Abstractive Multi-Document Summarization via Phrase Selection and Merging*

Lidong Bing[§] **Piji Li**[♮] **Yi Liao**[♮] **Wai Lam**[♮]
Weiwei Guo[†] **Rebecca J. Passonneau**[‡]
[§]Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA USA
[♮]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong
[†]Yahoo Labs, Sunnyvale, CA, USA
[‡]Center for Computational Learning Systems, Columbia University, New York, NY, USA
[§]lbing@cs.cmu.edu, [♮]{pjli, yliao, wlam}@se.cuhk.edu.hk
[†]wguo@yahoo-inc.com, [‡]becky@ccls.columbia.edu

Abstract

We propose an abstraction-based multi-document summarization framework that can construct new sentences by exploring more fine-grained syntactic units than sentences, namely, noun/verb phrases. Different from existing abstraction-based approaches, our method first constructs a pool of concepts and facts represented by phrases from the input documents. Then new sentences are generated by selecting and merging informative phrases to maximize the salience of phrases and meanwhile satisfy the sentence construction constraints. We employ integer linear optimization for conducting phrase selection and merging simultaneously in order to achieve the global optimal solution for a summary. Experimental results on the benchmark data set TAC 2011 show that our framework outperforms the state-of-the-art models under automated pyramid evaluation metric, and achieves reasonably well results on manual linguistic quality evaluation.

1 Introduction

Existing multi-document summarization (MDS) methods fall in three categories: extraction-based, compression-based and abstraction-based. Most summarization systems adopt the **extraction-based** approach which selects some original sentences from the source documents to create a short summary (Erkan and Radev, 2004; Wan et al., 2007). However, the restriction that the whole sentence should be selected potentially yields some overlapping information in the summary. To this end, some researchers apply compression on the selected sentences by deleting words or phrases (Knight and Marcu, 2000; Lin, 2003; Zajic et al., 2006; Harabagiu and Lacatusu, 2010; Li et al., 2015), which is the **compression-based** method. Yet, these compressive summarization models cannot merge facts from different source sentences, because all the words in a summary sentence are solely from one source sentence.

In fact, previous investigations show that human-written summaries are more abstractive, which can be regarded as a result of sentence aggregation and fusion (Cheung and Penn, 2013; Jing and McKeown, 2000). Some works, albeit less popular, have studied **abstraction-based** approach that can construct a sentence whose fragments come from different source sentences. One important work developed by Barzilay and McKeown (2005) employed sentence fusion, followed by (Filippova and Strube, 2008; Filippova, 2010). These works first conduct clustering on sentences to compute the salience of topical themes. Then, sentence fusion is applied within each cluster of related sentences to generate a new sentence containing common information units of the sentences. The abstractive-based approaches gather information across sentence boundary, and hence have the potential to cover more content in a more concise manner.

In this paper, we propose an abstractive MDS framework that can construct new sentences by

* The work described in this paper is substantially supported by grants from the Research and Development Grant of Huawei Technologies Co. Ltd (YB2013090068/TH138232) and the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Codes: 413510 and 14203414).
The work was done when Weiwei Guo was in Columbia University

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1587–1597,
Beijing, China, July 26-31, 2015.

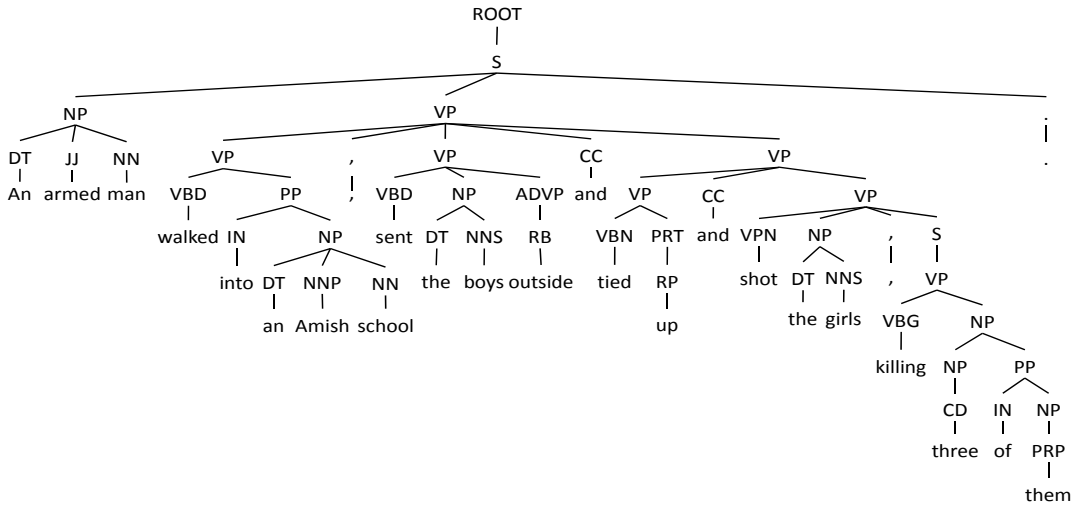

Figure 1: The constituency tree of a sentence from a news document.

exploring more fine-grained syntactic units than sentences, namely, noun/verb phrases (NPs/VPs). This idea is based on two observations. First, the major constituent phrases loosely correspond to the concepts and facts. After reading a set of documents describing the same topic or event, a person digests these documents as key concepts and facts in his/her mind, such as "*an armed man*" and "*walked into an Amish school*" from Figure 1. Second, a summary writer re-organizes the key concepts and facts to form new sentences for the summary. Accordingly, our proposed framework has two major components corresponding to the above observations. The first component creates a pool of concepts and facts represented by NPs and VPs from the input documents. A salience score is computed for each phrase by exploiting redundancy of the document content in a global manner. The second component constructs new sentences by selecting and merging phrases based on their salience scores, and ensures the validity of new sentences using a integer linear optimization model.

The contribution of this paper is two folds. (1) We extract NPs/VPs from constituency trees to represent key concepts/facts, and merge them to construct new sentences, which allows more summary content units (SCUs) (Nenkova and Passonneau, 2004) to be included in a sentence by breaking the original sentence boundaries. (2) The designed optimization framework for addressing the problem is unique and effective. Our optimization algorithm **simultaneously** selects and merges a set of phrases that maximize the number of covered SCUs in a summary. Meanwhile, since the basic unit is phrases, we design compatibility relations among NPs and VPs, as well as other optimization constraints, to ensure that the generated sentences contain correct facts. Compared with the sentence fusion approaches that compute salience scores of sentence clusters, our proposed framework explores a more fine-grained textual unit (i.e., phrases), and maximizes the salience of selected phrases in a global manner.

2 Description of Our Framework

We first introduce how to extract NPs and VPs from constituency trees, and subsequently calculate salience scores for them. Then we formulate the sentence generation task as an optimization problem, and design constraints. In the end, we perform several post-processing steps to improve the order and the readability of the generated sentences.

2.1 Phrase Salience Calculation

The first component decomposes the sentences in documents into a set of noun phrases (NPs) derived from the subject parts of a constituency tree and a set of verb-object phrases (VPs), representing potential key concepts and key facts, respectively. These phrases will serve as the basic elements for sentence generation.

We employ Stanford parser (Klein and Manning, 2003) to obtain a constituency tree for each input sentence. After that, we extract NPs and VPs from the tree as follows: (1) The NPs and VPs that are the direct children of the sentence node (repre-

sented by the **S** node) are extracted. (2) VPs (NPs) in a path on which all the nodes are VPs (NPs) are also recursively extracted and regarded as having the same parent node **S**. Recursive operation in the second step will only be carried out in two levels since the phrases in the lower levels may not be able to convey a complete fact. Take the tree in Figure 1 as an example, the corresponding sentence is decomposed into phrases "*An armed man*", "*walked into an Amish school, sent the boys outside and tied up and shot the girls, killing three of them*", "*walked into an Amish school*", "*sent the boys outside*", and "*tied up and shot the girls, killing three of them*". [1] Because of the recursive operation, the extracted phrases may have overlaps. Later, we will show how to avoid such overlapping in phrase selection.

A salience score is calculated for each phrase to indicate its importance. Different types of salience can be incorporated in our framework, such as position-based method (Yih et al., 2007), statistical feature based method (Woodsend and Lapata, 2012), concept-based method (Li et al., 2011), etc. One key characteristic of our approach is that the considered basic units are phrases instead of sentences. Such finer granularity leaves more room for better global salience score by potentially covering more distinct facts. In our implementation, we adopt a concept-based weight incorporating the position information. The concept set is designated to be the union set of unigrams, bigrams, and named entities in the documents. We remove stopwords and perform lemmatization before extracting unigrams and bigrams. The position-based term frequency is used in the concept weighting scheme. When counting the frequency, each occurrence of a concept in an input document is weighted with the paragraph position. The weight larger than 1 is given to the concept occurrences in the first few paragraphs. Specifically, the weight of the first paragraph is B and the weight decreases as the position of the paragraph increases from the beginning of the document. The weighting function is:

$$H(p) = \begin{cases} \rho^p * B & \text{if } p < -(\log B / \log \rho) \\ 1 & \text{otherwise} \end{cases}, \quad (1)$$

where p is the position of the paragraph starting from 0, from beginning of the document, and ρ is a positive constant and smaller than 1. Then, the salience of a phrase is calculated as the summed weights of its concepts.

2.2 New Sentence Construction Model

The construction of new sentences is formulated as an optimization problem which is able to simultaneously generate a group of sentences. Each new sentence is composed of one NP and at least one VP, where the NP and VPs may come from different source sentences. In the process of new sentence generation, the compatibility relation between NP and VP and a variety of summarization requirements are jointly considered.

2.2.1 Compatibility Relation

Compatibility relation is designed to indicate whether an NP and a VP can be used to form a new sentence. For example, the NP "*Police*" from another sentence should not be the subject of the VP "*sent the boys outside*" extracted from Figure 1. We use some heuristics to find compatibility, and then expand the compatibility relation to more phrases by extracting coreference.

To find coreference NPs (different mentions for the same entity), we first conduct coreference resolution for each document with Stanford coreference resolution package (Lee et al., 2013). We adopt those resolution rules that are able to achieve high quality and address our need for summarization. In particular, Sieve 1, 2, 3, 4, 5, 9, and 10 in the package are used. A set of clusters are obtained and each cluster contains the mentions that refer to the same entity in a document. The clusters from different documents in the same topic are merged by matching the named entities. After merging, the mentions that are not NPs extracted in the phrase extraction step are removed in each cluster. Two NPs in the same cluster are determined as alternative of each other.

To find alternative VPs, Jaccard Index is employed as the similarity measure. Specifically, each VP is represented as a set of its concepts and the index value is calculated for each pair of VPs. If the value is larger than a threshold, the two VPs are determined as alternative of each other.

[1] We only consider the recursive operation for a VP with more than one parallel sub-VPs, such as the highest VP in Figure 1. The sub-VPs following modal, link or auxiliary verbs are not extracted as individual VPs. In addition, we also extract the clauses functioning as subjects of sentences as NPs, such as "that clause". Note that we also mention such clauses as "noun phrase" although their syntactic labels could be "SBAR" or "S".

We then define an indicator matrix $\Gamma_{|\mathbf{N}||\mathbf{V}|}$, in which $\Gamma[i,j]=1$ if an NP N_i and a VP V_j come from the same node **S** in the constituency tree, otherwise, $\Gamma[i,j]=0$. Let $\tilde{\mathbf{N}}_i$ and $\tilde{\mathbf{V}}_i$ represent the alternative phrases of N_i and V_i as described above. The compatibility matrix $\tilde{\Gamma}_{|\mathbf{N}||\mathbf{V}|}$ is defined as follows:

$$\tilde{\Gamma}[p,q]=\begin{cases}1 & \text{if } N_p \in \tilde{\mathbf{N}}_i \wedge \Gamma[i,q]=1\\ 1 & \text{if } V_q \in \tilde{\mathbf{V}}_j \wedge \Gamma[p,j]=1\\ 1 & \text{if } \Gamma[p,q]=1\\ 0 & \text{otherwise}\end{cases} \quad (2)$$

where $\tilde{\Gamma}[p,q]=1$ means N_p and V_q are compatible/permitted for constructing a new sentence. $\tilde{\Gamma}$ is the final compatibility matrix that we use in the optimization. The first case of Equation 2 implies that if N_p and N_i are coreferent, N_p can replace N_i and serve as the subject of N_i's VP (i.e., V_q). The second case implies that if V_q is very similar to V_j, V_q can be concatenated to V_j's NP (i.e., N_p).

2.2.2 Phrase-based Content Optimization

The overall objective function of our optimization formulation to select NPs and VPs is defined as:

$$\max\{\sum_i \alpha_i S_i^N - \sum_{i<j}\alpha_{ij}(S_i^N+S_j^N)R_{ij}^N + \sum_i \beta_i S_i^V - \sum_{i<j}\beta_{ij}(S_i^V+S_j^V)R_{ij}^V\}, \quad (3)$$

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). If N_i and N_j are coreferent, $R_{ij}^N=1$. Otherwise, the similarity is calculated with the above Jaccard Index based method. The notations are summarized in Table 1.

Specifically, we maximize the salience score of the selected NPs and VPs as indicated by the first and the third terms in Equation 3, and penalize the selection of similar NP pairs and similar VP pairs as indicated by the second and the fourth terms. Meanwhile, the phrase selection is governed by a set of constraints so that the selected phrases can generate valid sentences. The constraints will be explained in details in Section 2.2.3.

One characteristic of our objective function is that NPs and VPs are treated differently, i.e., there are different selection/penalty terms for NP and VP. Such design enables us to avoid the false penalty between an NP and a VP. For example, the algorithm produces two sentences: the first sentence is "*the gunman shot ...*" with an NP "*the gunman*", and the other sentence has a VP "*confirmed the gunman died*". Obviously, we should not penalize the redundancy between them, because mentioning the gunman is necessary in both sentences.

Notation	Description
N_i, V_i	Noun phrase i and verb phrase i
α_i, β_i	Selection indicators of N_i and V_i
α_{ij}, β_{ij}	Co-occurrence indicators of pairs (N_i, N_j) and (V_i, V_j)
S_i^N, S_i^V	Salience scores of N_i and V_i
R_{ij}^N, R_{ij}^V	Similarity of pair (N_i, N_j) and pair (V_i, V_j)
$\Gamma_{\|\mathbf{N}\|\|\mathbf{V}\|}$	$\Gamma[i,j]=1$ if N_i and V_j are from the same sentence
$\tilde{\mathbf{N}}_i$, $\tilde{\mathbf{V}}_i$	The alternative phrases of N_i and V_i
$\tilde{\Gamma}_{\|\mathbf{N}\|\|\mathbf{V}\|}$	$\tilde{\Gamma}[i,j]=1$ means N_i and V_j are compatible for being used to construct a new sentence
$\tilde{\gamma}_{ij}$	Sentence generation indicator for N_i and V_j if $\tilde{\Gamma}[i,j]=1$

Table 1: Notations.

2.2.3 Sentence Generation Constraints

To summarize the related sentences in the documents, human writers usually merge the important facts in different VPs about the same entity into a single sentence, and omit the trivial facts. Also, the same entity is likely to be described by coreferent NPs. Therefore, in our approach, only one NP is selected and employed as the subject of the newly generated sentence, which is then concatenated with the merged facts (i.e., VPs). If the compatibility entry $\tilde{\Gamma}[i,j]$ for N_i and V_j is 1, we define a sentence generation indicator $\tilde{\gamma}_{ij}$ to indicate whether both N_i and V_j are selected to construct a new sentence in the summary.

We design the following groups of constraints to realize our aim of phrase selection and new sentence construction. The objective function and constraints are linear, therefore the problem can be solved by existing Integer Linear Programming (ILP) solvers such as simplex algorithm (Dantzig and Thapa, 1997).

`NP validity`. To maintain the consistency between the selection indicator α and the compatibility entry $\tilde{\Gamma}$ for NP N_i, we introduce two constraints as follows:

$$\forall i,j,\alpha_i \geq \tilde{\gamma}_{ij};\ \ \forall i, \sum_j \tilde{\gamma}_{ij} \geq \alpha_i. \quad (4)$$

These two constraints work together to ensure the valid assignment of α according to the compatibility entry $\tilde{\Gamma}$.

VP legality. Similarly, the following requirement guarantees the consistency between the selection indicator β and the compatibility entry $\tilde{\Gamma}$ for selected VP V_i:

$$\forall j, \sum_i \tilde{\gamma}_{ij} = \beta_j. \quad (5)$$

The above two constraints jointly ensure that the selected NPs and VPs are able to form new summary sentences according to the values of sentence generation indicators.

Not i-within-i. Two phrases in the same path of a constituency tree cannot be chosen at the same time:

$$\begin{array}{c} \text{if } \exists V_k \rightsquigarrow V_j, \text{then } \beta_k + \beta_j \leq 1, \\ \text{if } \exists N_k \rightsquigarrow N_j, \text{then } \alpha_k + \alpha_j \leq 1. \end{array} \quad (6)$$

For example, "*walked into an Amish school, sent the boys outside and tied up and shot the girls, killing three of them*" and "*walked into an Amish school*" cannot be both incorporated in the summary, because of the obvious redundancy.

Phrase co-occurrence. These constraints control the co-occurrence relation of NPs or VPs. For NPs, we introduce three constraints:

$$\alpha_{ij} - \alpha_i \leq 0, \quad (7)$$
$$\alpha_{ij} - \alpha_j \leq 0, \quad (8)$$
$$\alpha_i + \alpha_j - \alpha_{ij} \leq 1. \quad (9)$$

Constraints 7 to 9 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:

$$\beta_{ij} - \beta_i \leq 0, \quad (10)$$
$$\beta_{ij} - \beta_j \leq 0, \quad (11)$$
$$\beta_i + \beta_j - \beta_{ij} \leq 1. \quad (12)$$

Sentence number. In abstractive summarization, we do not prefer to generate many short sentences. This is controlled by:

$$\sum_i \alpha_i \leq K, \quad (13)$$

where K is the maximum number of sentences.

Short sentence avoidance. We do not select the VPs from very short sentences because a short sentence normally cannot convey a complete key fact (Woodsend and Lapata, 2012).

$$\text{if } l(\mathbf{S}) < M, V_i \in \mathbf{S}, \text{then } \beta_i = 0, \quad (14)$$

where M is the threshold of the sentence length.

Pronoun avoidance. We exclude the NPs that are pronouns from being selected as the subject of the new sentences. As previously observed (Woodsend and Lapata, 2012), pronouns are normally not used by human summary writers. It is because the summary is short and the narration relation of sentences is relatively simple so that pronouns are not needed. Moreover, in automatic summary, pronouns will cause ambiguity in the summary, especially when the sentence order is automatically determined. Therefore, we model the constraint as:

$$\text{if } N_i \; is \; pronoun, \text{then } \alpha_i = 0. \quad (15)$$

Length constraint. The overall length of the selected NPs and VPs is no larger than a limit L:

$$\sum_i \{l(N_i) * \alpha_i\} + \sum_j \{l(V_j) * \beta_j\} \leq L, \quad (16)$$

where $l()$ is the word-based length of a phrase.

2.3 Postprocessing

Recall that we require that one NP and at least one VP compose a sentence. Thus, we form a raw sentence with a selected NP as the subject followed by the corresponding selected VPs that are indicated by sentence generation indicator $\tilde{\gamma}_{ij}$ having the value 1. The VPs in a summary sentence are ordered according to their natural order if they come from the same document. Otherwise, they are ordered according to the timestamps of the corresponding documents. After that, if the total length is smaller than L, we add conjunctions such as "and" and "then" to concatenate the VPs for improving the readability of the newly generated sentences. The pseudo-timestamp of a sentence is defined as the earliest timestamp of its VPs and the sentences are ordered based on their pseudo-timestamps.

2.4 Relation to Existing MDS Approaches

Many existing extraction-based and compression-based MDS approaches could be regarded as special cases under our framework: (1) To simulate

extraction-based summarization, we just need to constrain that the highest NP and the highest VP from the same sentence are selected simultaneously. In addition, no NPs and VPs in lower levels can be selected. Thus, the output only contains the original sentences of the source documents. (2) To simulate compression-based summarization, we can adapt our framework to conduct sentence selection and sentence compression in a joint manner. Specifically, we only need to restrict that the NP and VPs of a summary sentence must come from the same original sentence.

3 Experiments

3.1 Experimental Setup

The data set of traditional summarization task in Text Analysis Conference (TAC) 2011 is used to evaluate the performance of our approach. This data set is the latest one and it contains 44 topics. Each topic falls into one of 5 predefined event categories and contains 10 related news documents. There are four writers to write model summaries for each topic.

The data set of traditional summarization task in TAC 2010 is employed as the development/tuning data set. This data set contains 46 topics from the same predefined categories. Each topic also has 10 documents and 4 model summaries.

Based on the tuning set, the key parameters of our model are set as follows. The constants B and ρ in the weighting function are set to 6 and 0.5 repectively. The similarity threshold in obtaining the alternative VPs is 0.75. We did not observe significant difference between cosine similarity and Jaccard Index.

We mainly evaluate the system by pyramid evaluation. To gain a comprehensive understanding, we also evaluate by ROUGE evaluation and manual linguistic quality evaluation.

3.2 Results with Pyramid Evaluation

The pyramid evaluation metric (Nenkova and Passonneau, 2004) involves semantic matching of summary content units (SCUs) so as to recognize alternate realizations of the same meaning. Different weights are assigned to SCUs based on their frequency in model summaries. A weighted inventory of SCUs named a pyramid is created, which constitutes a resource for investigating alternate realizations of the same meaning. Such property makes pyramid method more suitable to evaluate summaries. Another widely used evaluation metric is ROUGE (Lin and Hovy, 2003) and it evaluates summaries from word overlapping perspective. Because of the strict string matching, it ignores the semantic content units and performs better when larger sets of model summaries are available. In contrast to ROUGE, pyramid scoring is robust with as few as four model summaries (Nenkova and Passonneau, 2004). Therefore, in recent summarization evaluation workshops such as TAC, the pyramid is used as the major metric.

System	Auto-pyr (Th: .6)	Auto-pyr (Th: .65)	Rank in TAC 2011
Our	0.905	0.793	NA
22	0.878	0.775	1
43	0.875	0.756	2
17	0.860	0.741	3

Table 2: Comparison with the top 3 systems in TAC 2011.

Since manual pyramid evaluation is time-consuming, and the exact evaluation scores are not reproducible especially when the assessors for our results are different from those of TAC, we employ the automated version of pyramid proposed in (Passonneau et al., 2013). The automated pyramid scoring procedure relies on distributional semantics to assign SCUs to a target summary. Specifically, all n-grams within sentence bounds are extracted, and converted into 100 dimension latent topical vectors via a weighted matrix factorization model (Guo and Diab, 2012). Similarly, the contributors and the label of an SCU are transformed into 100 dimensional vector representations. An SCU is assigned to a summary if there exists an n-gram such that the similarity score between the SCU low dimensional vector and the n-gram low dimensional vector exceeds a threshold. Passonneau et al. (2013) showed that the distributional similarity based method produces automated scores that correlate well with manual pyramid scores, yielding more accurate pyramid scores than string matching based automated methods (Harnly et al., 2005). In this paper, we adopt the same setting as in (Passonneau et al., 2013): a 100 dimension matrix factorization model is learned on a domain independent corpus, which is drawn from sense definitions of WordNet and Wiktionary[2], and Brown corpus. We exper-

[2]http://en.wiktionary.org/

	ROUGE-2			ROUGE-SU4		
System	P	R	F1	P	R	F1
Our	0.117	0.117	0.117	0.148	0.147	0.148
22	0.112	0.114	0.113	0.147	0.150	0.148
43	0.132	0.135	0.134	0.162	0.166	0.164
17	0.128	0.131	0.129	0.157	0.160	0.159

Table 3: Performance under ROUGE metric.

iment with 2 threshold values, i.e., 0.6 and 0.65, similar to those used in (Passonneau et al., 2013).

The top three systems in TAC 2011 evaluated with manual pyramid score were System 22 (Li et al., 2011), 43, and 17 (Ng et al., 2011). Table 2 shows the comparison with them under the automated pyramid evaluation. Our method achieves the best results in both thresholds, which means that our method is able to find more semantic content units (SCUs) than the state-of-the-art system in TAC 2011. In addition, paired t-test (with $p < 0.01$) comparing our model with the best system in TAC 2011, i.e., System 22, shows that the performance of our model is significantly better. It is worth noting that the three systems used additional external linguistic resources: System 22 used a Wikipedia corpus for providing domain knowledge, System 17 and 43 defined some category-specific features. Without any domain adaption, our framework can still achieve encouraging performance.

We calculate Pearson's correlation to measure how well the automatic pyramid approximates the manual pyramid scores for 50 system submissions in TAC 2011. The values are 0.91 and 0.93 for thresholds 0.6 and 0.65 respectively. It demonstrates that the automated pyramid is reliable to differentiate the performance of different methods.

3.3 Results with ROUGE Evaluation

As mentioned above, we favor the pyramid evaluation over the ROUGE score because it can measure the summary quality beyond simply string matching. Here, we also provide ROUGE score for our reference. ROUGE-1.5.5 package[3] is employed with the same parameters as in TAC. The results are summarized in Table 3. Our performance is slightly better than System 22, and it is not as good as System 43 and 17. The reason is that System 43 and 17 used category-specific features and trained the feature weights with the category information in TAC 2010 data. These features help them select better category-specific content for the summary. However, the usability of such features depends on the availability of predefined categories in the summarization task, as well as the availability of training data with the same predefined categories for estimating feature weights. Therefore, the adaptability of these methods is limited to some extent. In contrast, our framework does not define any category-specific feature and only uses TAC 2010 data to tune the parameters for general summarization purpose.

[3]http://www.berouge.com/Pages/default.aspx

3.4 Linguistic Quality Evaluation

The linguistic quality of summaries is evaluated using the five linguistic quality questions on grammaticality (Q1), non-redundancy (Q2), referential clarity (Q3), focus (Q4), and coherence (Q5) in Document Understanding Conferences (DUC). A Likert scale with five levels is employed with 5 being very good with 1 being very poor. A summary was blindly evaluated by three assessors on each question. System 22 performed better than System 43 and 17 in TAC 2011 on the evaluation of readability, which is an aggregation of the above questions. Considering the intensive labor force of manual assessment, we only conduct comparison with System 22.

The results are given in Table 4. On average, the two systems perform very closely. System 22 is an extraction-based method that picks the original sentences, hence it achieves higher score in Q1 grammaticality, while our approach has some new sentences with grammar mistakes, which is a common problem for abstractive methods and deserves more future research effort. For Q4 focus, our score is higher than System 22, which reveals that our summary sentences are relatively more cohesive. The score of Q3 referential clarity shows that the referential relation is basically clear in our summaries, even when new sentences are automatically generated. In general, ignoring the grammaticality scores, our system still performs better than System 22. Specifically, the average scores of our system and System 22 on the last four questions are 3.37 and 3.33 respectively.

4 Qualitative Results

4.1 Analysis of Summary Sentence Type

There are three types of sentences in the summaries generated by our framework, namely, new

System	Q1	Q2	Q3	Q4	Q5	AVG
Our	3.67	3.50	3.90	3.23	2.83	3.43
22	4.13	3.50	3.97	2.97	2.87	3.49

Table 4: Evaluation of linguistic quality.

sentences, compressed sentences, and original sentences. A new sentence is constructed by merging the phrases from different original sentences. A compressed sentence is generated by deleting phrases from an original sentence. An original sentence in the summary is directly extracted from the input documents.

The percentage of different types of sentences in our summaries is calculated. About 33% of the summary sentences are newly constructed. This demonstrates that our framework has good capability of merging phrases from the original sentences so as to convey more information in compacted summaries. In addition, about 44% of the summary sentences are generated by compression. It shows a unique characteristic of our framework: sentence construction and sentence compression are conducted in a unified model.

4.2 Case Study

Table 5 shows the summary of the first topic, i.e., "*Amish Shooting*", by our framework. The summary sentence ID and the sentence type are given in the form of "`[summary sentence ID: sentence type]`". Each selected phrase and the original sentence ID where the phrase originated are given in the form of "`{selected phrase (original sentence ID)}`". There are three compressed sentences with IDs 1, 2, and 4, one new sentence with ID 3, and two original sentences with IDs 5 and 6.

The new sentence is constructed from the following original sentences in which the extracted NPs and VPs are indicated with colored parentheses:

(84): On Monday morning, (NP Charles Carl Roberts IV) (VP (VP entered the West Nickel Mines Amish School in Lancaster County) and (VP shot 10 girls), (VP killing five)).
(85): (NP Roberts) (VP killed himself as police stormed the building).
(150): (NP Roberts) (VP left what they described as rambling notes for his family).

[1:C] {An armed man (25)} {walked into an Amish school (25)} {tied up and shot the girls, killing three of them. (25)} [2:C] {A man who laid siege to a one-room Amish schoolhouse (64)} {told his wife shortly before opening fire that he had molested two young girls who were his relatives decades ago (64)} {was tormented by dreams of molesting again. (64)} [3:N] {Charles Carl Roberts IV (84)} {killed himself as police stormed the building (85)} {left what they described as rambling notes for his family. (150)} [4:C] {The gunman (145)} {was not Amish (145)} {had not attended the school. (145)} [5:O] {The shootings (148)} {occurred about 10:45 a.m. (148)} [6:O] {Police (149)} {could offer no explanation for the killings. (149)}

Table 5: The summary of "*Amish Shooting*" topic.

The NPs of these sentences are coreferent so that some of their VPs are merged and concatenated with one NP, i.e., "*Charles Carl Roberts IV*".

The summary sentences with IDs 1, 2, and 4 are compressions from the following original sentences respectively:

(25): (NP An armed man) (VP (VP walked into an Amish school), (VP sent the boys outside) and (VP tied up and shot the girls, killing three of them)), (NP authorities) (VP said).
(64): (NP (NP A man) who laid siege to a one-room Amish schoolhouse), (VP killing five girls), (VP (VP told his wife shortly before opening fire that he had molested two young girls who were his relatives decades ago) and (VP was tormented by "dreams of molesting again")), (NP authorities) (VP said Tue).
(145): According to media reports, (NP the gunman) (VP (VP was not Amish) and (VP had not attended the school)).

Some uncritical information is excluded from the summary sentences, such as "*sent the boys outside*", "*authorities said*", etc. In addition, the VP "*killing five girls*" of the original sentence with ID 64 is also excluded since it has significant redundancy with the summary sentence with ID 1.

5 Related Work

Existing multi-document summarization (MDS) works can be classified into three categories:

extraction-based approaches, compression-based approaches, and abstraction-based approaches.

Extraction-based approaches are the most studied of the three. Early studies mainly followed a greedy strategy in sentence selection (Çelikyilmaz and Hakkani-Tür, 2011; Goldstein et al., 2000; Wan et al., 2007). Each sentence in the documents is firstly assigned a salience score. Then, sentence selection is performed by greedily selecting the sentence with the largest salience score among the remaining ones. The redundancy is controlled during the selection by penalizing the remaining ones according to their similarity with the selected sentences. An obvious drawback of such greedy strategy is that it is easily trapped in local optima. Later, unified models are proposed to conduct sentence selection and redundancy control simultaneously (McDonald, 2007; Filatova and Hatzivassiloglou, 2004; Yih et al., 2007; Gillick et al., 2007; Lin and Bilmes, 2010; Lin and Bilmes, 2012; Sipos et al., 2012). However, extraction-based approaches are unable to evaluate the salience and control the redundancy on the granularity finer than sentences. Thus, the selected sentences may still contain unimportant or redundant phrases.

Compression-based approaches have been investigated to alleviate the above limitation. As a natural extension of the extractive method, the early works adopted a two-step approach (Lin, 2003; Zajic et al., 2006; Gillick and Favre, 2009). The first step selects the sentences, and the second step removes the unimportant or redundant units from the sentences. Recently, integrated models have been proposed that jointly conduct sentence extraction and compression (Martins and Smith, 2009; Woodsend and Lapata, 2010; Almeida and Martins, 2013; Berg-Kirkpatrick et al., 2011; Li et al., 2015). Note that our model also jointly conducts phrase selection and phrase merging (new sentence generation). Nonetheless, compressive methods are unable to merge the related facts from different sentences.

On the other hand, abstraction-based approaches can generate new sentences based on the facts from different source sentences. In addition to the previously mentioned sentence fusion work, new directions have been explored. Researchers developed an information extraction based approach that extracts *information items* (Genest and Lapalme, 2011) or *abstraction schemes* (Genest and Lapalme, 2012) as components for generating sentences. Summary revision was also investigated to improve the quality of automatic summary by rewriting the noun phrases or people references in the summaries (Nenkova, 2008; Siddharthan et al., 2011). Sentence generation with word graph was applied for summarizing customer opinions and chat conversations (Ganesan et al., 2010; Mehdad et al., 2014).

Recently, the factors of information certainty and timeline in MDS task were explored (Ng et al., 2014; Wan and Zhang, 2014; Yan et al., 2011). Researchers also explored some variants of the typical MDS setting, such as query-chain focused summarization that combines aspects of update summarization and query-focused summarization (Baumel et al., 2014), and hierarchical summarization that scales up MDS to summarize a large set of documents (Christensen et al., 2014). A data-driven method for mining sentence structures on large news archive was proposed and utilized to summarize unseen news events (Pighin et al., 2014). Moreover, some works (Liu et al., 2012; Kågebäck et al., 2014; Denil et al., 2014; Cao et al., 2015) utilized deep learning techniques to tackle some summarization tasks.

6 Conclusions and Future Work

We propose an abstractive MDS framework that constructs new sentences by exploring more fine-grained syntactic units, namely, noun phrases and verb phrases. The designed optimization framework operates on the summary level so that more complementary semantic content units can be incorporated. The phrase selection and merging is done simultaneously to achieve global optimal. Meanwhile, the constructed sentences should satisfy the constraints related to summarization requirements such as NP/VP compatibility. Experimental results on TAC 2011 summarization data set show that our framework outperforms the top systems in TAC 2011 under the pyramid metric. For future work, one aspect is to enhance the grammar quality of the generated new sentences and compressed sentences. Another aspect is to improve time efficiency of our framework, and its major bottleneck is the time consuming ILP optimzation.

References

Miguel Almeida and Andre Martins. 2013. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *ACL*, pages 196–206.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Comput. Linguist.*, 31(3):297–328.

Tal Baumel, Raphael Cohen, and Michael Elhadad. 2014. Query-chain focused summarization. In *ACL*, pages 913–922.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *HLT*, pages 481–490.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *AAAI*.

Asli Çelikyilmaz and Dilek Hakkani-Tür. 2011. Concept-based classification for multi-document summarization. In *ICASSP*, pages 5540–5543.

Jackie Chi Kit Cheung and Gerald Penn. 2013. Towards robust abstractive multi-document summarization: A caseframe analysis of centrality and domain. In *ACL*, pages 1233–1242.

Janara Christensen, Stephen Soderland, Gagan Bansal, and Mausam. 2014. Hierarchical summarization: Scaling up multi-document summarization. In *ACL*, pages 902–912.

George B. Dantzig and Mukund N. Thapa. 1997. *Linear Programming 1: Introduction*. Springer-Verlag New York, Inc.

Misha Denil, Alban Demiraj, Nal Kalchbrenner, Phil Blunsom, and Nando de Freitas. 2014. Modelling, visualising and summarising documents with a single convolutional neural network. *arXiv preprint arXiv:1406.3830*.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. In *COLING*.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *EMNLP*, pages 177–185.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *COLING*, pages 322–330.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *COLING*, pages 340–348.

Pierre-Etienne Genest and Guy Lapalme. 2011. Framework for abstractive summarization using text-to-text generation. In *MTTG*, pages 64–73.

Pierre-Etienne Genest and Guy Lapalme. 2012. Fully abstractive approach to guided summarization. In *ACL*, pages 354–358.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Workshop on ILP for NLP*, pages 10–18.

Dan Gillick, Benoit Favre, and Dilek Hakkani-tür. 2007. The icsi summarization system at tac 2008. In *Proc. of Text Understanding Conference*.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLP-AutoSum*, pages 40–48.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *ACL*, pages 864–872.

Sanda Harabagiu and Finley Lacatusu. 2010. Using topic themes for multi-document summarization. *ACM Trans. Inf. Syst.*, 28(3):13:1–13:47.

Aaron Harnly, Ani Nenkova, Rebecca Passonneau, and Owen Rambow. 2005. Automation of summary evaluation by the pyramid method. In *RANLP*.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and paste based text summarization. In *NAACL*, pages 178–185.

Mikael Kågebäck, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. 2014. Extractive summarization using continuous vector space models. In *CVSC@EACL*, pages 31–39.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *ACL*, pages 423–430.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *AAAI-IAAI*, pages 703–710.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Comput. Linguist.*, 39(4):885–916.

Huiying Li, Yue Hu, Zeyuan Li, Xiaojun Wan, and Jianguo Xiao. 2011. Pkutm participation in tac2011. In *Proceedings of TAC*.

Piji Li, Lidong Bing, Wai Lam, Hang Li, and Yi Liao. 2015. Reader-aware multi-document summarization via sparse coding. In *IJCAI*.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *HLT*, pages 912–920.

Hui Lin and Jeff Bilmes. 2012. Learning mixtures of submodular shells with application to document summarization. In *UAI*, pages 479–490.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL*, pages 71–78.

Chin-Yew Lin. 2003. Improving summarization performance by sentence compression: a pilot study. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, pages 1–8. Association for Computational Linguistics.

Yan Liu, Sheng-hua Zhong, and Wenjie Li. 2012. Query-oriented multi-document summarization via unsupervised deep learning. In *AAAI*.

André F. T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Workshop on ILP for NLP*, pages 1–9.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *ECIR*, pages 557–564.

Yashar Mehdad, Giuseppe Carenini, and Raymond T. Ng. 2014. Abstractive summarization of spoken and written conversations based on phrasal queries. In *ACL*, pages 1220–1230.

Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL*, pages 145–152.

Ani Nenkova. 2008. Entity-driven rewrite for multi-document summarization. In *Third International Joint Conference on Natural Language Processing, IJCNLP*, pages 118–125.

Jun-Ping Ng, Praveen Bysani, Ziheng Lin, Min yen Kan, and Chew lim Tan. 2011. Swing: Exploiting category-specific information for guided summarization. In *Proceedings of TAC*.

Jun-Ping Ng, Yan Chen, Min-Yen Kan, and Zhoujun Li. 2014. Exploiting timelines to enhance multi-document summarization. In *ACL*, pages 923–933.

Rebecca J. Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. 2013. Automated pyramid scoring of summaries using distributional semantics. In *ACL (2)*, pages 143–147.

Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *ACL*, pages 892–901.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2011. Information status distinctions and referring expressions: An empirical study of references to people in news summaries. *Comput. Linguist.*, 37(4):811–842.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Large-margin learning of submodular summarization models. In *EACL*, pages 224–233.

Xiaojun Wan and Jianmin Zhang. 2014. Ctsum: Extracting more certain summaries for news articles. In *SIGIR*, pages 787–796.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI*, pages 2903–2908.

Kristian Woodsend and Mirella Lapata. 2010. Automatic generation of story highlights. In *ACL*, pages 565–574.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *EMNLP-CoNLL*, pages 233–243.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *SIGIR*, pages 745–754.

Wen-tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. 2007. Multi-document summarization by maximizing informative content-words. In *IJCAI*, pages 1776–1782.

David M. Zajic, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. 2006. Sentence compression as a component of a multi-document summarization system. In *DUC at NLT/NAACL 2006*.