Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Salience Estimation via Variational Auto-Encoders for Multi-Document Summarization*

Piji Li,† Zihao Wang,† Wai Lam,† Zhaochun Ren,‡ Lidong Bing§
†Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Hong Kong
‡University College London, London, UK
§ AI Platform Department, Tencent Inc., Shenzhen, China
†{pjli, zhwang, wlam}@se.cuhk.edu.hk, ‡zhaochun.ren@ucl.ac.uk, §lyndonbing@tencent.com

Abstract

We propose a new unsupervised sentence salience framework for Multi-Document Summarization (MDS), which can be divided into two components: latent semantic modeling and salience estimation. For latent semantic modeling, a neural generative model called Variational Auto-Encoders (VAEs) is employed to describe the observed sentences and the corresponding latent semantic representations. Neural variational inference is used for the posterior inference of the latent variables. For salience estimation, we propose an unsupervised data reconstruction framework, which jointly considers the reconstruction for latent semantic space and observed term vector space. Therefore, we can capture the salience of sentences from these two different and complementary vector spaces. Thereafter, the VAEs-based latent semantic model is integrated into the sentence salience estimation component in a unified fashion, and the whole framework can be trained jointly by back-propagation via multi-task learning. Experimental results on the benchmark datasets DUC and TAC show that our framework achieves better performance than the state-of-the-art models.

Introduction

Multi-Document Summarization (MDS), aiming at automatically generating a brief, well-organized summary for a topic which describes an event with a set of documents from different sources, has been studied extensively. (Goldstein et al. 2000; Erkan and Radev 2004; Wan, Yang, and Xiao 2007; Nenkova and McKeown 2012; Min, Chew, and Tan 2012; Bing et al. 2015). Summarization approaches can be grouped into two classes: extraction-based methods and abstraction-based methods. For both classes, salience estimation plays a critical role in improving the performance. Considering the scalability restriction of labeling MDS datasets, some works adopt unsupervised data reconstruction methods to conduct salience estimation and achieve comparable results (He et al. 2012; Liu, Yu, and Deng 2015; Yao, Wan, and Xiao 2015; Li et al. 2015; Ren et al. 2016; Song et al. 2017). After investigating these works, we observe that they mainly use Bag-of-Words (BoWs) vectors in sentence representation and reconstruction loss function. On the other hand, some research works (Le and Mikolov 2014; Kim 2014) have demonstrated that distributed representations outperform BoWs in modeling sentence and document semantics. In this paper, instead of using BoWs vectors, we explore a distributed representation for modeling the latent semantics of sentences for the MDS task. We propose a framework based on probabilistic generative models to describe the observed sentences and latent semantic vectors.

Given a topic (event) composed of a set of documents, we build a distributed latent semantic vector to model each sentence with a generative framework, where each sentence is generated from an unobserved latent semantic space. Another characteristic is that the generative process employs a neural network conditioned on the input text approximating the distributions over the latent semantic vector. Markov Chain Monte Carlo (MCMC) sampling and Variational Inference (VI) are the most common methods used in generative models (Jordan et al. 1999; Wainwright and Jordan 2008; Blei, Kucukelbir, and McAuliffe 2016). Nevertheless, some integrals of the marginal likelihood are intractable due to the continuous latent variables and neural network based generative modeling. Standard variational inference methods such as mean-field algorithms (Xing, Jordan, and Russell 2002) cannot be used. Moreover, MCMC based sampling methods are too slow to extend to large-scale machine learning tasks. Recently, Variational Autoencoders (VAEs) (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014) and Generative Adversarial Networks (GANs) (Goodfellow et al. 2014; Radford, Metz, and Chintala 2015) have been proposed that can handle the inference problem associated with complex generative modeling frameworks. In our work, we employ VAEs as the basic framework for the generative model. In fact, some works (Miao, Yu, and Blunsom 2015; Chung et al. 2015) have demonstrated that VAEs outperform the general Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) in generating high-level semantic representations.

To address the sentence salience estimation problem for MDS, we propose an unsupervised data reconstruction framework which jointly reconstructs the latent semantic

*The work described in this paper is supported by grants from the Research and Development Grant of Huawei Technologies Co. Ltd (YB2015100076/TH1510257) and the Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14203414).

space and the observed term vector space. The basic idea behind the data reconstruction is that each original sentence can be reconstructed using a linear combination of several other representative sentences. These representative sentences are able to capture different aspects implied in the event, such as "what happened", "damages", "countermeasures", etc. We name the vectors which are used to represent the aspect sentences as aspect vectors. Then, salience estimation can be conducted during the reconstruction process using aspect vectors. Based on the spirit of generative model and data reconstruction process, we design several latent aspect vectors and use them to reconstruct the whole original latent semantic space. In parallel with such idea, we also design some aspect term vectors which are used to reconstruct the original observed term vector space. Thereafter, the VAEs-based latent semantic model is integrated into the sentence salience estimation component in a unified fashion, and the whole framework can be trained jointly by back-propagation via multi-task learning. After estimating the sentence salience, we employ a phrase merging based unified optimization framework to generate a final summary.

Our contributions are as follows: (1) We propose a VAEs-based generative model to conduct the latent semantic modeling for sentences. To the best of our knowledge, there is no other work exploring the use of VAEs for summarization related tasks. (2) In our framework, salience estimation is conducted by jointly considering the latent semantic space and the observed input term vector space, which can draw richer information from these two different and complementary spaces. (3) The VAEs-based generative model and salience estimation component are integrated into a unified framework, which can be trained simultaneously in a multi-task learning fashion using back-propagation. (4) Experimental results on the benchmark data sets DUC and TAC show that our framework achieves better performance than the state-of-the-art models.

Overview of Our Proposed Framework

As shown in Figure 1, our sentence salience framework has two main components: (1) latent semantic modeling; (2) salience estimation. To tackle the latent semantic modeling problem, a VAEs-based generative model is designed to project sentences from the term vector space to the latent semantic space. Consider a dataset $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^n\}$ consisting of n sentences from all the documents in a topic (event), represented by BoWs term vectors. The left part of Figure 1 illustrates a VAEs-based component implemented as a feed-forward neural network for associating a latent semantic vector $\mathbf{z}^i \in \mathbb{R}^K$ with each sentence $\mathbf{x}^i \in \mathbb{R}^{|V|}$, where V is the term dictionary. Based on generative modeling, a latent semantic vector $\mathbf{z}^i \in \mathbb{R}^K$ is generated from some prior distribution $p_\theta(\mathbf{z}^i)$. Then the sentence term vector $\mathbf{x}^i$ is generated from a conditional distribution $p_\theta(\mathbf{x}^i|\mathbf{z}^i)$. To find the parameter θ, the reparameterization trick is applied to obtain a differentiable estimator of the variational lower bound. Then back-propagation can be employed to train the neural network. For sentence salience estimation, we propose **VAEs-A**, an unsupervised data reconstruction framework with the alignment mechanism for aspect vector discovery. The general idea is shown in the right part of Figure 1. Note that $\{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^n\}$ and $\{\mathbf{z}^1, \mathbf{z}^2, \cdots, \mathbf{z}^n\}$ are exactly the same vectors as those depicted in the left part of Figure 1. We design some latent aspect vectors $\mathbf{S_z}$ for capturing the latent aspect information of a topic. The corresponding aspect term vectors $\mathbf{S_x}$ are generated according to the conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$. By reconstructing the original sentence term vectors $\mathbf{X}$ and the corresponding latent semantic vectors $\mathbf{Z}$ using $\mathbf{S_x}$ and $\mathbf{S_z}$ jointly, the sentence salience can be estimated from the optimized coefficient matrix. Finally, inspired by (Bing et al. 2015), a phrase-based unified numerical optimization framework is employed to conduct the summary generation.

Sentence Salience Framework

Latent Semantic Modeling

VAEs-based latent semantic modeling can be viewed as an instance of unsupervised learning, which can be divided into two parts: inference (variational-encoder) and generation (variational-decoder). Recall that the dictionary is V. As shown in the left part of Figure 1, for each sentence term vector $\mathbf{x} \in \mathbb{R}^{|V|}$, the variational-encoder can map it to a latent semantic vector $\mathbf{z} \in \mathbb{R}^K$, which can be used to generate the original sentence term vector via the variational-decoder component. The target is to maximize the probability of each $\mathbf{x}$ in the dataset based on the generation process according to:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z} \tag{1}$$

For the purpose of solving the intractable integral of the marginal likelihood as shown in Equation 1, a recognition model $q_\phi(\mathbf{z}|\mathbf{x})$ is introduced as the approximation to the intractable of true posterior $p_\theta(\mathbf{z}|\mathbf{x})$. It is obvious that $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ can be regarded as a probabilistic encoder and decoder respectively. The recognition model parameters ϕ and the generative model parameters θ can be learnt jointly. The aim is to reduce the Kulllback-Leibler divergence (KL) between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z}|\mathbf{x})$:

$$\begin{aligned} D_{KL}[q_\varphi(z|x)\|p_\theta(z|x)] &= \int_z q_\varphi(z|x) \log \frac{q_\varphi(z|x)}{p_\theta(z|x)} dz \\ &= \mathbb{E}_{q_\varphi(z|x)}[\log q_\varphi(z|x) - \log p_\theta(z|x)] \end{aligned} \tag{2}$$

By applying Bayes rule to $p_\theta(\mathbf{z}|\mathbf{x})$:

$$\begin{aligned} D_{KL}[q_\varphi(z|x)\|p_\theta(z|x)] = \log p_\theta(x) \ + \\ \mathbb{E}_{q_\varphi(z|x)}[\log q_\varphi(z|x) - \log p_\theta(x|z) - \log p_\theta(z)] \end{aligned} \tag{3}$$

We can extract $\log p_\theta(\mathbf{x})$ from the expectation, transfer the expectation term $\mathbb{E}_{q_\varphi(z|x)}$ back to KL-divergence, and rearrange all the terms. Then we yield:

$$\begin{aligned} \log p_\theta(x) = \ & D_{KL}[q_\varphi(z|x)\|p_\theta(z|x)] \\ & + \mathbb{E}_{q_\varphi(z|x)}[\log p_\theta(x|z)] \\ & - D_{KL}[q_\varphi(z|x)\|p_\theta(z)] \end{aligned} \tag{4}$$

Let $\mathcal{L}(\theta, \varphi; \mathbf{x})$ represent the last two terms from the right part of Equation 4:

$$\mathcal{L}(\theta, \varphi; x) = \mathbb{E}_{q_\varphi(z|x)}[\log p_\theta(x|z)] - D_{KL}[q_\varphi(z|x)\|p_\theta(z)] \tag{5}$$

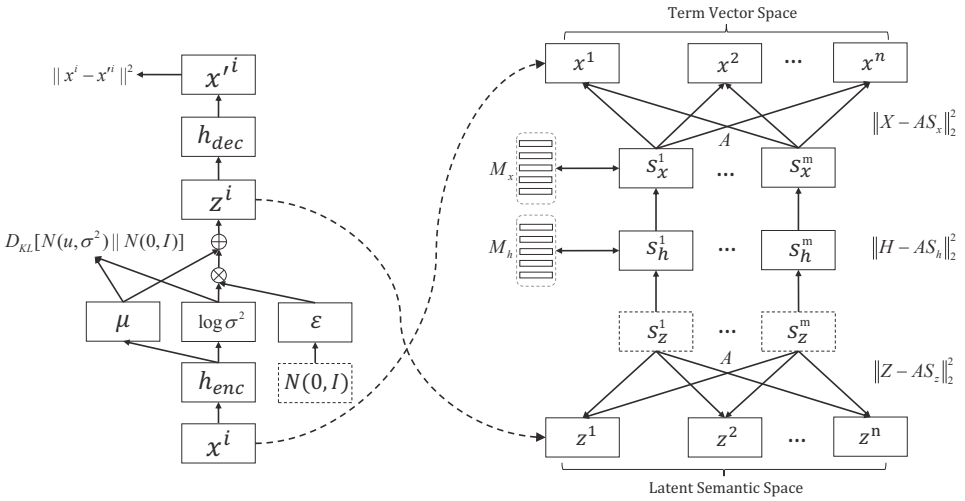

Figure 1: Our proposed sentence salience framework. **Left**: Latent semantic modeling via variation auto-encoders for sentence $\mathbf{x}^i$. **Right**: Salience estimation by a data reconstruction method during the variation-decoding process. $\mathbf{x}$ is the sentence term vector, and $\mathbf{z}$ is the corresponding latent semantic vector. $\mathbf{S_z}$ are the latent aspect vectors. $\mathbf{S_h}$ and $\mathbf{S_x}$ are hidden vectors and the output aspect term vectors. $\mathbf{M_h}$ and $\mathbf{M_x}$ are two memories used to refine $\mathbf{S_h}$ and $\mathbf{S_x}$ based on the neural alignment mechanism. $\mathbf{A}$ is a reconstruction coefficient matrix which contains the sentence salience information.

Because the first KL-divergence term of Equation 4 is non-negative, so we have $\log p_\theta(x) \geq \mathcal{L}(\theta, \varphi; \mathbf{x})$, which means that $\mathcal{L}(\theta, \varphi; \mathbf{x})$ is a lower bound (the objective to be maximized) on the marginal likelihood. In order to differentiate and optimize the lower bound $\mathcal{L}(\theta, \varphi; \mathbf{x})$, following the core idea of VAEs, we use a neural network framework for the probabilistic encoder $q_\phi(\mathbf{z}|\mathbf{x})$ for better approximation.

Similar to previous works (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014; Gregor et al. 2015), we assume that both the prior and posterior of the latent variables are Gaussian, i.e., $p_\theta(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ and $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the variational mean and standard deviation respectively, which can be calculated with a multilayer perceptron (MLP). Precisely, given the term vector representation of an input sentence $\mathbf{x}$, we first project it to a hidden space:

$$h_{enc} = relu(W_{xh}x + b_{xh}) \tag{6}$$

where $h_{enc} \in \mathbb{R}^{d_h}$, W_{xh} and b_{xh} are the neural parameters. $relu(x) = max(0, x)$ is the activation function.

Then the Gaussian parameters $\boldsymbol{\mu} \in \mathbb{R}^K$ and $\boldsymbol{\sigma} \in \mathbb{R}^K$ can be obtained via a linear transformation based on h_{enc}:

$$\begin{aligned} \mu &= W_{h\mu}h_{enc} + b_{h\mu} \\ \log(\sigma^2) &= W_{h\sigma}h_{enc} + b_{h\sigma} \end{aligned} \tag{7}$$

The latent semantic vector $\mathbf{z} \in \mathbb{R}^K$ can be calculated using the reparameterization trick:

$$\varepsilon \sim \mathcal{N}(0, \mathbf{I}), \;\; z = \mu + \sigma \otimes \varepsilon \tag{8}$$

where $\varepsilon \in \mathbb{R}^K$ is an auxiliary noise variable. It is obvious that the mapping from $\mathbf{x}$ to $\mathbf{z}$ is similar with the process of general auto-encoder. Therefore this process can be named variational-encoding process.

Given the latent semantic vector $\mathbf{z}$, a new term vector $\mathbf{x}'$ is generated via the conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$. Under the neural network framework, the generation process is similar with the decoding process of the typical auto-encoder model:

$$h_{dec} = relu(W_{zh}z + b_{zh}) \tag{9}$$

$$x' = sigmoid(W_{hx}h_{dec} + b_{hx}) \tag{10}$$

Finally, based on the reparameterization trick in Equation 8, we can get the analytical representation of the variational lower bound $\mathcal{L}(\theta, \varphi; \mathbf{x})$:

$$\log p(x|z) = \sum_{i=1}^{|V|} x_i \log x'_i + (1 - x_i) \cdot \log(1 - x'_i)$$

$$-D_{KL}[q_\varphi(z|x)\|p_\theta(z)] = \frac{1}{2}\sum_{i=1}^{K}(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2)$$

In this work we let $p_\theta(\mathbf{x}|\mathbf{z})$ be a multivariate Bernoulli. All the parameters $\{\mathbf{W}, \mathbf{b}\}$ can be learnt using the back-propagation method.

Salience Estimation

The right part of Figure 1 depicts the general framework for salience estimation. Note that $\mathbf{x}^i$ and $\mathbf{z}^i$ are the same vectors as those in the left part of Figure 1. Considering the spirit of summarization, we design a set of latent aspect vectors $\mathbf{S_z}$ from the latent space which can be regarded as the representatives of the whole semantic space. Inspired by previous works (He et al. 2012; Yao, Wan, and Xiao 2015; Li et al. 2015; Ren et al. 2016), we propose an unsupervised data reconstruction framework, named **VAEs-A**, for sentence salience estimation. The main idea is to jointly consider the reconstruction for latent semantic space and observed term vector space. This framework can capture the salience of sentences from these two different and complementary vector spaces.

VAEs-A Assume that $\mathbf{S}_z = \{\mathbf{s}_z^1, \mathbf{s}_z^2, \cdots, \mathbf{s}_z^m\}$ are m latent aspect vectors used for reconstructing all the latent semantic vectors $\mathbf{Z} = \{\mathbf{z}^1, \mathbf{z}^2, \cdots, \mathbf{z}^n\}$, and $m \ll n$. Recall that n is the number of original sentences. Here, we do not use the standard probabilistic sampling methods, instead we propose a more efficient and straightforward estimation method based on a neural network, which can be trained using back-propagation. More specifically, $\mathbf{S}_z$ is initialized using values from [-0.1, 0.1] randomly. Thereafter, the variational-decoding progress of VAEs can map the latent aspect vector $\mathbf{S}_z$ to $\mathbf{S}_h$, and then produce m new aspect term vectors $\mathbf{S}_x$:

$$s_h = relu(W_{zh} s_z + b_{zh}) \tag{11}$$

$$s_x = sigmoid(W_{hx} s_h + b_{hx}) \tag{12}$$

where the neural parameters $\mathbf{W}$ and $\mathbf{b}$ are shared from the decoder of VAEs.

Although VAEs are able to generate high-level abstract latent semantic representations for sentences, they may not be sufficient for generating high-quality sentence term vectors. The top-down generating process may lose detailed information (Li, Zhu, and Zhang 2016). In order to address this problem and to estimate the sentence salience more precisely, we add an alignment mechanism (Bahdanau, Cho, and Bengio 2015; Luong, Pham, and Manning 2015) to the decoding hidden layer and output layer respectively. The purpose of the alignment mechanism is to recall the lost detailed information from the sentence term vector memory $\mathbf{M_x} = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^n\}$ and the encoder hidden state memory $\mathbf{M_h} = \{\mathbf{h_{enc}^1}, \mathbf{h_{enc}^2}, \cdots, \mathbf{h_{enc}^n}\}$.

For each decoder hidden state s_h^i, we align it with each encoder hidden state $h_{enc}^j \in M_h$ by an alignment vector $a^h \in \mathbb{R}^n$. $a_{i,j}^h$ is derived by comparing s_h^i with each input sentence hidden state h_{enc}^i:

$$\begin{aligned} a_{i,j}^h &= \frac{\exp(e_{i,j}^h)}{\sum_{j'} \exp(e_{i,j'}^h)} \\ e_{i,j}^h &= v_{ha}^T \tanh(W_{ha} h_{enc}^j + U_{ha} s_h^i) \end{aligned} \tag{13}$$

The alignment vector $a_{i,j}^h$ captures much more detailed information from the source hidden space when generating the new representations. Based on the alignment vectors $\{a_{i,j}^h\}$, we can create a context vector c_h^i by linearly blending the sentence hidden states $h_{enc}^{j'}$:

$$c_h^i = \sum\nolimits_{j'} a_{i,j'}^h h_{enc}^{j'} \tag{14}$$

Then the output hidden state can be updated based on the context vector:

$$\tilde{s}_h^i = \tanh(W_{ch}^h c_h^i + W_{hh}^a s_h^i) \tag{15}$$

And a temporal output vector is generated according to:

$$\tilde{s}_x^i = sigmoid(W_{hx} \tilde{s}_h^i + b_{hx}) \tag{16}$$

Besides the alignment mechanism on the hidden layer, we also directly add alignment on the output layer, which can capture more nuanced and subtle difference information from the BoWs term vector space. The alignment is conducted by comparing $\tilde{s}_x^i$ with each observed term vector $x^j \in M_x$:

$$\begin{aligned} a_{i,j}^x &= \frac{\exp(e_{i,j}^x)}{\sum_{j'} \exp(e_{i,j'}^x)} \\ e_{i,j}^x &= \tilde{s}_x^i \cdot x^j \end{aligned} \tag{17}$$

where $\cdot$ in the inner product operation. Then the output context vector is computed as:

$$c_x^i = \sum\nolimits_j a_{i,j}^x x^j \tag{18}$$

To update the output vector, we develop a different method from that of the hidden alignments. Specifically we use a weighted combination of the context vectors and the original outputs with $\omega_a \in [0, 1]$:

$$s_x^i = \omega_a c_x^i + (1 - \omega_a) \tilde{s}_x^i \tag{19}$$

Intuitively, $\mathbf{S}_z$, $\mathbf{S}_h$, and $\mathbf{S}_x$ can be used to reconstruct the space to which they belong respectively. Let $\mathbf{A} \in \mathbb{R}^{n \times m}$ be the reconstruction coefficient matrix. Specifically, we do not create the new variable $\mathbf{A}$ here. Instead, we represent it using the decoder output layer alignment matrix $\mathbf{A} = \{a_{i,j}^x\}$, then refine it during optimization. We define the magnitude of each row of $\mathbf{A}$ as the salience scores for the corresponding sentences.

The optimization objective contains three reconstruction terms, jointly considering the latent semantic reconstruction and the term vector space reconstruction:

$$\mathcal{L}_A = \lambda_z \left\| Z - AS_z \right\| + \lambda_h \left\| H - AS_h \right\| + \lambda_x \left\| X - AS_x \right\|$$

This objective is integrated with the variational lower bound of VAEs and optimized in a multi-task learning fashion.

VAEs-Zero We also investigate a simpler VAEs-based model named VAEs-Zero which can also conduct salience estimation. Recall the reparameterization trick, the prior and posterior of the latent semantic vector $\mathbf{z}$ are both from Gaussian, and the vectors drawn from the zero mean will hold larger probability density. Based on this idea, we can generate a term vector $\mathbf{s}_x \in \mathbb{R}^{|V|}$ from a special latent semantic vector $\mathbf{s}_z = \mathbf{0}$ via the variational-decoding process. Intuitively, $\mathbf{s}_x$ contains richer information than the other vectors, which should be distilled as the summary information. Therefore, we assume that sentences which are more similar with $\mathbf{s}_x$ hold larger salience values. For each sentence $\mathbf{x}^i \in \mathbf{X}$, we use the cosine similarity as the salience estimation:

$$a^i = \frac{x^i \cdot s_x}{\|x^i\| \, \|s_x\|} \tag{20}$$

Interestingly, $\mathbf{s}_x$ can also be treated as the word salience information, so it can be employed to conduct the keyword extraction task.

Multi-Task Learning

As mentioned before, we integrate VAEs-based latent semantic modeling and salience estimation into a unified framework. Then the new optimization objective is:

$$\mathcal{J} = \min_{\Theta}(-\mathcal{L}(\theta, \varphi; x) + \lambda \mathcal{L}_{\text{salience}}) \tag{21}$$

where Θ is a set of all the parameters related to this task. $\mathcal{L}_{salience}$ is the reconstruction loss function for VAEs-A or VAEs-Zero. The whole framework can be trained using back-propagation efficiently. After the training, we calculate the magnitude of each row of $\mathbf{A}$ as the salience score for each corresponding sentence, which will be fed into a phrase-based optimization framework to generate a summary.

Summary Generation

Inspired by the phrase-based model in Bing et al. (2015) and Li et al. (2015), we refine this model to consider the salience information obtained by our VAEs-based salience estimation framework. Based on the parsed constituency tree for each input sentence, we extract the noun-phrases (NPs) and verb-phrases (VPs). The salience S_i of a phrase P_i is defined as:

$$S_i = \{\sum_{t \in P_i} tf(t) / \sum_{t \in Topic} tf(t)\} \times a_i, \quad (22)$$

where a_i is the salience of the sentence containing P_i; $tf(t)$ be the frequency of the concept t (unigram/bigram) in the whole topic. Thus, S_i inherits the salience of its sentence, and also considers the importance of its concepts.

The overall objective function of this optimization formulation for selecting salient NPs and VPs is formulated as an integer linear programming (ILP) problem:

$$\begin{aligned} \max\{&\sum_i \alpha_i S_i^N - \sum_{i<j} \alpha_{ij}(S_i^N + S_j^N) R_{ij}^N \\ &+ \sum_i \beta_i S_i^V - \sum_{i<j} \beta_{ij}(S_i^V + S_j^V) R_{ij}^V\} \end{aligned} \quad (23)$$

where α_i and β_i are selection indicators for the NP N_i and the VP V_i, respectively. S_i^N and S_i^V are the salience scores of N_i and V_i. α_{ij} and β_{ij} are co-occurrence indicators of pairs (N_i, N_j) and (V_i, V_j). R_{ij}^N and R_{ij}^V are the similarity of pairs (N_i, N_j) and (V_i, V_j). The similarity is calculated by the Jaccard Index based method. Specifically, this objective maximizes the salience score of the selected phrases, and penalizes the selection of similar phrase pairs.

In order to obtain coherent summaries with good readability, we add some constraints into the ILP framework, such as phrase co-occurrence constraint which control the co-occurrence relation of NPs or VPs: For NPs, we introduce three constraints:

$$\alpha_{ij} - \alpha_i \leq 0, \quad (24)$$

$$\alpha_{ij} - \alpha_j \leq 0, \quad (25)$$

$$\alpha_i + \alpha_j - \alpha_{ij} \leq 1. \quad (26)$$

Constraints 24 to 26 ensure a valid solution of NP selection. The first two constraints state that if the units N_i and N_j co-occur in the summary (i.e., $\alpha_{ij} = 1$), then we have to include them individually (i.e., $\alpha_i = 1$ and $\alpha_j = 1$). The third constraint is the inverse of the first two. Similarly, the constraints for VPs are as follows:

$$\beta_{ij} - \beta_i \leq 0, \quad (27)$$

$$\beta_{ij} - \beta_j \leq 0, \quad (28)$$

$$\beta_i + \beta_j - \beta_{ij} \leq 1. \quad (29)$$

Other constraints include sentence number, summary length, phrase co-occurrence, etc. For details, please refer to Woodsend and Lapata (2012), Bing et al. (2015), and Li et al. (2015). The objective function and constraints are linear. Therefore the optimization can be solved by existing ILP solvers such as simplex algorithms (Dantzig and Thapa 2006). In the implementation, we use a package called lp_solve[1].

[1]http://lpsolve.sourceforge.net/5.5/

Table 1: Results on DUC 2006.

System	Rouge-1	Rouge-2	Rouge-SU4
Random	0.280	0.046	0.088
Lead	0.308	0.048	0.087
MDS-Sparse	0.340	0.052	0.107
DSDR	0.377	0.073	0.117
RA-MDS	0.391	0.081	0.136
ABS-Phrase	0.392	0.082	0.137
VAEs-Zero	0.382	0.080	0.135
VAEs-A	**0.396**	**0.089**	**0.143**

Table 2: Results on DUC 2007.

System	Rouge-1	Rouge-2	Rouge-SU4
Random	0.302	0.046	0.088
Lead	0.312	0.058	0.102
MDS-Sparse	0.353	0.055	0.112
DSDR	0.398	0.087	0.137
RA-MDS	0.408	0.097	0.150
ABS-Phrase	0.419	0.103	0.156
VAEs-Zero	0.416	0.106	0.158
VAEs-A	**0.421**	**0.110**	**0.164**

Table 3: Results on TAC 2011.

System	Rouge-1	Rouge-2	Rouge-SU4
Random	0.303	0.045	0.090
Lead	0.315	0.071	0.103
PKUTM	0.396	0.113	0.148
RA-MDS	0.400	0.117	0.151
ABS-Phrase	0.393	0.117	0.148
VAEs-Zero	0.388	0.113	0.145
VAEs-A	**0.405**	**0.122**	**0.155**

Experiments and Results

Datasets

The standard MDS datasets from DUC and TAC are used in our experiments. DUC 2006 and DUC 2007 contain 50 and 45 topics respectively. Each topic has 25 news documents and 4 model summaries. The length of the model summary is limited to 250 words. TAC 2011 is the latest standard summarization benchmark data set and it contains 44 topics. Each topic contains 10 related news documents and 4 model summaries. TAC 2010 is used as the parameter tuning data set of our TAC evaluation. The length of the model summary is limited to 100 words.

Evaluation Metric

We use ROUGE score as our evaluation metric (Lin 2004) with standard options[2]. F-measures of ROUGE-1, ROUGE-2 and ROUGE-SU4 are reported.

Settings

For text processing, the input sentences are represented as BoWs vectors with dimension $|V|$. The dictionary V is cre-

[2]ROUGE-1.5.5.pl -n 4 -w 1.2 -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0

Table 4: Top-10 terms extracted from each topic according to the output of VAEs-A

Topic 1	Topic 2	Topic 3
Roberts	China	food
amish	earthquake	recall
girl	Sichuan	pet
school	province	cat
Miller	tuesday	dog
family	million	company
child	relief	menu
police	people	sell
kill	government	product

ated using unigrams, bigrams and named entity terms. n is the number of sentences in all the documents of a topic (event). For the number of aspects, we let $m = 5$. For the neural network framework, we set the hidden size $d_h = 500$ and the latent size $K = 100$. For the optimization objective, we let $\lambda_z = 1$, $\lambda_h = 400$, $\lambda_x = 800$, and $\lambda = 1$. Adam (Kingma and Ba 2014) is used for gradient based optimization with a learning rate 0.001. Our neural network based framework is implemented using Theano (Bastien et al. 2012) on a single GPU[3].

Results and Discussions

To compare the performance of our framework with previous methods, our first priority is to get the summaries produced by their systems (or get their code to produce summaries by ourselves). Then we run ROUGE evaluation on them with the same option.

We compare our system with several summarization baselines and existing unsupervised methods. **Random** baseline selects sentences randomly for each topic. **Lead** baseline (Wasson 1998) ranks the news chronologically and extracts the leading sentences one by one. Three other unsupervised existing methods based on sparse coding are also compared, namely, **DSDR** (He et al. 2012), **MDS-Sparse** (Liu, Yu, and Deng 2015), and **RA-MDS** (Li et al. 2015). **ABS-Phrase** (Bing et al. 2015) generates abstractive summaries using phrase-based optimization framework with weighted term frequency as salience estimation. Moreover, we would like to mention that **SpOpt** (Yao, Wan, and Xiao 2015) also presents some good results in their paper, however, it is difficult to rebuild their system to faithfully reproduce their results.

As shown in Table 1 and Table 2, our system achieves the best results on all the ROUGE metrics. It demonstrates that VAEs based latent semantic modeling and jointly semantic space reconstruction can improve the MDS performance considerably. It is worth to note that VAEs-Zero also achieves comparable performance. Although it is not as good as VAEs-A, it is better than most of the existing methods. Therefore, VAEs based latent semantic modeling can benefit the MDS performance. Besides those **unsupervised** models, to our knowledge, the method presented in Wang et al. (2013) achieved the best performance on DUC 2007. The reason is that it uses **supervised** learning framework to train the sentence compression and document summarization models. In the evaluation, it provides two supervised learning based sentence selection methods: Support Vector Regression (SVR) and LambdaMART. SVR obtains 0.095 and 0.147 on Rouge-2 and Rouge-SU4 respectively. LambdaMART obtains 0.123 and 0.156. Our framework, which is unsupervised, outperforms SVR and achieves similar results compared with LambdaMART.

For the data set TAC 2011, besides the above mentioned baselines, we compare our framework with several more top systems: **PKUTM** (Li et al. 2011) employs manifold-ranking for sentence scoring and selection; Table 3 shows that our performance is better than both PKUTM. It is worth noting that PKUTM used a Wikipedia corpus for providing domain knowledge. The method **SWING** (Min, Chew, and Tan 2012) is the best TAC 2011 system. However, our results are not as good as SWING. The reason is that SWING uses category-specific features and trains the feature weights with the category information of TAC 2010 data in a supervised manner. These features help them select better category-specific content for the summary. In contrast, our model is **unsupervised**, and we only use TAC 2010 for general parameter tuning purpose.

We mention that $\mathbf{S_z}$ and $\mathbf{S_x}$ represent different aspects of an event. To validate this idea, we take the topic "Pet Food Recall" in TAC 2011 and extract some keywords from each aspect. **Aspect-1** contains words "*Nutro, purchase, dozen, drop, 60, timing, protein, research*", **Aspect-2** is "*Sarah, Tuite, source, protein, Food, and, Drug Administration*", and **Aspect-3** is "*food, company, recall, pet, menu, cat, product, foods, dog*". It demonstrates that our framework is able to capture the main aspects of a topic. Moreover, we find that the magnitude of $\mathbf{S_x}$ can represent the word salience information. We select 3 topics from TAC 2011: "Amish Shooting", "Earthquake Sichuan", and "Pet Food Recall". For each topic, we sort the dictionary terms according to their salience scores, and extract the top-10 terms, as shown in Table 4. We can see that the top-10 terms reveal the most important information of each topic. For the topic "Amish Shooting", we notice a sentence from the golden summary: "On October 2, 2006, a gunman, Charles Roberts, entered an Amish school near Lancaster, PA, took the children hostage, killed five girls and wounded seven other children before killing himself." It is obvious that the top-10 terms can capture the main semantics.

Conclusions

We propose an new unsupervised Multi-Document Summarization (MDS) framework. First, a VAEs based generative model is employed to map the sentence from term vector space to latent semantic space. Then an unsupervised data reconstruction model is proposed to conduct salience estimation, by jointly reconstructing latent semantic space and observed term vector space using aspect related vectors. Experimental results on the benchmark data sets DUC and TAC show that our framework achieves better performance than the state-of-the-art models.

[3]Tesla K80, 1 Kepler GK210 is used, 2496 Cuda cores, 12G GDDR5 memory.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Bastien, F.; Lamblin, P.; Pascanu, R.; Bergstra, J.; Goodfellow, I.; Bergeron, A.; Bouchard, N.; Warde-Farley, D.; and Bengio, Y. 2012. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

Bing, L.; Li, P.; Liao, Y.; Lam, W.; Guo, W.; and Passonneau, R. 2015. Abstractive multi-document summarization via phrase selection and merging. In *ACL*, 1587–1597.

Blei, D. M.; Kucukelbir, A.; and McAuliffe, J. D. 2016. Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*.

Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A. C.; and Bengio, Y. 2015. A recurrent latent variable model for sequential data. In *NIPS*, 2980–2988.

Dantzig, G. B., and Thapa, M. N. 2006. *Linear programming 1: introduction*. Springer Science & Business Media.

Erkan, G., and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR* 457–479.

Goldstein, J.; Mittal, V.; Carbonell, J.; and Kantrowitz, M. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLPWorkshop*, 40–48.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D.; and Wierstra, D. 2015. Draw: A recurrent neural network for image generation. In *ICML*, 1462–1471.

He, Z.; Chen, C.; Bu, J.; Wang, C.; Zhang, L.; Cai, D.; and He, X. 2012. Document summarization based on data reconstruction. In *AAAI*, 620–626.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T. S.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine learning* 37(2):183–233.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, 1746–1751.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *ICLR*.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *ICML*, 1188–1196.

Li, H.; Hu, Y.; Li, Z.; Wan, X.; and Xiao, J. 2011. Pkutm participation in tac2011. In *TAC*.

Li, P.; Bing, L.; Lam, W.; Li, H.; and Liao, Y. 2015. Reader-aware multi-document summarization via sparse coding. In *IJCAI*, 1270–1276.

Li, C.; Zhu, J.; and Zhang, B. 2016. Learning to generate with memory. In *ICML*, 1177–1186.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8.

Liu, H.; Yu, H.; and Deng, Z.-H. 2015. Multi-document summarization based on two-level sparse representation model. In *AAAI*, 196–202.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, 1412–1421.

Miao, Y.; Yu, L.; and Blunsom, P. 2015. Neural variational inference for text processing. *arXiv preprint arXiv:1511.06038*.

Min, Z. L.; Chew, Y. K.; and Tan, L. 2012. Exploiting category-specific information for multi-document summarization. *COLING* 2093–2108.

Nenkova, A., and McKeown, K. 2012. A survey of text summarization techniques. In *Mining Text Data*. Springer. 43–76.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Ren, Z.; Song, H.; Li, P.; Liang, S.; Ma, J.; and de Rijke, M. 2016. Using sparse coding for answer summarization in non-factoid community question-answering. In *SIGIR Workshop: Web Question Answering, Beyond Factoids*.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 1278–1286.

Song, H.; Ren, Z.; Li, P.; Liang, S.; Ma, J.; and de Rijke, M. 2017. Summarizing answers in non-factoid community question-answering. In *WSDM*.

Wainwright, M. J., and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1(1-2):1–305.

Wan, X.; Yang, J.; and Xiao, J. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI*, volume 7, 2903–2908.

Wang, L.; Raghavan, H.; Castelli, V.; Florian, R.; and Cardie, C. 2013. A sentence compression based framework to query-focused multi-document summarization. In *ACL*, 1384–1394.

Wasson, M. 1998. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In *ACL*, 1364–1368.

Woodsend, K., and Lapata, M. 2012. Multiple aspect summarization using integer linear programming. In *EMNLP-CNLL*, 233–243.

Xing, E. P.; Jordan, M. I.; and Russell, S. 2002. A generalized mean field algorithm for variational inference in exponential families. In *UAI*, 583–591. Morgan Kaufmann Publishers Inc.

Yao, J.-g.; Wan, X.; and Xiao, J. 2015. Compressive document summarization via sparse optimization. In *IJCAI*, 1376–1382.