Optimizing Sentence Modeling and Selection for Document Summarization

Wenpeng Yin\textsuperscript{1} and Yulong Pei\textsuperscript{2}
\textsuperscript{1}The Center for Information and Language Processing, University of Munich
wenpeng@cis.uni-muenchen.de
\textsuperscript{2}School of Computer Science, Carnegie Mellon University
yulongp@cs.cmu.edu

Abstract

Extractive document summarization aims to conclude given documents by extracting some salient sentences. Often, it faces two challenges: 1) how to model the information redundancy among candidate sentences; 2) how to select the most appropriate sentences. This paper attempts to build a strong summarizer \textit{DivSelect+CNNLM} by presenting new algorithms to optimize each of them. Concretely, it proposes \textit{CNNLM}, a novel neural network language model (NNLM) based on convolutional neural network (CNN), to project sentences into dense distributed representations, then models sentence redundancy by cosine similarity. Afterwards, it formulates the selection process as an optimization problem, constructing a diversified selection process (\textit{DivSelect}) with the aim of selecting some sentences which have high prestige, meantime, are dis-similar with each other. Experimental results on DUC2002 and DUC2004 benchmark data sets demonstrate the effectiveness of our approach.

1 Introduction

Automatic document summarization, aiming at concluding given documents by a piece of concise text, has been intensively studied in recent decades. Existing extraction-based summarization systems were mostly implemented by developing a selection model to choose sentences from a candidate set [Erkan and Radev, 2004a; Wan and Yang, 2008; Pei et al., 2012]. Usually, those selection models considered sentences based on their centrality or prestige in a connectivity network.

Naturally, the performance of such a process depends considerably on two aspects: 1) how to model the information overlapping among candidate sentences; 2) how to pick out the most salient sentences. The first aspect suffers mainly from information deficiency, lexical co-referencing, implicit semantic and other flexible linguistic usages in sentences, which make traditional counting-based approaches hard to discover true semantic or topical relation between sentences. The second one is seldom achieved by considering a global optimization objective in respect of prestige, diversity etc.

This paper tries to build a better summarizer through optimizing the two aspects separately. In view of the successful applications of representation learning by deep neural network in lots of natural language processing (NLP) tasks [Collobert et al., 2011; Blunsom et al., 2014; Le and Mikolov, 2014], we propose a novel representation learning approach \textit{CNNLM} based on convolutional neural network (CNN) [LeCun et al., 1998]. Previous CNN-based methods were designed for specific classification tasks, hence they could only learn biased sentence representations such as sentiment-focused [Blunsom et al., 2014], subjectivity-focused [Kim, 2014] etc. However, unbiased sentence representations are needed in this generic summarization task. Hence, the novelty of \textit{CNNLM} lies in that \textit{n-gram language model (LM)} is leveraged to convert traditional CNN architecture into an unsupervised learning regime. To be concrete, given a sentence, \textit{CNNLM} extracts the sentence representation by hierarchical neural network, then combines that sentence representation with the representations of context words to predict the next word. This kind of prediction-based NNLMs are shown more powerful than traditional counting-based methods in representation learning [Baroni et al., 2014]. Sentence representations enable to calculate pairwise sentence similarities, therefore we can build a connectivity graph and use PageRank [Page et al., 1999] to derive the sentence prestige.

A good summary is supposed to have a low degree of redundancy. Hence, beyond prestige, diversity has also been recognized as a crucial objective in selection. However, some pioneering work suffered from a severe problem that top-ranked sentences usually share much redundant information. Although there exist some researches like [Aliguliyev, 2006; Wan, 2008] that can control redundancy via some strategies, such as clustering and MMR (Maximum Marginal Relevance), few approaches could combine the two properties, i.e., prestige and diversity, into a unified selection process.

Recently, diversified ranking has attracted much attention. For example, Mei et al., [2010] developed vertex-reinforced random walk over an adjacent graph to conduct a comprehensive quantification of objects with regard to their prestige as well as diversities. Tong et al., [2011] and He et al., [2012] introduced optimization viewpoints to solve diversified ranking problem for query-oriented situations. Inspired, this paper proposes a diversified selection algorithm \textit{DivSelect} for generic document summarization from an optimization perspective.
Our model automatically balances the prestige and diversity of the early-selected sentences in a principled way. The integrated algorithm DivSelect+CNNLM, combining innovations on both aspects, gets promising results on DUC2002 and DUC2004 benchmark data sets.

2 Related Work

Prior work with similar idea, namely modeling sentence similarity as well as optimizing selection process, occupies a large proportion. Considering the pioneering work LexRank [Erkan and Radev, 2004b] and LexPageRank [Erkan and Radev, 2004a], both computed cosine similarity based on TF-IDF (Term Frequency-Inverse Document Frequency) matrix first, then used ranking algorithm PageRank [Page et al., 1999] to calculate sentence prestige. In addition, another typical ranking algorithm in web mining HITS [Kleinberg, 1999] is also popularly studied in document summarization [Wan, 2008; Wan and Yang, 2008].

Lots of subsequent work attempted to make progresses in either sentence similarity calculation or selection strategy or even both. Aliguliyev [2009] presented a method to measure dissimilarity between sentences using the normalized google distance [Ciliberti and Vitani, 2007], then performed sentence clustering and selected the most distinctive sentences from each cluster to form summaries. Chali and Joty [2008] studied query-biased summarization, where sentence similarity was based on n-gram overlap, longest common subsequence (LCS), skip-bigram overlap and so on, then sentences selection in each cluster depended on similarities towards the query. In [Wang et al., 2008], sentence similarity was composed of pairwise word similarity from WordNet [Fellbaum, 1998]. Yin et al., [2012a] combined LCS, weighted LCS, skip-bigram statistic with word semantic similarity derived by Latent Dirichlet Allocation [Blei et al., 2003] for sentence similarity learning, and exploited traditional PageRank to select sentences. Obviously, a trend in learning sentence similarity is trying to keep order information, meanwhile integrating word similarity further. Inspired, we develop convolutional neural network to learn sentence representation which is supposed to be able to absorb the global information of sentence structure as well as a high-level abstraction of word semantics.

Furthermore, lots of researchers have attempted to extend the traditional graph-based models. For example, Pei et al., [2012] decomposed traditional PageRank graph into multiple sub-graphs on the basis of topic distribution, finally ranked sentences by averaging over all sub-graphs. Yin et al., [2012b] built sentence connection network as a tensor where heterogeneous relations were actually different latent topics, then tried to co-rank sentences and topics simultaneously. In addition, Wan et al., [2007] integrated sentence-to-sentence, word-to-word, and sentence-to-word graphs into one comprehensive network, and proposed a reinforcement ranking algorithm to select prestigious sentences and keywords together. In a word, such kind of algorithms generally hold a similar rationale: taking global information into consideration rather than relying only on vertex-specific information. Therefore, they have been proved effective in summarization task. However, they rarely integrated diversity into a unified ranking process, and consequently had to resort to some extra strategies to achieve redundancy reduction. Contrarily, our proposal DivSelect is able to produce a diversified top-k ranking list which facilitates the sentence selection for summary generation without extra steps.

3 CNNLM: Optimizing Sentence Modeling

First we provide some notation conventions: bold-face lowercase letters denote vectors, bold-face upper-case letters denote matrices, and calligraphic upper-case letters denote sets.

Convolutional neural network (CNN) is good at extracting global features of the input (sentence here) [LeCun et al., 1998]. We propose a new framework CNNLM based on CNN to learn sentence representations, then compute sentence similarity via cosine measure.

![Figure 1: CNNLM for learning sentence representations.](image)

Concretely, our CNNLM, as illustrated in Figure 1, includes:

**Input Layer:** A sentence with \( s \) words: \( w_0, w_1, \ldots, w_{s-1} \). Each word \( w_i \) is denoted by an initialized vector \( w_i \in \mathbb{R}^d \) (“300” in Figure 1 denotes representation of dimension 300).

**Convolution Layer:** a convolution layer uses sliding filters to extract features of local phrases in the sentence. The filter width \( l \), i.e., phrase length, is a parameter. We first concatenate the representations of \( l \) consecutive words \((w_i, w_{i+1}, \ldots, w_{i+l-1})\) as \( c_i \in \mathbb{R}^d \) (\( 0 \leq i \leq s-l \)), then generate the representation of this phrase as \( u_i \in \mathbb{R}^d \) using a tanh activation function and a linear projection matrix \( \mathbf{W} \in \mathbb{R}^{d \times d} \) which is the same across all phrases in the sentence, as:

\[
u_i = \tanh(\mathbf{W} \cdot c_i + b)
\]

**Max-pooling Layer:** Based on representations of all regional phrases, max-pooling layer extracts the maximum value from each dimension to form sentence representation \( x \in \mathbb{R}^d \). Let \( u_{i,j} \) denote the value of \( j \)th dimension in phrase representation \( u_i \), the \( j \)th value in \( x \) can be derived by:

\[
x_j = \max(u_{0,j}, u_{1,j}, u_{2,j}, \ldots, u_{s-l,j})
\]
3.1 Unsupervised Training

Most applications of CNN are implemented to extract sentence features for specific classification task, which requires supervised training and a label for each training sample [Blunsom et al., 2014; Kim, 2014]. Contrarily, here we need unbiased sentence representations, and labels are unavailable nor necessary. In order to train the CNN in an unsupervised scheme, we form a n-gram language model by element-wisely averaging sentence representation and the representations of context words to predict the next word, as depicted 3.1 Unsupervised Training in Figure 1. This process resembles the CBOW scheme in word2vec [Mikolov et al., 2013] which has no global sentence features to help prediction, resembles also the PV-DM model in [Le and Mikolov, 2014] while our sentence representation is derived by CNN.

We employ noise-contrastive estimation (NCE) [Mnih and Teh, 2012] to compute the cost: the model learns to discriminate between true next words and noise words. NCE allows us to fit unnormalized models, making the training time effectively independent of the vocabulary size.

Generally, this new model uses not only the preceding context but also the whole sentence to predict the next word. As no labels are needed, such a training framework enables to produce general sentence representations, no longer sentiment-biased nor other task-specific representations as some literature did. Finally, a sentence adjacent graph based on sentence similarity is built for next phase.

4 DivSelect: Optimizing Sentence Selection

In this section, we discuss in detail our DivSelect model. First, we present an objective function to measure the quality of a sentence set with size k that conveys both the prestige and the diversity; then conduct some theoretical analysis regarding its challenges and properties.

Specifically, assume a set \( N \) of \( n \) sentences \( \{x_1, x_2, \ldots, x_n\} \) for the target document collection, let \( S \) denote the \( n \times n \) symmetric similarity matrix obtained in the above section, where \( S_{i,j} \) is the entry of \( S \) in the \( i \)th row and the \( j \)th column (\( i, j = 1, \ldots, n \)).

For a given \( S \), we exploit PageRank algorithm to derive the prestige vector, presented as \( p \), for all sentences in \( N \). Our goal is to identify an objective subset \( C \) of \( k \) sentences which are prestigious and diverse to each other. Here the positive integer \( k \) is the budget of the desired sentence set.

4.1 Objective Function

As our goal conducts, we use \( Q(C) \) in Equation 3 to measure the quality of a random collection \( C \) which contains \( k \) sentences. Naturally, “quality” here means the overall prestige and dis-similarity within that set.

\[
\arg \max_{|C|=k} Q(C) = \alpha \sum_{i \in C} p_i^2 - \sum_{i,j \in C} p_i S_{i,j} p_j \tag{3}
\]

where \( \alpha \) is a positive regularization parameter that defines the trade-off between the two terms, i.e., prestige and diversity, and \( p_i \) is the \( i \)th entry of prestige vector \( p \).

Apparently, our objective function prefers a set which values high-prestige sentences while penalizing them if they exhibit similar content. Note that in the second half of Equation 3, we take into consideration two sentences’ mutual similarity as well as their respective prestige while penalizing redundancy between them. It is different with the work in [Tong et al., 2011] which only paid attention to the pairwise similarity and one object’s score. Here, we argue that the same similarity values have different degrees of damages, depending on the prestige of the sentence pair linked by that similarity. The prestige of a pair of sentences could be treated as their possibilities of being selected. So, given a fixed similarity score, the bigger the overall prestige of two sentences, the heavier the penalization to them.

4.2 Challenges of Objective Function

Our task is essentially a subset selection problem to find the optimal \( k \) sentences that maximize Equation 3. Note that the Densest \( k \)-Subgraph (D\&S) problem is NP-hard [Feige et al., 2001]. Unfortunately, the following derivation proves that our proposal is equivalent to the D\&S problem, in other words, it is also NP-hard to find the optimal solution.

Given an undirected connectivity graph \( G = (N, E) \) with the affinity matrix \( M \), where \( N \) and \( E \) are the node set and edge set, respectively. \( M \) is a \( |N| \times |N| \) symmetric matrix with entries being 0 or 1. Based on these notations, the D\&S problem is defined as following formula:

\[
\mathcal{R} = \arg \max_{|R|=k} \sum_{i,j \in R} M_{i,j} \tag{4}
\]

Now, look at above described D\&S definition with another angle. Define matrix \( \mathcal{M} = I - M \), then task in Equation 4 could be described as following representation:

\[
\mathcal{R} = \arg \min_{|R|=k} \sum_{i,j \in R} \mathcal{M}_{i,j} \tag{5}
\]

Note that \( \sum_{i,j=1}^{|N|} M_{i,j} = |N|^2 - |E| \) is a constant. Let set \( \mathcal{C} = N \setminus \mathcal{R} \), then, Equation 5 is further equivalent to

\[
\arg \max_{|\mathcal{C}|=|\mathcal{N}\setminus\mathcal{R}|} \sum_{i \in \mathcal{R}, j \in \mathcal{C}} M_{i,j} + \sum_{i \in \mathcal{C}, j \in \mathcal{C}} M_{i,j} = \arg \max_{|\mathcal{C}|=|\mathcal{N}\setminus\mathcal{R}|} \sum_{i \in \mathcal{R}, j \in \mathcal{C}} M_{i,j} + \sum_{i \in \mathcal{C}, j \in \mathcal{C}} M_{i,j} \tag{6}
\]

Next, we attempt to elaborate that Equation 6 can be treated as an instance of the optimization task in Equation 3 under following constraints: let the similarity matrix \( S \) in Equation 3 be \( M \), the prestige distribution \( p \) be \( \mathbf{1}_{|N| \times 1} \), and the parameter \( \alpha \) be 2. Under such settings, the objective function in Equation 3 becomes

\[
\arg \max_{|\mathcal{C}|=k} Q(C) = \alpha \sum_{i \in \mathcal{C}} p_i^2 - \sum_{i,j \in \mathcal{C}} p_i S_{i,j} p_j = 2 \sum_{i \in \mathcal{C}} \sum_{j=1}^{|N|} p_i M_{i,j} p_j - \sum_{i,j \in \mathcal{C}} p_i M_{i,j} p_j \quad \text{PageRank} \tag{7}
\]

\[
= 2 \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} p_i M_{i,j} p_j + \sum_{i,j \in \mathcal{C}} p_i M_{i,j} p_j \quad \text{symmetry of } M \tag{8}
\]

\[
= 2 \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{C}} M_{i,j} + \sum_{i,j \in \mathcal{C}} M_{i,j} \quad \text{(dfn. of } p) \tag{9}
\]
which is equivalent to the objective function in Equation 6. Thus, we have proved that the objective function in Equation 3 is an NP-hard problem.

### 4.3 Diminishing Returns Property of \( Q(C) \)

Above subsection have elaborated that our proposed process is an NP-hard problem. Then, what is the condition that enables us to find a near-optimal solution? Here, we give the so-called diminishing returns property of \( Q(C) \), which is summarized in Theorem 1. The intuitive explanation of diminishing returns property is as follows: (a) by \( P_1 \), it suggests that the objective value is 0 if we get an empty set; (b) by \( P_2 \), if we add more sentences to the current subset, the overall quality of the ranking list does not decrease; and (c) by \( P_3 \), the marginal gain of adding new sentences is relatively small if we already have a large subset.

**Theorem 1. Diminishing Returns Property of \( Q(C) \).** Let \( \emptyset \) be an empty set, \( C_1, C_2 \) and \( L \) be three sets, s.t., \( C_1 \subseteq C_2 \subseteq N \) and \( L = C_2 \setminus C_1 \). The objective function presented in Equation 3 has the following properties:

1. **(P_1) Monotonicity.** For any \( \alpha \geq 2 \), the objective function \( Q(C) \) is monotonically non-decreasing w.r.t \( C \);
2. **(P_2) Submodularity.** For any \( \alpha > 0 \), the objective function \( Q(C) \) is submodular w.r.t \( C \).

**Proof of (P_1).** It is obviously held by the definition of \( Q(C) \).

**Proof of (P_2).** In view of \( L = C_2 \setminus C_1 \) and the formula in Equation 3, we have

\[
Q(C_2) - Q(C_1) = 2 \sum_{i \in C_2} p_i^2 - \sum_{i,j \in C_2} p_i S_{i,j} p_j - (2 \sum_{i \in C_1} p_i^2 - \sum_{i,j \in C_1} p_i S_{i,j} p_j)
\]

\[
= 2 \sum_{i \in C_2} p_i^2 - \sum_{i,j \in C_2} p_i S_{i,j} p_j - 2 \sum_{i \in C_1} p_i^2 + \sum_{i,j \in C_1} p_i S_{i,j} p_j
\]

\[
= 2 \sum_{i \in L} p_i^2 - (\sum_{i \in C_1} p_i S_{i,j} p_j + \sum_{i,j \in C_2} p_i S_{i,j} p_j)
\]

\[
= (\sum_{i \in L} p_i^2 - \sum_{i \in C_1} p_i S_{i,j} p_j + \sum_{i \in L \setminus C_1} p_i S_{i,j} p_j)
\]

\[
(8)
\]

The first half of Equation 8 satisfies

\[
\sum_{j \in L} p_j^2 - \sum_{i \in C_1} p_i S_{i,j} p_j
\]

\[
= \sum_{j \in L \setminus C_1} p_j S_{i,j} p_j - \sum_{i \in C_1} p_i S_{i,j} p_j \quad \text{(PageRank)}
\]

\[
= \sum_{j \in L \setminus C_1} p_j S_{i,j} p_j - \sum_{i \in C_1} p_i S_{i,j} p_j \geq 0 \quad (C_1 \subseteq N)
\]

Similarly, we can prove that the second half of Equation 8 is also not less than 0. Putting Equations (8-9) together, we have that \( Q(C_2) \geq Q(C_1) \), which completes the proof of \( P_2 \).

**Proof of (P_3).** Let a new set \( C_3 \) s.t., \( C_3 \cap C_2 = \emptyset \). We have

\[
Q(C_2 \cup C_3) - Q(C_2) = (2 \sum_{i \in C_2 \cup C_3} p_i^2 - \sum_{i,j \in C_2 \cup C_3} p_i S_{i,j} p_j - (2 \sum_{i \in C_2} p_i^2 - \sum_{i,j \in C_2} p_i S_{i,j} p_j)
\]

\[
= 2 \sum_{i \in C_3} p_i^2 - (\sum_{i,j \in C_2} p_i S_{i,j} p_j - \sum_{i,j \in C_2} p_i S_{i,j} p_j)
\]

\[
= 2 \sum_{i \in C_3} p_i^2 - (2 \sum_{i,j \in C_3} p_i S_{i,j} p_j + \sum_{i,j \in C_3} p_i S_{i,j} p_j)
\]

Similarly, we could have

\[
Q(C_1 \cup C_3) - Q(C_1) = 2 \sum_{i \in C_3} p_i^2 - (2 \sum_{i \in C_3} p_i S_{i,j} p_j + \sum_{i,j \in C_3} p_i S_{i,j} p_j)
\]

**Accordingly, putting Equations (10-11) together produces**

\[
Q(C_1 \cup C_3) - Q(C_1) - (Q(C_2 \cup C_3) - Q(C_2))
\]

\[
= 2(\sum_{i \in C_3} p_i S_{i,j} p_j - \sum_{i \in C_3} p_i S_{i,j} p_j)
\]

\[
\geq 0 \quad (C_1 \subseteq C_2)
\]

which completes the proof of \( P_3 \).

### 5 The Proposed Optimization Algorithm

Given the affinity matrix \( S \) of a large collection of sentences, and the budget \( k \), we aim to find a subset of \( k \) sentences that maximizes the function \( Q(\cdot) \) described in Equation 3. In this section, we first present our iterative algorithm. Subsequently, its near-optimality and complexity will be analyzed.

#### 5.1 Algorithm Description

Our proposed near-optimal solution is presented in Alg. 1. In step 1, based on the similarity matrix \( S \), we derive the sentence prestige via widely-used PageRank. Then, we perform some initialization (step 2). Note that “\( \emptyset \)” denotes the element-wise product between two vectors. In the process of **for** \( \cdots \) **end for** (steps 3-7), we select \( k \) sentences one-by-one as follows. At each time, we add the one with the highest score from \( s \) into set \( C \), then use this sentence to update (or penalize) the scores of all sentences (step 6). Intuitively, the score vector \( s \) keeps the marginal contribution of each sentence for the quality given the currently selected subset \( C \). It also can be seen that at each iteration, the values of such marginal contribution either keeps unchanged or decreases, which is consistent with \( P_3 \) of Theorem 1.

To be specific, the initialization of \( s \) in step 2 originates from \( s_i = Q(x_i) - Q(\emptyset) = \alpha p_i^2 - p_i S_{i,j} p_j = (\alpha - 1)p_i^2 \).

Namely, we first select a sentence with the highest prestige. The updating of \( s \) in step 6 corresponds to \( Q(C' \cup x_i) - Q(C') \), where \( C' \) denotes a temporary non-empty set. Similar with Equation 11, we could easily get that \( Q(C' \cup x_i) - Q(C') = \alpha p_i^2 - p_i S_{i,j} p_j - 2 \sum_{j \in C'} p_j S_{i,j} p_j \).
We conduct experiments on the data sets DUC2002\(^1\) and DUC2004\(^2\) in which generic multi-document summarization has been one of the fundamental tasks (i.e., task 2 in DUC2002 and task 2 in DUC2004). Each task has a gold standard data set consisting of document sets and reference summaries. Table 1 gives a short summary of above data sets. Documents are pre-processed by segmenting sentences and splitting words. Stop words are removed and the remaining words are stemmed using Porter stemmer\(^3\).

<table>
<thead>
<tr>
<th></th>
<th>DUC2002</th>
<th>DUC2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Task2</td>
<td>Task2</td>
</tr>
<tr>
<td>Number of documents</td>
<td>567</td>
<td>500</td>
</tr>
<tr>
<td>Number of clusters</td>
<td>59</td>
<td>50</td>
</tr>
<tr>
<td>Data source</td>
<td>TREC</td>
<td>TDT</td>
</tr>
<tr>
<td>Summary length</td>
<td>200 words</td>
<td>665 bytes</td>
</tr>
</tbody>
</table>

Table 1: Summary of data sets

\(^1\)http://www-nlpir.nist.gov/projects/duc/data/2002_data.html
\(^3\)http://tartarus.org/martin/PorterStemmer/

6.1 Data Set and Setup

6.2 Compared Methods

In this work, we compare our DivSelect+CNRLM with following representative methods: 1) Random: select sentences randomly from document set to construct summaries. 2) Lead: take the first sentences one by one from the last document in the collection, where documents are assumed to be ordered chronologically. 3) LexRank\[^{[Erkan and Radev, 2004b]}\]: it used PageRank to find prestigious sentences and exploited MMR greedy algorithm to keep low redundancy. Hence, this method is also able to control a balance of high prestige and low redundancy to some extent. 4) DivRank\[^{[Wang et al., 2008]}\]: this is a typical Markov walk based diversified ranking algorithm. It uses accumulative visit times to skew the original adjacent graph, aiming to increase the score gap of similar objects. We choose some sentences with high DivRank scores to produce summaries. 5) SNMF\[^{[Wang et al., 2008]}\]: it used symmetric non-negative matrix factorization (SNMF) to cluster sentences into groups, then selected sentences from each group for summary generation. Note that the original work\[^{[Wang et al., 2008]}\] is for topic-biased summarization. Namely, the ranking score of each sentence is influenced by the topic-sentence affinity. As we focus on generic summarization, therefore there is no topic-focused bias in our implementation.

The above baselines try to cover typical summarization approaches. For example, Random and Lead represent feature-based methods, LexRank represents graph-based methods, SNMF represents clustering-based methods, and DivRank is representative of diversified ranking approaches. It is worth mentioning that our algorithm is unsupervised. Thus, we do not consider literature concerning supervised methods.

We use the officially adopted ROUGE\[^{[Lin, 2004]}\] (version 1.5.5) toolkit\(^4\) for evaluation. ROUGE measures summary quality through counting overlapping units such as the n-gram, word sequences and word pairs between the candidate summary (produced by various algorithms) and the reference summary (produced by humans). Here we report the average F-measure scores of ROUGE-1, ROUGE-2 and ROUGE-SU4, which base on uni-gram match, bi-gram match, and unigram plus skip-bigram match with maximum skip distance of 4 between the candidate summary and the reference summary, respectively.

**CNLM setup.** DUC data is relatively small for training a neural network. In experiments, we first pre-train CNRLM on one million sentences from English Gigawords\[^{[Robert, 2009]}\], then further train it on DUC data to learn representation for each sentence in DUC. Additionally, like some literature did, pre-trained word representations by\[^{[Mikolov et al., 2013]}\]\(^5\) are used to initialize the input layer of Figure 1 and fine-tuned during training. The filter width \(l = 5\). Five left context words are used in “average” layer. For each true example, 10 noise words are sampled in NCE. All words, phrases and sentences have 300-dimensional representations.

5.2 Algorithm Analysis

Now, we first analyze why our Alg.1 is a near-optimal solution of our defined objective function. Here is a lemma:

**Lemma 1. Near-Optimality of Alg.1.** Let \(C\) be the sentence subset obtained by our algorithm, and \(C^*\) be the theoretically optimal subset. Hence, \(|C| = k\) and \(C^* = \arg \max_{|C| = k} Q(C)\). We have \((1 - 1/e)Q(C^*) \leq Q(C) \leq Q(C^*)\), where \(e\) is the base of the natural logarithm.

**Proof.** Indicated by\[^{[He et al., 2012; Nemhauser et al., 1978]}\], the breach of the proof is to verify that for any sentence \(x_i \notin C, s_i = Q(C \cup x_i) - Q(C)\). Apparently, the remaining part of the proof directly follows the diminishing returns property of the objective function in Theorem 1. We omit the proof details for brevity.

6 Experiments

6.1 Data Set and Setup

6.2 Compared Methods

In this work, we compare our DivSelect+CNRLM with following representative methods: 1) Random: select sentences randomly from document set to construct summaries. 2) Lead: take the first sentences one by one from the last document in the collection, where documents are assumed to be ordered chronologically. 3) LexRank\[^{[Erkan and Radev, 2004b]}\]: it used PageRank to find prestigious sentences and exploited MMR greedy algorithm to keep low redundancy. Hence, this method is also able to control a balance of high prestige and low redundancy to some extent. 4) DivRank\[^{[Wang et al., 2008]}\]: this is a typical Markov walk based diversified ranking algorithm. It uses accumulative visit times to skew the original adjacent graph, aiming to increase the score gap of similar objects. We choose some sentences with high DivRank scores to produce summaries. 5) SNMF\[^{[Wang et al., 2008]}\]: it used symmetric non-negative matrix factorization (SNMF) to cluster sentences into groups, then selected sentences from each group for summary generation. Note that the original work\[^{[Wang et al., 2008]}\] is for topic-biased summarization. Namely, the ranking score of each sentence is influenced by the topic-sentence affinity. As we focus on generic summarization, therefore there is no topic-focused bias in our implementation.

The above baselines try to cover typical summarization approaches. For example, Random and Lead represent feature-based methods, LexRank represents graph-based methods, SNMF represents clustering-based methods, and DivRank is representative of diversified ranking approaches. It is worth mentioning that our algorithm is unsupervised. Thus, we do not consider literature concerning supervised methods.
6.3 Experimental Results and Analysis

The experimental results are concluded in Tables 2-3. As those statistics indicate, our proposed method DivSelect+CNNLM performs best. Among all the systems, selecting leading sentences or randomly show the poorest performance on both data sets. It is easily understood that these two baselines could not discover truly prestigious sentences, let alone controlling redundancy. In addition, the performances of neither SNMF nor LexRank are promising. Although SNMF reduces information redundancy via clustering, the calculation of sentence prestige within a group/cluster may be affected by the available neighbors. Apparently, the most competitive method is DivRank, which determines a balanced score for the diversity and prestige properties of sentences with a vertex-reinforced random walk. However, such kind of random walk essentially changes the structure of information network. Consequently, the relevance between sentences, which fully depends on the primitive network structure, may not be well captured [Du et al., 2011]. Therefore, DivRank focuses on diversity, at the cost of sacrificing the prestige of the entire ranking list. Whereas, DivSelect+CNNLM keeps the adjacent graph stable, and at each step, striving to add a new sentence which could improve the value of the objective function to the most extent.

<table>
<thead>
<tr>
<th>Systems</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-SU4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.38469</td>
<td>0.11705</td>
<td>0.18007</td>
</tr>
<tr>
<td>Lead</td>
<td>0.39860</td>
<td>0.16042</td>
<td>0.20315</td>
</tr>
<tr>
<td>LexRank</td>
<td>0.47366</td>
<td>0.23105</td>
<td>0.25839</td>
</tr>
<tr>
<td>SNMF</td>
<td>0.48783</td>
<td>0.24929</td>
<td>0.27103</td>
</tr>
<tr>
<td>DivRank</td>
<td>0.48825</td>
<td>0.25361</td>
<td>0.27644</td>
</tr>
<tr>
<td>DivSelect+CNNLM</td>
<td><strong>0.51013</strong></td>
<td><strong>0.26972</strong></td>
<td><strong>0.29431</strong></td>
</tr>
</tbody>
</table>

Table 2: F-measure comparison on DUC2002.

<table>
<thead>
<tr>
<th>Systems</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-SU4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.31857</td>
<td>0.06299</td>
<td>0.11780</td>
</tr>
<tr>
<td>Lead</td>
<td>0.33182</td>
<td>0.06348</td>
<td>0.10582</td>
</tr>
<tr>
<td>LexRank</td>
<td>0.38071</td>
<td>0.08319</td>
<td>0.13032</td>
</tr>
<tr>
<td>SNMF</td>
<td>0.38325</td>
<td>0.09217</td>
<td>0.13316</td>
</tr>
<tr>
<td>DivRank</td>
<td>0.38851</td>
<td>0.09555</td>
<td>0.13827</td>
</tr>
<tr>
<td>DivSelect+CNNLM</td>
<td><strong>0.40907</strong></td>
<td><strong>0.10723</strong></td>
<td><strong>0.14969</strong></td>
</tr>
</tbody>
</table>

Table 3: F-measure comparison on DUC2004

6.4 Competitiveness of CNNLM

Above subsection has indicated the superiority of our DivSelect+CNNLM algorithm. However, it could not not demonstrate the robustness of DivSelect optimization framework. Will an alternative sentence modeling endow the optimization algorithm such excellent performance either? To answer these questions, we keep DivSelect framework stable while replacing CNNLM with below three representative approaches: 1) VSM: cosine measure based on TF-IDF matrix; 2) LSA: first exert SVD factorization on the sentence-word TF-IDF matrix to learn sentence representation, then compute cosine similarity; 3) LCS: [Yin et al., 2012a], it combines LCS, WLCS, skip-bigrams and word semantic similarities; 4) PV: Paragraph vector by Le and Mikolov [2014]

<table>
<thead>
<tr>
<th>Systems</th>
<th>DUC2002</th>
<th>DUC2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>DivSelect+VSM</td>
<td>0.48666</td>
<td>0.38319</td>
</tr>
<tr>
<td>DivSelect+LSA</td>
<td>0.48921</td>
<td>0.38857</td>
</tr>
<tr>
<td>DivSelect+LCS</td>
<td>0.48936</td>
<td>0.38903</td>
</tr>
<tr>
<td>DivSelect+PV</td>
<td>0.50478</td>
<td>0.39720</td>
</tr>
<tr>
<td>DivSelect+CNNLM</td>
<td><strong>0.51013</strong></td>
<td><strong>0.40907</strong></td>
</tr>
</tbody>
</table>

Table 4: Performance of similarity measures.

From Table 4, we can see that the combination of DivSelect and CNNLM performs best. Unexpectedly, system VSM cannot provide much contribution to our DivSelect framework for possibly it not only fails to capture the high-level semantics, but loses the sentence structure information. “DivSelect+LSA” and “DivSelect+LCS” get very comparable performances. It should be due to that LSA is good at extracting latent semantics while LCS (including WLCS,skip-bigrams) succeeds to take structure feature into account. That seems having given good explanations for the good performance of our proposed “DivSelect+CNNLM”: actually, the CNNLM is a co-training structure for learning sentence representations and word representations. Word representations derived by NNLMs have been proved owning high-quality semantics [Mikolov et al., 2013]. More importantly, the filters of CNN are used to extract local features, like the function of longest common subsequence, while max-pooling is able to extract the globally dominant features by considering all local features comprehensively. Hence, using CNNLM to learn sentence representation seems a promising approach.

7 Conclusions

Like some prior work, this paper also split the summarization task into two subtasks: calculation of sentence similarity and sentence selection. We developed an unsupervised CNN scheme to learn sentence representations, and proposed a new sentence selection algorithm DivSelect to balance sentence prestige and diversity. Experimental results on DUC2002 and DUC2004 data sets are very promising.

Acknowledgments

The authors thank the anonymous reviewers for constructive comments.

References

