Unity in Diversity: Learning Distributed Heterogeneous Sentence Representation for Extractive Summarization

by

Abhishek Singh, Manish Gupta, Vasudeva Varma

in

32nd AAAI Conference on Artificial Intelligence (AAAI-18)
(*AAAI-18*)

New Orleans, USA

Report No: IIIT/TR/2018/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
February 2018

Unity in Diversity: Learning Distributed Heterogeneous Sentence Representation for Extractive Summarization

Abhishek Kumar Singh, Manish Gupta* and Vasudeva Varma
IIIT Hyderabad, India
abhishek.singh@research.iiit.ac.in, manish.gupta@iiit.ac.in, vv@iiit.ac.in

Abstract

Automated multi-document extractive text summarization is a widely studied research problem in the field of natural language understanding. Such extractive mechanisms compute in some form the worthiness of a sentence to be included into the summary. While the conventional approaches rely on human crafted document-independent features to generate a summary, we develop a data-driven novel summary system called HNet, which exploits the various semantic and compositional aspects latent in a sentence to capture document independent features. The network learns sentence representation in a way that, salient sentences are closer in the vector space than non-salient sentences. This semantic and compositional feature vector is then concatenated with the document-dependent features for sentence ranking. Experiments on the DUC benchmark datasets (DUC-2001, DUC-2002 and DUC-2004) indicate that our model shows significant performance gain of around 1.5-2 points in terms of ROUGE score compared with the state-of-the-art baselines.

1 Introduction

The rapid growth of online news over the web has generated an epochal change in the way we retrieve, analyze and consume data. The readers now have access to a huge amount of information on the web. For a human, understanding large documents and assimilating crucial information out of it is often a laborious and time-consuming task. Motivation to make a concise representation of huge text while retaining the core meaning of the original text has led to the development of various automated summarization systems. These systems provide users filtered, high-quality concise content to work at unprecedented scale and speed. Summarization methods are mainly classified into two categories: *extractive* and *abstractive*. Extractive methods aim to select salient phrases, sentences or elements from the text while abstractive techniques focus on generating summaries from scratch without the constraint of reusing phrases from the original text.

The majority of literature on text summarization is dedicated to extractive summarization approach. Previous methods can be predominantly categorized as (1) greedy approaches (e.g. (Carbonell and Goldstein, 1998)), (2) graph based approaches (e.g. (Erkan and Radev, 2004)) and (3) constraint optimization based approaches (e.g. (McDonald, 2007)). These approaches rely mainly on a set of features which were manually crafted. Recently, few efforts have been made towards data-driven learning approaches for extractive summarization using neural networks. Kågebäck et al. (2014) used recursive autoencoders to summarize documents, achieving good performance on the Opinosis (Ganesan, Zhai, and Han, 2010) dataset. Cao et al. (2015b) used convolution neural networks for addressing the problem of learning summary prior representation for multi-document extractive summarization. Cheng and Lapata (2016) introduced attention based neural encoder-decoder model for extractive single document summarization trained on a large corpus of news articles collected from the Daily Mail. Their work focuses on sentence-level as well as the word-level extractive summarization of individual documents using encoder-decoder architecture. Singh, Gupta, and Varma (2017) proposed a combination of memory network and convolutional BLSTM (Bidirectional Long Short Term Memory) network to learn better unified document representation which jointly captures n-gram features, sentential information and the notion of the summary worthiness of sentences leading to better summary generation.

Most successful multi-document summarization systems use extractive methods. Sentence extraction is a crucial step in such a system. The idea is to find a representative subset of sentences, which contains the information of the entire set. Thus, sentence ranking is imperative in finding such an informative subset, which sets our focus to sentence-level summarization. The performance of the summarization system using sentence ranking approach is profoundly determined by the feature engineering, irrespective of the ranking models (Osborne, 2002; Conroy et al., 2004; Galley, 2006; Li et al., 2007). Features are broadly classified as: (a) document-dependent features (e.g., position, term frequency), and (b) document-independent features (e.g., length, stop-word ratio, word polarity). Document independent features often reveal the aspect that a sentence can be considered summary worthy irrespective of which document it is present in. Consider the following example.

1. Six killed, eight wounded in a shooting at Quebec City.
2. It was the shooting that killed six people and injured eight

*The author is also a Principal Applied Researcher at Microsoft.

people at a Quebec City mosque.

While the former sentence conveys prominent information in concrete terms, the latter is a more verbose way of portrayal with similar meaning. In the case of multi-document summary, the former sentence is the best candidate, as it is a concise representation keeping important information intact. This intuition was called as summary prior nature by Cao et al. (2015b), and can be captured by learning better document independent features.

We aim to learn a better sentence representation that incorporates both document dependent features as well as document independent features to capture the notion of saliency of a sentence. Since the sentence representation comprises of two different kinds of features, we call it a heterogeneous representation. Contrary to the orthodox method of painstakingly engineering document independent features, we propose a model with a Convolutional Sentence Tree Indexer (CSTI), a novel data-driven neural network for capturing semantic and compositional aspects in a sentence. CSTI slides over the input sequence to produce higher-level representation by compressing all the input information into a single representation vector of the root node in the constructed binary tree. We present details in Section 3. Final sentence representation obtained by concatenating the transformed document dependent features and the features obtained from CSTI (document independent features) is used under a regression framework for sentence ranking.

Deep neural networks perform better in the case of huge training data. However, non-availability of large multi-document summarization corpus makes learning challenging for deep networks and often results procured are not of high quality. To overcome this issue, we use transfer learning approach where we first train the network on single document summarization corpus (Cheng and Lapata, 2016) and then fine-tune the network with the multi-document datasets. We summarize our key contributions below.

1. We propose CSTI, a novel method to encode semantic and compositional features latent in a sentence which can be combined with document dependent features to learn a better heterogeneous sentence representation for capturing the notion of summary worthiness of a sentence.
2. Further, we propose a novel Siamese CSTI (Siam-CSTI) model for effectively identifying redundant sentences during the sentence selection process.
3. We use transfer learning method to overcome the problem of lack of data for multi-document summarization.
4. We experimentally show that our method outperforms the basic systems and several competitive baselines. Our model achieves significant performance gain on the DUC 2001, 2002 and 2004 multi-document summarization datasets.

2 Related Work

Extractive document summarization has been traditionally connected to the task of sentence ranking. Sentence ranking models by Osborne (2002); Conroy et al. (2004); Galley (2006); Li et al. (2007) are dependent on the human-crafted features. Shen et al. (2007) modeled extractive document summarization as a sequence classification problem using Conditional Random Fields. Our approach is different from theirs as we use a data-driven approach to automatically acquire document-independent features for representing sentences without the need of manually crafted document independent features. Hong and Nenkova (2014) built a summarization system using advanced document-independent features which can be seen as an attempt to capture better sentence representation. These features are often hand-crafted and fail to capture various semantic aspects. Summarization system CTSUM (Wan and Zhang, 2014) attempts to rank sentences using certainty score. However, certainty score alone is not enough to reveal all possible latent semantic aspects. Ren et al. (2016) develop a redundancy aware sentence regression framework for multi-document extractive summarization. They model importance and redundancy simultaneously by evaluating the relative importance of a sentence given a set of selected sentences. Along with single sentence features they incorporate additional features derived from the sentence relations. They manually crafted *sentence importance features* and *sentence relation features* while we use deep neural network for getting automatic document-independent features.

Recursive Neural Networks are known to model compositionality in natural language over trees. The tree structure is predefined by a syntactic parser (Socher et al., 2013) and each non-leaf tree node is associated with a node composition function. Socher et al. (2013) also proposed Tensor networks as composition function for sentence level sentiment analysis tasks. Recently, Zhu, Sobihani, and Guo (2015) introduced S-LSTM which extends LSTM units to compose tree nodes in a recursive fashion. Neural Tree Indexer (NTI), an extension of S-LSTM was proposed for natural language inference and QA task (Yu and Munkhdalai, 2017). In our work we introduce a CSTI, an enhanced version of NTI adapted for summarization task. Unlike NTI, our model uses (a) CNNs that can slide over inputs to produce higher-level representations, and (b) BLSTM as the primary composition function.

3 Proposed Model

Our architecture intends to learn a better representation of a sentence with consideration of both document-dependent and document-independent features in order to measure the worthiness of a sentence in the summary. The proposed system architecture is illustrated in Figure 1. The principal components of our model architecture are as follows.

1. *CSTI*: captures local (word n-grams and phrase level), global (sequential and compositional dependencies between phrases) information and the notion of saliency of a sentence. Details in Section 3.1.
2. *Extractor*: extracts document dependent features from the given sentence. Details in Section 3.2.
3. *Regression Layer*: predicts sentence scores and thus, helps in the sentence ranking process.

CSTI provides an embedding which incorporates document-independent features. Final unified sentence embedding

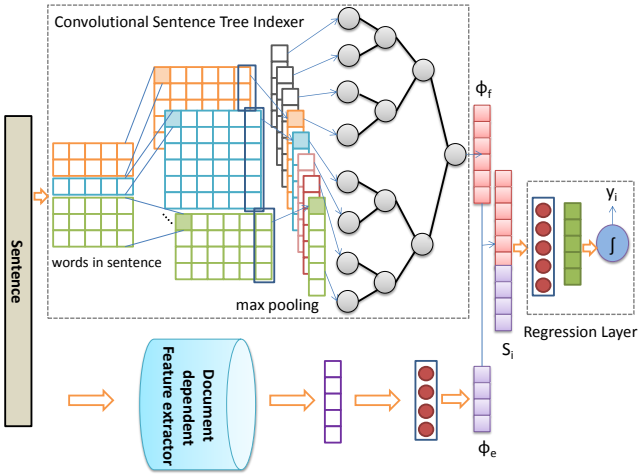

Figure 1: The System Architecture of HNet. After max pool operation padding vectors (represented in black color) are added to form a full binary tree.

is obtained by concatenating embedding from CSTI and document-dependent features, which is then forwarded through the regression layer to obtain saliency score of a sentence. Since the model makes use of the heterogeneous representation of the sentence, we name our model as Heterogeneous Net (HNet). In this section, we first describe the CSTI and then present details of the extractor and the regression layer.

3.1 Convolutional Sentence Tree Indexer (CSTI)

We focus on learning a hierarchical sentence representation that not only incorporates phrase level features and global sentence level information but it should also include the notion of saliency of a sentence. The hierarchical nature of our model reflects the fact that sentences are generated from words, phrases and often have some sequential and compositional dependencies among these units. Therefore, we use an architecture to obtain a representation with minimum information loss such that the global information gets discovered and the local information remains preserved.

CSTI comprises of: (a) *Convolutional Encoder*: We use a Convolution Neural Network (CNN) with multiple filters to automatically capture set of phrase (n-grams) based features followed by a max-over-time pooling operation to obtain a set of feature vectors. We do this because phrases with different lengths can exhibit the same characteristics of summary prior nature. (b) *Bidirectional Long Short Term Memory Tree Indexer (BLSTM Tree Indexer)* to obtain a comprehensive set of document-independent features incorporating semantic and compositional aspects in a sentence. We use BLSTM Tree Indexer because: (a) it models conditional and compositional power of sequential RNNs and syntactic tree based recursive neural nets, and (b) it is a robust syntactic parsing-independent tree structure model and does not require a parse tree structure.

Convolutional Encoder For first level sentence encoding, we choose convolution neural network for the following reasons: (1) it is easily trainable without long-term dependencies, (2) it handles sentences of variable length inherently and is able to learn compressed representation of n-grams effectively, (3) previous research has shown that it can be successfully used for sentence-level classification tasks such as sentiment analysis (Kim, 2014).

Conventional convolution neural network uses convolution operation over various word embeddings which is then followed by a max pooling operation. Suppose, d dimensional word embedding of the i^{th} word in the sentence is w_i, and let $w_{i:i+n}$ denote the concatenation of word embeddings $w_i, ..., w_{i+n}$. Then, convolution operation over a window of c words using a filter of $\theta_t^c \in \mathbb{R}^{m \times cd}$ yields new features with m dimensions. Convolution operation is written as follows.

$$f_i^c = tanh(\theta_t^c \times w_{i:i+c-1} + b) \quad (1)$$

Here b is the bias term. We obtain a feature map F^c by applying filter θ_t^c over all possible windows of c words in the sentence of length N.

$$F^c = [f_1^c, f_2^c, ..., f_{N-c+1}^c] \quad (2)$$

Our intention is to capture the most prominent features in the feature map. Hence, we used max-over-time pooling operation (Collobert et al., 2011) to acquire final features for a filter of fixed window size. To exploit several latent features from phrase based information, we used multiple filters of different window widths. Let $\theta_t^1, \theta_t^2, ..., \theta_t^k$ be k filters for window sizes from 1 to k then we have k feature maps $F^1, F^2, ..., F^k$. Applying max-over-time pooling operation helps to get most salient features. They seem to capture the phrase-level information nicely. The first level features ϕ_1 obtained from convolution network can be denoted as follows.

$$\phi_1 = \{max\{F^1\}, max\{F^2\}, ..., max\{F^k\}\} \quad (3)$$

We use an enhanced convolution network which is different from the one used for sentence classification task (Kim, 2014) or for learning the prior summary task (Cao et al., 2015b). Kim (2014) reserves all representation generated by filters to a fully connected layer which ignores relations among phrases with different lengths. Cao et al. (2015b) tried to capture this relation by performing two-stage max-over-time pooling operation. Unlike these models, our model captures the relation among different length phrases by passing the representations generated after max-over-time pooling operation to the BLSTM Tree Indexer network. Representation thus obtained also incorporates the latent temporal and compositional dependencies among variable length phrases.

BLSTM Tree Indexer (BTI) Sequential LSTMs are known to learn syntactic structure (conditional transition) from natural language. However their generalization to unseen text is relatively poor in comparison with models that exploit syntactic tree structure (Bowman, Manning, and Potts, 2015). BLSTM Tree Indexer leverages the sequential power of LSTMs and the compositional power of recursive

models, without the need of a parse tree. The model constructs a binary tree by processing the input sequences with its node function in a bottom-up fashion. It compresses all the input information into a single representation vector of the root node. This representation seems to capture both semantic and compositional aspects in the sentence.

The output of the convolutional encoder is padded with padding vectors to form a full binary tree and fed as input to the BLSTM Tree Indexer. The input set consists of a sequence of vectors (ϕ_1). BTI can be a full n-ary tree structure. To reduce computational complexity, we have implemented binary tree form of BTI in our study. It has two types of transformation functions: (a) a non-leaf node composition function $f^{node}(h^1, ..., h^q)$ and (b) a leaf node transformation function $f^{leaf}(\phi_1^j)$, where ϕ_1^j is j^{th} feature vector from set ϕ_1. $f^{node}(h^1, ..., h^q)$ is a composition function of the representation of its child nodes $h^1, ..., h^q$, where q is the total number of child nodes of this non-leaf node. $f^{leaf}(\phi_1^j)$ is some non-linear transformation of the input vector ϕ_1^j.

As we use the binary tree form of BTI, a non-leaf node can only take two direct child nodes, i.e., $q = 2$. Hence, the function $f^{node}(h^l, h^r)$ learns a composition over its left child node h^l and right child node h^r. The node and the leaf node functions are actually parameterized neural networks.

We present our approach for the two types of transformation functions in the following.

Leaf Node Transformation: We use a MLP (Multi-Layer Perceptron) with $ReLU$ function (for non-linear transformation) for the leaf node function f^{leaf} as follows.

$$h_j = ReLU(MLP(\phi_1^j; \theta)) \tag{4}$$

where ϕ_1^j is input sequence fed to the multi-layer perceptron, θ is the learning parameter and h_j is the vector representation for the leaf node.

Non-Leaf Node Composition: A Bidirectional LSTM (BLSTM) is used as the composition function $f^{node}(h^l, h^r)$ to get the representation of the parent node. BLSTM processes the input both in the forward order as well in the reverse order, allowing to combine future and past information in every time step. It comprises of two LSTM layers processing the input separately to produce $\overrightarrow{h}$, $\overrightarrow{c}$, the hidden and cell states of an LSTM processing the input in the forward order, and $\overleftarrow{h}$ and $\overleftarrow{c}$, the hidden and the cell states of an LSTM processing the input in reverse order. Both, $\overrightarrow{h}$ and $\overleftarrow{h}$, are then combined to produce output sequence of the BLSTM layer. Let h_t^l, h_t^r, c_t^l and c_t^r be the vector representations and cell states for left and right children. A BLSTM computes a parent node representation h_{t+1}^p and a node cell state c_{t+1}^p as follows.

Forward order:

$$\overrightarrow{i_{t+1}} = \sigma(W_1\overrightarrow{h_t^l} + W_2\overrightarrow{h_t^r} + W_3\overrightarrow{c_t^l}) + W_4\overrightarrow{c_t^r} \tag{5}$$

$$\overrightarrow{f_{t+1}^l} = \sigma(W_5\overrightarrow{h_t^l} + W_6\overrightarrow{h_t^r} + W_7\overrightarrow{c_t^l}) + W_8\overrightarrow{c_t^r} \tag{6}$$

$$\overrightarrow{f_{t+1}^r} = \sigma(W_9\overrightarrow{h_t^l} + W_{10}\overrightarrow{h_t^r} + W_{11}\overrightarrow{c_t^l}) + W_{12}\overrightarrow{c_t^r} \tag{7}$$

$$\overrightarrow{c_{t+1}^p} = \overrightarrow{f_{t+1}^l} \odot \overrightarrow{c_t^l} + \overrightarrow{f_{t+1}^r} \odot \overrightarrow{c_t^r} + \overrightarrow{i_{t+1}} \odot \tanh(W_{13}\overrightarrow{h_t^l} + W_{14}\overrightarrow{h_t^r}) \tag{8}$$

$$\overrightarrow{o_{t+1}} = \sigma(W_{15}\overrightarrow{h_t^l} + W_{16}\overrightarrow{h_t^r} + W_{17}\overrightarrow{c_{t+1}^p}) \tag{9}$$

$$\overrightarrow{h_{t+1}^p} = \overrightarrow{o_{t+1}} \odot \tanh(\overrightarrow{c_{t+1}^p}) \tag{10}$$

Similarly, in the reverse order we obtain $\overleftarrow{h_{t+1}^p}$ and $\overleftarrow{c_{t+1}^p}$. Finally, we combine them to obtain the vectors c_{t+1}^p and h_{t+1}^p as follows.

$$c_{t+1}^p = mean(\overrightarrow{c_{t+1}^p}, \overleftarrow{c_{t+1}^p}), \quad h_{t+1}^p = mean(\overrightarrow{h_{t+1}^p}, \overleftarrow{h_{t+1}^p})$$

where W_1, ..., $W_{17} \in \mathbb{R}^{k \times k}$ ($k = N - c + 1$) are trainable weights. For brevity we eliminated the bias terms. σ and $\odot$ denote the element-wise sigmoid function and the element-wise vector multiplication respectively. Each non-leaf node computes its representation by composing its children representation using the above set of equations. This representation is passed towards the root in a bottom-up fashion to construct the tree representation. The vector representation of the root h^{root} (also referred as ϕ_f) incorporates semantic and compositional aspects latent in a sentence.

3.2 Extractor

Besides just using the document independent features, we also intend to use document dependent features in learning better sentence representation for saliency estimation of sentences. Our feature set includes the following: (1) The position of the sentence, (2) The averaged cluster frequency values of words in the sentence, (3) The average term frequency values of the words in the sentence, (4) The average word IDF values in the sentence, divided by sentence length, and (5) The maximal word IDF score in the sentence.

We choose these features for the following reasons: (1) They tend to impart some contextual knowledge. (2) They are often simple to calculate and have been extensively used in previous research (Cao et al., 2015a,b).

3.3 Regression Layer

We follow traditional supervised learning approach for sentence ranking (Carbonell and Goldstein, 1998; Li et al., 2007). The regression layer at the end of the architecture aims to assign scores to a sentence. Sentences are ranked based on the score predicted by the regression layer. Since our approach focuses on learning a better sentence representation embracing both document-independent and document-dependent features, we concatenate the document-independent features obtained from the CSTI net with the transformed extracted document-dependent features (using a dense layer). Let ϕ_e be the transformed extracted document-dependent features and let S_i denote the heterogeneous sentence embedding of the i^{th} sentence. Thus, $S_i = [\phi_f, \phi_e]$.

The sentence worthiness is scored by ROUGE-2 (Lin and Hovy, 2003) (without stop words) and our model tries to estimate this score. Given sentence i, the final sentence representation S_i is used in the regression layer to score saliency as $y_i = \sigma(W^T \times S_i)$ where W are the regression weights and σ is the softmax function. The softmax function gives a

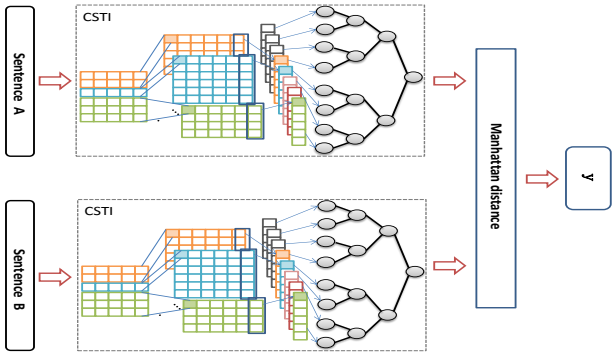

Figure 2: Siamese CSTI Architecture

nice distribution over the range $[0, 1]$ which makes it suitable to imitate ROUGE score.

This score is used to rank sentences. Higher the score, higher is the chance of the sentence to be included in the generated summary.

3.4 Removing Redundant Sentences

A good summary should be informative with non-redundant content. We generate the final summary by choosing top ranked sentences taking into account the redundancy among the selected sentences. The sentences are sorted in descending order of saliency scores. To identify whether the next candidate sentence is redundant, we compare it with all the sentences in the summary generated so far. We introduce a Siamese CSTI (Siam-CSTI) network for identifying redundant sentences. Figure 2 shows the Siam-CSTI architecture. The base network consists of the CSTI net. Weight parameters are tied for the base network. Two CSTI nets feed their output to a distance metric layer. We experiment with cosine, Euclidean and Manhattan distances and empirically find that the Manhattan distance seems to perform better in our case.

Siam-CSTI network is trained for sentence similarity task on the SICK data (Marelli et al., 2014). We present dataset details in Section 4.1. Formally, we consider a supervised learning setting where each training example consists of a pair of sequences $(x^a_1, ..., x^a_{T_a})$, $(x^b_1, ..., x^b_{T_b})$ of fixed-size vectors (each x^a_i, $x^b_j \in \mathbb{R}^d$ is d-dimensional word vector) along with a single label y for the pair. The sequences may be of different lengths $T_a \neq T_b$ and the sequence lengths can vary from example to example. The similarity function g is based on the Manhattan distance metric as follows.

$$g(h^a_{T_a}, h^b_{T_b}) = \exp(-||h^a_{T_a} - h^b_{T_b}||_1) \in [0, 1] \qquad (11)$$

where $h^a_{T_a}$, $h^b_{T_b}$ are the learned representations of the sequences $x^a_{T_a}$, $x^b_{T_b}$ respectively such that $h^a_{T_a}$ and $h^b_{T_b}$ are closer in the vector space if $x^a_{T_a}$ and $x^b_{T_b}$ are similar otherwise they reside far apart. Mean Squared Error (MSE) is used as loss function (after rescaling the training set relatedness labels to lie in $[0, 1]$). The Siam-CSTI model trained on paired examples seems to learn a highly structured space of sentence representations by exploiting the sequential and recursive power of CSTI that captures rich semantics. Similar sentences ($y = 1$) are considered as redundant sentences and non-similar sentences ($y = 0$) are considered as non-redundant sentences. The final summary is generated by iteratively picking up a sentence from the set of previously sorted sentences and adding it to current summary if the picked sentence is non-redundant.

Ranking by Dependency Tree-LSTM	Tree	M	S
a woman is slicing potatoes			
• a woman is cutting potatoes	4.82	4.87	4.91
• potatoes are being sliced by a woman	4.70	4.38	4.68
• tofu is being sliced by a woman	4.39	3.51	3.62
a boy is waving at some young runners from the ocean			
• a group of men is playing with a ball on the beach	3.79	3.13	2.68
• a young boy wearing a red swimsuit is jumping out of a blue kiddies pool	3.37	3.48	3.29
• the man is tossing a kid into the swimming pool that is near the ocean	3.19	2.26	1.87

Table 1: Most similar sentences (from 1000-sentence subsample) in the SICK test data according to the Tree-LSTM. Tree/M/S denote relatedness (with the sentence preceding each group) predicted by the Tree-LSTM/MaLSTM/Siam-CSTI.

4 Experimental Setup

We experiment with our CSTI and Siam-CSTI based summarization model (HNet) for the task of multi-document summarization. In this section, we present our experimental setup for assessing the performance of our system. We discuss the corpora used for training and evaluation and provide implementation details of our approach.

4.1 Datasets

Initial training of our model is done on the Daily Mail corpus, used for the task of single document summarization by Cheng and Lapata (2016). Overall, we have 193986 training documents, 12147 validation documents and 10350 test documents in the corpus. For the purpose of training, we created a sentence and its ROUGE-2 score pairs from this corpus. Sentences which are part of the summary get high ROUGE scores than non-summary sentences. We experiment on DUC 2001-2004 datasets which are used for generic multi-document summarization task. These documents are from newswires which are grouped into several thematic clusters. The full DUC data set can be availed by request at `http://duc.nist.gov/data.html`. The DUC 2001, 2002 and 2004 datasets consist of 11295, 15878 and 13070 sentences respectively. The SICK dataset which contains 9927 sentence pairs with a 5,000/4,927 training/test split (Marelli et al., 2014) was used for training the Siam-CSTI net. Each pair has a relatedness label $\in [1, 5]$ corresponding to the average relatedness judged by 10 different individuals.

4.2 Implementation Details

We fine tuned our model on DUC datasets after initial training on Daily Mail corpus. DUC 2003 data is used as de-

velopment set and we perform a 3-fold cross-validation on DUC 2001, 2002 and 2004 datasets with two years of data as training set and one year of data as the test set. The word vectors were initialized with 250-dimensional pre-trained embeddings (Mikolov et al., 2013). The embeddings for "out of vocabulary" words were set to zero vector. The size of the hidden units of BLSTM was set to 150. After tuning on the validation set, we fix the dimension m of the latent features from convolutional encoder as 125 and window size $k = 5$ for HNet system. We use Adam (Kingma and Ba, 2014) as the optimizer with mini batches of size 35. Learning rates are set to $\{0.009, 0.0009\}$. For our network, we use regularization dropout of $\{0.2, 0.5\}$.

4.3 Baseline Methods

In this section of the paper, we describe several summarization baseline systems that we choose to compare against our system. These baselines include best peer systems (PeerT, Peer26, and Peer65) which participated in DUC data evaluations, state-of-the-art summarization results on DUC 2001, 2002 and 2004 corpus respectively. We select the systems that performed best on DUC 2001, 2002, 2004 datasets, which are: (1) R2N2 (Cao et al., 2015a), (2) ClusterCMRW (Wan and Yang, 2008), (3) REGSUM (Hong and Nenkova, 2014), (4) PriorSum (Cao et al., 2015b), and (5) RASR (Ren et al., 2016). The R2N2 system uses a recursive neural network to rank sentences by automatically learning to weigh hand-crafted features. ClusterCMRW system leverages the cluster-level information and incorporates this information into a graph-based ranking algorithm. REGSUM follows a word regression approach for doing better estimation of word importance which leads to better extractive summaries. PriorSum captures summary prior nature by exploiting phrase based information. RASR uses regression framework that simultaneously learns the model importance and redundancy information by calculating the relative gain of a sentence with respect to given set of selected sentences. Further, we use LexRank (Erkan and Radev, 2004) as a baseline to compare performance level of regression approaches. We also compare with StandardCNN and Reg_Manual. StandardCNN consists of just conventional CNNs with fixed window size for learning sentence representation. Reg_Manual is used as a baseline system to explore and understand the effects of learned sentence representation prior to the summary. It adopts human-compiled document-independent features: (a) NUMBER (if a number exists), (b) NENTITY (if named entities exist), and (c) STOPRATIO (the ratio of stopwords). It combines these features with document dependent features and tunes the feature weights through LIBLINEAR[1] support vector regression.

5 Results and Analysis

In this section, we compare the performance of our system against various summarization baselines using ROUGE-1 (unigram match) and ROUGE-2 (bigram match) measures. We also attempt to analyze our system trained with different approaches with intuition and empirical evidence presented in the form of tables and graphs. Lastly, we conclude this section by presenting examples of sentences selected for summaries by the proposed system.

We carried out extensive experiments with diverse settings in order to evaluate our system. In doing so, we created several variations of HNet model which are: (1) HNet-B: uses Convolutional BLSTM as sentence encoder instead of CSTI. (2) HNet-B(T): refers to HNet-B model which is trained with transfer learning approach, i.e., the model was first trained on Daily Mail dataset (Cheng and Lapata, 2016) and was then fine-tuned on multi-document DUC datasets. (3) HNet: refers to our proposed model with CSTI as sentence encoder for sentence ranking and Siam-CSTI as redundancy identifier for sentence selection task. (4) HNet(T): refers to HNet model which was first trained on Daily Mail dataset (Cheng and Lapata, 2016) and was then fine-tuned on multi-document DUC datasets. (5) HNet$^-$: refers to the HNet model when the embedding from the extractor (ϕ_e) is made zero.

It is evident from the results presented in Table 2 that our basic systems HNet-B and HNet-B (T) significantly outperform (T-test with p-value=0.05) state-of-the-art summarization systems R2N2, Cluster-CMRW, REGSUM, PriorSum, and RASR. This is encouraging because despite having not so complex deep network architecture the HNet-B system is able to learn efficient document-dependent semantic features. It also outperforms the Reg_Manual baseline which uses human-compiled features for obtaining the document-independent features and the graph-based summarization system LexRank. From the results shown in Table 2 it is clear that HNet-B outperforms the StandardCNN baseline. This is due to the fact that the additional BLSTM network used in HNet-B helps in learning temporal (sequential) dependencies among variable length phrases exploiting past as well as future context. Finally, our proposed model (HNet/HNet (T)) significantly outperforms our basic systems HNet-B/HNet-B (T) which supports the fact that HNet is equipped with suitable deep network architecture for procuring latent semantic features (document-independent features) from a sentence. In the following, we analyze different aspects of the proposed system.

Contribution of Document Independent Features To explore the contribution of the learned document independent features towards the saliency estimation of a sentence prior to the summary, we follow a simple approach. For each sentence, we ignore document dependent features by setting the ϕ_e vector to $\mathbf{0}$, and then applying the regression transform to calculate the saliency score. We refer to this model as HNet$^-$. This setting helps us in analyzing the intuitive features latent in our heterogeneous representation of the sentence without consideration of the contextual features. After comparing results of HNet$^-$ and HNet in Table 2, we observe a difference of around 3–4 points and 1–2 points in terms of ROUGE-1 and ROUGE-2 scores respectively. The drop in points has resulted due to the absence of document dependent features. Therefore, we can conclude that document independent features have a major contribution to-

[1] http://www.csie.ntu.edu.tw/~cjlin/liblinear

2001			2002			2004		
System	**ROUGE-1**	**ROUGE-2**	**System**	**ROUGE-1**	**ROUGE-2**	**System**	**ROUGE-1**	**ROUGE-2**
PeerT	33.03	7.86	Peer26	35.15	7.64	Peer65	37.88	9.18
R2N2	35.88	7.64	ClusterCMRW	38.55	8.65	REGSUM	38.57	9.75
LexRank	33.43	6.09	LexRank	35.29	7.54	LexRank	37.87	8.88
Reg_Manual	35.95	7.86	Reg_Manual	35.81	8.32	Reg_Manual	38.24	9.74
StandardCNN	35.19	7.63	StandardCNN	35.73	8.69	StandardCNN	37.9	9.93
PriorSum	35.98	7.89	PriorSum	36.63	8.97	PriorSum	38.91	10.07
RASR	36.31	8.49	RASR	37.8	9.61	RASR	36.6	10.57
HNet-B	36.82	8.64	HNet-B	38.79	9.43	HNet-B	39.27	10.85
HNet-B(T)	37.69	9.12	HNet-B(T)	39.52	9.69	HNet-B(T)	39.9	11.08
HNet	37.21	8.96	HNet	39.17	9.61	HNet	39.54	10.94
HNet$^-$	34.51	7.88	HNet$^-$	35.86	8.24	HNet$^-$	35.66	9.37
HNet(T)	**38.18**	**9.43**	HNet(T)	**39.94**	**9.92**	HNet(T)	**40.34**	**11.29**

Table 2: Comparison Results (%) on DUC Datasets

wards saliency estimation of a sentence. This experiment also supports the need of document dependent features as incorporating them results in significant increase in ROUGE scores as provided in Table 2.

Significance of BTI in CSTI (HNet Model) After performing rigorous experiments, we observe that the use of BTI as part of CSTI significantly enhances the performance of the HNet system. This fact is evident when we compare HNet performance against StandardCNN and PriorSum as they use only CNN for obtaining semantic representation of a sentence. The performance improvement is better reflected in the case of HNet(T) system because of increase in the training data. Adding BLSTM Tree Indexer increases the number of parameters to be learned in the network. The more the training data the better the robustness of the system. HNet also outperforms (T-test with p-value=0.04) HNet-B. This is due to the fact that BTI constructs a full binary tree by processing the input sequence with its node functions in a bottom-up fashion. It compresses all the input information into a single representation vector of the root. This representation seems to capture the sequential and recursive dependencies among various units (words/phrases) of the sentence.

Significance of Siam-CSTI in Sentence Selection From Table 1 it is evident that Siam-CSTI performs better (T-test with p-value=0.02) than similar state-of-the-art architectures: TreeLSTM (Tai, Socher, and Manning, 2015) and MaLSTM (Mueller and Thyagarajan, 2016) for sentence similarity task. We also experimented with basic TF-IDF cosine similarity and empirically found the superior performance of Siam-CSTI. The network seems to exploit the sequential and recursive aspects of the sentences to learn a rich set of semantics that help in identifying similar sentences.

Contribution of Transfer Learning Method The fact that increase in training data results in better performance as the system becomes more robust motivated us to pre-train the HNet-B and HNet systems on Daily Mail dataset first and then fine-tune the systems to multi-document summarization setting. We refer to these systems as HNet-B(T) and HNet(T). Table 2 shows the improvement in results for these systems in terms of ROUGE-1 and ROUGE-2 scores on DUC benchmark datasets. HNet(T) is the best performing system amongst the HNet variants.

Examples of Sentences Selected by HNet(T): In Table 3, we provide examples of some high scored sentences and low scored sentences selected by our HNet(T) system. From Table 3, we observe that the learned representation high-scores the sentences that consist of more facts (named entities, numbers etc.) and low-scores the sentences that contain more stop-words and/or are informal and so often fail to provide rich facts.

High scored	• The largest tanker spill in history resulted from the July 19, 1979, collision off Tobago of the supertankers Atlantic Empress and Aegean Captain, in which 300,000 tons more than 80 million gallons of oil was lost. • If the approximate 200,000 illegal aliens were not counted, the county would loose an estimated $56 million a year in federal revenue and lose representatives in Congress.
Low scored	• His coach and physician had also testified at the inquiry. • The House had twice rejected efforts to exclude aliens. • However, that oil burned as well as spilled. • The new growth will attract a larger variety of birds and other animal life to the area.

Table 3: Example Sentences Selected by HNet(T)

6 Conclusions

We proposed a novel deep neural network to learn sentence representations combining both document-dependent and document-independent aspects. The architecture consists of a CSTI which acts as a sentence encoder, an extractor module which extracts document dependent features, a Siam-CSTI net which identifies redundant sentences, and a regression layer which performs sentence saliency scoring. The proposed system discovers various inherent semantic and compositional aspects as part of document-independent features. We also showed that the use of transfer learning approach helps in overcoming the learning issues faced by the network due to the shortage of training data for multi-document summarization. Experimental results on DUC 2001, 2002, and 2004 datasets confirmed that our system outperforms several state-of-the-art baselines.

References

Bowman, S. R.; Manning, C. D.; and Potts, C. 2015. Tree-Structured Composition in Neural Networks with-

out Tree-Structured Architectures. In *NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

Cao, Z.; Wei, F.; Dong, L.; Li, S.; and Zhou, M. 2015a. Ranking with Recursive Neural Networks and Its Application to Multi-Document Summarization. In *AAAI*, 2153–2159.

Cao, Z.; Wei, F.; Li, S.; Li, W.; Zhou, M.; and Wang, H. 2015b. Learning Summary Prior Representation for Extractive Summarization. In *ACL (2)*, 829–833.

Carbonell, J., and Goldstein, J. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *SIGIR*, 335–336.

Cheng, J., and Lapata, M. 2016. Neural Summarization by Extracting Sentences and Words. In *ACL*.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Conroy, J. M.; Schlesinger, J. D.; Goldstein, J.; and Oleary, D. P. 2004. Left-Brain/Right-Brain Multi-Document Summarization. In *Proc. of the Document Understanding Conf. (DUC 2004)*.

Erkan, G., and Radev, D. R. 2004. Lexrank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research* 22:457–479.

Galley, M. 2006. A Skip-Chain Conditional Random Field for Ranking Meeting Utterances by Importance. In *EMNLP*, 364–372. Association for Computational Linguistics.

Ganesan, K.; Zhai, C.; and Han, J. 2010. Opinosis: A Graph-based Approach to Abstractive Summarization of Highly Redundant Opinions. In *Proc. of the 23rd Intl. Conf. on Computational Linguistics*, 340–348. Association for Computational Linguistics.

Hong, K., and Nenkova, A. 2014. Improving the Estimation of Word Importance for News Multi-Document Summarization. In *EACL*, 712–721.

Kågebäck, M.; Mogren, O.; Tahmasebi, N.; and Dubhashi, D. 2014. Extractive Summarization using Continuous Vector Space Models. In *Proc. of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL*, 31–39. Citeseer.

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. *arXiv preprint arXiv:1412.6980*.

Li, S.; Ouyang, Y.; Wang, W.; and Sun, B. 2007. Multi-Document Summarization using Support Vector Regression. In *Proc. of DUC*. Citeseer.

Lin, C.-Y., and Hovy, E. 2003. Automatic Evaluation of Summaries using N-Gram Co-occurrence Statistics. In *NAACL-HLT*, 71–78. Association for Computational Linguistics.

Marelli, M.; Bentivogli, L.; Baroni, M.; Bernardi, R.; Menini, S.; and Zamparelli, R. 2014. SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment. In *Proc. of the 8th Intl. Workshop on Semantic Evaluation*, 1–8.

McDonald, R. 2007. A Study of Global Inference Algorithms in Multi-Document Summarization. In *European Conf. on Information Retrieval*, 557–564. Springer.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, 3111–3119.

Mueller, J., and Thyagarajan, A. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI*, 2786–2792.

Osborne, M. 2002. Using Maximum Entropy for Sentence Extraction. In *Proc. of the ACL-02 Workshop on Automatic Summarization-Volume 4*, 1–8. Association for Computational Linguistics.

Ren, P.; Wei, F.; Chen, Z.; Ma, J.; and Zhou, M. 2016. A Redundancy-Aware Sentence Regression Framework for Extractive Summarization. In *COLING*, 33–43.

Shen, D.; Sun, J.; Li, H.; Yang, Q.; and Chen, Z. 2007. Document Summarization Using Conditional Random Fields. In *IJCAI*, 2862–2867.

Singh, A. K.; Gupta, M.; and Varma, V. 2017. Hybrid MemNet for Extractive Summarization. In *Proc. of the 2017 ACM on Conf. on Information and Knowledge Management*, 2303–2306.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank. In *Proc. of the 2013 Conf. on Empirical Methods in Natural Language Processing*, 1631–1642.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *ACL*, 1556–1566.

Wan, X., and Yang, J. 2008. Multi-Document Summarization using Cluster-based Link Analysis. In *SIGIR*, 299–306.

Wan, X., and Zhang, J. 2014. CTSUM: Extracting more Certain Summaries for News Articles. In *SIGIR*, 787–796.

Yu, H., and Munkhdalai, T. 2017. Neural Tree Indexers for Text Understanding. In *EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, 11–21.

Zhu, X.; Sobihani, P.; and Guo, H. 2015. Long Short-Term Memory over Recursive Structures. In *ICML*, 1604–1612.