Improving Multi-documents Summarization by Sentence Compression based on Expanded Constituent Parse Trees

Chen Li[1], Yang Liu[1], Fei Liu[2], Lin Zhao[3], Fuliang Weng[3]
[1] Computer Science Department, The University of Texas at Dallas
Richardson, TX 75080, USA
[2] School of Computer Science, Carnegie Mellon University
Pittsburgh, PA 15213, USA
[3] Research and Technology Center, Robert Bosch LLC
Palo Alto, California 94304, USA
{chenli,yangl@hlt.utdallas.edu}
{feiliu@cs.cmu.edu}
{lin.zhao,fuliang.weng@us.bosch.com}

Abstract

In this paper, we focus on the problem of using sentence compression techniques to improve multi-document summarization. We propose an innovative sentence compression method by considering every node in the constituent parse tree and deciding its status – remove or retain. Integer liner programming with discriminative training is used to solve the problem. Under this model, we incorporate various constraints to improve the linguistic quality of the compressed sentences. Then we utilize a pipeline summarization framework where sentences are first compressed by our proposed compression model to obtain top-n candidates and then a sentence selection module is used to generate the final summary. Compared with state-of-the-art algorithms, our model has similar ROUGE-2 scores but better linguistic quality on TAC data.

1 Introduction

Automatic summarization can be broadly divided into two categories: extractive and abstractive summarization. Extractive summarization focuses on selecting 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.

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 (Lin, 2003; Zajic et al., 2007; Vanderwende et al., 2007; Wang et al., 2013). Another line of work uses joint compression and summarization. Such methods have been shown to achieve promising performance (Daumé, 2006; Chali and Hasan, 2012; Almeida and Martins, 2013; Qian and Liu, 2013), but they are typically computationally expensive.

In this study, we propose an innovative sentence compression model based on expanded constituent parse trees. Our model uses integer linear programming (ILP) to search the entire space of compression, and is discriminatively trained. It is built based on the discriminative sentence compression model from (McDonald, 2006) and (Clarke and Lapata, 2008), but our method uses an expanded constituent parse tree rather than only the leaf nodes in previous work. Therefore we can extract rich features for every node in the constituent parser tree. This is an advantage of tree-based compression technique (Knight and Marcu, 2000; Galley and McKeown, 2007; Wang et al., 2013). Similar to (Li et al., 2013a), we use a pipeline summarization framework where multiple compression candidates are generated for each pre-selected important sentence, and then an ILP-

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 691–701,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

based summarization model is used to select the final compressed sentences. We evaluate our proposed method on the TAC 2008 and 2011 data sets using the standard ROUGE metric (Lin, 2004) and human evaluation of the linguistic quality. Our results show that using our proposed sentence compression model in the summarization system can yield significant performance gain in linguistic quality, without losing much performance on the ROUGE metric.

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. Supervised approaches include the Bayesian classifier (Kupiec et al., 1995), maximum entropy (Osborne, 2002), skip-chain 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 methods include using integer linear programming (ILP) and submodular functions to solve the optimization problem (Gillick et al., 2009; Li et al., 2013b; Lin and Bilmes, 2010).

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 sentence compression model with a summary sentence pre-selection or post-selection step.

Many studies have explored the joint sentence compression and selection setting. Martins and Smith (2009) jointly performed sentence extraction and compression by solving an ILP problem. Berg-Kirkpatrick et al. (2011) proposed an approach to score the candidate summaries according to a combined linear model of extractive sentence selection and compression. They trained the model using a margin-based objective whose loss captures the final summary quality. Woodsend and Lapata (2012) presented another 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) incorporated semantic role information in the ILP model.

Our work is closely related with the pipeline approach, where sentence-based extractive summarization is followed or proceeded by sentence compression. There have been many studies on sentence compression, independent of the summarization task. McDonald (2006) firstly introduced a discriminative sentence compression model to directly optimize the quality of the compressed sentences produced. Clarke and Lapata (2008) improved the above discriminative model by using ILP in decoding, making it convenient to add constraints to preserve grammatical structure. Nomoto (2007) treated the compression task as a sequence labeling problem and used CRF for it. Thadani and McKeown (2013) presented an approach for discriminative sentence compression that jointly produces sequential and syntactic representations for output text. Filippova and Altun (2013) presented a method to automatically build a sentence compression corpus with hundreds of thousands of instances on which deletion-based compression algorithms can be trained.

In addition to the work on sentence compression as a stand-alone task, prior studies have also investigated compression for the summarization task. Knight and Marcu (2000) utilized the noisy channel and decision tree method to perform sentence compression in the summarization task. Lin (2003) showed that pure syntactic-based compression may not significantly improve the summarization performance. Zajic et al. (2007) compared two sentence compression approaches for multi-document summarization, including a 'parse-and-trim' and a noisy-channel approach. Galanis and Androutsopoulos (2010) used the maximum entropy model to generate the candidate compressions by removing branches from the source sentences. Woodsend and Lapata (2010) presented a joint content selection and compression model for single-document summarization. They operated over a phrase-based representation of the source document which they obtained by merging information from PCFG parse trees and dependency graphs. Liu and Liu (2013) adopted the CRF-based sentence compression approach for summarizing spoken documents. Unlike the word-based operation, some of these models e.g (Knight and Marcu, 2000; Siddharthan et al., 2004; Turner and Charniak, 2005; Galanis and Androutsopoulos, 2010; Wang et al., 2013), are tree-based approaches that operate on the parse trees and thus the compression decision can be made for a constituent, instead of a single word.

3 Sentence Compression Method

Sentence compression is a task of producing a summary for a single sentence. The compressed sentence should be shorter, contain important content from the original sentence, and be grammatical. In some sense, sentence compression can be described as a 'scaled down version of the text summarization problem' (Knight and Marcu, 2002). Here similar to much previous work on sentence compression, we just focus on how to remove/select words in the original sentence without using operation like rewriting sentence.

3.1 Discriminative Compression Model by ILP

McDonald (2006) presented a discriminative compression model, and Clarke and Lapata (2008) improved it by using ILP for decoding. Since our proposed method is based upon this model, in the following we briefly describe it first. Details can be found in (Clarke and Lapata, 2008). In this model, the following score function is used to evaluate each compression candidate:

$$s(\mathbf{x},\mathbf{y}) = \sum_{j=2}^{|\mathbf{y}|} s(\mathbf{x}, L(y_{j-1}), L(y_j)) \qquad (1)$$

where $\mathbf{x} = x_1x_2, ..., x_n$ represents an original sentence and $\mathbf{y} = y_1y_2, ..., y_m$ denotes a compressed sentence. Because the sentence compression problem is defined as a word deletion task, y_j must occur in $\mathbf{x}$. Function $L(y_i) \in [1...n]$ maps word y_i in the compression to the word index in the original sentence $\mathbf{x}$. Note that $L(y_i) < L(y_{i+1})$ is required, that is, each word in x can only occur at most once in compression y. In this model, a first order Markov assumption is used for the score function. Decoding this model is to find the combination of bigrams that maximizes the score function in Eq (1). Clarke and Lapata (2008) introduced the following variables and used ILP to solve it:

$$\delta_i = \begin{cases} 1 & if\ x_i\ is\ in\ the\ compression \\ 0 & otherwise \end{cases} \qquad \forall i \in [1..n]$$

$$\alpha_i = \begin{cases} 1 & if\ x_i\ starts\ the\ compression \\ 0 & otherwise \end{cases} \qquad \forall i \in [1..n]$$

$$\beta_i = \begin{cases} 1 & if\ x_i\ ends\ the\ compression \\ 0 & otherwise \end{cases} \qquad \forall i \in [1..n]$$

$$\gamma_{ij} = \begin{cases} 1 & if\ x_i, x_j\ are\ in\ the\ compression \\ 0 & otherwise \end{cases} \qquad \forall i \in [1..n-1] \forall j \in [i+1..n]$$

Using these variables, the objective function can be defined as:

$$\begin{aligned} \max z \quad &= \sum_{i=1}^{n} \alpha_i \cdot s(\mathbf{x}, 0, i) \\ &+ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \gamma_{ij} \cdot s(\mathbf{x}, i, j) \\ &+ \sum_{i=1}^{n} \beta_i \cdot s(\mathbf{x}, i, n+1) \qquad (2) \end{aligned}$$

The following four basic constraints are used to make the compressed result reasonable:

$$\sum_{i=1}^{n} \alpha_i = 1 \qquad (3)$$

$$\delta_j - \alpha_j - \sum_{i=1}^{j} \gamma_{ij} = 0 \quad \forall j \in [1..n] \quad (4)$$

$$\delta_i - \sum_{j=i+1}^{n} \gamma_{ij} - \beta_i = 0 \quad \forall i \in [1..n] \quad (5)$$

$$\sum_{i=1}^{n} \beta_i = 1 \qquad (6)$$

Formula (3) and (6) denote that exactly one word can begin or end a sentence. Formula (4) means if a word is in the compressed sentence, it must either start the compression or follow another word; formula (5) represents if a word is in the compressed sentence, it must either end the sentence or be followed by another word.

Furthermore, discriminative models are used for the score function:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{j=2}^{|\mathbf{y}|} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, L(y_{j-1}), L(y_j)) \qquad (7)$$

High dimensional features are used and their corresponding weights are trained discriminatively.

Above is the basic supervised ILP formulation for sentence compression. Linguistically and semantically motivated constraints can be added in the ILP model to ensure the correct grammar structure in the compressed sentence. For example, Clarke and Lapata (2008) forced the introducing term of prepositional phrases and subordinate clauses to be included in the compression if any word from within that syntactic constituent is also included, and vice versa.

3.2 Compression Model based on Expanded Constituent Parse Tree

In the above ILP model, variables are defined for each word in the sentence, and the task is to predict each word's status. In this paper, we propose to adopt the above ILP framework, but operate directly on the nodes in the constituent parse tree, rather than just the words (leaf nodes in the tree). This way we can remove or retain a chunk of the sentence rather than isolated words, which we expect can improve the readability and grammar correctness of the compressed sentences.

The top part of Fig1 is a standard constituent parse tree. For some levels of the tree, the nodes at that same level can not represent a sentence. We extend the parse tree by duplicating non-POS constituents so that leaf nodes (words and their corresponding POS tags) are aligned at the bottom level as shown in bottom of as Fig1. In the example tree, the solid lines represent relationship of nodes from the original parse tree, the long dot lines denote the extension of the duplication nodes from the upper level to the lower level, and the nodes at the same level are connected (arrowed lines) to represent that is a sequence. Based on this expanded constituent parse tree, we can consider every level as a 'sentence' and the tokens are POS tags and parse tree labels. We apply the above compression model in Section 3.1 on every level to decide every node's status in the final compressed sentence. In order to make the compressed parsed tree reasonable, we model the relationship of nodes between adjacent levels as following: if the parent node is labeled as removed, all of its children will be removed; one node will retain if at least one of its children is kept.

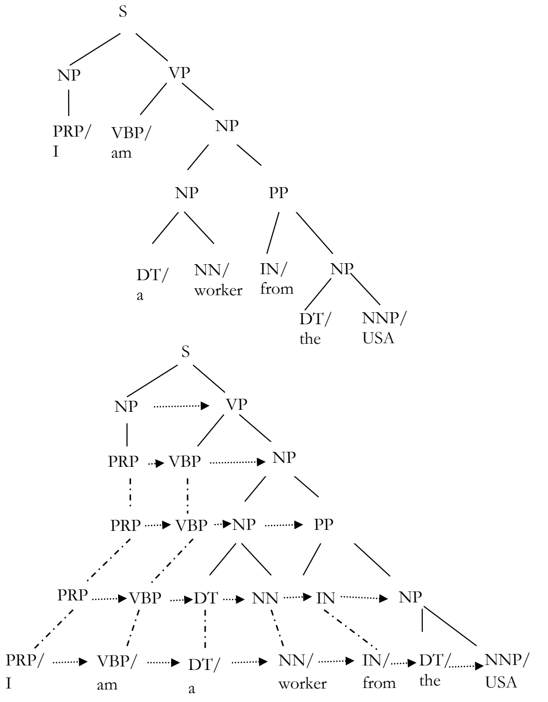

Figure 1: A regular constituent parse tree and its Expanded constituent tree.

Therefore, the objective function in the new ILP formulation is:

$$\begin{aligned} \max z \quad &= \sum_{l=1}^{height} \Big(\sum_{i=1}^{n_l} \alpha_i^l \cdot s(\mathbf{x}, 0, l_i) \\ &+ \sum_{i=1}^{n_l - 1} \sum_{j=i+1}^{n_l} \gamma_{ij}^l \cdot s(\mathbf{x}, l_i, l_j) \\ &+ \sum_{i=1}^{n_l} \beta_i^l \cdot s(\mathbf{x}, l_i, n_l + 1) \Big) \qquad (8) \end{aligned}$$

where height is the depth for a parse tree (starting from level 1 for the tree), and n_l means the length of level l (for example, $n_5 = 6$ in the example in Fig1). Then every level will have a set of parameters $\delta_i^l, \alpha_i^l, \beta_i^l$, and γ_{ij}^l, and the corresponding constraints as shown in Formula (3) to (6). The relationship between nodes from adjacent levels can be expressed as:

$$\delta_i^l \geq \delta_j^{(l+1)} \qquad (9)$$

$$\delta_i^l \leq \sum \delta_j^{(l+1)} \qquad (10)$$

in which node j at level $(l+1)$ is the child of node

i at level l. In addition, $1 \leq l \leq height - 1$, $1 \leq i \leq n_l$ and $1 \leq j \leq n_{l+1}$.

3.3 Linguistically Motivated Constraints

In our proposed model, we can jointly decide the status of every node in the constituent parse tree at the same time. One advantage is that we can add constraints based on internal nodes or relationship in the parse tree, rather than only using the relationship based on words. In addition to the constraints proposed in (Clarke and Lapata, 2008), we introduce more linguistically motivated constraints to keep the compressed sentence more grammatically correct. The following describes the constraints we used based on the constituent parse tree.

- If a node's label is 'SBAR', its parent's label is 'NP' and its first child's label is 'WHNP' or 'WHPP' or 'IN', then if we can find a noun in the left siblings of 'SBAR', this subordinate clause could be an attributive clause or appositive clause. Therefore the found noun node should be included in the compression if the 'SBAR' is also included, because the node 'SBAR' decorates the noun. For example, the top part of Fig 2 is part of expanded constituent parse tree of sentence 'Those who knew David were all dead.' The nodes in ellipse should share the same status.

- If a node's label is 'SBAR', its parent's label is 'VP' and its first child's label is 'WHNP', then if we can find a verb in the left siblings of 'SBAR', this subordinate clause could be an objective clause. Therefore, the found verb node should be included in the compression if the 'SBAR' node is also included, because the node 'SBAR' is the object of that verb. An example is shown in the bottom part of Fig 2. The nodes in ellipse should share the same status.

- If a node's label is 'SBAR', its parent's label is 'VP' and its first child's label is 'WHADVP', then if the first leaf for this node is a wh-word (e.g., 'where, when, why') or 'how', this clause may be an objective clause (when the word is 'why, how, where') or attributive clause (when the word is 'where') or adverbial clause (when the word is 'when'). Therefore, similar to above, if a verb or noun is found in the left siblings of 'SBAR', the found verb or noun node should be included in the compression if the 'SBAR' node is also included.

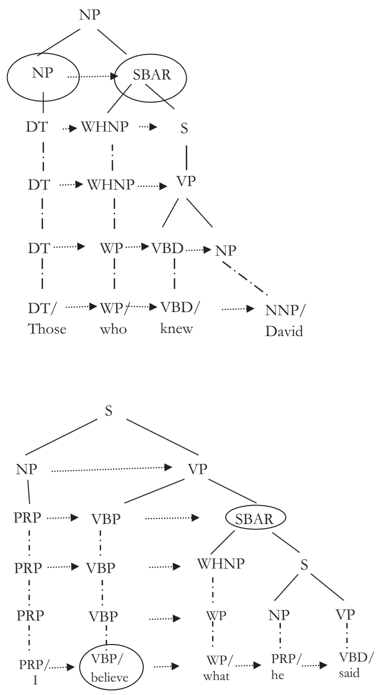

Figure 2: Expanded constituent parse tree for examples.

- If a node's label is 'SBAR' and its parent's label is 'ADJP', then if we can find a 'JJ', 'JJR', or 'JJS' in the left siblings of 'SBAR', the 'SBAR' node should be included in the compression if the found 'JJ', 'JJR' or 'JJS' node is also included because the node 'SBAR' is decorated by the adjective.

- The node with a label of 'PRN' can be removed without other constraints.

We also include some other constraints based on the Stanford dependency parse tree. Table 1 lists the dependency relations we considered.

- For type I relations, the parent and child node with those relationships should have the same value in the compressed result (both are kept or removed).

- For type II relations, if the child node in those relations is retained in the compressed sentence, the parent node should be also retained.

	Dependency Relation	Example
I	prt: phrase verb particle	They shut down the station. prt(shut,down)
	prep: prepositional modifier	He lives in a small village. prep(lives,in)
	pobj: object of a preposition	I sat on the chair. pobj(on,chair)
	nsubj: nominal subject	The boy is cute. nsubj(cute,boy)
	cop: copula	Bill is big. cop(big,is)
II	partmod: participial modifier	Truffles picked during the spring are tasty. partmod(truffles,picked)
	nn: noun compound modifier	Oil price futures. nn(futures,oil)
	acomp: adjectival complement	She looks very beautiful. acomp(looks,beautiful)
III	pcomp: prepositional complement	He felt sad after learning that tragedy. pcomp(after,learning)
	ccomp: clausal complement	I am certain that he did it. ccomp(certain,did)
	tmod: temporal modifier	Last night I swam in the pool. tmod(swam,night)

Table 1: Some dependency relations used for extra constraints. All the examples are from (Marneffe and Manning, 2002)

- For type III relations, if the parent node in these relations is retained, the child node should be kept as well.

3.4 Features

So far we have defined the decoding process and related constraints used in decoding. These all rely on the score function $s(\mathbf{x}, \mathbf{y}) = \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, L(y_{j-1}), L(y_j))$ for every level in the constituent parse tree. We included all the features introduced in (Clarke and Lapata, 2008) (those features are designed for leaves). Table 2 lists the additional features we used in our system.

General Features for Every Node
1. individual node label and concatenation of a pair of nodes 2. distance of two nodes at the same level 3. is the node at beginning or end at that level? 4. do the two nodes have the same parent? 5. if two nodes do not have the same parent, then is the left node the rightmost child of its parent? is the right node the leftmost child of its parent? 6. combination of parent label if the node pair are not under the same parent 7. number of node's children: 1/0/>1 8. depth of nodes in the parse tree
Extra Features for Leaf nodes
1. word itself and concatenation of two words 2. POS and concatenation of two words' POS 3. whether the word is a stopword 4. node's named entity tag 5. dependency relationship between two leaves

Table 2: Features used in our system besides those used in (Clarke and Lapata, 2008).

3.5 Learning

To learn the feature weights during training, we perform ILP decoding on every sentence in the training set, to find the best hypothesis for each node in the expanded constituent parse tree. If the hypothesis is incorrect, we update the feature weights using the structured perceptron learning strategy (Collins, 2002). The reference label for every node in the expanded constituent parse tree is obtained automatically from the bottom to the top of the tree. Since every leaf node (word) is human annotated (removed or retain), we annotate the internal nodes as removed if all of its children are removed. Otherwise, the node is annotated as retained.

During perceptron training, a fixed learning rate is used and parameters are averaged to prevent overfitting. In our experiment, we observe stable convergence using the held-out development corpus, with best performance usually obtained around 10-20 epochs.

4 Summarization System

Similar to (Li et al., 2013a), our summarization system is , which consists of three key components: an initial sentence pre-selection module to select some important sentence candidates; the above compression model to generate n-best compressions for each sentence; and then an ILP summarization method to select the best summary sentences from the multiple compressed sentences.

The sentence pre-selection model is a simple supervised support vector regression (SVR) model that predicts a salience score for each sentence and selects the top ranked sentences for further processing (compression and summarization). The target value for each sentence during training is the ROUGE-2 score between the sentence and the human written abstracts. We use three common features: (1) sentence position in the document; (2) sentence length; and (3) interpolated n-gram document frequency as introduced in (Ng et al., 2012).

The final sentence selection process follows the

ILP method introduced in (Gillick et al., 2009). Word bi-grams are used as concepts, and their document frequency is used as weights. Since we use multiple compressions for one sentence, an additional constraint is used: for each sentence, only one of its n-best compressions may be included in the summary.

For the compression module, using the ILP method described above only finds the best compression result for a given sentence. To generate n-best compression candidates, we use an iterative approach – we add one more constraints to prevent it from generating the same answer every time after getting one solution.

5 Experimental Results

5.1 Experimental Setup

Summarization Data For summarization experiments, we use the standard TAC data sets[1], which have been used in the NIST competitions. In particular, we used the TAC 2010 data set as training data for the SVR sentence pre-selection model, TAC 2009 data set as development set for parameter tuning, and the TAC 2008 and 2011 data as the test set for reporting the final summarization results. The training data for the sentence compression module in the summarization system is summary guided compression corpus annotated by (Li et al., 2013a) using TAC2010 data. In the compression module, for each word we also used its document level feature.[2]

Compression Data We also evaluate our compression model using the data set from (Clarke and Lapata, 2008). It includes 82 newswire articles with manually produced compression for each sentence. We use the same partitions as (Martins and Smith, 2009), i.e., 1,188 sentences for training and 441 for testing.

Data Processing We use Stanford CoreNLP toolkit[3] to tokenize the sentences, extract name entity tags, and generate the dependency parse tree. Berkeley Parser (Petrov et al., 2006) is adopted to obtain the constituent parse tree for every sentence and POS tag for every token. We use Pocket CRF[4] to implement the CRF sentence compression model. SVMlight[5] is used for the summary sentence pre-selection model. Gurobi ILP solver[6] does all ILP decoding.

5.2 Summarization Results

We compare our summarization system against four recent studies, which have reported some of the highest published results on this task. Berg-Kirkpatrick et al. (2011) introduced a joint model for sentence extraction and compression. Woodsend and Lapata (2012) learned individual summary aspects from data, e.g., informativeness, succinctness, grammaticalness, stylistic writing conventions, and jointly optimized the outcome in an ILP framework. Ng et al. (2012) exploited category-specific information for multi-document summarization. Almeida and Martins (2013) proposed compressive summarization method by dual decomposition and multi-task learning. Our summarization framework is the same as (Li et al., 2013a), except they used a CRF-based compression model. In addition to the four previous studies, we also report the best achieved results in the TAC competitions.

Table 3 shows the summarization results of our method and others. The top part contains the results for TAC 2008 data and bottom part is for TAC 2011 data. 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. In addition, we evaluate the linguistic quality (LQ) of the summaries for our system and (Li et al., 2013a).[7] The linguistic quality consists of two parts. One evaluates the grammar quality within a sentence. For this, annotators marked if a compressed sentence is grammatically correct. Typical grammar errors include lack of verb or subordinate clause. The other evaluates the coherence between sentences, including the order of sentences and irrelevant sentences. We invited 3 English native speakers to do this evaluation. They gave every compressed sentence a grammar score and a coherence score for

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

[2] Document level features for a word include information such as the word's document frequency in a topic. These features cannot be extracted from a single sentence, as in the standard sentence compression task, and are related to the document summarization task.

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

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

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

[6] http://www.gurobi.com

[7] We chose to evaluate the linguistic quality for this system because of two reasons: one is that we have an implementation of that method; the other more important one is that it has the highest reported ROUGE results among the compared methods.

System	R-2	R-SU4	Gram	Cohere
TAC'08 Best System	11.03	13.96	n/a	n/a
(Berg-Kirk et al., 2011)	11.70	14.38	n/a	n/a
(Woodsend et al., 2012)	11.37	14.47	n/a	n/a
(Almeida et al.,2013)	12.30	15.18	n/a	n/a
(Li et al., 2013a)	**12.35**	15.27	3.81	3.41
Our System	12.23	**15.47**	**4.29**	**4.11**
TAC'11 Best System	13.44	16.51	n/a	n/a
(Ng et al., 2012)	13.93	16.83	n/a	n/a
(Li et al., 2013a)	**14.40**	**16.89**	3.67	3.32
Our System	14.04	16.67	**4.18**	**4.07**

Table 3: Summarization results on the TAC 2008 and 2011 data sets.

each topic. The score is scaled and ranges from 1 (bad) to 5 (good). Therefore, in table 3, the grammar score is the average score for each sentence and coherence score is the average for each topic. We measure annotators' agreement in the following way: we consider the scores from each annotator as a distribution and we find that these three distributions are not statistically significantly different each other ($p > 0.05$ based on paired t-test).

We can see from the table that in general, our system achieves better ROUGE results than most previous work except (Li et al., 2013a) on both TAC 2008 and TAC 2011 data. However, our system's linguistic quality is better than (Li et al., 2013a). The CRF-based compression model used in (Li et al., 2013a) can not well model the grammar. Particularly, our results (ROUGE-2) are statistically significantly ($p < 0.05$) higher than TAC08 Best system, but are not statistically significant compared with (Li et al., 2013a) ($p > 0.05$). The pattern is similar in TAC 2011 data. Our result (R-2) is statistically significantly ($p < 0.05$) better than TAC11 Best system, but not statistically ($p > 0.05$) significantly different from (Li et al., 2013a). However, for the grammar and coherence score, our results are statistically significantly ($p < 0.05$) than (Li et al., 2013a). All the above statistics are based on paired t-test.

5.3 Compression Results

The results above show that our summarization system is competitive. In this section we focus on the evaluation of our proposed compression method. We compare our compression system against four other models. HedgeTrimmer in Dorr et al. (2003) applied a variety of linguistically-motivated heuristics to guide the sentences compression; McDonald (2006) used the output of two parsers as features in a discriminative model that decomposes over pairs of consecutive words; Martins and Smith (2009) built the compression model in the dependency parse and utilized the relationship between the head and modifier to preserve the grammar relationship; Wang et al. (2013) developed a novel beam search decoder using the tree-based compression model on the constituent parse tree, which could find the most probable compression efficiently.

System	C Rate (%)	Uni-F1	Rel-F1
HedgeTrimmer	57.64	0.64	0.50
McDonald (2006)	70.95	0.77	0.55
Martins (2009)	71.35	0.77	0.56
Wang (2013)	68.06	0.79	0.59
Our System	71.19	0.77	0.58

Table 4: Sentence compression results. The human compression rate of the test set is 69%.

Table 4 shows the compression results of various systems, along with the compression ratio (C Rate) of the system output. We adopt the compression metrics as used in (Martins and Smith, 2009) that measures the macro F-measure for the retained unigrams (Uni-F1), and the one used in (Clarke and Lapata, 2008) that calculates the F1 score of the grammatical relations labeled by (Briscoe and Carroll, 2002) (Rel-F1). We can see that our proposed compression method performs well, similar to the state-of-the-art systems.

To evaluate the power of using the expanded parse tree in our model, we conducted another experiment where we only consider the bottom level of the constituent parse tree. In some sense, this could be considered as the system in (Clarke and Lapata, 2008). Furthermore, we use two different setups: one uses the lexical features (about the words) and the other does not. Table 5 shows the results using the data in (Clarke and Lapata, 2008). For a comparison, we also include the results using the CRF-based compression model (the one used in (Nomoto, 2007; Li et al., 2013a)). We report results using both the automatically calculated compression metrics and the linguistic quality score. Three English native speaker annotators were asked to judge two aspects of the compressed sentence compared with the gold result: one is the content that looks at whether the important words are kept and the other is the grammar score which evaluates the sentence's readability. Each of these two scores ranges from 1(bad) to 5(good).

Table 5 shows that when using lexical features, our system has statistically significantly ($p < 0.05$) higher Grammar value and content importance value than the CRF and the leaves only system. When no lexical features are used, default system can achieve statistically significantly ($p < 0.01$) higher results than the CRF and the leaves only system.

We can see that using the expanded parse tree performs better than using the leaves only, especially when lexical features are not used. In addition, we observe that our proposed compression method is more generalizable than the CRF-based model. When our system does not use lexical features in the leaves, it achieves better performance than the CRF-based model. This is important since such a model is more robust and may be used in multiple domains, whereas a model relying on lexical information may suffer more from domain mismatch. From the table we can see our proposed tree based compression method consistently has better linguistic quality. On the other hand, the CRF compression model is the most computationally efficient one among these three compression methods. It is about 200 times faster than our model using the expanded parse tree. Table 6 shows some examples using different methods.

System	C Rate(%)	Uni-F1	Rel-F1	Gram	Imp
Using lexical features					
CRF	79.98	0.80	0.51	3.9	4.0
ILP(I)	80.54	0.79	0.57	4.0	4.2
ILP(II)	79.90	0.80	0.57	4.2	4.4
No lexical features					
CRF	77.75	0.78	0.51	3.35	3.5
ILP(I)	77.77	0.78	0.56	3.7	3.9
ILP(II)	77.78	0.80	0.58	4.1	4.2

Table 5: Sentence compression results: effect of lexical features and expanded parse tree. ILP(I) represents the system using only bottom nodes in constituent parse tree. ILP(II) is our system. Imp means the content importance value.

Using lexical features
Source: Apart from drugs, detectives believe money is laundered from a variety of black market deals involving arms and high technology. **Human compress**: detectives believe money is laundered from a variety of black market deals. **CRF result** : Apart from drugs detectives believe money is laundered from a black market deals involving arms and technology. **ILP(I) Result**: detectives believe money is laundered from a variety of black deals involving arms. **ILP(II) Result**: detectives believe money is laundered from black market deals.

No lexical features
Source: Mrs Allan's son disappeared in May 1989, after a party during his back packing trip across North America. **Human compress**: Mrs Allan's son disappeared in 1989, after a party during his trip across North America. **CRF result** : Mrs Allan's son disappeared May 1989, after during his packing trip across North America. **ILP(I) Result**: Mrs Allan's son disappeared in May, 1989, after a party during his packing trip across North America . **ILP(II) Result**: Mrs Allan's son disappeared in May 1989, after a party during his trip.

Table 6: Examples of original sentences and their compressed sentences from different systems.

6 Conclusion

In this paper, we propose a discriminative ILP sentence compression model based on the expanded constituent parse tree, which aims to improve the linguistic quality of the compressed sentences in the summarization task. Linguistically motivated constraints are incorporated to improve the sentence quality. We conduct experiments on the TAC 2008 and 2011 summarization data sets and show that by incorporating this sentence compression model, our summarization system can yield significant performance gain in linguistic quality without losing much ROUGE results. The analysis of the compression module also demonstrates its competitiveness, in particular the better linguistic quality and less reliance on lexical cues.

Acknowledgments

We thank the anonymous reviewers for their detailed and insightful comments on earlier drafts of this paper. 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 authors 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*.

Ted Briscoe and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of LREC*.

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*.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*.

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

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of NAACL*.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of EMNLP*.

Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Proceedings of NAACL*.

Michel Galley and Kathleen McKeown. 2007. Lexicalized markov grammars for sentence compression. In *Processings of NAACL*.

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. In *Proceedings of AAAI*.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

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

Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. 2013a. Document summarization via guided sentence compression. In *Proceedings of the EMNLP*.

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

Marie-Catherine de Marneffe and Christopher D Manning. 2002. Stanford typed dependencies manual.

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*.

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, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL*.

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

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of Coling*.

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*.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*.

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. 2010. Automatic generation of story highlights. 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*.