; Radev, Jing, Stys, & Tam, 2004), some on graphs (Wan, 2009; Wan & Xiao, 2009; Wei, Li, Lu, & He, 2010), and some on supervised learning (Cao, Qin, Liu, Tsai, & Li, 2007; Ouyang, Li, Li, & Lu, 2011). After obtaining a list of ranked sentences, it is then important to select a subset of sentences to form a good summary that includes diverse information within a length limit. Goldstein, Mittal, Carbonell, and Kantrowitz (2000) were among the first to propose global models using the maximum marginal relevance (MMR) criteria. The models score sentences under consideration as a weighted combination of relevance plus redundancy with sentences already in the summary. Currently, greedy MMR style algorithms are the standard algorithms in document summarization. McDonald (2007) proposed to replace the greedy search of MMR with a globally optimal formulation, where the basic MMR framework can be expressed as a knapsack packing problem, and an integer linear program (ILP) solver can be used to maximize the resulting objective function.

This paper presents a new method following the extractive style to summarize documents using deep techniques. Deep learning models the learning task using deep architectures composed of multiple layers of parameterized nonlinear modules. These models have been proved outstanding in feature extraction of visual data. To our knowledge, this is the first attempt that utilizes deep learning in query-oriented multi-document summarization task. Different from the existing methods, we neither directly rank the textual units based on the relevance to the topic or query, nor directly improve diversity or remove redundancy. The proposed deep learning algorithm is partitioned into three stages: concept extraction, reconstruction validation, and summary generation. In the concept extraction stage, hidden layers are used to abstract the documents layer by layer using greedy layer-wise extraction algorithm. The second stage of reconstruction validation intends to reconstruct the data distribution by fine-tuning the whole deep architecture globally. Finally, dynamic programming (DP) is utilized to maximize the importance of the summary with the length constraint. A novel framework with several new algorithms is proposed in the following part.

2. Related work on deep learning

Different from shallow learning models, deep learning is learning multiple levels of representation and abstraction so as to extract more senses out of data. Besides evidence from neuroscience, theoretical analyses from machine learning also confirmed that deep models are more compact and expressive than shallow models in representing most learning functions, especially highly variable ones. Many empirical validations are also reported to support that deep architectures are promising in solving hard learning problems (Larochelle, Erhan, Courville, Bergstra, & Bengio, 2007). Moreover, theoretical analysis shows that deep architectures are more efficient than shallow circuits such as a typical support vector machine (SVM), because the former can represent most common functions, especially highly-variable learning functions compactly and effectively.

However, it is difficult to learn the parameters of deep architectures with multiple hidden layers containing trainable weights at all levels. Back propagation, a well-known computationally efficient model for multilayer neural networks, also suffers from insufficient labeled data, high computational cost, and poor local optima when working under a deep model (Hinton, 2007). To reduce the difficulty of deep learning, Hinton and Salakhutdinov (2006) proposed deep belief network (DBN), i.e. a densely-connected, directed belief net with multiple hidden layers. DBN partitions the learning procedure to two stages: to abstract input information layer by layer and to fine-tune the whole deep network to the ultimate learning target (Hinton, Osindero, & Teh, 2006; Salakhutdinov & Hinton, 2007). The network pairs each feed-forward layer with a feed-back layer that attempts to reconstruct the input of the layer from the output. Such layer-wise generative models are implemented by a family of Restricted Boltzmann Machines (RBMs) (Smolensky, 1986). After a greedy unsupervised learning to each pair of layers, the lower-level features are progressively combined into more compact high-level representations. In the second stage, the whole deep network is refined using a contrastive version of the "wake-sleep" algorithm via a global gradient-based optimization strategy. Owing to this two-stage fast greedy learning, DBN exhibits notable performance in dimensionality reduction (Liu, Xu, Tsang, & Luo, 2009) and classification (Cao, Yu, Luo, & Huang, 2009) for different applications, such as image generation (Dahl, Ranzato, Mohamed, & Hinton, 2010), and audio event classification (Ballan, Bazzica, Bertini, Bimbo, & Serra, 2009).

The conference version of our work is the first attempt of deep learning methods for the query-oriented multi-document summarization task (Liu, Zhong, & Li, 2012). After we proposed deep learning models for document summarization task, more and more recent work focused on deep learning based methods. For example, Cao, Wei, Dong, Li, and Zhou (2015) introduced a ConvNet model to support introspection of the document structure. Their model is used to identify and extract task-specific salient sentences from documents. Denil, Demiraj, and Freditas (2014) developed a ranking framework upon Recursive Neural Networks to rank sentences for multi-document summarization. It formulates the sentence ranking task as a hierarchical regression process, which simultaneously measures the salience of a sentence and its constituents in the parsing tree.

3. Basic idea of proposed model

Humans do not have difficulty with summarizing documents based on given queries. Query-oriented multi-document summarization, however, has remained a well-known challenge in natural language processing in the past fifteen years of extensive research. In the evaluation of the summarization tasks in the Document Understanding Conference (DUC), the summaries created by human peers are much better than those extracted automatically. Motivated by this fact, we aimed at designing a proper deep architecture and corresponding unsupervised learning algorithms for query-oriented multi-document summarization. Latest research findings from neuroscience suggest that the deep learning model is consistent with the physical structure of human neocortex, evolution of intelligence, and propagation of information in the human neocortex. Thus, it has great potential to provide human-like judgment using a human-like system in tasks of natural language processing. A discussion of the deep learning model from three aspects is presented in the following sections.

(1) The deep architecture is identical to the multi-layer physical structure of the human cerebral cortex. The neocortex, which is associated with many cognitive abilities, has a complex multi-layer hierarchy (Lee & Mumford, 2003). The laminar structure and a multi-layer illustration of the neocortex are shown in Fig. 1. Fig. 1(a) is the laminar structure of the

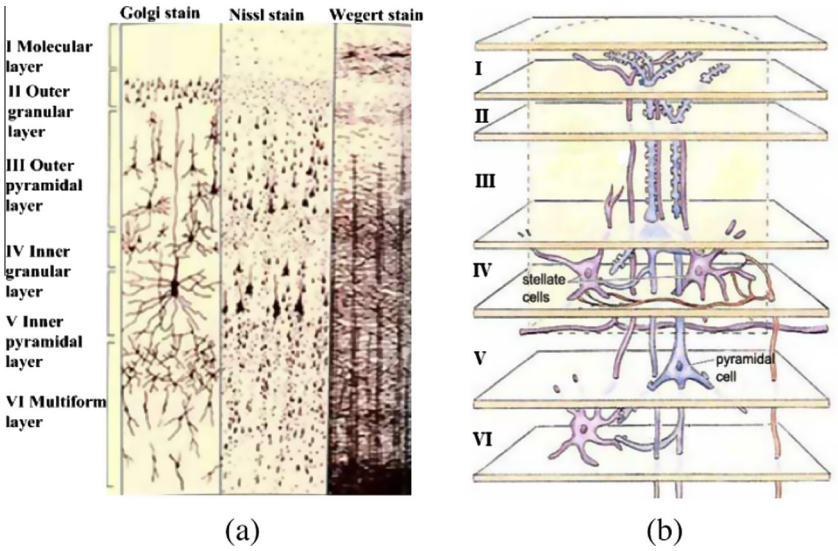


Fig. 1. Multi-layer structure of the cerebral cortex. (a) Laminar structure of the neocortex. (b) Multi-layer illustration of the neocortex.

neocortex, and (b) is multi-layer illustration of the neocortex. We can find the neocortex can be roughly differentiated into six functionally distinct horizontal layers from Molecular layer I to Multiform layer VI (Leuba & Kraftsik, 1994). Many different neocortex areas are involved in lexical-semantic processing, and therefore, dozens of cortical layers are involved in generating even the simplest lexical-semantic processing.

(2) The development of intelligence follows the multi-layer structure. From an evolutionary viewpoint, the phylogenetically most recent part of the brain is the neocortex. In humans and other primates, starting from catarhinians, the multi-layers structure began to appear in the neocortex (Barton, 1996). Therefore, a deep multi-layers architecture may represent the result of human intelligence evolution. It thus provides a possible way to achieve the ultimate goal of natural language processing, to enable the computer to understand the human (natural) languages.

(3) The manner in which data is delivered in a deep architecture is a good simulation of the information propagation in the neocortex. The deep architecture contains multi-layer generative models with bottom-up and top-down connections. Bottom-up connections can be used to infer the high-level representations that would have generated an observed set of low-level features. Top-down connections can be used to generate low-level features from high-level representations. Single cell recordings and the reciprocal connectivity between cortical areas both suggest a hierarchy of progressively more complex features in which each layer can influence layers below it (Felleman & Van Essen, 1991).

Based on the above features, the deep model is chosen in this paper for query-oriented multi-document summarization. To better adapt the document data and the query-oriented multi-document summarization application, we propose a novel unsupervised deep learning model called query-oriented deep extraction (QODE) with a new deep architecture and a new unsupervised deep learning algorithm. To our knowledge, our paper is the first attempt to utilize deep learning in tasks of query-oriented multi-document summarization.

Fig. 2 shows the deep architecture of the QODE technique. The feature vector $\mathbf{f}^d = \left[f_1^d, f_2^d, \ldots, f_v^d, \ldots, f_V^d\right]$, the *tf* value of word in

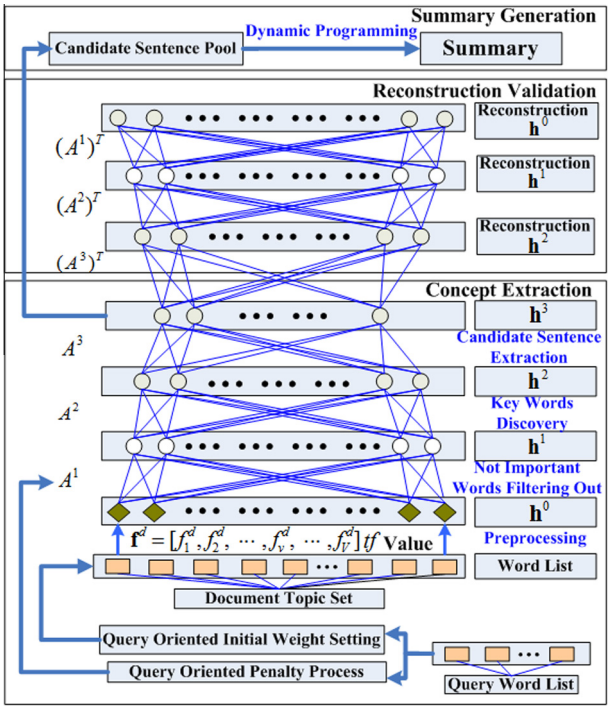


Fig. 2. Deep architecture of QODE technique.

the vocabulary of **D** calculated in document $\mathbf{d}_m$, is input into deep architecture. *V* is the length of the vocabulary of **D**. The output is a summary $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t, \ldots, \mathbf{s}_T]$. Similar with DBN, Restricted Boltzmann Machines (RBMs) are used as building blocks. The deep architecture of QODE is designed with a reference to the human neocortex and human perception. The most obvious characteristic of QODE is its multi-layer structure, which is faithful to the physiology and anatomy of the human cortex. As discussed earlier, the functional areas of the neocortex could be roughly differentiated into six horizontal layers, including the classic lexical-semantic processing areas, such as: Broca's and Wernicke's areas, and other important lexical-semantic processing areas.

Based on this new deep architecture, the whole deep learning algorithm can be partitioned into three stages: concept extraction,

reconstruction validation, and summary generation. In the concept extraction stage, three hidden layers of H^1, H^2, and H^3 are used to abstract the documents layer by layer using greedy layer-wise extraction algorithm. In our implementation, the hidden layer H^1 is used to filter out words appearing accidentally. The hidden layer H^2 is supposed to discover key words. The second stage of reconstruction validation intends to reconstruct the data distribution by fine-tuning the whole deep architecture globally. Finally, dynamic programming (DP) is utilized to maximize the importance of the summary with the length constraint. After these three stages, the final optimized summary S^* is generated. In the following session, we will discuss the detailed learning procedure of each stage.

3.1. Query-oriented concept extraction

First, we generate a vocabulary with length V based on words appearing in the document topic set $\mathbf{D}$. The feature vectors $\mathbf{f}^D = [f_1^D, f_2^D, \ldots, f_v^D, \ldots, f_V^D]$ of the document set $\mathbf{D}$ and $\mathbf{f}^d = [f_1^d, f_2^d, \ldots, f_v^d, \ldots, f_V^d]$ of the single document $\mathbf{d}_m$ are calculated. Here, f_v^D means the *tf* value of vth word in the vocabulary of $\mathbf{D}$ calculated in all documents. f_v^d means the *tf* value of vth word in the vocabulary of $\mathbf{D}$ calculated in $\mathbf{d}_m$.

Then, $\mathbf{f}^d$ is input to the deep architecture as the visible layer H^0 to construct a RBM with hidden layer H^1. The energy of the state $(\mathbf{h}^0, \mathbf{h}^1)$ in the first RBM is:

$$E(\mathbf{h}^0, \mathbf{h}^1; \theta^1) = -\left((\mathbf{h}^0)^T A^1 \mathbf{h}^1 + (b^1)^T \mathbf{h}^0 + (c^1)^T \mathbf{h}^1\right) = -\sum_{i=1}^{K_0}\sum_{j=1}^{K_1} h_i^0 A_{ij}^1 h_j^1 - \sum_{i=1}^{K_0} b_i^1 h_i^0 - \sum_{j=1}^{K_1} c_j^1 h_j \quad (1)$$

where $\theta^1 = (A^1, b^1, c^1)$ are the model parameters between layer H^0 and layer H^1. A_{ij}^1 is the symmetric interaction term between visible unit i in H^0 and hidden unit j in H^1. b_i^1 is the ith bias of layer H^0 and c_j^1 is the jth bias of layer H^1. K_0 is the number of unit in H^0 and K_1 is the number of unit in H^1. So the first RBM has the following joint distribution:

$$P(\mathbf{h}^0, \mathbf{h}^1; \theta^1) = \frac{1}{Z} e^{-E(\mathbf{h}^0, \mathbf{h}^1; \theta^1)} = \left(e^{-E(\mathbf{h}^0, \mathbf{h}^1; \theta^1)}\right) \Big/ \left(\sum_{\mathbf{h}^0}\sum_{\mathbf{h}^1} e^{-E(\mathbf{h}^0, \mathbf{h}^1; \theta^1)}\right) \quad (2)$$

where Z is the normalization constant. And the probability of the model assigned to a visible vector to $\mathbf{h}^0$ in H^0 is:

$$P(\mathbf{h}^0) = \frac{1}{Z}\sum_{\mathbf{h}^1} e^{-E(\mathbf{h}^0, \mathbf{h}^1; \theta^1)} = \left(\sum_{\mathbf{h}^1} e^{-E(\mathbf{h}^0, \mathbf{h}^1; \theta^1)}\right) \Big/ \left(\sum_{\mathbf{h}^0}\sum_{\mathbf{h}^1} e^{-E(\mathbf{h}^0, \mathbf{h}^1; \theta^1)}\right) \quad (3)$$

And the log-likelihood of $P(\mathbf{h}^0)$ is:

$$\log P(\mathbf{h}^0) = \log \sum_{\mathbf{h}^1} e^{-E(\mathbf{h}^0, \mathbf{h}^1; \theta^1)} - \log \sum_{\mathbf{h}^0}\sum_{\mathbf{h}^1} e^{-E(\mathbf{h}^0, \mathbf{h}^1; \theta^1)} \quad (4)$$

Gibbs sampling from an RBM proceeds by sampling $\mathbf{h}^1$ given $\mathbf{h}^0$, then $\mathbf{h}^0$ given $\mathbf{h}^1$, etc. The conditional distributions over input state $\mathbf{h}^0$ in visible layer H^0 and hidden state $\mathbf{h}^1$ in hidden layer H^1 are given by Eqs. (5) and (6), where $\sigma(x) = 1/(1 + \mathbf{exp}(-x))$.

$$p(\mathbf{h}^1|\mathbf{h}^0) = \prod_j p(\mathrm{h}_j^1|\mathbf{h}^0), \quad p(\mathrm{h}_j^1 = 1|\mathbf{h}^0) = \sigma\left(\sum_i A_{ij}\mathrm{h}_i^0 + a_j\right) \quad (5)$$

$$p(\mathbf{h}^0|\mathbf{h}^1) = \prod_i p(\mathrm{h}_i^0|\mathbf{h}^1), \quad p(\mathrm{h}_i^0 = 1|\mathbf{h}^1) = \sigma\left(\sum_j A_{ij}\mathrm{h}_j^1 + b_i\right) \quad (6)$$

Denote $\mathbf{h}^1(k)$ for the kth $\mathbf{h}^1$ sample from the chain, starting at $k = 0$ with $\mathbf{h}^1(0)$, which is the input observation for the RBM and $(\mathbf{h}^1(k), \mathbf{h}^0(k))$ for $k \to \infty$ is a sample from the Markov chain. So we could calculate the derivative of Eq. (4) with respect to the parameter $\theta^1 = (A^1, b^1, c^1)$ below:

$$\frac{\partial \log p(\mathbf{h}^1(0))}{\partial \theta^1} = -\sum_{\mathbf{h}^0(0)} p(\mathbf{h}^0(0)|\mathbf{h}^1(0)) \frac{\partial E(\mathbf{h}^1(0), \mathbf{h}^0(0))}{\partial \theta^1} + \sum_{\mathbf{h}^1(k)}\sum_{\mathbf{h}^0(k)} p(\mathbf{h}^1(k), \mathbf{h}^0(k)) \frac{\partial E(\mathbf{h}^1(k), \mathbf{h}^0(k))}{\partial \theta^1} \quad (7)$$

The idea of Contrastive Divergence (Hinton, 2002) algorithm using the difference between two Kullback–Liebler divergences is to take k small (typically $k = 1$) to run the claim for only one step. When $k = 1$, the derivative to the model parameter A^1 can be obtained by Eq. (8),

$$\begin{aligned}\frac{\partial \log P(\mathbf{h}^1(0))}{\partial A^1} &= -\sum_{\mathbf{h}^0(0)} P(\mathbf{h}^0(0)|\mathbf{h}^1(0)) \frac{\partial E(\mathbf{h}^1(0), \mathbf{h}^0(0))}{\partial A^1} + \sum_{\mathbf{h}^1(1)}\sum_{\mathbf{h}^0(1)} P(\mathbf{h}^1(1), \mathbf{h}^0(1)) \frac{\partial E(\mathbf{h}^1(1), \mathbf{h}^0(1))}{\partial A^1} \\ &= \langle \mathbf{h}^1(0)^T \mathbf{h}^0(0) \rangle_{data} - \langle \mathbf{h}^1(1)^T \mathbf{h}^0(1) \rangle_{recon}\end{aligned} \quad (8)$$

where $\langle \cdot \rangle_{data}$ denotes an expectation with respect to the data distribution and $\langle \cdot \rangle_{recon}$ denotes the "reconstruction" distribution of data after one step. This leads to a simple learning rule for performing stochastic steepest ascent in the log probability of the training data in Eqs. (9) and (10).

$$\Delta A^1 = \varepsilon_A \left(\langle \mathbf{h}^1(0)^T \mathbf{h}^0(0) \rangle_{data} - \langle \mathbf{h}^1(1)^T \mathbf{h}^0(1) \rangle_{recon}\right) \quad (9)$$

$$\begin{aligned}A^1 &= \vartheta A^1 + \Delta A^1 \\ &= \vartheta A^1 + \varepsilon_A \left(\langle \mathbf{h}^1(0)^T \mathbf{h}^0(0) \rangle_{data} - \langle \mathbf{h}^1(1)^T \mathbf{h}^0(1) \rangle_{recon}\right)\end{aligned} \quad (10)$$

Other parameters in θ^1 update function could be calculated in a similar manner,

$$b^1 = \vartheta b^1 + \Delta b^1 = \vartheta b^1 + \varepsilon_b (\mathbf{h}^0(0) - \mathbf{h}^0(1)) \quad (11)$$

$$c^1 = \vartheta c^1 + \Delta c^1 = \vartheta c^1 + \varepsilon_c (\mathbf{h}^1(0) - \mathbf{h}^1(1)) \quad (12)$$

where ϑ is the momentum and ε_A, ε_b, ε_c are the learning rate.

To integrate query information for document summarization, we have two different processes including: query oriented initial weight setting and query oriented penalty process. In classical deep network, the parameter matrix A^1 is initialized to small random values chosen from a zero-mean Gaussian with a standard deviation of about 0.01. Different from it, we strengthen the influence from query as Eq. (13) after random initialization setting if the ith node word v_i in H^0 belongs to the query.

$$A_{ij}^1 = \mathbf{max}(\mathbf{A}^1) \quad \text{if } v_i \in \mathbf{q} \quad (13)$$

In the penalty process, the reconstruction error in query word is penalized more than others as below, where γ is the penalty factor.

$$\Delta A_{ij}^1 = \gamma \Delta A_{ij}^1 \quad \text{if } v_i \in \mathbf{q} \quad (14)$$

The above discussion is based on single document $\mathbf{d}_m$ for the first layer. Similar operations can be performed to the higher layer RBMs based on all documents in topic set $\mathbf{D}$.

After the concept extraction based on the deep architectures, the importance matrix AF is defined as Eq. (15). The element AF_{in} of AF is the importance of ith word in the vocabulary to the nth node of hidden layer H^3, where K_3 is the number of unit in H^3,

A^1, A^2, A^3 are the symmetric interaction term in layer pairs, and they are dot multiplied together to obtain the importance matrix.

$$AF = \left[\underbrace{\left(\mathbf{f}^D\right)^T, \left(\mathbf{f}^D\right)^T, \ldots, \left(\mathbf{f}^D\right)^T}_{K_3}, \ldots, \left(\mathbf{f}^D\right)^T\right](A^1A^2A^3) \tag{15}$$

In our implementation, hidden layer H^3 is assumed to extract the candidate sentences for the summary. Certainly, we could extract the candidate sentences of every node in H^3 only depending on how many unions of key words are in them according to AF_{in}. In our technique, after the reconstruction validation globally adjust the deep network to find optimum parameters, the DP is utilized to maximize the query oriented importance of generated summary with the constraint of summary length.

3.2. Reconstruction validation for global adjustment

In the concept extraction, we use greedy layer-by-layer algorithm to learn a deep model. In the second stage, we use backpropagation through the whole deep model to fine-tune the parameters $\theta = [A, b, c]$ for optimal reconstruction.

The greedy layer-by-layer query-oriented concept extraction stage has performed a global search for a sensible and good region in the whole parameter space. Therefore, after achieving that, we already construct a good data concept extraction model. Backpropagation is well known as a better local fine-tuning model than global search. So backpropagation is utilized to adjust the entire deep network in order to find good local optimum parameters $\theta^* = [A^*, b^*, c^*]$ which is used in summary generation via DP. The learning algorithm in this stage is used to minimize the cross-entropy error $\left[-\sum_v f_v \mathbf{log}\hat{f_v} - \sum_v (1-f_v)\mathbf{log}(1-\hat{f_v})\right]$, where f_v is the *tf* value of vth word and $\hat{f_v}$ is the *tf* value of its reconstruction.

3.3. Summary generation via dynamic programming

In this stage, dynamic programming (DP) is utilized to maximize the importance of the summary with the length constraint.

After the optimum parameters are obtained in the reconstruction validation, we use them to calculate the importance matrix *AF* by Eq. (15). Then we extract ten words with largest AF_{in} value in every nth node of Hidden layer H^3. The set of these unions of words are denoted as **UN**. The importance of every sentence In_t is calculated by Eq. (16), where λ is the query word importance factor, μ_i is the word in sentence $\mathbf{s}_t$. And the importance of the generated summary could be denoted as $\mathrm{In} = \sum_t \mathrm{In}_t$.

$$\mathrm{In}_t = \sum_i \omega_i, \begin{cases} \omega_i = \lambda & \text{if } (\mu_i \in \mathbf{UN}) \cap (\mu_i \in \mathbf{q}) \\ \omega_i = 1 & \text{if } \mu_i \in \mathbf{UN} \\ \omega_i = 0 & \text{others} \end{cases} \tag{16}$$

Taken the limited length of summary $\mathrm{N_S}$ into consideration, the summary length Le is defined as below, where l_t is the length of sentence $\mathbf{s}_t$.

$$\mathrm{Le} = l_1 + \cdots + l_t + \cdots + l_T \leqslant \mathrm{N_S} \tag{17}$$

Based on the analysis above, we could obtain the objective function aiming to optimize with the constraint below. Because the task of Document Understanding Conference is to produce query-oriented multi-document summarization with allowance of 250 words, our paper then also sets $\mathrm{N_S}$ as equal to 250.

$$max\ \mathrm{In} = \sum_t \mathrm{In}_t, \quad s.t.\ \mathrm{Le} \leqslant \mathrm{N_S} \tag{18}$$

In context of mathematical optimization method, DP refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. The optimization problem in (18) is classical knapsack problem which is often solved by DP. So we use DP to find the optimum solution.

The DP function is denoted in Eq. (19). Here, $f_K(\lambda_K)$ is the maximum of the summary importance in stage K. K is the stage variable to describe the current sentence. The state variable λ_K is the remaining length before K starts. The decision variable u_K is the choice whether or not to put the current sentence $\mathbf{s}_t$ into the summary.

$$\begin{cases} f_K(\lambda_K) = \max\{u_K \mathrm{In}_K + f_{K-1}(\lambda_{K-1})\} \\ \lambda_K = \lambda_{K+1} - u_{K+1} l_{K+1}, \quad K = t,\ 1 \leqslant t \leqslant T \\ \lambda_0 = 0,\ \lambda_T = 250,\ f_0(\lambda_0) = 0 \end{cases} \tag{19}$$

After we solve the Eq. (19) by positive sequence method of DP, we obtain the optimized summary $S^* = \{\mathbf{s}_1^*, \mathbf{s}_2^*, \ldots, \mathbf{s}_t^*, \ldots, \mathbf{s}_T^*\}$, where $\mathbf{s}_t^*$ is the optimized sentence.

4. Empirical validation

4.1. Evaluation setup

In this section, we conduct several experiments for multi-document summarization task evaluation in the Document Understanding Conference (DUC) on three open benchmark dataset DUC 2005, DUC 2006 and DUC 2007. There are altogether 50 topics in DUC 2005, 50 in DUC 2006, and 45 in DUC 2007. Each DUC topic consists a topic description and a relevant document set. The components of the topics are almost the same except for the source and number of documents. Each DUC 2005 topic includes 25–50 related documents selected from the Los Angeles Times and Financial Times of London, while each topic in DUC 2006 and DUC 2007 is composed of exactly 25 documents from Associated Press, New York Times, and Xinhua Newswire.

The task of DUC is to produce query-oriented multi-document summarization with generous allowance of 250 words. As a preprocessing step, the stop words in each sentence are removed and the remaining words are stemmed using the Porter's stemmer (Porter, 1980). In the evaluation step, the ROUGE (Lin, 2004) toolkit (i.e. ROUGEeval-1.5.5 in this study) that has been widely adopted by DUC tasks is selected. It measures summary quality by counting overlapping units such as the n-gram, word sequences and word pairs between the candidate summary and the reference summary. ROUGE-N is an n-gram recall measure computed as follows:

$$\text{ROUGE-}N = \frac{\sum_{S\in\{\text{Ref Sum}\}}\sum_{n\text{-}gram\in S} Count_{match}(n\text{-}gram)}{\sum_{S\in\{\text{Ref Sum}\}}\sum_{n\text{-}gram\in S} Count(n\text{-}gram)} \tag{20}$$

where n stands for the length of the n-gram, and $Count_{match}(n\text{-}gram)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. $Count(n\text{-}gram)$ is the number of n-grams in the reference summaries.

In performance comparison of three open dataset, we provide the results of the average recall scores of ROUGE-1, ROUGE-2 and ROUGE-SU4, along with the 95% confidential intervals within the square brackets. We compare the performance of QODE with other representative multi-document summarization algorithms. They are graph-based sentence ranking algorithm (Wan, 2009; Wan & Xiao, 2009; Wei et al., 2010) supervised learning based sentence ranking algorithms including Support Vector Classification (Vapnik, 1995), Ranking SVM (Joachims, 2002), and Regression (Ouyang et al., 2011), and also classical relevance and redundancy based selection algorithms including greedy search (Filatova & Hatzivassiloglou, 2004), maximum marginal relevance (MMR) (Goldstein et al., 2000), integer linear program (ILP) (McDonald, 2007), and the NIST baseline system (Dang, 2005).

Table 1
Performance of QODE with a comparison to representative algorithms on the DUC 2005 dataset.

System	ROUGE-1	ROUGE-2	ROUGE-SU4
QODE	0.3751 (0.3687–0.3809)	**0.0775 (0.07341–0.08136)**	**0.1341 (0.1303–0.1378)**
Graph-based (Wan, 2009)	**0.3839**	0.0737	0.1317
Graph-based (Wan & Xiao, 2009)	0.3718	0.0676	0.1293
Graph-based (Wei, Li, Lu, & He, 2010)	–	0.0771	0.1337
Support Vector Classification (Vapnik, 1995)	0.3663 (0.3569–0.3757)	0.0701 (0.0677–0.0736)	0.1243 (0.1202–0.1382)
Ranking SVM (Joachims, 2002)	0.3702	0.0711	0.1299
Regression (Ouyang, Li, Li, & Lu, 2011)	0.3770 (0.3713–0.3828)	0.0761 (0.0727–0.0793)	0.1329 (0.1294–0.1363)
Greedy search (Filatova & Hatzivassiloglou, 2004)	0.3560	0.0610	–
MMR (Goldstein et al., 2000)	0.3701	0.0701	0.1289
ILP (McDonald, 2007)	0.3580	0.0610	–
NIST baseline	–	0.0403	0.0872

Table 2
Query oriented contribution analysis.

Method			ROUGE-1	ROUGE-2	ROUGE-SU4
1	2	3			
√	√	√	0.3751	0.0775	0.1341
√	√		0.3731	0.0742	0.1315
	√	√	0.3734	0.0755	0.1329
√		√	0.3704	0.0740	0.1301

1. Query oriented initial weight setting, 2. Query oriented penalty process, 3. Summary importance maximization by DP.

4.2. Performance comparison

First, we compare the performance of the proposed techniques with other representative ones on three standard datasets based on ROUGE scores. Results of DUC 2005 listed in Table 1 shows that our new algorithm outperforms most of existing algorithms. In the proposed QODE technique, we integrate query information in concept extraction, layer-wise reconstruction, and summary generation. Table 2 presents the query oriented contribution analysis in three stages. Each step has its own contribution to the final summary generation.

In Tables 3 and 4, we provide the performance comparison on DUC 2006 and DUC 2007. As an unsupervised learning algorithm, the performance of QODE is comparable to the supervised learning based algorithms. Therefore, we can still conclude that our system is able to achieve state-of-the-art performances giving the sufficient results in various rounds.

From the performance comparison based on ROUGE scores, we could also find that the supervised learning method which is integrated with the training stage could achieve best performance in some specific dataset. For example, supervised learning based regression (Ouyang et al., 2011) obtained best ROUGE-SU4 on DUC 2006; supervised learning based regression algorithm (Joachims, 2002) got best ROUGE-1 and ROUGE-2 on DUC 2007. But we can still find, as an unsupervised learning algorithm, the proposed QODE achieved best performance of ROUGE-2 & ROUGE-SU4 on DUC 2005, ROUGE-2 on DUC 2006, and ROUGE-SU4 on DUC 2007. To other cases, the performance of QODE is comparable to the supervised learning based algorithms. Therefore, we can still conclude that our system is able to achieve state-of-the-art performances giving the sufficient results in various rounds.

As we know, the proposed method does not assume the existence of a training stage, which means we need not the labeled data (manually created summaries). Compared with unlabeled

Table 3
Performance of QODE with a comparison to representative algorithms on the DUC 2006 dataset.

System	ROUGE-1	ROUGE-2	ROUGE-SU4
QODE	0.4015 (0.3957–0.41076)	**0.0928 (0.0884–0.0972)**	0.1479 (0.1440–0.1521)
Graph-based (Wan, 2009)	**0.4101**	0.0886	0.1420
Graph-based (Wan & Xiao, 2009)	0.4031	0.0851	0.1400
Graph-based (Wei, Li, Lu, & He, 2010)	–	0.0899	0.1427
Support Vector Classification (Vapnik, 1995)	–	0.0834 (0.0793–0.0876)	0.1387 (0.1344–0.1428)
Ranking SVM (Joachims, 2002)	–	0.0890 (0.0852–0.0928)	0.1443 (0.1403–0.1477)
Regression (Ouyang, Li, Li, & Lu, 2011)	–	0.0926 (0.0883–0.0969)	**0.1485 (0.1443–0.1525)**
NIST baseline	–	0.0491	0.0962

Table 4
Performance of QODE with a comparison to representative algorithms on the DUC 2007 dataset.

System	ROUGE-1	ROUGE-2	ROUGE-SU4
QODE	0.4295 (0.4233–0.4355)	0.1163 (0.1119–0.1205)	**0.1685 (0.1645–0.1723)**
Graph-based (Wan, 2009)	0.4204	0.1030	0.1460
Graph-based (Wan & Xiao, 2009)	–	0.1123	0.1682
Graph-based (Wei, Li, Lu, & He, 2010)	0.4211 (0.4152–0.4274)	0.1103 (0.1063–0.1144)	0.1628 (0.1588–0.1670)
Support Vector Classification (Vapnik, 1995)	–	0.1075 (0.1032–0.1120)	0.1616 (0.1573–0.1659)
Ranking SVM (Joachims, 2002)	**0.4301 (0.4237–0.4365)**	**0.1175 (0.1134–0.1219)**	0.1682 (0.1642–0.1725)
NIST baseline	0.3091	0.0599	0.1036

Table 5
Performance comparison of typical QODE with different structures on the DUC 2005 dataset.

System	ROUGE-1	ROUGE-2	ROUGE-SU4
QODE	**0.3751 (0.3687–0.3809)**	**0.0775 (0.07341–0.08136)**	**0.1341 (0.1303–0.1378)**
QODE_30-H^3	0.3681 (0.3623–0.3738)	0.0750 (0.07124–0.07864)	0.1306 (0.1271–0. 1321)
QODE_no-dp	0.3699 (0.3638–0.3755)	0.0746 (0.07104–0.07790)	0.1308 (0.1275–0.1341)

data, in real-world applications, the labeled information are always insufficient. When relying on the efforts of experienced human experts, manually created summaries are often difficult, expensive, or time consuming to obtain. By contrast, with the growing availability of a large number of documents from text corpora, abundant unlabeled data are available. Thus, it is much more practical to construct a document summarization model based on unsupervised learning framework. It is one of important advantages of the proposed method.

4.3. Parameter analysis

There are numerical meta-parameters in the proposed techniques. For the parameters related with the deep model, such as learning rate and the momentum, we follow the general setting (Hinton, 2010) for the sake of simplicity. The structure of deep learning model is another set of parameters. Different with existing deep learning techniques that determine the structure such as the number of hidden layers based on intuition, we intend to provide more meaningful architecture by considering the characters of document summary tasks. In our implementation, the hidden layer H^1 is used to filter out words appearing accidentally and 1000 hidden units are used in this paper. Hidden layer H^2 is supposed to discover the key words, so the number of hidden units depends on the length of summary. In our experiment, the length is predetermined by the DUC tasks with allowance of 250 words, so we use 250 hidden units in H^2. Hidden layer H^3 is assumed to extract the candidate sentences for the summary. If the length of the summary is equal to 250 words, 10-hidden-unit is a reasonable setting of H^3. In Table 5, we provide the performance comparison of typical QODE with different structures on the DUC 2005 dataset. Because the number of hidden units in H^3 directly determines the number of candidate sentences extracted by our model, we first change the number from 10 to 30 in hidden layer H^3. From Table 5, we can find the ROUGE results of QODE_30-H^3 are worse than typical QODE. Moreover, in Table 5, we also list the performance of QODE without dynamic programming (QODE_no-dp), which means the candidate sentences will be directly used to generate as summaries. It is obvious that we could find the typical QODE is still better than QODE_no-dp. It means the dynamic programming stage is useful to obtain better summaries.

For the parameter used in dynamic programming, we discuss the influence to ROUGE results on DUC 2005 from query word importance λ. Fig. 3(a) shows the value of ROUGE-2 and Fig. 3(b) shows the value of ROUGE-SU4 when λ varies from 1 to 3. At most time, the proposed technique has the best performance. ROUGE-2 and ROUGE-SU4 peak together when λ is equal to 2.5. Similar to DUC 2005, the peak points of ROUGE-2 and ROUGE-SU4 curves can be obtained when λ is equal to 2.5 on DUC 2006 and DUC 2007.

4.4. Physical information in deep network analysis

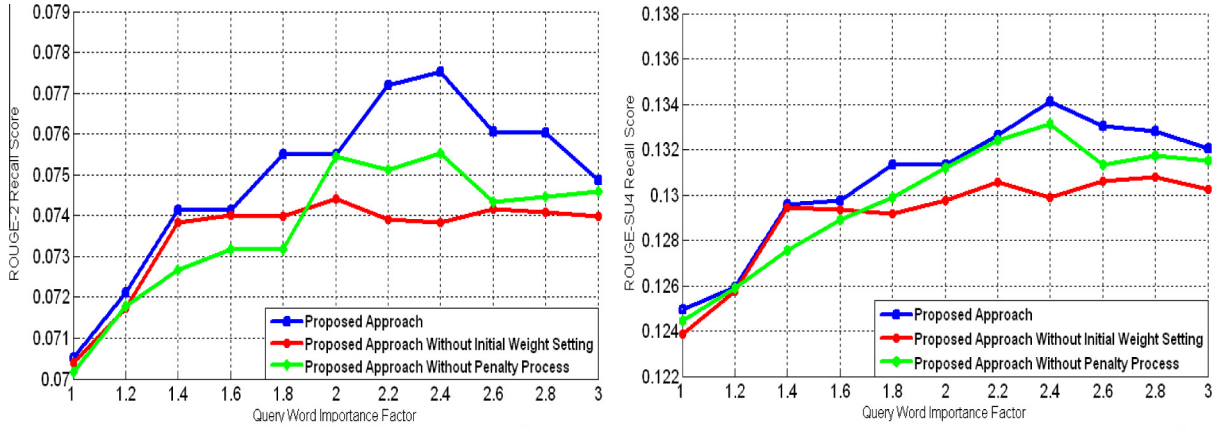
Furthermore, we want to demonstrate the rational of the proposed techniques, whether QODE really has advanced extraction ability. To demonstrate the extraction ability of proposed QODE, we analyze the information coverage in every layer using one document set D376e that contains 26 documents and 9 human summaries.

For dataset D376e, the number of nodes in layer H^0 is equal to 2032. In our experiment, we set the number of hidden nodes in layer H^1 to 1000. So we keep 1000 words pushed out by H^1 and calculate how many of them appear in human summaries. We also calculate the percentage according to the filtering out 1032 words. From Table 6, obviously, deep networks intend to find the informative words.

In layer H^2, the number of hidden layer is reduced to 250. As previously, we calculate how many words pushed out by H^2 appear in human summaries in Table 7. The word coverage of human summary is about 40%, which is nearly doubled to layer H^1. For the convenience of comparison, we randomly select 250 words from 2032 nodes and calculate that how many of them appear in human summaries. We repeat the experiments ten times and calculate the average percentage. A comparison between these two results confirms the extraction ability of our proposed techniques again.

There are ten hidden units in layer H^3 that correspond to the ten sentences appearing in the 250-words summary. In Table 8, we list ten candidate sentences related with corresponding nodes. To compare with human summary, the ID numbers of human summary which has similar sentence are also listed. Therefore, by inheriting the distinguishing extraction ability from the deep learning model, the proposed QODE pushes out important concepts layer by layer effectively.

From Tables 6–8, we have demonstrated the extraction ability of QODE by calculating the information coverage in each layer. As we known, by relying on the layer-wise information representation of RBM, deep learning methods have the ability to learn the



(a) ROUGE-2 Recall performance vs. λ (b) ROUGE-SU4 Recall performance vs. λ

Fig. 3. Performance comparison vs. λ.

Table 6
The statistical analysis of words in layer H^1.

Words	Numbers	In human summary	Percentage
Filtering out words	1032	65	6.3
Remaining words	1000	211	21.2

multi-level representation of data and overcome the "vanishing gradient problem" when layers become deeper. In the proposed algorithm, we do not just satisfy to make use of the advantages of deep learning models. In the proposed QODE, we integrated query information in initial weight setting, penalty procession, and summary importance maximization. As shown in Table 2, each step has its contribution to the final summary generation. Thus, it could explain why and how the rate of information coverage is high in every layer of the model.

4.5. Crowdsourcing experiment on query word

In this paper, we propose a deep learning model to automatically summarize query-oriented multi-documents based on given queries. To integrate query information into the final multi-documents summary, in our paper, three different processes are included. The query oriented initial weight setting strengthens the influence from query word according to set a maximum value rather than a random initial value to the connection matrix A^1. In the query oriented penalty process, the reconstruction error in query word is penalized more than others. In summary generation step, the query word is set with a higher weight in dynamic programming to maximize the importance of the summary. In our previous empirical validation, all of them demonstrate its own contribution to the final summary generation. In our preliminary work (Liu et al., 2012), we focus on the query-oriented deep learning model rather than discuss the influence from different query words. In Section 4.5, we try to analyze the influence of query words based on the crowdsourcing experiment on Amazon's Mechanical Turk (MTurk) platform.

Table 7
The statistical analysis of words in layer H^2.

Words	Number	In human summary	Percentage
Random words	250	34	13.6
Key words	250	99	39.6

Crowdsourcing is a distributed model that assigns tasks traditionally undertaken by employees or contractors to an undefined crowd (Brabham, 2008). As we known, MTurk has become popular as a source of experimental data. It offers a potential paradigm for engaging a large number of users for low time and monetary costs. Tasks can involve any kind of efforts, such as participating in

Table 8
Candidate sentence extracted in layer H^3.

Sentence with union of key words in automatically extracted summary	Id of human's summary
An **international war** crimes **tribunal** covering the former **Yugoslavia** formally opens in The Hague **today** with a request for the extradiction from **Germany** of a Bosnian Serb **alleged** to have killed three Moslem **prisoners**	A, B, C, D, E, G, H, I, J
The extradiction is important to the **tribunal** – the first **international war** crimes **court** since the Nuremberg trials after the second **world war** – because it has no power to try **suspects** in absentia	B, C, D, E, G, H, I, J
World News in Brief: **Court rules** on **border**	A, C, D, E, G, H, I, J
The International **Court** of Justice in The Hague ruled in Chad's favor in a 20-year **border** dispute with **Libya** which has **caused** two **wars**	B, D, E, H, I, J
Maybe we'll go full circle; the World **Court** can condemn this action and then the **Soviets** can defy that body, just as the **United States** defied the **court's** condemnation of our **embargo** of Nicaragua	C, D, G, I, J
Ever since the Reagan Administration walked out of the Hague to protest Nicaragua's **claim** of illegality in U.S. aid to the Contras, the **State** Department has opposed submitting to the **World Court** any **case** that involves the use of military force	H, I, J
They refused to appear in the **World Court** 10 **years ago** when Washington sought the release of American hostages in Tehran	H, I, J
A **year** after **Noriega's** capture, the **court** was still hearing arguments on whether Bush could be subpoenaed and the **World Court** was in preliminary hearings on Panama's complaint	J
After six **months** of uproar, the U.S. district **court** judge in Miami **ordered** that the **case** proceed to trial	Null
Mr. Edwin Williamson, a **legal** adviser to the U.S. **State** Department who will address the court later in the proceedings, said **yesterday**, 'This (**court**) action in no way inhibits what the Security Council is doing'	Null

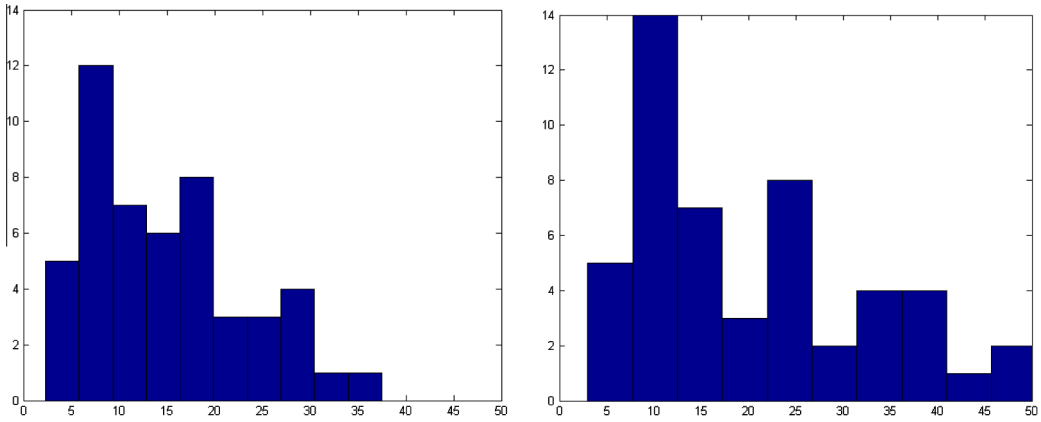


Fig. 4. Histogram of the average repeated numbers of each query in ground truth summary for every topic.

surveys, performing experiments, or answering certain specialized questions. Researchers have adopted MTurk to conduct user studies on image annotation (Loni et al., 2014), document relevance (Bashir et al., 2013). In our studies, I am investigating the importance of query word on multi-documents summarization based on MTurk crowdsourcing platform.

In the task, workers are required to read all news articles for specific topic and then select the high relevant word items from all queries words based on these news articles. The qualification requirement for this task is number of HITS approved should be greater than or equal to 100. We allowed each HIT to be completed by 10 unique workers. Since in DUC 2005 dataset, we had 50 topics to select the relevant words per behavior and had 10 unique workers select each document per behavior, 500 HITs were created. The monetary reward was based on an effective hourly wage of $4. In total, 500 HITs were created with $667 for an estimated 167 h of work. For the actual experiment, 38 workers were granted a qualification to access the HITs. And all of 500 HITS completed by those qualified workers were approved without rejections.

Fig. 4 shows the histogram of the average repeated numbers of each query in ground truth summary for every topic. To demonstrate the differences between different requirements, we provide two cases of histogram. To Fig. 4(a), we only calculate the repeated numbers for the query which is more than two workers select. To Fig. 4(b), the remaining query words are those words which are more than the average numbers of selection. From these two figures, we can find most of selected word items appear more than ten times in the reference summary in both of cases. It supports that the query words selected by MTurk workers are repeated many times in reference summary. Thus, these highly related words can be considered as the true important queries. Moreover, we count the repeated number for every word in ground truth summary and provide the average ranking order for each query word. The average ranking order for the first case is 4.76, and the average ranking order for the second case is 5.69. From these results, we could easily know that the selected query words are more important than others. In different processes of query-oriented setting, we also tried to use the good query words (selected by more than two workers) instead of all queries in our model. But we find that these settings cannot help us to achieve a better performance in final summary. It means some of query words are not thought to be close related with summary. But indeed, they have some underlying relationships with the topic and can also contribute to summarization task.

5. Conclusions

In this paper, we proposed a novel deep learning model for query-oriented multi-documents summarization. The main contributions of the work are summarized as follows: (1) Our work is the first attempt of deep learning methods for the query-oriented multi-document summarization task; (2) By inheriting the outstanding abstraction ability of deep learning methods, a novel framework is proposed to push out important concepts layer by layer effectively; (3) Under unsupervised learning framework, the proposed method demonstrates excellent extraction ability and better summary quality even compared with some representative supervised learning methods; (4) We provide the statistical analysis of query words based on Amazon's Mechanical Turk (MTurk) crowdsourcing platform.

We have already shown that the proposed algorithm is applicable to multi-document summarization task in our experiments. Actually, it also has good impacts and practical implications in a wide range of real-world applications in information retrieval and natural language processing. Furthermore, the proposed algorithm does not need the training stage, which makes it potentially suitable for industry application where the label information is limited.

Although the proposed algorithm has performed better than existing methods in query-oriented multi-documents summarization task, there exists much room for improvement. From the performance comparison of Table 4 on the DUC 2007 dataset, we can observe that the recall measure of the proposed method is relatively low than the ranking SVM algorithm. As we know, the proposed method does not assume the existence of a training corpus. We believe that the recall measure would be better if we take the supervised training stage into consideration. Hence, how to improve the performance of the proposed method by integrating the effective classifier such as SVM, is the first future work we need to consider. Another meaningful future work is to improve the efficiency of the proposed method in order to make sure the current algorithm can be transplanted on the portable devices. Last but not least, we would like to incorporate the compression decisions into our method and apply it in abstractive document summarization task.

Acknowledgment

This research was supported by National Natural Science Foundation of China (NSFC) 61373122.

References

Ballan, L., Bazzica, A., Bertini, M., Bimbo, A. D., & Serra, G. (2009). Deep networks for audio event classification in soccer videos. *Proceedings of the 2009 IEEE international conference on multimedia and expo* (pp. 474–477).

Barton, R. A. (1996). Neocortex size and behavioural ecology in primates. *Proceedings of Royal Society of London B: Biological Sciences, 263*, 173–177.

Bashir, M., Anderton, J., Wu, J., Golbus, P. B., Pavlu, V., & Aslam, J. A. (2013). A document rating system for preference judgements. *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval* (pp. 909–912).

Baxendale, P. B. (1958). Machine-made index for technical literature—An experiment. *IBM Journal of Research Development*, 354–361.

Berger, A., & Mittal, V. (2000). Query-relevant summarization using FAQs. *Proceedings of the 38th annual meeting on association for computational linguistics* (pp. 294–301).

Brabham, D. (2008). Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence, 14*(1), 75.

Cao, Z., Qin, T., Liu, T., Tsai, M., & Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. *Proceedings of the 24th international conference on machine learning* (pp. 129–136).

Cao, Z., Wei, F., Dong, L., Li, S., & Zhou, M. (2015). Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of 29th AAAI conference on artificial intelligence* (pp. 1–7).

Cao, L., Yu, J., Luo, J., & Huang, T. S. (2009). Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression. In *Proceedings of the 17th ACM international conference on multimedia* (pp. 125–134).

Chen, E. K., Yang, X. K., Zha, H. Y., Zhang, R., & Zhang, W. J. (2008). Learning object classes from image thumbnails through deep neural networks. In *Proceedings of the 2008 international conference on acoustics, speech and signal processing* (pp. 829–832).

Dahl, G., Ranzato, M., Mohamed, A., & Hinton, G. E. (2010). Generating more realistic images using gated MRF's. In *Proceedings of 24th Annual conference on neural information processing systems* (pp. 1–9).

Dang, H. T. (2005). Overview of DUC 2005. In *Proceedings of DUC 2005*. <http://www-nlpir.nist.gov/projects/duc/pubs/2005papers/OVERVIEW05.pdf>.

Denil, M., Demiraj, A., Freditas, N. (2014). Extraction of salient sentences from labelled documents. Eprint Arxiv (pp. 1–9).

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM, 16*(2), 264–285.

Felleman, D. J., & Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex, 1*(1), 1–47.

Filatova, E., & Hatzivassiloglou, V. (2004). A formal model for information selection in multisentence text extraction. In *Proceedings of the 20th international conference on computational linguistics*. Article No. 397.

Goldstein, J., Mittal, V., Carbonell, J., & Kantrowitz, M. (2000). Multi-document summarization by sentence extraction. *Proceedings of the 2000 ANLP/NAACL workshop on automatic summarization* (Vol. 4, pp. 40–48). .

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*, 1711–1800.

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in Cognitive Sciences, 11*, 428–434.

Hinton, G. E. (2010). A practical guide to training restricted Boltzmann machine. Technical report (pp. 1–21). UTML TR 2010–003, University of Toronto.

Hinton, G. E., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*, 1527–1554.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*, 504–507.

Jin, F., Huang, M. L., & Zhu, X. Y. (2010). A comparative study on ranking and selection strategies for multi-document summarization. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 525–533).

Joachims, T. (2002). Optimizing search engines using click through data. In *Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 133–142).

Khanpour, H. (2009). *Sentence extraction for summarization and notetaking* (PhD. Diss.). University of Malaya

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on machine learning* (pp. 473–480).

Lee, T., & Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *Journal of the Optical Society of America A: Optics, Image Science, and Vision, 20*(7), 1434–1448.

Leuba, G., & Kraftsik, R. (1994). Changes in volume, surface estimate, three-dimensional shape and total number of neurons of the human primary visual-cortex from midgestation until old age. *Anatomy and Embryology, 190*(4), 351–366.

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Proceedings of workshop on text summarization branches out, post-conference workshop of ACL* (pp. 74–81).

Lin, H., & Bilmes, J. (2010). Multi-document summarization via budgeted maximization of submodular functions. In *Proceedings of the 2010 annual conference of the North American chapter of the association for computational linguistics* (pp. 912–920).

Liu, Y., Xu, D., Tsang, I. W., & Luo, J. (2009). Using large-scale web data to facilitate textual query based retrieval of consumer photos. In *Proceedings of the 17th ACM international conference on multimedia* (pp. 55–64).

Liu, Y., Zhong, S.-H., & Li, W. J. (2012). Query-oriented multi-document summarization via unsupervised deep learning. In *Proceedings of 26th AAAI conference on artificial intelligence* (pp. 1699–1705).

Loni, B., Cheung, L. Y., Riegler, M., Bozzon, A., Gottlieb, L., & Larson, M. (2014). Fashion 10000: An enriched social image dataset for fashion and clothing. In *Proceedings of 5th ACM multimedia system conference* (pp. 41–46).

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research Development, 2*(2), 159–165.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. *Advances in Information Retrieval, Lecture Notes in Computer Science, 4425*, 557–564.

Ouyang, Y., Li, W. J., Li, S. J., & Lu, Q. (2011). Applying regression models to query-focused multi-document summarization. *Information Processing and Management, 47*(2), 227–237.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Radev, D. R., Jing, H. Y., Stys, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing and Management, 40*(6), 919–938.

Salakhutdinov, R. R., & Hinton, G. E. (2007). Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*.

Shen, D., Sun, J. T., Li, H., Yang, Q., & Chen, Z. (2007). Document summarization using conditional random fields. In *Proceedings of the 17th international conference on artificial intelligence* (pp. 2863–2867).

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1*, 194–281.

Song, W., Choi, L. C., Park, S. C., & Ding, X. F. (2011). Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization. *Expert System with Applications, 38*(8), 9112–9121.

Tang, J., Yao, L., & Chen, D. (2009). Multi-topic based query-oriented summarization. In *Proceedings of 2009 SLAM international conference on data mining* (pp. 1148–1159).

Vapnik, V. N. (1995). *The nature of statistical learning theory* (2nd ed.). Springer.

Wan, X. (2009). Topic analysis for topic-focused multi-document summarization. In *Proceedings of the 18th ACM conference on information and knowledge management* (pp. 1609–1612).

Wan, X., & Xiao, J. (2009). Graph-based multi-modality learning for topic-focused multi-document summarization. In *Proceedings of 19th international conference on artificial intelligence* (pp. 1586–1591).

Wei, F., Li, W. J., Lu, Q., & He, Y. X. (2010). A document-sensitive graph model for multi-document summarization. *Knowledge and Information Systems, 22*(2), 245–259.

Wong, K. F., Wu, M. J., & Li, W. J. (2008). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd international conference on computational linguistics* (pp. 985–992).