Document Summarization via Guided Sentence Compression

Chen Li[1], **Fei Liu**[2], **Fuliang Weng**[2], **Yang Liu**[1]
[1] Computer Science Department, The University of Texas at Dallas
Richardson, Texas 75080, USA
[2] Research and Technology Center, Robert Bosch LLC
Palo Alto, California 94304, USA
{chenli,yangl@hlt.utdallas.edu}
{fei.liu, fuliang.weng@us.bosch.com}

Abstract

Joint compression and summarization has been used recently to generate high quality summaries. However, such word-based joint optimization is computationally expensive. In this paper we adopt the 'sentence compression + sentence selection' pipeline approach for compressive summarization, but propose to perform *summary guided compression*, rather than generic sentence-based compression. To create an annotated corpus, the human annotators were asked to compress sentences while explicitly given the important summary words in the sentences. Using this corpus, we train a supervised sentence compression model using a set of word-, syntax-, and document-level features. During summarization, we use multiple compressed sentences in the integer linear programming framework to select salient summary sentences. Our results on the TAC 2008 and 2011 summarization data sets show that by incorporating the guided sentence compression model, our summarization system can yield significant performance gain as compared to the state-of-the-art.

1 Introduction

Automatic summarization can be broadly divided into two categories: extractive and abstractive summarization. Extractive summarization focuses on selecting the salient sentences from the document collection and concatenating them to form a summary; while abstractive summarization is generally considered more difficult, involving sophisticated techniques for meaning representation, content planning, surface realization, etc., and the "true abstractive summarization remains a researcher's dream" (Radev et al., 2002).

There has been a surge of interest in recent years on generating compressed document summaries as a viable step towards abstractive summarization. These compressive summaries often contain more information than sentence-based extractive summaries since they can remove insignificant sentence constituents and make space for more salient information that is otherwise dropped due to the summary length constraint. Two general strategies have been used for compressive summarization. One is a pipeline approach, where sentence-based extractive summarization is followed or proceeded by sentence compression (Knight and Marcu, 2000; Lin, 2003; Zajic et al., 2007; Wang et al., 2013). Another line of work uses joint compression and summarization. They have been shown to achieve promising performance (Daumé, 2006; Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Chali and Hasan, 2012; Almeida and Martins, 2013; Qian and Liu, 2013). One popular approach for such joint compression and summarization is via integer linear programming (ILP). However, since words are the units in the optimization framework, solving this ILP problem can be expensive.

In this study, we use the pipeline compression and summarization method because of its computational efficiency. Prior work using such pipeline methods simply uses generic sentence-based compression for each sentence in the documents, no matter whether compression is done before or after summary sentence extraction. We propose to use sum-

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 490–500,
Seattle, Washington, USA, 18-21 October 2013.

mary guided compression combined with ILP-based sentence selection for summarization in this paper. We create a compression corpus for this purpose. Using human summaries for a set of documents, we identify salient words in the sentences. During annotation, the human annotators are given these salient words and asked to generate compressed sentences. We expect such "guided" sentence compression is beneficial for the pipeline compression and summarization task. In addition, previous research on joint modeling for compression and summarization suggested that the labeled extraction and compression data sets would be helpful for learning a better joint model (Daumé, 2006; Martins and Smith, 2009). We hope that our work on this guided compression will also be of benefit to the future joint modeling studies.

Using our created compression data, we train a supervised compression model using a variety of word-, sentence-, and document-level features. During summarization, we generate multiple compression candidates for each sentence, and use the ILP framework to select compressed summary sentences. In addition, we also propose to apply a preselection step to select some important sentences, which can both speed up the summarization system and improve performance. We evaluate our proposed summarization approach on the TAC 2008 and 2011 data sets using the standard ROUGE metric (Lin, 2004). Our results show that by incorporating a guided sentence compression model, our summarization system can yield significant performance gain as compared to the state-of-the-art reported results.

2 Related Work

Summarization research has seen great development over the last fifty years (Nenkova and McKeown, 2011). Compared to the abstractive counterpart, extractive summarization has received considerable attention due to its clear problem formulation – to extract a set of salient and non-redundant sentences from the given document set. Both unsupervised and supervised approaches have been explored for sentence selection. The supervised approaches include the Bayesian classifier (Kupiec et al., 1995), maximum entropy (Osborne, 2002), skip-chain conditional random fields (CRF) (Galley, 2006), discriminative reranking (Aker et al., 2010), among others.

The extractive summary sentence selection problem can also be formulated in an optimization framework. Previous approaches include the integer linear programming (ILP) and submodular functions, which are used to solve the optimization problem. In particular, Gillick et al. (2009) proposed a concept-based ILP approach for summarization. Li et al. (2013) improved it by using supervised stragety to estimate concept weight in ILP framework. In (Lin and Bilmes, 2010), the authors model the sentence selection problem as maximizing a submodular function under a budget constraint. A greedy algorithm is proposed to efficiently approximate the solution to this NP-hard problem.

Compressive summarization receives increasing attention in recent years, since it offers a viable step towards abstractive summarization. The compressed summaries can be generated through a joint model of the sentence selection and compression processes, or through a pipeline approach that integrates a generic sentence compression model with a summary sentence pre-selection or post-selection step.

Many studies explore the joint sentence compression and selection setting. Martins and Smith (2009) jointly perform sentence extraction and compression by solving an ILP problem; Berg-Kirkpatrick et al. (2011) propose an approach to score the candidate summaries according to a combined linear model of extractive sentence selection and compression. They train the model using a margin-based objective whose loss captures the final summary quality. Woodsend and Lapata (2012) present a method where the summary's informativeness, succinctness, and grammaticality are learned separately from data but optimized jointly using an ILP setup; Yoshikawa et al. (2012) incorporate semantic role information in the ILP model; Chali and Hasan (2012) investigate three strategies in compressive summarization: compression before extraction, after extraction, or joint compression and extraction in one global optimization framework. These joint models offer a promise for high quality summaries, but they often have high computational cost. Qian and Liu (2013) propose a graph-cut based method that improves the speed of joint compression and summarization.

The pipeline approach, where sentence-based extractive summarization is followed or proceeded by sentence compression, is also popular. Knight and Marcu (2000) utilize the noisy channel and decision tree method to perform sentence compression; Lin (2003) shows that pure syntactic-based compression may not improve the system performance; Zajic et al. (2007) compare two sentence compression approaches for multi-document summarization, including a 'parse-and-trim' and a noisy-channel approach; Galanis and Androutsopoulos (2010) use the maximum entropy model to generate the candidate compressions by removing the branches from the source sentences; Liu and Liu (2013) couple the sentence compression and extraction approaches for summarizing the spoken documents; Wang et al. (2013) design a series of learning-based compression models built on parse trees, and integrate them in query-focused multi-document summarization. Prior studies often rely heavily on the generic sentence compression approaches (McDonald, 2006; Nomoto, 2007; Clarke and Lapata, 2008; Thadani and McKeown, 2013) for compressing the sentences in the documents, yet a generic compression system may not be the best fit for the summarization purpose.

In this paper, we adopt the pipeline-based compressive summarization framework, but propose a novel *guided compression* method that is catered to the summarization task. We expect this approach to take advantage of the efficient pipeline processing while producing satisfying results as the joint models. We train a supervised guided compression model to produce n-best compressions for each sentence, and use an ILP formulation to select the best set of summary sentences. In addition, we propose to apply a sentence pre-selection step to further accelerate the processing and enhance the performance.

3 Guided Compression Corpus

The goal of guided sentence compression is to create compressed sentences that are grammatically correct and contain the important information that we would like to preserve in the final summary. Following the compression literature (Clarke and Lapata, 2008), the compression task is defined as a word deletion problem, that is, the human annotators (and also automatic compression systems) are allowed to only remove words from the original sentence to form a compression. The key difference between our proposed guided compression with generic sentence compression is that, we provide guidance to the human compression process by specifying a set of "important words" that we wish to keep for each sentence. We expect this kind of summary oriented compression would benefit the ultimate summarization task. Take the sentence shown in Table 1 as an example. For generic sentence compression, there may be multiple 'good' human compressions for this sentence, such as those listed in the table. Without guidance, a human annotator (or automatic system) is likely to use option A or B; however, if "18 hours" appears in the summary, then we want to provide this guidance in the compression process, hence option C may be the best compression choice. This guided compression therefore avoids removing the salient words that are important to the final summary.

Original Sentence:
The gas leak was contained Monday afternoon , nearly 18 hours after it was reported , Statoil spokesman Oeivind Reinertsen said .
Compression A:
The gas leak was contained
Compression B:
The gas leak was contained ***Monday afternoon***
Compression C:
The gas leak was contained ***nearly 18 hours after it was reported***

Table 1: Example sentence and three compressions.

To generate the guided compression corpus, we use the TAC 2010 data set[1] that was used for the multi-document summarization task. There are 46 topics. Each has 10 news documents, and also four human-created abstractive reference summaries. Since annotating all the sentences in this data set is time consuming and some sentences are not very important for the summarization task, we choose a set of sentences that are highly related to the human abstracts for annotation. We compare each sentence with the four human abstracts using the ROUGE-2 metric (Lin, 2004), and the sentences

[1] http://www.nist.gov/tac/2010/

Original Sentence: He said ***Vietnam veterans*** are ***presumed to have been exposed to Agent Orange*** and veterans with any of the ***10 diseases*** is presumed to have contracted it from the exposure , without individual proof .
Guided Compression: Vietnam veterans are presumed to have been exposed to Agent Orange.
Original Sentence: The ***province has*** limited the number of trees to be chopped down in the forest area in northwest Yunnan and has ***stopped building sugar factories in the Xishuangbanna region to preserve*** the only ***tropical rain forest*** in the country located there .
Guided Compression: province has stopped building sugar factories in the Xishuangbanna region to preserve tropical rain forest.

Table 2: Example original sentences and their guided compressions. The "guiding words" are italicized and marked in red.

with the highest scores are selected.

In annotation, human annotators are provided with important 'guiding words' (highlighted in the annotation interface) that we want to preserve in the sentences. We calculate the word overlap between a sentence and each of those sentences in the human abstracts, and use a set of heuristic rules to determine the "guiding words" in a sentence: the longest consecutive word overlaps (greater than 2 words) in each sentence pair are first selected; the rest overlaps that contain 2 or more words (excluding the stopwords) are also selected. We suggest the human annotators to use their best judgment to keep the guiding words as many as possible while compressing the sentence.

We use the Amazon Mechanical Turk (AMT) for data annotation[2]. In total, we select 1,150 sentences from the TAC news documents. They are grouped into about 230 human intelligence tasks (HITs) with 5 sentences in each HIT. A sentence was compressed by 3 human annotatorsand we select the shortest candidate as the goldstandard compression for each sentence. In Table 2, we show two example sentences, their guiding words (bold), and the human compressions. The first example shows that giving up some guiding words is acceptable, since more unnecessary words will be included in order to accommodate all the guiding words; the second example shows that the guided compression can lead to more aggressive word deletions since the constituents that are not important to the summary will be deleted even though they contain salient information by themselves.

For our compression corpus, which contains 1,150 sentences and their guided compressions, the average compression rate, as measured by the percentage of dropped words, is about 50%. This compression ratio is higher compared to other generic sentence compression corpora, in which the word deletion rate ranges from 24% to 34% depending on different text genres and annotation guidelines (Clarke and Lapata, 2008; Liu and Liu, 2009). This suggests that the annotators can remove words more aggressively when they are provided with a limited set of guiding words.

4 Summarization System

Our summarization system consists of three key components: we train a supervised guided compression model using our created compression data, with a variety of features.then we use this model to generate n-best compressions for each sentence; we feed the multiple compressed sentences to the ILP framework to select the best summary sentences. In addition, we propose a sentence pre-selection step that can both speed up the summarization system and improve the performance.

4.1 Guided Sentence Compression

Sentence compression has been explored in previous studies using both supervised and unsupervised approaches, including the noisy-channel and decision tree model (Knight and Marcu, 2000; Turner and Charniak, 2005), discriminative learning (McDonald, 2006), integer linear programming (Clarke and Lapata, 2008; Thadani and McKeown, 2013), conditional random fields (CRF) (Nomoto, 2007; Liu and Liu, 2013), etc. In this paper, we employ the CRF-based compression approach due to its proved performance and its flexibility to integrate different levels of discriminative features. Under this framework, sentence compression is formulated as a sequence labeling problem, where each word is

[2]http://www.mturk.com

labeled as either "0" (retained) or "1" (removed). We develop different levels of features to capture word-specific characteristics, sentence related information, and document level importance. Most of the features are extracted based only on the sentence to be compressed. However, we introduce a few document level features. These are designed to capture the word and sentence significance within the given document collection and are thus expected to be more summary related.

Word and sentence features:

- **Word n-grams**: identity of the current word and two words before and after, as well as all the bigrams and trigrams that can be formed by the adjacent words and the current word.
- **POS n-grams**: same as the word n-grams, but use the part-of-speech tags instead.
- **Named entity tags**: binary features representing whether the current word is a person, location, or temporal expression. We use the Stanford CoreNLP tools[3] for named entity tagging.
- **Stopwords**: whether the current word is a stopword or not.
- **Conjunction features**: (1) conjunction of the current word with its relative position in the sentence; (2) conjunction of the NER tag with its relative position.
- **Syntactic features**: We obtain the syntactic parsing tree using the Berkeley Parser (Petrov and Klein, 2007), then obtain the following features: (1) the last sentence constituent tag in the path from the root to the word; (2) depth: length of the path starting from the root node to the word; (3) normalized depth: depth divided by the longest path in the parsing tree; (4) whether the word is under an SBAR node; (5) depth and normalized depth of the SBAR node if the word is under an SBAR node;
- **Dependency features:** We employ the Penn2Malt toolkit [4] to convert the parse result from the Berkeley parser to the dependency parsing tree, and use these dependency features: (1) dependency relations such as 'AMOD' (adjective modifier), 'NMOD' (noun modifier), etc. (2) whether the word has a child, left child, or right child in the dependency tree.

Document-level features:

- **Sentence salience score:** We use a simple regression model to estimate a salience score for each sentence (more details in Section 4.3), which represents the importance of the sentence in the document. This score is discretized into four binary features according to the average sentence salience.
- **Unigram document frequency:** this is the current word's document frequency based on the 10 documents associated with each topic.
- **Bigram document frequency:** document frequency for the two bigrams, the current word and its previous or next word.

Some of the above features were employed in related sentence compression studies (Nomoto, 2007; Liu and Liu, 2013). In addition to these features, we explored other related features, including the absolute position of the current word, whether the word appears in the corresponding topic title and descriptions, conjunction of the syntactic tag with the tree depth, etc.; however, these features did not lead to improved performance. We train the CRF model with the Pocket CRF toolkit[5] using the guided compression corpus collected in Section 3. During summarization, we apply the model to a given sentence to generate its n-best guided compressions and use them in the following summarization step.

4.2 Summary Sentence Selection

The sentence selection process is similar to the standard sentence-based extractive summarization, except that the input to the selection module is a list of compressed sentences in our work. Many extractive summarization approaches can be applied for this purpose. In this work, we choose the integer linear programming (ILP) method, specifically, the concept-based ILP framework introduced in (Gillick

[3] http://nlp.stanford.edu/software/corenlp.shtml

[4] http://stp.lingfil.uu.se/~nivre/research/Penn2Malt.html

[5] http://sourceforge.net/projects/pocket-crf-1/

et al., 2009), mainly because it yields best performance in the TAC evaluation tasks. This ILP approach aims to extract sentences that can cover as many important concepts as possible, while ensuring the summary length is within a given constraint. We follow the study in (Gillick et al., 2009) to use word bi-grams as concepts, and assign a weight to each bi-gram using its document frequency in the given document collection for a test topic. Two differences are between our ILP setup and that in (Gillick et al., 2009). First, since we use multiple compressions for one sentence, we need to introduce an additional constraint: for each sentence, only one of the n-best compressions may be included in the summary. Second, we optimize a joint score of the concept coverage and the sentence salience. The formal ILP formulation is shown below:

$$\max \quad \sum_i w_i c_i + \sum_j v_j \sum_k s_{jk} \tag{1}$$

$$s.t. \quad \sum_k s_{jk} \leq 1 \forall j \tag{2}$$

$$s_{jk} Occ_{i\_jk} \leq c_i \tag{3}$$

$$\sum_{jk} s_{jk} Occ_{i\_jk} \geq c_i \tag{4}$$

$$\sum_{jk} l_{jk} s_{jk} \leq L \tag{5}$$

$$c_i \in \{0,1\} \ \forall i \tag{6}$$

$$s_{jk} \in \{0,1\} \ \forall j,k \tag{7}$$

where c_i and s_{jk} are binary variables indicating the presence of a concept and a sentence respectively; s_{jk} denotes the k^{th} candidate compression of the j^{th} sentence; w_i represents the weight of the concept; v_j is the sentence salience score of the j^{th} sentence, predicted using a regression model (Section 4.3), and all of its compressed candidates share this value. (1) is the new objective function we use that combines the coverage of the concepts and the sentence salience scores. (2) represents our additional constraint, which requires that for each sentence j, only one candidate compression will be chosen. Occ_{i_jk} represents the occurrence of concept i in the sentence s_{jk}. Inequalities (3) and (4) associate the sentences and the concepts. Constraint (5) controls the summary length, as measured by the total number of words in the summary. We use an open source ILP solver[6].

4.3 Sentence Pre-selection

The above ILP method can offer an exact solution to the defined objective function. However, ILP is computationally expensive when the formulation involves large quantities of variables, i.e, when we have many sentences and a large number of candidate compressions for each sentence. We therefore propose to apply a sentence pre-selection step before the compression. This kind of selection step has been used in previous ILP-based summarization systems (Berg-Kirkpatrick et al., 2011; Gillick et al., 2009). In this work, we propose to use a simple supervised support vector regression (SVR) model (Ng et al., 2012) to predict a salience score for each sentence and select the top ranked sentences for further processing (compression and summarization).

To train the SVR model, the target value for each sentence is the ROUGE-2 score between the sentence and the four human abstracts (this same value is used for sentence selection in corpus annotation (Section 3)). We employ three commonly used features: (1) sentence position in the document; (2) sentence length as indicated by a binary feature: it takes the value of 0 if the number of words in the sentence is greater than 50 or less than 10, otherwise the feature value is 1; (3) interpolated n-gram document frequency as introduced in (Ng et al., 2012), which is a weighted linear combination of the document frequency of the unigrams and bigrams contained in the sentence:

$$f(s) = \frac{\alpha \sum_{w_u \in S} DF(w_u) + (1-\alpha) \sum_{w_b \in S} DF(w_b)}{|S|}$$

where w_u and w_b represent the unigrams and bigrams contained in the sentence S; α is a balancing factor; $|S|$ denotes the number of words in the sentence.

The SVR model was trained using the SVMlight toolkit[7]. Using this model, we can predict a salience score (V_j in Eq 1) for each sentence and only select the top n sentences and supply them to the compression and summarization steps. In practice, using a fixed n may not be a good choice since the number

[6] http://www.gnu.org/software/glpk/

[7] http://svmlight.joachims.org/

of sentences varies greatly for different topics. We therefore set n heuristically based on the total number of sentences m for each topic: n=15 if $m > 150$; n=10 if $m < 100$; n=$0.1 * m$ otherwise.

5 Experimental Results

5.1 Experimental Setup

For our experiments, we use the standard TAC data sets[8], which have been used in the NIST competitions and in other summarization studies. In particular, we used the TAC 2010 data set for creating the guided compression corpus and training the SVR pre-selection model, the TAC 2009 data set as development set for parameter tuning, and the TAC 2008 and 2011 data sets as the test set for reporting the final summarization results.

We compare our pipeline summarization system against three recent studies, which have reported some of the highest published results on this task. Berg-Kirkpatrick et al. (2011) introduce a joint model for sentence extraction and compression. The model is trained using a margin-based objective whose loss captures the end summary quality; Woodsend and Lapata (2012) learn individual summary aspects from data, e.g., informativeness, succinctness, grammaticality, stylistic writing conventions, and jointly optimize the outcome in an integer linear programming framework. Ng et al. (2012) exploit category-specific information for multi-document summarization. In addition to the three previous studies, we also report the best achieved results in the TAC competitions.

5.2 Summarization Results

In Table 3 and Table 4, we present the results of our system and the aforementioned summarization studies. We use the ROUGE evaluation metrics (Lin, 2004), with R-2 measuring the bigram overlap between the system and reference summaries and R-SU4 measuring the skip-bigram with the maximum gap length of 4. "Our System" uses the pipeline setting including the three components described in Section 4. We use the SVR-based approach to preselect a set of sentences from the document set; these sentences are further fed to the guided compression module that produces n-best compressions for each sentence; the ILP-based framework is then used to select the summary sentences from these compressions.

[8] http://www.nist.gov/tac/data/index.html

System	R-2	R-SU4	CompR
TAC'08 Best System	11.03	13.96	n/a
(Berg-Kirkpatrick et al., 2011)	11.70	14.38	n/a
(Woodsend et al., 2012)	11.37	14.47	n/a
Our System	**12.35**†	**15.27**†	43.06%
Our System w/o Pre-selection	12.02	14.98	55.69%
Our System w/ Generic Comp	10.88	13.79	30.90%

Table 3: Results on the TAC 2008 data set. "Our System" uses the SVR-based sentence pre-selection + guided compression + ILP-based summary sentence selection. "Our System w/ Generic Comp" uses the pre-selection + generic compression + ILP summary sentence selection setting. "CompR" represents the compression ratio, i.e., percentage of dropped words. † represents our system outperforms the best previous result at the 95% significance level.

System	R-2	R-SU4	CompR
TAC'11 Best System	13.44	16.51	n/a
(Ng et al., 2012)	13.93	16.83	n/a
Our System	**14.40**	**16.89**	39.90%
Our System w/o Pre-selection	13.74	16.5	53.81%
Our System w/ Generic Comp	13.08	16.23	30.10%

Table 4: Results on the TAC 2011 data set. The systems use the same settings as for the TAC 2008 data set.

We can see from the table that in general, our system achieves considerably better results compared to the state-of-the-art on both the TAC 2008 and 2011 data sets. On the TAC 2008 data set, our system outperforms the best reported result at the 95% significance level; on the TAC 2011 data set, our system also yields considerable performance gain though not exceed the 95% significance level. In the following, we show more detailed analysis to study the effect of different system parameters.

With or without sentence pre-selection. First we evaluate the impact of sentence pre-selection step. In Table 3 and Table 4, we include the results when this step is not used ("Our System w/o Pre-selection"). That is, all of the sentences in the documents (excluding those containing less than 5 words) are compressed and used in the ILP-

based summary sentence selection module. We can see that although sentence pre-selection removes some sentences from consideration in the later summarization step, it actually significantly improves system performance. In the TAC 2008 data set, each topic contains averagely 210 sentences; while the pre-selection step chooses 13 sentences among them. These numbers are 185 and 12 for the TAC 2011 data set. Table 5 shows the average running time of each topic in TAC 2011 data for the two systems, with or without the pre-selection step. Here we fix the number of compressions to 100 in both cases for fair comparison. We can see the selection step greatly accelerates the system processing. When applying the pre-selection step, fewer sentences are used in the compression and summarization, this means we are able to use more compression candidates for each sentence (considering the complexity of ILP module). Using the TAC 2009 as development set, we tuned the number of candidate compressions generated for each sentence. Without pre-selection, we used the 100-best candidates generated from the compression model; with pre-selection, we are able to increase the number to 200-best candidate compressions and still maintain reasonable computational cost. These are the numbers used in the results in Table 3 and 4. Using more compressions helps improve summarization performance. We also notice that the compression ratios are quite different when using sentence pre-selection vs. not. This suggests that in the important sentences (those are kept after pre-selection), there is more summary related information and thus the compression model keeps more words in them (lower compression ratio).

System	Compressed Sentences	Number of Compressions	Running Time (sec)
w/o Pre-selection	185	100	3.9
w/ Pre-selection	12	100	0.85

Table 5: Average running time of our system, w/ or w/o the sentence pre-selection step. Experiments conducted on the TAC 2011 data set. Running time refers only to the execution time of the ILP module for each topic.

Number of compression candidates. This parameter (denoted as n) also impacts system performance. Figure 1 shows the R-2 scores of the two systems (with and without the sentence pre-selection step) when using different number of compressions for each sentence. In general, we find that the R-2 scores do not change much when n is large enough. For example, the 'with pre-selection' system can achieve relatively stable R-2 scores on the TAC 2008 data set (ranging from 12.2 to 12.4) when m is greater than 140; similarly, the R-2 scores on the TAC 2011 data is over 14.2 when m is greater than 100. Without the pre-selection step, the scores are less stable in regard to the changing of the m value, since the large amount of sentences plus a high volume of the compression candidates may incur huge computational cost to the ILP solver. This is also the reason that in Figure 1, for the system without pre-selection, we only vary n from 1 to 100. In general, we also notice that given more compression candidates, the R-2 score is still improving, as indicated by Figure 1. The improved performance of 'with pre-selection' over 'without pre-selection' is partly because fewer sentences are used and thus we are able to increase the number of compression candidates for these sentences in the ILP sentence extraction module.

Quality of sentence compression training data. In order to illustrate the contribution of our summary-guided sentence compression component, we train a generic sentence compression model and use this in our compression and summarization pipeline. The generic compression model was trained using the Edinburgh sentence compression corpus (Clarke and Lapata, 2008), which contains 1370 sentences collected from news articles. This data set has been widely used in other summarization studies (Martins and Smith, 2009). Each sentence has 3 compressions and we choose the shortest compression as the reference. The average compression rate of this corpus is about 28%, lower than that in our summary guided compression data. Note that in generic sentence compression, we only use those word and sentence features described in Section 4.1, not the document-level features since they are not available for the Edinburgh data set. Results of our system using the generic compression model (with sentence pre-selection) are shown in the last row of Table 3 and Table 4. We can see that the system with this generic compression model performs

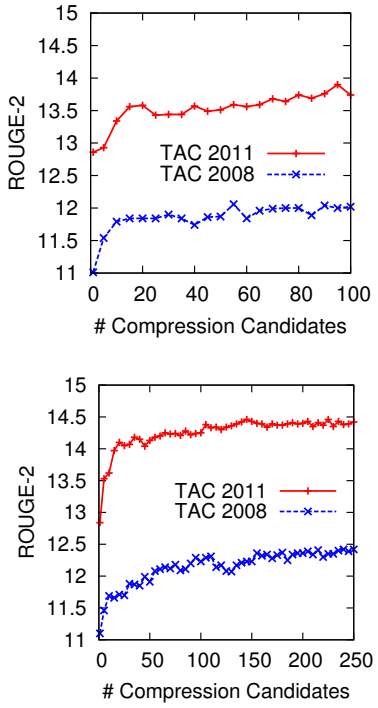

Figure 1: R-2 scores of the two systems (without and with the sentence pre-selection step) when using different number of compressions for each sentence.

worse than ours, and is also inferior to the TAC best performing system on both data sets, which signifies the importance of our proposed summary guided sentence compression approach. We can also see there is a difference in the compression ratio in the system generated compressions when using different compression corpora to train the compression models. The resulting compression ratio patterns are consistent with those in the training data, that is, using our guided compression corpus our system compressed sentences more aggressively.

Learning curve of guided compression. Since we use a supervised compression model, we further consider the relationship between the summarization performance and the number of sentence pairs used for training the guided compression model. In total, there are 1150 training sentence pairs in our corpus. We incrementally add 100 sentence pairs each time and plot the learning curve in Figure 2. In the compression step, we generate only the 1-best compression candidate in order to remove the impact caused by the downstream summary sentence selection module. As seen from Figure 2, increasing the compression training data generally improves summarization performance, although there are also fluctuations. When adding more training sentence pairs, the system performance is likely to further increase.

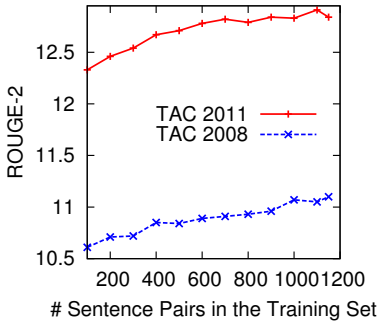

Figure 2: ROUGE-2 scores when using different number of sentences to train the guided compression model.

6 Conclusion and Future Work

In this paper, we propose a pipeline summarization approach that combines a novel *guided compression* model with ILP-based summary sentence selection. We create a guided compression corpus, where the human annotators were explicitly informed about the important summary words during the compression annotation. We then train a supervised compression model to capture the guided compression process using a set of word-, sentence-, and document-level features. We conduct experiments on the TAC 2008 and 2011 summarization data sets and show that by incorporating the guided sentence compression model, our summarization system can yield significant performance gain as compared to the state-of-the-art. In future, we would like to further explore the reinforcement relationship between keywords and summaries (Wan et al., 2007), improve the readability of the sentences generated from the guided compression system, and report results using multiple evaluation metrics (Nenkova et al., 2007; Louis and Nenkova, 2012) as well as performing human evaluations.

Acknowledgments

Part of this work was done during the first author's internship in Bosch Research and Technology Center. The work is also partially supported by NSF award IIS-0845484 and DARPA Contract No. FA8750-13-2-0041. Any opinions, findings, and conclusions or recommendations expressed are those of the author and do not necessarily reflect the views of the funding agencies.

References

Ahmet Aker, Trevor Cohn, and Robert Gaizauskas. 2010. Multi-document summarization using a* search and discriminative training. In *Proceedings of EMNLP*.

Miguel B. Almeida and Andre F. T. Martins. 2013. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of ACL*.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of ACL*.

Yllias Chali and Sadid A. Hasan. 2012. On the effectiveness of using sentence compression models for query-focused multi-document summarization. In *Proceedings of COLING*.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression an integer linear programming approach. *Journal of Artificial Intelligence Research*.

Hal Daumé. 2006. Practical structured learning techniques for natural language processing. *Ph.D. thesis, University of Southern California*.

Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Michel Galley. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of EMNLP*.

Dan Gillick, Benoit Favre, Dilek Hakkani-Tur, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The icsi/utd summarization system at tac 2009. In *Proceedings of TAC*.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of SIGIR*.

Chen Li, Xian Qian, and Yang Liu. 2013. Using supervised bigram-based ilp for extractive summarization. In *Proceedings of ACL*.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Proceedings of NAACL*.

Chin-Yew Lin. 2003. Improving summarization performance by sentence compression - A pilot study. In *Proceeding of the Sixth International Workshop on Information Retrieval with Asian Language*.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Proceedings of ACL*.

Fei Liu and Yang Liu. 2009. From extractive to abstractive meeting summaries: Can it be done by sentence compression? In *Proceedings of ACL-IJCNLP*.

Fei Liu and Yang Liu. 2013. Towards abstractive speech summarization: Exploring unsupervised and supervised approaches for spoken utterance compression. *IEEE Transactions on Audio, Speech, and Language Processing*.

Annie Louis and Ani Nenkova. 2012. Automatically assessing machine summary content with a gold-standard. *Computational Linguistics*.

Andre F. T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the ACL Workshop on Integer Linear Programming for Natural Language Processing*.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of EACL*.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*.

Jun-Ping Ng, Praveen Bysani, Ziheng Lin, Min-Yen Kan, and Chew-Lim Tan. 2012. Exploiting category-specific information for multi-document summarization. In *Proceedings of COLING*.

Tadashi Nomoto. 2007. Discriminative sentence compression with conditional random fields. *Information Processing and Management*.

Miles Osborne. 2002. Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT-NAACL*.

Xian Qian and Yang Liu. 2013. Fast joint compression and summarization via graph cuts. In *Proceedings of EMNLP*.

Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. In *Computational Linguistics*.

Kapil Thadani and Kathleen McKeown. 2013. Sentence compression with joint structural inference. In *Proceedings of CoNLL*.

Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of ACL*.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of ACL*.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of ACL*.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of EMNLP-CoNLL*.

Katsumasa Yoshikawa, Tsutomu Hirao, Ryu Iida, and Manabu Okumura. 2012. Sentence compression with semantic role constraints. In *Proceedings of ACL*.

David Zajic, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. In *Information Processing and Management*.