Summarizing Answers in Non-Factoid Community Question-Answering

Hongya Song[†*]
hongya.song.sdu@gmail.com

Zhaochun Ren[‡*]
zhaochun.ren@ucl.ac.uk

Shangsong Liang[‡*]
shangsong.liang@ucl.ac.uk

Piji Li[§]
pjli@se.cuhk.edu.hk

Jun Ma[†]
majun@sdu.edu.cn

Maarten de Rijke[II]
derijke@uva.nl

[†]Shandong University, Jinan, China
[‡]University College London, London, United Kingdom
[§]The Chinese University of Hong Kong, Hong Kong, China
[II]University of Amsterdam, Amsterdam, The Netherlands

ABSTRACT

We aim at summarizing answers in community question-answering (CQA). While most previous work focuses on factoid question-answering, we focus on the non-factoid question-answering. Unlike factoid CQA, non-factoid question-answering usually requires passages as answers. The *shortness*, *sparsity* and *diversity* of answers form interesting challenges for summarization. To tackle these challenges, we propose a sparse coding-based summarization strategy that includes three core ingredients: short document expansion, sentence vectorization, and a sparse-coding optimization framework. Specifically, we extend each answer in a question-answering thread to a more comprehensive representation via entity linking and sentence ranking strategies. From answers extended in this manner, each sentence is represented as a feature vector trained from a short text convolutional neural network model. We then use these sentence representations to estimate the saliency of candidate sentences via a sparse-coding framework that jointly considers candidate sentences and Wikipedia sentences as reconstruction items. Given the saliency vectors for all candidate sentences, we extract sentences to generate an answer summary based on a maximal marginal relevance algorithm. Experimental results on a benchmark data collection confirm the effectiveness of our proposed method in answer summarization of non-factoid CQA, and moreover, its significant improvement compared to state-of-the-art baselines in terms of ROUGE metrics.

Keywords

Community question-answering; Sparse coding; Short text processing; Document summarization

1. INTRODUCTION

In recent years, we have witnessed a rapid growth in the number of users of community question-answering (CQA). In the wake of this development, an increasing number of approaches to CQA retrieval have been proposed, addressing a wide range of tasks, including answer ranking [47, 49, 50], answer extraction [51], multimedia QA [28], answer generation [46, 53], and question classification [9, 10]. There has been a very strong focus on factoid CQA, in which typically a single correct answer exists, e.g., "Where was X born?" In contrast, in non-factoid CQA, multiple sparse and diverse sentences may make up the answers together. However, their shortness, sparsity and diversity make it difficult to identify all of the information that together covers all aspects of a question. Document summarization is an effective way to summarize salient information and generate a relevant and diverse answer for a given input question, in particular, in the context of non-factoid question-answering [13, 43]. However, traditional document summarization methods face a number of challenges when used for summarizing non-factoid answers in CQA.

The task on which we focus in this paper is *answer summarization in non-factoid community question-answering* [43]. Unlike previous work on document summarization, answer summarization in non-factoid CQA focuses on collecting all relevant and meaningful sentences to the input question from candidate answers. Fig. 1 shows an example of an answer summary given a question about "how to cure indigestion." We can find that the summary extracts sentences from candidate answers and covers important and diverse aspects of the answers to the question. Compared to traditional multi-document summarization task that usually focuses on summarizing news articles, summarizing answers in non-factoid CQA faces specific challenges: (1) The *shortness* of answers in non-factoid CQA is an obstacle for document summarization methods in answer summarization. (2) The *sparsity* of syntactic and context information hinders the summarization process using traditional representation of short text, based on term frequency or latent topic modeling [34]. (3) Summarization in non-factoid CQA is a recall-oriented problem, in which we need to recall as much relevant information as possible. However, the *diverse* topic distribution of answers in non-factoid CQA makes it difficult to generate a summary with high recall.

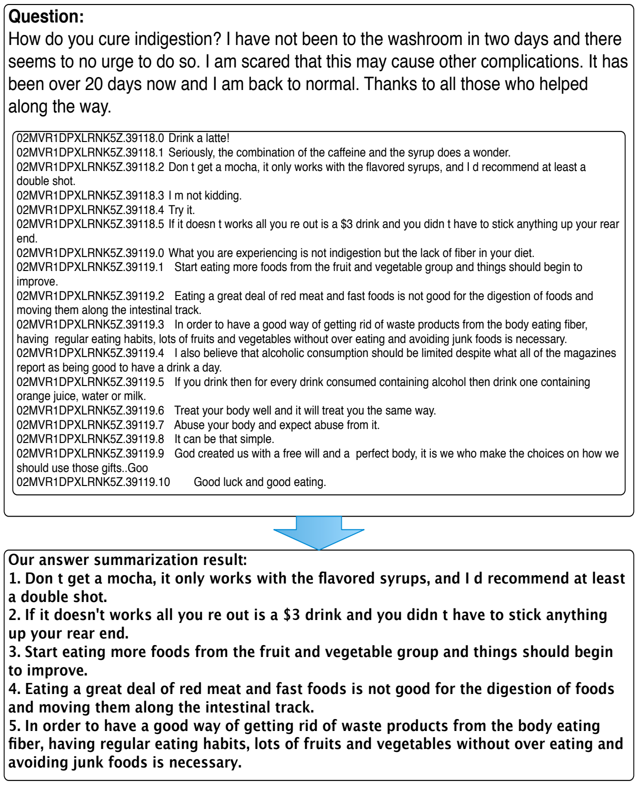
In this paper, we address the answer summarization problem

[*] These three authors contributed equally to the paper.

WSDM 2017, February 06 - 10, 2017, Cambridge, United Kingdom

ISBN 978-1-4503-4675-7/17/02...$15.00

DOI: http://dx.doi.org/10.1145/3018661.3018704



Question:
How do you cure indigestion? I have not been to the washroom in two days and there seems to no urge to do so. I am scared that this may cause other complications. It has been over 20 days now and I am back to normal. Thanks to all those who helped along the way.

02MVR1DPXLRNK5Z.39118.0 Drink a latte!
02MVR1DPXLRNK5Z.39118.1 Seriously, the combination of the caffeine and the syrup does a wonder.
02MVR1DPXLRNK5Z.39118.2 Don t get a mocha, it only works with the flavored syrups, and I d recommend at least a double shot.
02MVR1DPXLRNK5Z.39118.3 I m not kidding.
02MVR1DPXLRNK5Z.39118.4 Try it.
02MVR1DPXLRNK5Z.39118.5 If it doesn t works all you re out is a $3 drink and you didn t have to stick anything up your rear end.
02MVR1DPXLRNK5Z.39119.0 What you are experiencing is not indigestion but the lack of fiber in your diet.
02MVR1DPXLRNK5Z.39119.1 Start eating more foods from the fruit and vegetable group and things should begin to improve.
02MVR1DPXLRNK5Z.39119.2 Eating a great deal of red meat and fast foods is not good for the digestion of foods and moving them along the intestinal track.
02MVR1DPXLRNK5Z.39119.3 In order to have a good way of getting rid of waste products from the body eating fiber, having regular eating habits, lots of fruits and vegetables without over eating and avoiding junk foods is necessary.
02MVR1DPXLRNK5Z.39119.4 I also believe that alcoholic consumption should be limited despite what all of the magazines report as being good to have a drink a day.
02MVR1DPXLRNK5Z.39119.5 If you drink then for every drink consumed containing alcohol then drink one containing orange juice, water or milk.
02MVR1DPXLRNK5Z.39119.6 Treat your body well and it will treat you the same way.
02MVR1DPXLRNK5Z.39119.7 Abuse your body and expect abuse from it.
02MVR1DPXLRNK5Z.39119.8 It can be that simple.
02MVR1DPXLRNK5Z.39119.9 God created us with a free will and a perfect body, it is we who make the choices on how we should use those gifts..Goo
02MVR1DPXLRNK5Z.39119.10 Good luck and good eating.

Our answer summarization result:
1. Don t get a mocha, it only works with the flavored syrups, and I d recommend at least a double shot.
2. If it doesn't works all you re out is a $3 drink and you didn t have to stick anything up your rear end.
3. Start eating more foods from the fruit and vegetable group and things should begin to improve.
4. Eating a great deal of red meat and fast foods is not good for the digestion of foods and moving them along the intestinal track.
5. In order to have a good way of getting rid of waste products from the body eating fiber, having regular eating habits, lots of fruits and vegetables without over eating and avoiding junk foods is necessary.

Figure 1: An example answer summary for a question-answering thread about "how to cure indigestion." The answer summary extracts sentences from candidate answers.

in non-factoid CQA by utilizing a novel sparse-coding strategy. To address the shortness, sparsity, and diversity mentioned above, we proceed as follows. To tackle the *shortness* problem, previous works have proved that document expansion is effective in short text processing; Inspired by these works, in this paper, we employ an alternative strategy involving entity linking [29] and sentence ranking to collect and filter relevant information from Wikipedia. To tackle the challenge of *sparsity* in sentence presentation, we apply short text convolutional neural networks (CNNs) to model sentences given the input question. By considering the match between the question and the sentence, we learn an optimal sentence vector for each candidate sentence using stochastic gradient descent (SGD) and back propagation. To tackle the *diversity* challenge, we utilize sparse coding strategies that have been proved to be effective and efficient in summarizing sparse and diverse semantic units [18]. Given a question, the proposed sparse coding-based summarization framework can find a set of sentences which are used to reconstruct all the input sentences and hold the semantic diversity property in a very simple and elegant way, jointly considering candidate sentences from answers and sentences from auxiliary Wikipedia documents. We represent all the answer sentences and auxiliary Wikipedia sentences using distributed representations learnt from our CNNs' architecture, and utilize the coordinate descent method to optimize a loss function that jointly considers the candidate sentences' reconstruction error item and the Wikipedia sentences' reconstruction error item. We evaluate the performance of our proposed method on a benchmark dataset released by Tomasoni and Huang [43] using ROUGE metrics. Our experimental results show that the proposed sparse-coding based method is very effective in summarizing answers in non-factoid CQA, and moreover, significantly outperforms state-of-the-art baselines.

Our contributions in this paper can be summarized as follows:

- We address the task of summarizing answers to non-factoid questions in community question-answering by tackling the *shortness*, *sparsity* and *diversity* challenges.
- To address the *shortness* issue in the answer summarization task, we use a document expansion technique that can enrich short texts using Wikipedia articles.
- To tackle the *sparsity* problem in short texts, we apply convolutional neural networks to model sentences from candidate answers given the input question.
- To address the *diversity* challenge, we propose a new loss function within a sparse coding framework, by jointly considering the candidate sentences and auxiliary Wikipedia sentences as reconstruction items. We apply the coordinate descent method to optimize our proposed loss function.
- Experimental results on a benchmark dataset demonstrate the effectiveness of our proposed method and show that it significantly outperforms state-of-the-art baselines, in terms of ROUGE metrics.

In the remainder of the paper, we introduce related work in §2. We provide an overview of our method in §3 and a detailed account in §4. Then, §5 details our experimental setup and §6 presents the experimental results. Finally, §7 concludes the paper.

2. RELATED WORK

Our related work can be classed into two categories: multi-document summarization and community question-answering retrieval.

2.1 Multi-document summarization

Multi-document summarization (MDS) has been widely used to provide a brief digest of large numbers of relevant documents on the same topic [8]. MDS results can be divided into abstractive summaries and extractive summaries. Most existing work on MDS is based on the extractive format, where the target is to extract salient sentences to construct a summary. Both unsupervised and supervised based learning strategies have received a lot of attention. One of the most widely used unsupervised strategies is clustering with respect to the centroid of the sentences within a given set of documents [22, 31]. Many other recent publications on MDS employ graph-based ranking methods [8]. Wan and Yang [45] propose a theme-cluster strategy based on conditional Markov random walks. Celikyilmaz and Hakkani-Tur [4] consider the summarization task as a supervised prediction problem based on a two-step hybrid generative model, whereas the Pythy summarization system [44] learns a log-linear sentence ranking model by combining a set of semantic features. Several recent approaches to MDS view it as a combinatorial optimization problem, i.e., selecting a subset of sentences that maximizes an objective function under a length constraint [1]. Under this formulation, integer linear programming has been successfully applied to MDS [1, 25]. As to discriminative models, conditional random field-based algorithms [41] and structured SVM-based classifiers [17] have proved to be effective in extractive document summarization. Learning to rank models have also been employed for query-based MDS [40] and topic-focused MDS [59]. Recent studies on MDS have explored continuous sentence representations, including paragraph vectors [15], deep neural networks [37], and semantic volume maximization [54]. Li et al. [19] propose a neural generative model called Variational Auto-Encoders (VAEs) to describe the observed sentences and the corresponding latent semantic representations in MDS.

Recently, multi-document summarization is being applied to social media documents, e.g., tweets, blogs, and web forum posts [5, 7, 27, 32, 33]. Tweets summarization focuses on extracting a group of representative tweets out of a set of tweets [5, 35, 42]. MDS has been applied to web forums threads to select salient passages from candidate posts [33]. Previous work on blog summarization explores the effect of comments or social contexts on multi-document summarization [12, 18]. Li et al. [18] propose a reader-aware MDS paradigm that can generate compressive summaries jointly considering news documents and user comments. Answer summarization in community question-answering was first proposed by Zhou et al. [58], with a solution based on a head-modifier-relation triple representation of document content [11]. Tomasoni and Huang [43] exploit metadata to bias automatic answer summarization toward high quality information. However, all approaches so far neglect the *shortness*, *sparsity* and *diversity* challenges in answer summarization of non-factoid community question-answering.

2.2 Community question-answering retrieval

Research into community question-answering (CQA) is expanding due to the popularity of CQA on the web [57]. Recent research on community question-answering retrieval can be divided into factoid CQA research and non-factoid CQA research. Research on factoid community question-answering has seen a significant growth [9, 23, 24, 30, 38, 39, 53]. Feng et al. [9] present a question classifier using information distance. Several methods for answer sentence selection have been proposed with data from the TREC Question-Answering track [48, 51, 52]. Based on deep neural network architectures, several approaches have been proposed to select the best answer or question [30, 39, 53]. Severyn and Moschitti [39] propose a convolutional network for re-ranking pairs of a candidate answer and a question. Serban et al. [38] present a large question-answer pair corpus produced by applying a novel neural network architecture. Carmel et al. [3] take into account the syntactic function of the terms within CQA texts as an important factor affecting their importance for retrieval. Wang et al. [46] propose a retrieval-based automatic response model for short-text conversation. Omari et al. [30] promote CQA answers not only by their relevance to the question but also by diversity and novelty compared to other answers. Liu et al. [23] design a set function to re-rank questions given a user review. Zhao et al. [56] employ users' social networks for inferring a user model, and thus improve the performance of expert finding in CQA. Multimedia CQA has also received attention recently [24, 28].

Several approaches have already been proposed for the non-factoid CQA [43, 50]. Most recent work on non-factoid CQA retrieval focuses on the task of passage retrieval [6, 50]. Yang et al. [50] design semantic and context features for answer sentence retrieval in non-factoid CQA. Using a learning-to-rank approach, Chen et al. [6] find that searching relevant answer sentences can be leveraged in a feature-based framework by incorporating semantic features that make use of external resources.

Unlike previous research on non-factoid CQA, the aim of answer summarization is to explore all useful information from candidate answers given a question. As far as we know, very little work on this task has been reported in the literature.

3. OVERVIEW OF THE METHOD

Before outlining our approach to the answer summarization problem, we first list the notations we use in this paper. See Table 1.

For each non-factoid CQA thread, we suppose there exists a question q and a set of candidate answers $\mathcal{D} = \{d_1, d_2, \dots, d_D\}$, where each candidate answer $d \in \mathcal{D}$ can be represented as a set of sentences and D is the number of answers. We assume that, in total, there are S sentences in the CQA thread, i.e., $\mathcal{S} = \{s_1, s_2, \dots, s_S\}$. Our task is to extract sentences from $\mathcal{S}$ to construct the summary. To tackle the *shortness*, based on document expansion, we extend candidate answers by extracting S' relevant sentences, represented as $\mathcal{S}'$, from external knowledge material, e.g., Wikipedia. Totally, we have $\hat{S}$ sentences after document expansion, and we let $\hat{\mathcal{S}} = \mathcal{S} \cup \mathcal{S}'$. Using deep neural network models, each sentence in both $\mathcal{S}$ and $\mathcal{S}'$, i.e., $\hat{\mathcal{S}}$, can be represented as a m-dimensional vector. Therefore, after sentence vectorization, we have two vectors sets, $\mathcal{X}$ and $\mathcal{X}'$, corresponding to $\mathcal{S}$ and $\mathcal{S}'$, respectively.

Table 1: Glossary.

Symbol	Description
$\mathcal{D}$	candidate answers
$\mathcal{V}$	vocabulary in answers $\mathcal{D}$
$\mathcal{E}$	a knowledge base
$\mathcal{S}$	candidate sentences
$\mathcal{S}'$	auxiliary Wikipedia sentences
$\mathcal{X}$	candidate sentence vectors
$\mathcal{X}'$	auxiliary Wikipedia sentence vectors
$\mathcal{R}$	a summary of answers, $\mathcal{R} \subset \mathcal{S}$
$\mathcal{A}$	a saliency vector for candidate sentences
D	number of answers
S	number of sentences in answers
S'	number of auxiliary sentences from Wikipedia
$\hat{S}$	number of all sentences after document expansion
L	length limit of a summary of answers
s	a sentence
q	a question
e	an entity
v	a word $v \in \mathcal{V}$
x_s	a vector corresponding to a sentence s
sim_s	similarity between a sentence $s \in \hat{\mathcal{S}}$ and q
a_s	saliency score for a sentence s, $a_s \in \mathcal{A}$
m	dimensionality of a sentence vector
λ_s	parameter for a sentence s in sparse-coding

A sparse coding-based method is proposed to reconstruct the semantic space of an aspect, revealed by answer sentences $\mathcal{S}$. A saliency score $a_s \in [0, 1]$ is determined for a sentence s so as to define its contribution in constructing the semantic space of the topic from the answer content. For all candidate sentences $\mathcal{S}$, we determine a saliency vector $\mathcal{A} = [a_1, a_2, \dots, a_S]$, in which each item corresponds to a candidate sentence. Given a question q, a sentence set $\mathcal{S}$, and a target summary length L, the goal of answer summarization in CQA is to select a subset of sentences $\mathcal{R} \subseteq \mathcal{S}$ such that the total number of words in $\mathcal{R}$ is no more than L while maximizing the sum of their saliency scores, i.e., $\sum_{s \in \mathcal{R}} a_s$.

We provide a schematic overview of our method for performing answer summarization in Fig. 2. There are three main phases: (A) document expansion; (B) sentence representation; and (C) answer summarization. For all answers in $\mathcal{D}$, in phase (A) (see §4.1), we obtain a set of candidate sentences $\mathcal{S}$ with a set of auxiliary sentences $\mathcal{S}'$. In (B) (see §4.2), given the set of sentences $\mathcal{S} \cup \mathcal{S}'$, we employ convolutional neural networks to obtain feature vectors $\mathcal{X}$ and $\mathcal{X}'$ to represent candidate sentences and auxiliary sentences, respectively. Then, based on these sentence vectors, in (C) (see §4.3) we generate the final answer summary $\mathcal{R}$. We develop a sparse coding framework for this process: the first part in §4.3 optimizes the saliency vector $\mathcal{A}$, where each element in $\mathcal{A}$ is a saliency score for a candidate sentence; the second part in §4.3 generates the final answer summary $\mathcal{R}$ using maximal marginal relevance (MMR) [2].

4. METHOD

In this section, we detail the steps in our approach to answer summarization.

4.1 (A) Document expansion

4.1.1 Entity linking

Given a candidate answer d, the target of entity linking is to identify the entity e from a knowledge base $\mathcal{E}$ that is the most likely referent of each sentence in d. A *link candidate* $e \in \mathcal{E}$ links an *anchor* in d to a target g, where an anchor is an n-gram token in an answer and each target g is a Wikipedia article [29]. A target is identified by its unique title in Wikipedia. In the first step of entity linking, we identify as many *link candidates* as possible. We perform lexical matching of each n-gram anchor of answer d with the target texts found in Wikipedia, resulting in a set of *link candidates* $\mathcal{E}_d$ for each answer d. As the second step, after removing stop words, we combine words in each sentence $s \in d$, indicated as $\mathcal{V}_s$, and the given question q, indicated as $\mathcal{V}_q$, as a query $\mathcal{Q}_s = \{\mathcal{V}_s, \mathcal{V}_q\}$ together to search relevant Wikipedia articles. We employ BM25 [36] to rank link candidates in $\mathcal{E}_d$.

4.1.2 QA-based sentence ranking

Given a list of *link candidates*, we extract the most central sentences from the top three ranked Wikipedia articles. As in LexRank [8], Markov random walks are employed to iteratively optimize the ranked list of sentences, where each sentence may receives votes from other sentences. Suppose we collect a set of sentences, $\mathcal{S}_s$, after entity linking. Firstly, we build the similarity matrix M, where each item in M indicates the similarity between two sentences given $\mathcal{Q}_s = \{\mathcal{V}_s, \mathcal{V}_q\}$ as a query. Given two sentences s_i and s_j in $\mathcal{S}_s$, we compute the similarity $M_{i,j}$ between them as:

$$M_{i,j} = sim(s_i, s_j \mid \mathcal{Q}_s) / \sum_{s_x \in \mathcal{S}_s} sim(s_i, s_x \mid \mathcal{Q}_s). \tag{1}$$

At the beginning of the iterative process, an initial score for each sentence is set to $1/|\mathcal{S}_s|$, and at the l-th iteration, the score of s'_i is calculated as follows:

$$sco(s_i)^{(l)} = (1-\lambda_e) \sum_{i \neq j} M_{i,j} \cdot sco(s_j)^{(l-1)} + \lambda_e \frac{1}{|\mathcal{S}_s|}, \tag{2}$$

where $|\mathcal{S}_s|$ equals the number of sentences in Wikipedia documents that have been linked to the anchor text g (see §4.1.1) and the damping factor $\lambda_e = 0.15$. Then the transition matrix $\widetilde{M}$ equals to:

$$\widetilde{M} = (1-\lambda_e)M + \bar{e}\bar{e}^T \lambda_e / |\mathcal{S}_s|, \tag{3}$$

where $\bar{e}$ is a column vector with all items equal to 1. The iterative process will stop when it converges. Since $\widetilde{M}$ is a column stochastic matrix, the convergence of *sco* can be guaranteed, and a value of *sco* can be derived from the principle eigenvector of $\widetilde{M}$ [8]. We extract the top sentences, denoted as $\mathcal{E}_s$, from the resulting ranked list, and extend sentence s with those $|\mathcal{E}_s|$ sentences.

At the end of this document expansion phase, for the given set of candidate sentences $\mathcal{S}$ in a CQA thread totally we extract an auxiliary set of sentences $\mathcal{S}'$ from relevant Wikipedia articles.

4.2 (B) Sentence representation

Before sparse coding we transform the candidate sentences $\mathcal{S}$ and auxiliary sentences $\mathcal{S}'$ to basis vectors.

To tackle the *sparsity* in candidate answer sentences, inspired by [39], we apply the convolutional neural networks (CNNs) to map sentences to distributed representations, by jointly considering the syntactic and semantic information hidden in sentences and the given question.

4.2.1 Convolutional neural networks

In recent years, convolutional neural networks [14, 16] have been proved to be effective in many language processing tasks, such as text classification [14] and short text ranking [39], by efficiently embedding sentences into low-dimensional vector space.

Given a sentence $s \in \mathcal{S} \cup \mathcal{S}'$, for the i-th word v_i in s we set $z_i \in \mathbb{R}^k$ to be a k-dimensional word vector. Suppose there are n words in sentence s, then sentence s can be represented as a vector $z_{1:n}$:

$$z_{1:n} = z_1 \oplus z_2 \oplus \ldots \oplus z_n, \tag{4}$$

where $\oplus$ indicates the concatenation of two vectors. Let $z_{i:i+j}$ indicate the concatenation of word vectors z_i, z_{i+1}, …, z_{i+j-1}; for these j words within a window, we find a convolutional filter $w \in \mathbb{R}^{k \times j}$ to produce a new feature, i.e.,

$$c_i = f(w^T \cdot z_{i:i+j-1} + b).$$

Here, b is a bias parameter, and f is set to be a non-linear function, such as the hyperbolic tangent or rectified linear function, to learn non-linear decision boundaries. Using this method, we can produce a feature map $c = [c_1, c_2, \ldots, c_{i-j+1}]$. Then a max-over-time pooling operation is applied over the feature map c and the maximal element is extracted as the most important feature value, to be the feature corresponding to filter w. This pooling scheme naturally deals with variable sentence lengths. In CNNs, we can utilize multiple convolutional filters to get multiple features. Given m filters and the sentence s, CNNs are able to build a rich feature representation x_s, which forms the penultimate layer and is passed to a fully connected softmax layer. The softmax layer computes the probability distribution over the labels, i.e.,

$$p(y \mid x_s) = \frac{e^{x_s^T \theta_y}}{\sum_{y' \in \mathcal{Y}} e^{x_s^T \theta_{y'}}}, \tag{5}$$

where θ_y is a weight vector of the label y with $y = 1$ indicating that sentence s is included in the answer summary and $y = 0$ indicating not included, and $\mathcal{Y} = \{0, 1\}$ is the label set.

4.2.2 Feature vector generation

We consider each word in the given set of sentences as a k-dimensional vector, and initialize it by running `word2vec` tool [26] via the Yahoo Answers corpus.[1] For all sentences in $\mathcal{S} \cup \mathcal{S}'$ we first generate the corresponding sentence matrices respectively. As we have discussed in §4.2.1, we utilize CNNs with m filters in total to map a sentence $s \in \mathcal{S} \cup \mathcal{S}'$ to a m-dimensional vector x_s, and map the given question q to a vector x_q. Following [39], we calculate the similarity between these two vectors as follows:

$$sim(x_s, x_q) = x_q^T W x_s, \tag{6}$$

where W indicates a similarity weight parameter in CNNs that is optimized during model training. Combing the similarity between a sentence s and the given question q, we generate a joint layer

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l

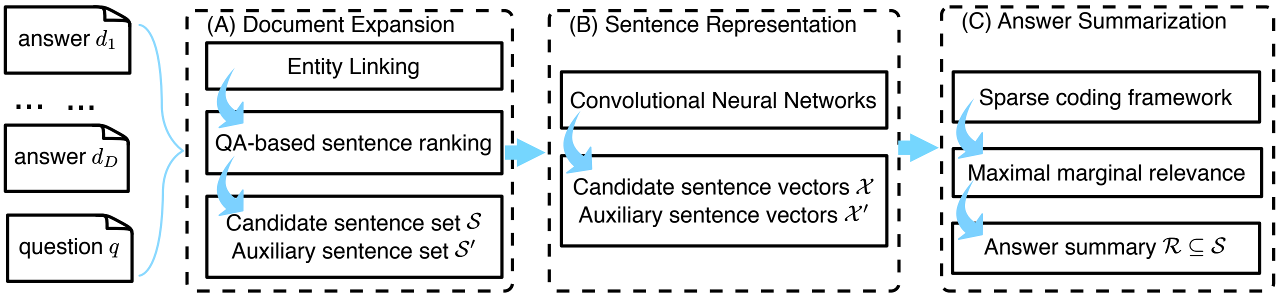

answer d_1
... ...
answer d_D
question q
(A) Document Expansion
Entity Linking
QA-based sentence ranking
Candidate sentence set $\mathcal{S}$
Auxiliary sentence set $\mathcal{S}'$
(B) Sentence Representation
Convolutional Neural Networks
Candidate sentence vectors $\mathcal{X}$
Auxiliary sentence vectors $\mathcal{X}'$
(C) Answer Summarization
Sparse coding framework
Maximal marginal relevance
Answer summary $\mathcal{R} \subseteq \mathcal{S}$

Figure 2: Overview of our approach to answer summarization. (A) indicates document expansion; (B) indicates sentence representation; and (C) refers to the answer summary generation process. Curved arrows indicate the output in each step; while straight arrows indicate processing directions.

that concatenates intermediate vectors into a new vector, i.e., $x_j = [x_s, sim(x_s, x_q), x_q]$. Based on this joint layer, the vector x_j is then passed through a fully connected hidden layer, which allows for modeling interactions between the components of the joined representation vector. We apply a non-linear function to transfer x_j into a new formulation, i.e.,

$$y_h = f(w^T \cdot x_j + b), \tag{7}$$

where w indicates an m-dimensional weight vector for the joint sentence vector x_j. Thereafter, y_h moves to the softmax layer, as described in §4.2.1, to generate a probabilistic distribution over class labels. Here, we set $y_s = 1$ if sentence s is included in the answer summary. We assume that auxiliary sentences for s have the same class label as s. We train the CNN model by optimizing an entropy-based loss function G, as follows:

$$G(\mathcal{Y}, \mathcal{S} \cup \mathcal{S}', q|\mathbf{\Theta}) = -\log \prod_{s \in \mathcal{S} \cup \mathcal{S}'} p(y_s|x_s, q) + \lambda_c \|\mathbf{\Theta}\|_2^2, \tag{8}$$

where $\mathbf{\Theta}$ indicates all parameters that are optimized by the CNN and λ_c is a free parameter. We optimize the parameters of the CNN using stochastic gradient descent (SGD), based on the back-propagation algorithm.

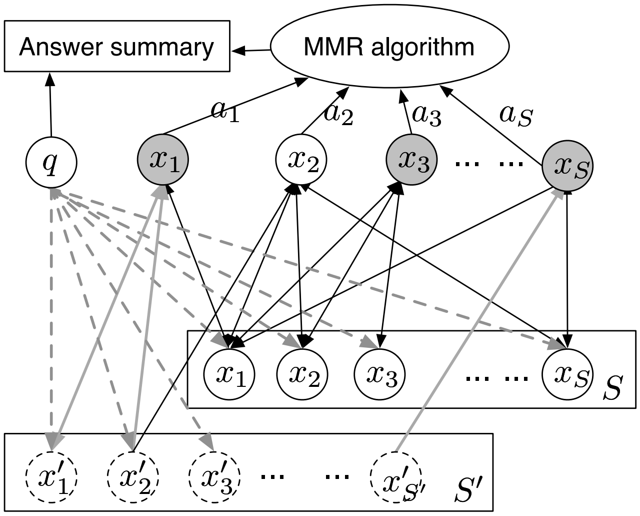


Figure 3: Overview of our sparse coding approach to non-factoid answer summarization. Circles indicate the question, candidate sentences, or auxiliary sentences; each dashed arrow indicates the correlations between the question and a sentence; each arrow reflects the correlations between two sentences.

4.3 (C) Answer summarization

We propose an unsupervised sparse coding-based framework to tackle the answer summarization problem in CQA. An overview of our framework is depicted in Fig. 3, in which boxes indicate the question or answer sentences. The grey boxes indicate sentences that are selected as the summary.

The aim of sparse coding is to find a linear combination of basis vectors to minimize the reconstruction error function [18]. Using the trained CNN model from §4.2, each sentence $s \in \mathcal{S} \cup \mathcal{S}'$ is mapped to an m-dimensional dense vector x_s. Given the candidate vectors $\mathcal{X}$ and auxiliary vectors $\mathcal{X}'$, the target of our sparse coding answer summarization strategy is to find a set of candidate sentence vectors $\mathcal{X}_\mathcal{R} \subset \mathcal{X}$ that can be used to reconstruct the candidate sentence vectors $\mathcal{X}$ and auxiliary sentence vectors $\mathcal{X}'$, i.e., the sentences included in the summary.

Following the idea of sparse coding, we regard each candidate sentence vector $x_i \in \mathcal{X}$ as a candidate basis vector, and all x_i's are employed to reconstruct the semantic space of the aspect, including $\mathcal{X}$ and $\mathcal{X}'$. We propose the following preliminary error formulation:

$$\frac{1}{2S} \sum_{i=1}^{S} \|x_i - \sum\nolimits_{j=1}^{|\mathcal{R}|} a_j x_j\|_2^2 + \tag{9}$$

$$\frac{1}{2S'} \sum_{i=1}^{S'} \|x'_i - \sum\nolimits_{j=1}^{|\mathcal{R}|} a_j x_j\|_2^2,$$

where the coefficient a_j is the saliency score for sentence vector x_j. To harness the characteristics of the summarization problem setting more effectively, we refine the preliminary error formulation. To utilize the information contained in the question, we compute the cosine similarity sim_i between vectors representing each candidate sentence x_i and the question vector x_q. Similarly, we introduce another parameter sim'_i to denote the cosine similarity between auxiliary sentence vector x'_i and x_q. Because the summary sentences are sparse, we impose a sparsity constraint, $\lambda_s > 0$, on the saliency score vector $\mathcal{A}$ using the L1-norm, as a scaling constant to determine its relative importance.

Putting things together, we arrive at the following loss function:

$$\begin{aligned} J = \min \frac{1}{2S} \sum\nolimits_{i=1}^{S} sim_i \cdot \|x_i - \sum\nolimits_{j=1}^{|\mathcal{R}|} a_j \cdot x_j\|_2^2 \qquad (10) \\ + \frac{1}{2S'} \sum\nolimits_{i=1}^{S'} sim'_i \cdot \|x'_i - \sum\nolimits_{j=1}^{|\mathcal{R}|} a_j \cdot x_j\|_2^2 + \lambda_s \cdot \|\mathcal{A}\|_1, \end{aligned}$$

subject to:

Algorithm 1: Coordinate descent algorithm for answer summarization

```
Input:
Question $q$, answer sentences $\mathcal{S} = \{s_1, s_2, ..., s_S\}$, cosine
similarity $sim_i$ between an answer sentence $s_i$ and $q$,
Wikipedia sentences $\mathcal{S}' = \{s'_1, s'_2, ..., s'_{S'}\}$, cosine similarity
$sim'_i$ between a Wikipedia sentence $s'_i$ and $q$, penalty
parameter $\lambda_s$, and stopping criterion $T$ and $\gamma$
Output: Saliency vector $\mathcal{A} \in \mathbb{R}^S$;
Initialize each $a \in \mathcal{A} \rightarrow 0$; $k \rightarrow 0$; $ite \rightarrow 0$;
Transfer sentences to basis vectors $\mathcal{X} = \{x_1, x_2, ..., x_S\}$;
while $ite < T$ do
    Reconstructing $\overline{x} = \sum_{i \in S} a_i^{ite} x_i$;
    Take partial derivatives:
    $\frac{\partial J}{\partial a_i} = \frac{1}{S} \sum_{j \in S} sim_j (x_j - \bar{x})^T \overrightarrow{x_i} - \frac{1}{S'} \sum_{j \in S} sim'_j (x'_j - \bar{x})^T \overrightarrow{x_i}$;
    Select the coordinate with maximum partial derivative:
    $i' = \arg\max_{i \in S} \left| \frac{\partial J}{\partial a_i} \right|$;
    Update the coordinate by soft-thresholding:
    $a_{i'}^{ite+1} = S_\lambda (a_{i'}^{ite} - \eta \frac{\partial J}{\partial a_{i'}})$;
    where $S_\lambda : a \rightarrow sign(a) \max(a - \lambda_s, 0)$;
    if $J_{\mathcal{A}^{ite+1}} - J_{\mathcal{A}^{ite}} < \gamma$ then
        break;
    end
    $ite \rightarrow ite + 1$;
end
```

1. $\forall a_j \in \mathcal{A}, a_j \geq 0$;
2. $\lambda_s > 0$;
3. $\sum_{s_j \in \mathcal{R}} |s_j| \leq L$;

where $|s_j|$ is the number of words in the sentence $s_j \in \mathcal{R}$. Based on our loss function, we formulate the task of answer summarization as an optimization problem. To learn the saliency vector $\mathcal{A}$, we utilize the coordinate descent method to iteratively optimize the target function about the saliency vector $\mathcal{A}$ until it converges. The details of the coordinate descent method are shown in Algorithm 1. Given a saliency score a_i for each candidate sentence $s \in \mathcal{S}$, we apply a maximal marginal relevance (MMR) algorithm [2] to select sentences according to their saliency score, which is shown in Algorithm 2. MMR incrementally computes the saliency-ranked list, and computes a maximal diversity ranking among candidate sentences.

In sum, our sparse coding strategy introduces several advantages. Firstly, sparse coding is a class of unsupervised methods, so no manual annotations for training data is needed. Secondly, the optimization procedure is modular leading to easily plug in different loss functions, so it can jointly minimize the reconstruct error of answer sentences and auxiliary sentences. Thirdly, our model incorporates semantic diversity naturally. In particular, it reduces the number of variables because the sparsity constraint can generate sparse salience scores, i.e., most of the sentences get a 0 score.

5. EXPERIMENTAL SETUP

Algorithm 2: Maximal marginal relevance for $\mathcal{R}$ Generation.

```
Input  :  $\mathcal{S}$, $\mathcal{X}$, $\mathcal{A}$, $L$;
Output:  $\mathcal{R}$;
Initialize: $\mathcal{R} = \emptyset$; Rank candidate sentences according to $\mathcal{A}$;
repeat
    Find a sentence $s$ by MMR with parameter $0 \leq \kappa \leq 1$, so
      that $s = \arg\max_{s \in \mathcal{S}/\mathcal{R}} [\kappa \cdot a_s - (1 - \kappa) \cdot \max sim_{s' \in \mathcal{R}} (x_s, x_{s'})]$;
    $\mathcal{R} = \mathcal{R} \cup s$;
until $|\mathcal{R}| > L$;
return $\mathcal{R}$.
```

In this section, we first formulate our research questions in §5.1 to guide our experiments. Then we describe our dataset in §5.2. The baselines and evaluation metrics are detailed in §5.3. The parameters for the proposed framework are listed in §5.4.

5.1 Research questions

To evaluate the effectiveness of our answer summarization method, we address the following research questions:

RQ1 How effective is our proposed method for answer summarization? Does it outperform state-of-the-art baselines?

RQ2 What is the impact of document expansion on the overall answer summarization task?

RQ3 What is the impact of different representations on the overall answer summarization task?

RQ4 What is the effect of the length constraint for the answer summary?

5.2 Dataset

We use a benchmark dataset released by Tomasoni and Huang [43]. Based on a Yahoo! Answers data collection with 89,814 question-answering threads, Tomasoni and Huang [43] removed factoid questions by applying a series of patterns for the identification of complex questions, and only leave non-factoid question-answering threads in the following patterns:

- Why, What is the reason [. . .]
- How to, do, does, did [. . .]
- How is, are, were, was, will [. . .]
- How could, can, would, should [. . .].

The ground truth of all these QA summaries is manually generated by selecting sentences subject to a fixed length limit of 250 words by human experts. In total, the dataset used in our experiments includes 100 non-factoid questions, 361 answers, 2,793 answer sentences, 59,321 words and 275 manually generated summaries.

5.3 Baselines and evaluation metrics

We write **SPQAS** for our sparse-coding based method as described in §4. To assess the effectiveness of our document expansion strategy, we write **SPQAS-AT** for our sparse-coding method restricted to the original answer texts only. We compare sentence representations based on our CNN-based architecture (in phase B) against word2vec as an alternative representation (**SPQAS +word2vec**). We perform comparisons between our methods and the following state-of-the-art baselines:

- **MaQAS**, [43], a metadata-aware question-answering summarization method;
- **LexRank** [8], a widely-used multi-document summarization model;
- **SVM**, A traditional supervised learning method that treats

question-answering summarization as a binary classification task;

- **CNN** [39], a state-of-the-art neural network architecture for question answer re-ranking;
- **BestAns**, a baseline that uses the top-ranked answer of the QA thread;
- **Random**, which extracts sentences randomly.

Following [43], we apply Porter stemming and set the length limit of the CQA answer summary to 250 words.

We adopt the ROUGE evaluation metrics [21], a widely-used recall-oriented group of metrics for document summarization that evaluates the overlap between a gold standard and candidate selections. We use ROUGE-1 (*unigram based method*), ROUGE-2 (*bigram based method*) and ROUGE-L (*longest common subsequence*) in our experiments. Statistical significance of observed differences between the performance of two approaches is tested using a two-tailed paired t-test and is denoted using $^\blacktriangle$ (or $^\blacktriangledown$) for strong significance for $\alpha = 0.01$; or $^\vartriangle$ (or $^\triangledown$) for weak significance for $\alpha = 0.05$.

5.4 Parameter settings

For the document expansion process (phase A), we set $\lambda_e = 0.15$, and extract 3 Wikipedia articles for each candidate sentence. For the vector generation process (phase B), we employ rectified linear units as the non-linear activation function, and the number of convolutional filter maps m is set to be 100. For the sparse coding model (phase C), we set the stop criteria $T = 500$, $\gamma = 0.0001$, and the learning rate $\eta = 1$ in Algorithm 1. For the sparsity item penalty, we set $\lambda_s = 0.005$.

6. EXPERIMENTAL RESULTS

In this section, we first consider the overall performance of our answer summarization method, SPQAS. Then we examine the impact of document expansion and of different sentence representations. Last we report on the impact of the length limit on answer summaries.

6.1 Overall performance

Firstly, we provide answer for **RQ1**. The evaluation results for non-factoid answer summarization using different models are summarized in Table 2. We find that SPQAS outperforms the other baselines, and significantly outperforms MaQAS in terms of all three ROUGE metrics. BestAns achieves the worst score since it suffers from a lack of diversity, which explains why we need to summarize all the answers for a given question. Since generic summarization methods neglect the correlation between a question and answers, the LexRank method does not perform well in the CQA answer summarization task. SPQAS also defeats SVM and CNN, two state-of-art supervised methods. While the difference between supervised methods (SVM and CNN) and unsupervised methods (LexRank and MaQAS) is always significant.

Table 2: Overall performance of all methods in answer summarization on ROUGE-1, ROUGE-2, and ROUGE-L. Statistically significant differences between SPQAS and the best performing baseline are marked in the upper right hand corner of SPQAS's scores.

Method	ROUGE-1	ROUGE-2	ROUGE-L
BestAns	0.473	0.390	0.463
Random	0.534	0.418	0.525
LexRank	0.584	0.438	0.565
MaQAS	0.674	0.588	0.663
SVM	0.699	0.633	0.692
CNN	0.700	0.614	0.690
SPQAS	**0.766**$^\blacktriangle$	**0.646**$^\blacktriangle$	**0.753**$^\blacktriangle$

We take a closer look at SPQAS vs. the three top performing baselines. Against MaQaS, SPQAS offers relative performance improvements of 13.6%, 9.8%, and 13.6%, respectively, for the ROUGE-1, ROUGE-2 and ROUGE-L metrics. SPQAS outperforms SVM by 9.6%, 2.1%, and 8.8%, respectively, for ROUGE-1, ROUGE-2 and ROUGE-L. And the relative gains for SPQAS over CNN are 9.4%, 5.2%, and 9.1%. The two supervised methods, SVM and CNN, both perform better than the unsupervised MaQAS method, and the difference with MaQaS is statistically significant. Threads on which SPQAS is outperformed by one of the top 3 performing baselines tend to have longer questions.

6.2 Document expansion

To answer **RQ2**, i.e., to determine the impact of document expansion, we consider a variation of SPQAS that is restricted to the original answer texts only (essentially skipping phase A). See Table 3 for the results. As can be seen, the expanded documents lead to a 3% increase for all three ROUGE metrics, and SPQAS statistically significantly outperforms SPQAS-AT, which confirms the merit of utilizing document expansion in SPQAS. To gain a better understanding of the differences, we visualize the results in Fig. 4. Each of the red dots and the blue crosses denotes one question answer thread in our dataset. A QA thread is represented as a red dot when SPQAS outperforms SPQAS-AT on the thread, otherwise a blue cross. Clearly, the red dots outnumber the blue crosses for each metric, confirming that document expansion is helpful, although it is not a silver bullet: threads that have multiple and implicit aspects in answers are typically favored by SPQAS-AT, whereas many threads that have only few explicit aspect in answers cannot perform well after document expansion.

Table 3: Effectiveness of document expansion. SPQAS-AT denotes SPQAS restricted to the original answer texts only. Statistically significant differences between SPQAS and SPQAS-AT are marked in the upper right hand corner of SPQAS's scores.

Method	ROUGE-1	ROUGE-2	ROUGE-L
SPQAS-AT	0.738	0.604	0.723
SPQAS	0.766$^\blacktriangle$	0.646$^\blacktriangle$	0.753$^\blacktriangle$

We further analyze the correlation between the length of answer texts and ROUGE scores; see Fig. 5. The average performance of the two methods appears to be similar when the total length of an answer thread is less than 800 words. SPQAS, i.e., with expanded texts, performs much better than SPQAS-AT with original answer texts only when the total length of the answer thread exceeds 800 words. As an aside, as the length of an answer thread increases, the performance of both methods, whether using expansions or not, decreases monotonously in terms of all metrics.

6.3 Sentence representations

To answer **RQ3**, we compare sentence representations based on our CNN-based architecture (in phase B) and word2vec as an alternative representation. Recall that SPQAS uses CNN. As an alternative, we consider SPQAS+word2vec, which produces sentence vectors using the idf-weighted sum of word vectors proposed in [26]

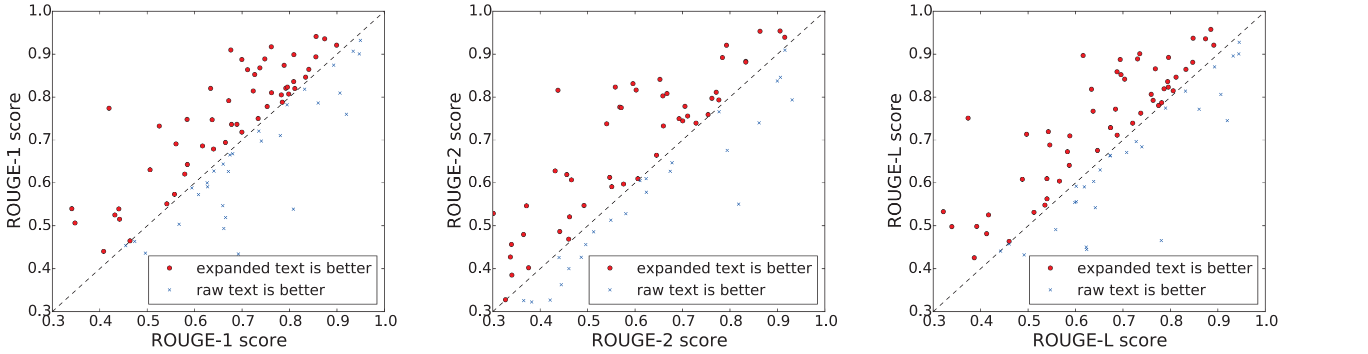


Figure 4: The ROUGE-1, ROUGE-2, and ROUGE-L scores of SPQAS and SPQAS without Wikipedia expansion. The red dots denote question answer threads where SPQAS outperforms SPQAS-Wiki, and vice versa for the blue crosses.

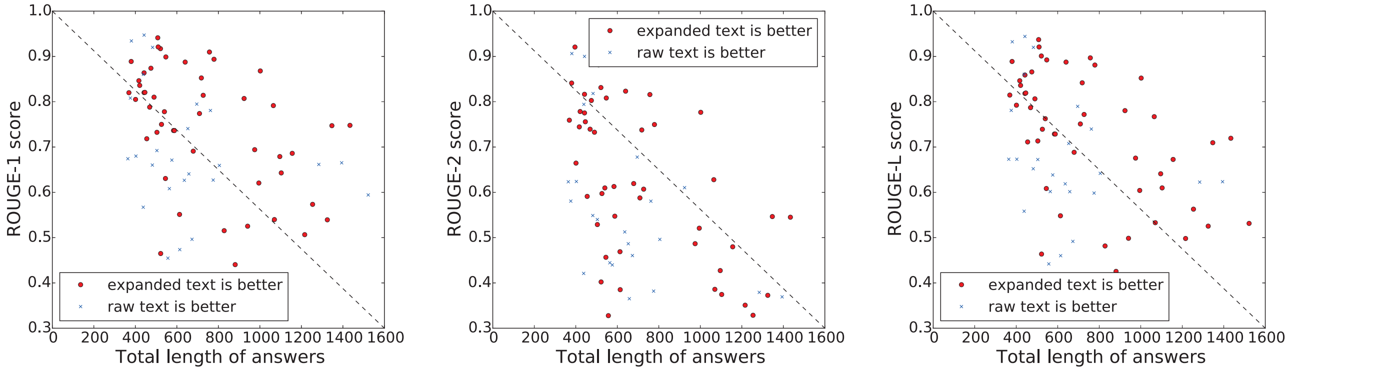


Figure 5: The correlation between the length of answer texts and ROUGE-1, ROUGE-2, and ROUGE-L scores. The red dots denote question answer threads where SPQAS are better than SPQAS-Wiki, and vice versa for the blue crosses.

for the questions, candidate answer sentences, and auxiliary Wikipedia sentences. These sentence vectors are subsequently fed to the rest of the SPQAS pipeline (phase C) to generate answer summaries. The ROUGE scores for the methods based on these two kinds of sentence vectors are listed in Table 4. The results based on idf-weighted word2vec sentence vectors are satisfactory, beating the baselines listed in Table 2, but the CNN-based sentence vectors always perform statistically significantly better. This illustrates that the CNN-based sentence modeling architecture has a positive impact on sentence representation. And it also suggests that the SPQAS pipeline offers good robustness and scalability.

Table 4: ROUGE results of different sentence representations. SPQAS+word2vec denotes SPQAS in which CNN-based sentence representations have been replaced by word2vec-based sentence representations. Statistically significant differences between SPQAS and SPQAS+word2vec are marked in the upper right hand corner of SPQAS's scores.

Method	ROUGE-1	ROUGE-2	ROUGE-L
SPQAS+word2vec	0.729	0.612	0.717
SPQAS	0.766$^{\blacktriangle}$	0.646$^{\blacktriangle}$	0.753$^{\blacktriangle}$

6.4 Length analysis

To answer **RQ4**, we vary the maximum allowed length of the answer summary and observe the performance of SPQAS; see Fig. 6, where we plot ROUGE-1 scores (recall, precision and F) for different length upper bounds. Generally, as we increase the upper bound on the length, recall rises gradually and precision decreases simultaneously, which is not surprising. A good balance between recall and precision can be found when the length constraint is set to between 200 and 300 words: this is where F-measure plateaus, with gains in recall balanced by losses in precision. Since short summaries (of equal quality) are preferred over longer ones, the

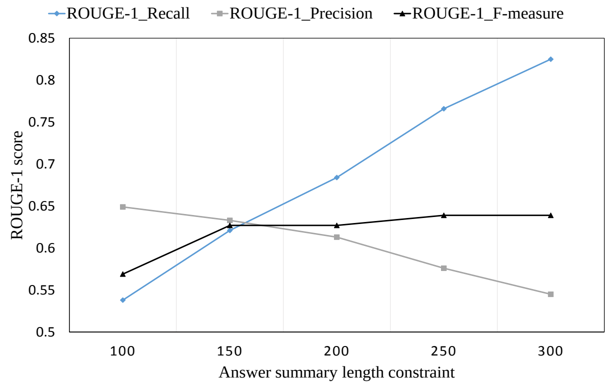


Figure 6: Correlation between summary length limit and ROUGE-1 scores.

main lesson here is that our default length of 250 words is a sensible choice, offering both enough information and good readability.

7. CONCLUSION AND FUTURE WORK

We consider the task of answer summarization for non-factoid community question-answering. We identify the main challenges: the shortness and sparsity of answers, and the diverse aspect distribution. We utilize a document expansion strategy to enrich short texts using Wikipedia articles. After generating the distributed sentence representations via convolutional neural networks, we prose a sparse coding-based framework to predict the saliency score of each candidate sentence, by jointly considering the candidate sentences and auxiliary Wikipedia sentences as reconstruction items in a novel loss function. We utilize a coordinate descent method to optimize our target loss function. We have demonstrated the effectiveness of our proposed method by showing significant improvements over multiple baselines tested with a benchmark dataset.

Limitations of our work include its ignorance of syntactic information and of semantic dependencies among answers. We also find that our method does not perform so well on answers with only few explicit aspect. Both are obvious areas for future work.

As to other areas of future work, given the positive impact of our sentence modeling techniques (in phase (B) of SPQAS), we believe there is more room to capture and exploit semantic information to further improve our summarization results. Secondly, transferring our method to the cross-language answer summarization and online answer summarization setting should be given new insights. Thirdly, it is interesting to consider a personalized summarization task on question-answering communities, based on dynamic clustering approaches [20, 55]. Fourthly, generating concise and abstractive answer summaries is worth considering. Finally, supervised and semi-supervised learning can be considered for improving the performance of answer summarization.

Acknowledgements. This work is supported by the Natural Science Foundation of China (61672322, 61672324, 61272240), the Natural Science Foundation of Shandong province (2016ZRE274-68, ZR2012FM037), the Fundamental Research Funds of Shandong University, Ahold Delhaize, Amsterdam Data Science, Blendle, the Bloomberg Research Grant program, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the ESF Research Network Program ELIAS, the Royal Dutch Academy of Sciences (KNAW) under the Elite Network Shifts project, the Microsoft Research Ph.D. program, the Netherlands eScience Center under project number 027.012.105, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.001.116, HOR-11-10, CI-14-25, 652.002.001, 612.-001.551, 652.001.003, 314-98-071, the Yahoo Faculty Research and Engagement Program, Yandex, and the EPSRC grant EP /K031-953/1 ("Early-Warning Sensing Systems for Infectious Diseases"). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

8. REFERENCES

[1] F. Boudin, H. Mougard, and B. Favre. Concept-based summarization using integer linear programming: From concept pruning to multiple optimal solutions. In *EMNLP*, 2015.

[2] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.

[3] D. Carmel, A. Mejer, Y. Pinter, and I. Szpektor. Improving term weighting for community question answering search using syntactic analysis. In *CIKM*, 2014.

[4] A. Celikyilmaz and D. Hakkani-Tur. A hybrid hierarchical model for multi-document summarization. In *ACL*, 2010.

[5] D. Chakrabarti and K. Punera. Event summarization using tweets. In *ICWSM*, 2011.

[6] R.-C. Chen, D. Spina, W. B. Croft, M. Sanderson, and F. Scholer. Harnessing semantics for answer sentence retrieval. In *the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval*, 2015.

[7] Y. Duan, F. Wei, C. Zhumin, Z. Ming, and Y. Shum. Twitter topic summarization by ranking tweets using social influence and content quality. In *COLING*, 2012.

[8] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[9] G. Feng, K. Xiong, Y. Tang, A. Cui, J. Bai, H. Li, Q. Yang, and M. Li. Question classification by approximating semantics. In *WWW*, 2015.

[10] K. Hacioglu and W. Ward. Question classification with support vector machines and error correcting codes. In *HLT-NAACL*, 2003.

[11] E. Hovy, C.-Y. Lin, and L. Zhou. A be-based multi-document summarizer with sentence compression. In *Proceedings of Multilingual Summarization Evaluation*, 2005.

[12] M. Hu, A. Sun, and E.-P. Lim. Comments-oriented document summarization: understanding documents with readers' feedback. In *SIGIR*, 2008.

[13] M. Keikha, J. H. Park, and W. B. Croft. Evaluating answer passages using summarization measures. In *SIGIR*. ACM, 2014.

[14] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[15] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

[17] L. Li, K. Zhou, G. Xue, H. Zha, and Y. Yu. Enhancing diversity, coverage and balance for summarization through structure learning. In *WWW*, 2009.

[18] P. Li, L. Bing, W. Lam, H. Li, and Y. Liao. Reader-aware multi-document summarization via sparse coding. In *IJCAI*, 2015.

[19] P. Li, Z. Wang, W. Lam, Z. Ren, and L. Bing. Salience estimation via variational auto-encoders for multi-document summarization. In *AAAI*, 2017.

[20] S. Liang, E. Yilmaz, and E. Kanoulas. Dynamic clustering of streaming short documents. In *KDD*, 2016.

[21] C. Lin. Rouge: A package for automatic evaluation of summaries. In *ACL*, 2004.

[22] C.-Y. Lin and E. Hovy. From single to multi-document summarization: A prototype system and its evaluation. In *ACL*, 2002.

[23] M. Liu, Y. Fang, D. H. Park, X. Hu, and Z. Yu. Retrieving non-redundant questions to summarize a product review. In *SIGIR*, 2016.

[24] L. Ma, Z. Lu, and H. Li. Learning to answer questions from image using convolutional neural network. In *AAAI*, 2016.

[25] R. McDonald. A study of global inference algorithms in multi-document summarization. In *ECIR*, 2007.

[26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, pages 3111–3119, 2013.

[27] J. Nichols, J. Mahmud, and C. Drews. Summarizing sporting events using twitter. In *IUI*, 2012.

[28] L. Nie, M. Wang, Y. Gao, Z.-J. Zha, and T.-S. Chua. Beyond text qa: Multimedia answer generation by harvesting web information. *IEEE Transactions on Multimedia*, 15(2): 426–441, 2013.

[29] D. Odijk, E. Meij, and M. de Rijke. Feeding the second screen: semantic linking based on subtitles. In *OAIR*, 2013.

[30] A. Omari, D. Carmel, O. Rokhlenko, and I. Szpektor. Novelty based ranking of human answers for community questions. In *SIGIR*, 2016.

[31] D. Radev et al. MEAD–A platform for multidocument multilingual text summarization. In *LREC*, 2004.

[32] Z. Ren and M. de Rijke. Summarizing contrastive themes via hierarchical non-parametric processes. In *SIGIR*, 2015.

[33] Z. Ren, J. Ma, S. Wang, and Y. Liu. Summarizing web forum threads based on a latent topic propagation process. In *CIKM*, 2011.

[34] Z. Ren, M.-H. Peetz, S. Liang, W. van Dolen, and M. de Rijke. Hierarchical multi-label classification of social text streams. In *SIGIR*, 2014.

[35] Z. Ren, O. Inel, L. Aroyo, and M. de Rijke. Time-aware multi-viewpoint summarization of multilingual social text streams. In *CIKM*, 2016.

[36] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at TREC-3. In *TREC*, 1994.

[37] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. In *EMNLP*, 2015.

[38] I. V. Serban et al. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *ACL*, 2016.

[39] A. Severyn and A. Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*, 2015.

[40] C. Shen and T. Li. Learning to rank for query-focused multi-document summarization. In *ICDM*, 2011.

[41] D. Shen, J. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *IJCAI*, 2007.

[42] L. Shou, Z. Wang, K. Chen, and G. Chen. Sumblr: continuous summarization of evolving tweet streams. In *SIGIR*, 2013.

[43] M. Tomasoni and M. Huang. Metadata-aware measures for answer summarization in community question answering. In *ACL*, 2010.

[44] K. Toutanova, C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende. The pythy summarization system: Microsoft research at DUC 2007. In *DUC*, 2007.

[45] X. Wan and J. Yang. Multi-document summarization using cluster-based link analysis. In *SIGIR*, 2008.

[46] H. Wang, Z. Lu, H. Li, and E. Chen. A dataset for research on short-text conversations. In *EMNLP*, 2013.

[47] M. Wang. A survey of answer extraction techniques in factoid question answering. *Computational Linguistics*, 1(1), 2006.

[48] M. Wang, N. A. Smith, and T. Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, 2007.

[49] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen. Cqarank: jointly model topics and expertise in community question answering. In *CIKM*, 2013.

[50] L. Yang, Q. Ai, D. Spina, R.-C. Chen, L. Pang, W. B. Croft, J. Guo, and F. Scholer. Beyond factoid QA: Effective methods for non-factoid answer sentence retrieval. In *ECIR*, 2016.

[51] X. Yao, B. Van Durme, C. Callison-Burch, and P. Clark. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*, 2013.

[52] W.-t. Yih, M.-W. Chang, C. Meek, and A. Pastusiak. Question answering using enhanced lexical semantic models. In *ACL*, 2013.

[53] J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, and X. Li. Neural generative question answering. In *IJCAI*, 2016.

[54] D. Yogatama, F. Liu, and N. A. Smith. Extractive summarization by maximizing semantic volume. In *EMNLP*, 2015.

[55] Y. Zhao, S. Liang, Z. Ren, J. Ma, E. Yilmaz, and M. de Rijke. Explainable user clustering in short text streams. In *SIGIR*, 2016.

[56] Z. Zhao, L. Zhang, X. He, and W. Ng. Expert finding for question answering via graph regularized matrix completion. *IEEE Transactions on Knowledge and Data Engineering*, 27 (4):993–1004, 2015.

[57] G. Zhou, T. He, J. Zhao, and P. Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *ACL*, 2015.

[58] L. Zhou, C.-Y. Lin, and E. Hovy. Summarizing answers for complicated questions. In *LREC*, 2006.

[59] Y. Zhu, Y. Lan, J. Guo, P. Du, and X. Cheng. A novel relational learning-to-rank approach for topic-focused multi-document summarization. In *ICDM*, 2013.