Fast Joint Compression and Summarization via Graph Cuts

Xian Qian and **Yang Liu**
The University of Texas at Dallas
800 W. Campbell Rd., Richardson, TX, USA
{qx,yangl}@hlt.utdallas.edu

Abstract

Extractive summarization typically uses sentences as summarization units. In contrast, joint compression and summarization can use smaller units such as words and phrases, resulting in summaries containing more information. The goal of compressive summarization is to find a subset of words that maximize the total score of concepts and cutting dependency arcs under the grammar constraints and summary length constraint. We propose an efficient decoding algorithm for fast compressive summarization using graph cuts. Our approach first relaxes the length constraint using Lagrangian relaxation. Then we propose to bound the relaxed objective function by the supermodular binary quadratic programming problem, which can be solved efficiently using graph max-flow/min-cut. Since finding the tightest lower bound suffers from local optimality, we use convex relaxation for initialization. Experimental results on TAC2008 dataset demonstrate our method achieves competitive ROUGE score and has good readability, while is much faster than the integer linear programming (ILP) method.

1 Introduction

Automatic multi-document summarization helps readers get the most important information from large amounts of texts. Summarization techniques can be roughly divided into two categories: extractive and abstractive. Extractive summarization casts the summarization task as a sentence selection problem: identifying important summary sentences from one or multiple documents. Many methods have been developed in the past decades, including supervised approaches that use classifiers to predict summary sentences, graph based approaches to rank the sentences, and recent global optimization methods such as integer linear programming (Gillick et al., 2008) (ILP) and submodular maximization methods (Lin and Bilmes, 2011). Though extractive summarization is popular because of its simplicity and high readability, it has limitations in that it selects each sentence as a whole, and thus may miss informative partial sentences.

To improve the informativeness, joint compression and summarization was proposed (Berg-Kirkpatrick et al., 2011), which uses words as summarization units, unlike extractive summarization where each sentence is a basic undecomposable unit. To achieve better readability, manually defined grammar constraints or automatically learned models based on syntax trees are added during the summarization process. Up to now, the state of the art compressive systems are based on integer linear programming (ILP). Because ILP suffers from exponential complexity, word-based compression summarization is an order of magnitude slower than sentence-based extraction.

One common way to solve an ILP problem is to use its LP relaxation and round the results. However Berg-Kirkpatrick et al. (2011) found that LP relaxation gave poor results, finding unacceptably suboptimal solutions. For speedup, they proposed a two stage method where they performed some sentence selection in the first step to reduce the number of candidates. Despite their empirical success, such

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1492–1502,
Seattle, Washington, USA, 18-21 October 2013.

a pruning approach has its inherent problem in that it may eliminate correct sentences in the first step. Recently, Almeida and Martins (2013) proposed a fast joint decoding algorithm based on dual decomposition. For fast convergence, they added quadratic penalty terms to alleviate the learning rate problem.

In this paper, we propose an efficient decoding algorithm for fast ILP based compressive summarization using graph cuts. Our assumption is that all concepts are word n-grams and non-negatively scored. The rationale for the non-negativity assumption is straightforward: the score of a concept reflects its informativeness, hence should be non-negative. Given a set of documents, each word is associated with a binary variable, indicating whether the word is selected in the summary. Our idea is to approximate the ILP as a binary quadratic programming problem where coefficients of all quadratic terms are non-negative. It is well known that such binary quadratic function is supermodular, and its maximum can be solved efficiently using graph max-flow/min-cut. Hence the key is to find the coefficients of the supermodular binary quadratic function (SBQF) so that its maximum is close to the optimal ILP objective function. Our solution consists of 3 steps. First, we show that the subtree deletion model and grammar constraints can be eliminated by adding SBQFs to the objective function. Second, we relax the summary length constraint using Lagrangian relaxation. Third, we propose a family of SBQFs that are lower bounds of the ILP objective function. Since finding the tightest lower bound suffers from local optimality, we choose to use convex relaxation for initialization. To demonstrate our technique, we conduct experiments on Text Analysis Conference (TAC) datasets using the same train/test splits as previous work (Berg-Kirkpatrick et al., 2011). We compare our approach with the state-of-the-art ILP based approach in terms of summary quality (ROUGE scores and sentence quality) and speed. Experimental results show that our proposed method achieves competitive performance with ILP, while about 100 times faster.

2 Compressive Summarization

2.1 Extractive Summarization

As our method is an approximation of ILP based method, we first briefly review the ILP based extractive summarization and compressive summarization. Gillick and Favre (2009) introduced the concept-based ILP for summarization. A concept is a basic semantic unit. They used word bigrams as such language concepts. Their system achieved the highest ROUGE score on the TAC 2009 evaluation. This approach selects sentences so that the total score of language concepts appearing in the summary is maximized. The association between the language concepts and sentences serves as the constraints, in addition to the summary length constraint.

Formally, given a set of sentences $\mathcal{S} = \{s_n\}_{n=1}^N$, extractive summarization can be represented by a binary vector $\mathbf{y}$, where y_n indicates whether sentence s_n is selected. Let $\mathcal{C} = \{c_1, \dots c_J\}$ denote the set of concepts in $\mathcal{S}$, e.g., word bigrams (Gillick and Favre, 2009). Each concept c_j is associated with a given score w_j and a binary variable v_j indicating if c_j is selected in the summary. Let n_{jk} denote the index of the sentence containing the k^{th} occurrence of concept c_j, and l_n denote the length of sentence s_n. The ILP based extractive summarization system can be formulated as below:

$$\begin{array}{lll} \max\limits_{\mathbf{y},\mathbf{v}} & \sum\limits_{j=1}^{J} w_j v_j & \\ \text{s.t.} & v_j = \bigcup\limits_k y_{n_{jk}} & 1 \le j \le J \\ & \sum\limits_{i=1}^{N} y_n l_n \le L & \\ & \mathbf{v}, \mathbf{y} \text{ are binary} & \end{array} \quad (1)$$

The first constraint is imposed by the relation between concept selection and sentence selection: selecting a sentence leads to the selection of all the concepts it contains, and selecting a concept only happens when it is present in at least one of the selected sentences. The second constraint is the summary length constraint.

As solving an ILP problem is generally NP-hard, pre-pruning of candidate concepts and sentences is necessary for efficient summarization. For example, the ICSI system (Gillick et al., 2008) removed the sentences that are too short or have non-overlap with the queries, and concepts with document frequency less than 3, resulting in 95.8 sentences and about 80 concepts per topic on the TAC2009 dataset. Therefore the actual scale of ILP is rather small after pruning (e.g., 176 variables and 372 constraints per topic). Empirical studies showed that such small scale ILP can be solved within a few seconds (Gillick and Favre, 2009).

2.2 Compressive Summarization

The quality of sentence-based extractive summarization is limited by the informativeness of the original sentences and the summary length constraint. To remove the unimportant part from a long sentence, sentence compression is proposed to generate more informative summaries (Liu and Liu, 2009; Li et al., 2013a). Recent studies show that joint sentence compression and extraction, namely compressive summarization, outperforms pipeline systems that run extractive summarization on the compressed sentences or compress selected summary sentences (Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Chali and Hasan, 2012). In Berg-Kirkpatrick et al. (2011), compressive summarization integrates the concept model for extractive summarization (Gillick and Favre, 2009) and subtree deletion model for sentence compression. The score of a compressive summary consists of two parts, scores of selected concepts, and scores of the broken arcs in the dependency parse trees. The selected words must satisfy the length constraint and grammar constraints that include subtree constraint and some manually defined hard constraints.

Formally, let $\mathbf{x} = x_1 \ldots x_I$ denote the word sequence of documents, where $s_1 = x_1, \ldots x_{l_1}$ corresponds to the first sentence, $s_2 = x_{l_1+1}, \ldots, x_{l_1+l_2}$ corresponds to the second sentence, and so on. A compressive summary can be represented by a binary vector $\mathbf{z}$, where z_i indicates whether word x_i is selected in the summary. Let a_{hm} denote the arc $x_h \rightarrow x_m$ in the dependency parse tree of the corresponding sentence containing words x_h and x_m, and $\mathcal{A} = \{a_{hm}\}$ denote the set of dependency arcs. The subtree constraint ensures that word x_m is selected only if its head x_h is selected. In order to guarantee the readability, grammar constraints are added to prohibit the breaks of some specific arcs. For example, Clarke and Lapata (2008) never deleted an arc whose dependency label is *SUB, OBJ, PMOD, SBAR* or *VC*. In this paper, we use $\mathcal{B} \subseteq \mathcal{A}$ to denote the set of these arcs that must not be broken in summarization. We use o_{jk} to denote the indices of words corresponding to the k^{th} occurrence of c_j. For example, suppose the j^{th} concept *European Union* appears twice in the document: $x_{22}x_{23} = x_{50}x_{51} =$*European Union*, then $o_{j1} = \{22, 23\}, o_{j2} = \{50, 51\}$.

The compressive summarization model can be formulated as an integer programming problem

$$\begin{aligned}
\max_{\mathbf{z},\mathbf{v}} \quad & \sum_{j=1}^{J} w_j \cdot v_j + \sum_{a_{hm}\in\mathcal{A}} w_{a_{hm}} z_h(1-z_m) \\
\text{s.t.} \quad & v_j = \bigcup_k \prod_{i\in o_{jk}} z_i \qquad \forall j \\
& \sum_i z_i \le L \\
& z_h \ge z_m \quad \forall a_{hm} \in \mathcal{A} \\
& z_h = z_m \quad \forall a_{hm} \in \mathcal{B} \\
& \mathbf{z}, \mathbf{v} \text{ are binary}
\end{aligned} \tag{2}$$

According to the subtree deletion model, the score of arc a_{hm} is included if $z_h = 1$ and $z_m = 0$, which can be formulated as $w_{a_{hm}} \cdot z_h(1 - z_m)$. The first constraint is similar to that in extractive summarization, that is, a concept is selected if and only if any of its occurrence is selected. The third and fourth constraints are the subtree constraints and manually defined grammar constraints respectively. In the rest of the paper, without loss of generality, we remove the fourth constraint by directly substituting one variable for the other.

Finding the optimal summary is generally NP-hard. Unlike extractive summarization where the scale of the problem (the number of sentences and concepts) is small, the number of variables in compressive summarization is linear in the number of words, which is usually thousands on the TAC datasets. Hence solving such a problem using ILP based decoding algorithms is not efficient especially when the document set is large.

3 Fast Decoding via Graph Cuts

In this section, we introduce our fast decoding algorithm. We assume that all the concepts are word n-grams, and their scores are non-negative. The non-negativity assumption can reduce the computational complexity, but is also reasonable: the score of a concept denotes its informativeness, hence should be non-negative. For example, Li et al. (2013b) proposed to use the estimated normalized frequencies of concepts as scores, which are essentially non-negative. The basic idea of our method is to approximate the above optimization problem (2) by the supermodular binary quadratic programming (SBQP) problem:

$$\begin{aligned} \max_{\mathbf{z}} \quad & \sum_i \beta_i z_i + \sum_{ij} \alpha_{ij} z_i z_j \\ \text{s.t.} \quad & \mathbf{z} \text{ is binary} \end{aligned} \tag{3}$$

where $\alpha_{ij} \geq 0$. It is known that such a binary quadratic function is supermodular, and its maximum can be solved efficiently using graph max-flow/min-cut (Billionnet and Minoux, 1985; Kolmogorov and Zabih, 2004). Now the problem is to find the optimal α, β for a good approximation.

3.1 Formulate Grammar Constraints and Subtree Deletion Model by SBQF

We show that the subtree deletion model can be formulated equivalently using SBQF. First, we can eliminate the constraint $z_h \geq z_m$ by adding a penalty term to the objective function. That is,

$$\begin{aligned} \max \quad & f(\mathbf{z}) \\ \text{s.t.} \quad & z_h \geq z_m \\ & \mathbf{z} \text{ is binary} \end{aligned}$$

is equivalent to

$$\begin{aligned} \max \quad & f(\mathbf{z}) - \infty(1 - z_h) z_m \\ \text{s.t.} \quad & \mathbf{z} \text{ is binary} \end{aligned}$$

We can see that the penalty term $-\infty(1 - z_h)z_m$ excludes $z_h = 0$, $z_m = 1$ from the feasible set, and for $z_h \geq z_m$, both problems have the same objective function value. Hence the two problems are equivalent. Notice that the coefficient of quadratic term in $-\infty(1 - z_h)z_m$ is positive, hence the penalty term is supermodular.

Now we eliminate the third constraint in problem (2) using the penalized objective function described above. Note that the fourth constraint has been eliminated by variable substitution, we have

$$\begin{aligned} \max_{\mathbf{z},\mathbf{v}} \quad & \sum_{j=1}^{J} w_j \cdot v_j + \sum_{a_{hm} \in \mathcal{A}} w_{a_{hm}} z_h (1 - z_m) \\ & -\infty \sum_{a_{hm} \in \mathcal{A}} (1 - z_h) z_m \\ \text{s.t.} \quad & v_j = \bigcup_k \prod_{i \in o_{jk}} z_i \qquad \forall j \\ & \sum_i z_i \leq L \\ & \mathbf{z}, \mathbf{v} \text{ are binary} \end{aligned} \tag{4}$$

We can see that for each arc a_{hm}, there must be a positive quadratic term $+\infty z_h z_m$ in the objective function, which guarantees the supermodularity of the objective function, no matter what $w_{a_{hm}}$ is.

3.2 Eliminate Length Constraint Using Lagrangian Relaxation

Problem (4) is NP-hard, because for any feasible $\mathbf{v}$, it is a SBQP with a length constraint. Since size constrained minimum cut problem is generally NP-hard (Nagano et al., 2011), Problem (4) can not be cast as a SBQP as long as P $\neq$ NP. One popular way to deal with the size constrained optimization problem is Lagrangian relaxation. We introduce Lagrangian multiplier λ to the length constraint in Problem (4), and get

$$\begin{aligned} \min_{\lambda} \max_{\mathbf{z},\mathbf{v}} \quad & \sum_{j=1}^{J} w_j \cdot v_j + \sum_{a_{hm} \in \mathcal{A}} w_{a_{hm}} z_h (1 - z_m) \\ & -\infty \sum_{a_{hm} \in \mathcal{A}} (1 - z_h) z_m \\ & +\lambda (L - \sum_i z_i) \\ \text{s.t.} \quad & v_j = \bigcup_k \prod_{i \in o_{jk}} z_i \qquad \forall j \\ & \lambda \geq 0 \\ & \mathbf{z}, \mathbf{v} \text{ are binary} \end{aligned} \tag{5}$$

We solve the relaxed problem iteratively. In each iteration, we fix λ and solve the inner maximization problem (details described below). The score of each word is penalized by λ — with larger λ, fewer words are selected. Hence the summary length can be adjusted by λ. The optimal λ can be found using binary search. We maintain an upper bound λ_{max} , and a lower bound λ_{min}, which is initially 0. In each iteration, we choose $\lambda = \frac{1}{2}(\lambda_{max} + \lambda_{min})$ and search the optimal $\mathbf{z}$. If the duality gap vanishes, i.e., $\lambda(L - \sum_i z_i) = 0$ and $\sum_i z_i \leq L$, then we get the global solution of Problem (4). Otherwise, if $\sum_i z_i > L$, then the current λ is too small, so we set $\lambda_{min} = \lambda$; otherwise, $\lambda > 0$ and $\sum_i z_i < L$, we set $\lambda_{max} = \lambda$. The search process terminates if $\lambda_{max} - \lambda_{min}$ is less than a predefined threshold.

3.3 Eliminate v Using Supermodular Relaxation

Now we consider the inner maximization Problem (5). It is still not a SBQP, since the objective function is not a linear function of $z_i z_j$. We propose to approximate the objective function using SBQP. Our solution consists of two steps. First we relax the first constraint of Problem (5) by bounding the objective function with a family of supermodular pseudo boolean functions (Boros and Hammer, 2002). Second we reformulate these pseudo boolean functions equivalently as quadratic functions.

Similar to the bounding strategy in (Qian and Liu, 2013), we relax the logical disjunction by linearization. Using the fact that for any binary vector $\mathbf{a}$, we have

$$\bigcup a_i = \max_{\mathbf{p} \in \Delta} \sum_i p_i a_i$$

where Δ denotes the probability simplex

$$\Delta = \{\mathbf{p} | \sum_k p_k = 1, p_k \geq 0\}$$

We have

$$\begin{aligned} v_j &= \bigcup_k \prod_{i \in o_{jk}} z_i \\ &= \max_{\mathbf{p}_j \in \Delta} \sum_k p_{jk} \prod_{i \in o_{jk}} z_i \end{aligned}$$

Plug the equation above into the objective function of Problem (5), we get the following optimization problem

$$\begin{aligned} \max_{\mathbf{z},\mathbf{p}} \quad & \sum_{j=1}^{J} \left(\sum_k p_{jk} w_j \prod_{i \in o_{jk}} z_i \right) \\ & + \sum_{a_{hm} \in \mathcal{A}} w_{a_{hm}} z_h (1 - z_m) \\ & -\infty \sum_{a_{hm} \in \mathcal{A}} (1 - z_h) z_m \\ & +\lambda(L - \sum_i z_i) \\ \text{s.t.} \quad & \mathbf{z} \text{ is binary} \qquad (6) \\ & \mathbf{p}_j \in \Delta \qquad \forall j \end{aligned}$$

Let $Q(\mathbf{p}, \mathbf{z})$ denote the objective function of Problem (6). Given $\mathbf{p}$, we can see that Q is a supermodular pseudo boolean function because coefficients of all non-linear terms are non-negative. Using the fact that for any binary vector $\mathbf{a} = [a_1, \ldots a_r]^T$, $a_i \in \{0, 1\}, 1 \leq i \leq r$,

$$\prod_{i=1}^{r} a_i = \max_{b \in \{0,1\}} \left(\sum_{i=1}^{r} a_i - r + 1 \right) b$$

(Freedman and Drineas, 2005), we get the following equivalent optimization problem of Problem (6)

$$\begin{aligned} \max_{\mathbf{z},\mathbf{p},\mathbf{q}} \quad & \sum_{j=1}^{J} \sum_k p_{jk} w_j q_{jk} \left(\sum_{i \in o_{jk}} z_i - |o_{jk}| + 1 \right) \\ & + \sum_{a_{hm} \in \mathcal{A}} w_{a_{hm}} z_h (1 - z_m) \\ & -\infty \sum_{a_{hm} \in \mathcal{A}} (1 - z_h) z_m \\ & +\lambda(L - \sum_i z_i) \\ \text{s.t.} \quad & \mathbf{z}, \mathbf{q} \text{ are binary} \qquad (7) \\ & \mathbf{p}_j \in \Delta \qquad \forall j \end{aligned}$$

where $|o_{jk}|$ is the size of o_{jk}.

Let $R(\mathbf{z}, \mathbf{p}, \mathbf{q})$ denote the objective function of Problem (7), to search the optimal point, we alternatively update $\mathbf{p}$ and $\mathbf{z}, \mathbf{q}$. First we initialize $\mathbf{p} = \mathbf{p}(0)$. In each iteration, we first fix $\mathbf{p}$. It is obvious that Problem (7) is a SBQP, hence the optimal $\mathbf{z}, \mathbf{q}$ can be solved efficiently using max-flow/min-cut. Then we fix $\mathbf{z}, \mathbf{q}$, and update $\mathbf{p}$ using projected subgradient. That is

$$\mathbf{p}_j^{\text{new}} = P_\Delta \left(\mathbf{p}_j + \frac{\partial R}{\partial \mathbf{p}_j} \alpha \right) \quad (8)$$

where $\alpha > 0$ is the step size in line search, and function $P_\Delta(q)$ denotes the projection of q onto the feasible set Δ

$$P_\Delta(\mathbf{q}) = \min_{\mathbf{p} \in \Delta} \|\mathbf{p} - \mathbf{q}\|_2$$

which can be solved efficiently by sorting (Duchi et al., 2008).

3.4 Initialize p Using Convex Relaxation

Since R is non-concave, searching its maximum using subgradient method suffers from local optimality. Though one can use techniques such as branch-and-bound for exact inference (Qian and Liu, 2013; Gormley and Eisner, 2013), here for fast decoding, we use convex relaxation to choose a good seed $\mathbf{p}(0)$. Recall that p_{jk} denotes the percentage of the k^{th} occurrence contributing to c_j. The larger p_{jk} is, the more likely the k^{th} occurrence is selected. To estimate such likelihood, we replace the binary constraint in extractive summarization (Problem (1)) by $\mathbf{0} \leq \mathbf{y}, \mathbf{v} \leq \mathbf{1}$, since solving a relaxed LP is much faster than ILP. Suppose $\mathbf{y}^*$ is the optimal solution for such a relaxed LP problem, we initialize $\mathbf{p}$ by

$$p_{jk} = \frac{y^*_{n_{jk}}}{\sum_k y^*_{n_{jk}}} \quad (9)$$

If for all k, $y^*_{n_{jk}} = 0$, then we initialize p_{jk} using uniform distribution

$$p_{jk} = \frac{1}{|o_j|}$$

where $|o_j|$ is the frequency of c_j.

3.5 Summary

For clarity, we summarize our decoding algorithm in Algorithm 1. Initial λ_{max} can be arbitrarily large. In our experiments, we set $\lambda_{max} = \sum_j w_j$, which empirically guarantees the summary length $\sum_i z_i \leq L$ when $\lambda = \lambda_{max}$. The choice of the step size for updating $\mathbf{p}$ is similar to the projected subgradient method in dual decomposition (Koo et al., 2010).

Algorithm 1 Compressive Summarization via Graph Cuts

```
Require: Scores of concepts $\{w_j\}$ and arcs $\{w_{a_{hm}}\}$, max summary length $L$.
Ensure: Compressive summarization $\mathbf{z}^*$, where $z_i$ indicates whether the $i^{th}$ word is selected.
  Solve the relaxed LP of Problem (1) (replace the binary constraint by $\mathbf{0} \leq \mathbf{y}, \mathbf{v} \leq \mathbf{1}$) to get $\mathbf{y}$.
  Initialize $\mathbf{p}(0)$ using Eq (9).
  Initialize sufficient large $\lambda_{max}$, and $\lambda_{min} = 0$
  while $\lambda_{max} - \lambda_{min} > \epsilon$ do
    Set $\lambda = \frac{1}{2}(\lambda_{min} + \lambda_{max})$
    Set $\mathbf{p} = \mathbf{p}(0)$.
    repeat
      Fix $\mathbf{p}$, solve Problem (7) to get $\mathbf{z}$ using max-flow/min-cut.
      Update $\mathbf{p}$ using Eq (8).
    until convergence
    if $\sum_i z_i > L$ then
      $\lambda_{min} = \lambda$
    else if $\sum_i z_i < L$ then
      $\lambda_{max} = \lambda$
    else
      break
    end if
  end while
```

4 Features and Hard Constraints

We choose discriminative models to learn the scores of concepts and arcs. For concept c_j, its score is

$$w_j = \theta_{\text{concept}}^T \mathbf{f}_{\text{concept}}(c_j)$$

where $\mathbf{f}_{\text{concept}}(c_j)$ is the feature vector of c_j, and θ_{concept} is the corresponding weight vector of feature $\mathbf{f}_{\text{concept}}(c_j)$. Similarly, score $w_{a_{hm}}$ is defined as

$$w_{a_{hm}} = \theta_{\text{arc}}^T \mathbf{f}_{\text{arc}}(a_{hm})$$

Though our algorithm can handle general word n-gram concepts, we restrict the concepts to word bigrams, which have been widely used recently in the sentence-based ILP extractive summarization systems. For a concept c_j, we define the following features, some of which have been used in previous work (Brandow et al., 1995; Aker and Gaizauskas, 2009; Edmundson, 1969; Radev, 2001; Li et al., 2013b). All of these features are non-negative.

- Term frequency: the frequency of c_j in the given topic.

- Stop word ratio: ratio of stop words in c_j. The value can be $\{0, 0.5, 1\}$.

- Similarity with topic title: the number of common words in these two strings, divided by the length of the longer string.

- Document ratio: percentage of documents containing c_j.

- Sentence ratio: percentage of sentences containing c_j.

- Sentence-title similarity: word unigram/bigrams cosine similarity between the sentence containing c_j and the topic title. For concepts appearing in multiple sentences, we choose the maximal similarity.

- Sentence-query similarity: word unigram/bigram cosine similarity between the sentence containing c_j and the topic query (concatenation of topic title and description). For concepts appearing in multiple sentences, we choose the maximal similarity.

- Sentence position: position of the sentence containing c_j in the document. For concepts appearing in multiple sentences, we choose the minimum.

- Sentence length: length of the sentence containing c_j. For concepts appearing in multiple sentences, we choose the maximum.

- Paragraph starter: binary feature indicating whether c_j appears in the first sentence of a paragraph.

For subtree deletion model, we define the following features for arc a_{hm}.

- POS tags of head word x_h and child word x_m and their concatenations.

- Dependency label of arc a_{hm} and its parent arc.

- Word x_m if x_m is a conjunction word or preposition word. Word x_h if x_m is a conjunction word or preposition word.

- Binary feature indicating whether the modifier x_m is a temporal word such as *Friday*.

We also define some hard constraints for subtree deletion to improve the readability of the generated compressed sentences.

- **C0** Arc a_{hm} can be cut only if one of the two conditions holds: (1) there is a comma, colon, or semicolon between the head and the modifier; (2) the modifier word is a preposition (POS tag is *IN*) or a wh-word, such as *what, who, whose* (corresponds to POS tag *IN, WDT, WP, WP$, WRB*).

- **C1** Arcs with dependency labels *SUB, OBJ, PRD, SBAR* or *VC* can not be cut.

- **C2** Arcs in set phrases like *so far, more than, according to* can not be cut.

- **C3** All arcs in coordinate structures can not be cut, such as *cats and dogs*.

Note that compared with previous work, our compression is more conservative. Constraint C0 allows only a small portion of arcs to be cut. This is based on our observation of the sentence compression corpus: removing preposition phrases (PP) or sub-clauses can greatly reduce the length of sentence, while hurting the readability little. Cutting other arcs like *NMOD* usually removes only one or two words, and possibly affects the sentence's readability.

5 Experimental Results

5.1 Experimental Setup

Due to the lack of training data for compressive summarization, we learn the subtree deletion model and the concept model separately. Specifically, the sentence compression dataset (Clarke and Lapata, 2008) (referred as CL08) is used for subtree deletion model training (θ_{arc}). A sentence pair in the corpus is kept for training the subtree deletion model if the compressed sentence can be derived by deleting subtrees from the parse tree of the original sentence. There are $3,178$ out of $5,739$ such pairs. The concept model (θ_{concept}) is learned from the TAC2009 dataset. We create the oracle extractive summaries with the maximal bigram recall as the reference summary. TAC2010 data is used as

	Corpus	Sent.	Words	Topics
Train	TAC2009	$4,216$	$117,304$	44
	CL08	$3,178$	$52,624$	N/A
Develop	TAC2010	$2,688$	$72,609$	46
Test	TAC2008	$4,518$	$123,946$	48

Table 1: Corpus statistics. Training data consist of two parts, TAC2009 for learning the concept model, CL08 (Clarke and Lapata, 2008) for learning the subtree deletion model.

development set for various parameter tuning. Table 1 has the descriptions of all the data used.

We choose averaged perceptron for fast training. The number of iterations is tuned on the development data. Remind that our algorithm is based on the assumption that scores of concepts are non-negative, $\forall j, w_j \geq 0$. We assume that feature vector $\mathbf{f}_{\text{concept}}$ is non-negative (e.g., term frequency, n-gram features), then $\theta_{\text{concept}} \geq 0$ is required to guarantee the non-negativity of w_j. Therefore, we project θ_{concept} onto the non-negative space after each iteration. Since training is offline, we use ILP based exact inference for accurate learning. [1]

To control the contributions of the concept model and the subtree deletion model, we introduce a parameter μ, and modify the original maximization problem (Problem 2) to:

$$\max_{\mathbf{z},\mathbf{v}} \quad \sum_{j=1}^{J} w_j \cdot v_j + \mu \times \sum_{a_{hm} \in \mathcal{A}} w_{a_{hm}} z_h (1 - z_m)$$

We tune μ on TAC2010 dataset. For max-flow/min-cut, in our experiments, we implemented the improved shortest augmented path (SAP) method (Edmonds and Karp, 1972).

For performance measure of the summaries, we use both ROUGE and linguistic quality. ROUGE has been widely used for summarization performance and can measure the informativeness of the summaries (content match between system and reference summaries). Since joint compression and summarization tends to pick isolated words to maximize the information coverage in the system generated summaries, it may have poor readability. Therefore we conduct human evaluation for the linguistic quality for various systems. The linguistic quality consists of two parts. One evaluates the grammar quality within a sentence. 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. For compressive summaries, we removed the sentences with grammar errors when evaluating coherence. The overall linguistic quality score is the combined score of the percentage of grammatically correct sentences and the correct ordering of the summary sentences. The score is scaled and ranges from 1 (bad) to 10 (good).

5.2 Results on the Development Set

We conducted a series of experiments on the development dataset to investigate the effect of the non-negative score assumption, SBQP approximation, and initialization. First, we build a standard ILP based compressive summarizer without the non-negative score assumption. We varied μ over $\{2^{-4}, 2^{-3}, \dots 2^{4}\}$ and selected the optimal $\mu = 2^{-2}$ according to both ROUGE-2 score and linguistic quality. This interpolation weight is used in all the other experiments.

To study the impact of the non-negative score assumption, we build another summarizer by replacing the concept model with the one trained under the non-negative constraint. We also compared three different initialization strategies for $\mathbf{p}$. The first one is uniform initialization, i.e., $p_{jk} = \frac{1}{|o_j|}$. The second one is a greedy approach, where extractive summarization is obtained by greedy search (i.e., add the top ranked sentence iteratively), then we use the corresponding $\mathbf{y}$ and Eq (9) to initialize $\mathbf{p}$. The last one is our convex relaxation method described in Section 3.4.

Table 2 shows the comparison results. For comparison, we also include the sentence-based ILP extractive summarization results. We can see that the impact of initial $\mathbf{p}$ is substantial. Using convex relaxation helps our method to survive from local optimality. The non-negativity assumption has very little effect on the standard compressive summarization (comparing the first two rows). This empirical result demonstrates the appropriateness of the assumption we use in our proposed method.

[1] we choose the GLPK as our ILP solver, which is used in (Berg-Kirkpatrick et al., 2011)

System	R-2	LQ
ILP ($\mu = 2^{-2}$)	11.22	6.3
ILP (Non Neg.)	11.18	6.4
Graph Cut (uniform)	9.54	5.9
Graph Cut (greedy)	10.13	6.2
Graph Cut (LP)	11.06	6.1
Sent Extractive	10.11	7.3

Table 2: Experimental results on development dataset. R-2 and LQ are short for ROUGE-2 score multiplied by 100, and linguistic quality respectively.

5.3 Results on Test Dataset

Table 3 shows the summarization results for various systems on the TAC2008 data set. We show both the summarization performance and the speed[2] of the system. The presented systems include our graph-cut based method, the ILP based compression and summarization, and the sentence-based extractive summarization. ILP 2-step refers to the 2-step fast decoding strategy proposed by (Berg-Kirkpatrick et al., 2011).

We also list the performance of some state-of-the-art systems, including the two ICSI systems (Gillick et al., 2008), the compressive summarization system of Berg-Kirkpatrick et al. (2011) (GBK'11), the multi-aspect ILP system of Woodsend and Lapata (2012)(WL'12) and the dual decomposition based system (Almeida and Martins, 2013) (AM'13). Note that for these referred systems since the linguistic quality results are not comparable due to different judgment methods. For our graph-cut based method, to study the tradeoff between the readability of the summary and the ROUGE scores, we present two versions for this method: one uses all the constraints (C0-C3), the other does not use C0.

We can see that our proposed method balanced speed and quality. Compared with ILP, we achieved competitive ROUGE scores, but with about 100x speedup. Our method is also faster than the 2-step ILP system. We also tried another state-of-the-art LP solver, Gurobi version 5.5[3], it achieves 0.17 seconds per topic, much faster than GLPK, but still slower than ours. Regarding the grammar constraints used in our system, from the two results for our graph-cut based method, we can see that adding constraint C0 significantly decreases the R-2 score but improves the language quality. This shows that word-based joint compression and summarization can improve ROUGE score; however, we need to keep in mind about linguistic quality and find a tradeoff between the ROUGE score and the linguistic quality. Almeida and Martins (2013) trained their model on extra corpora using multi-task learning, and achieved better results than ours. The results of our system and theirs showed that Lagrangian relaxation based method combined with combinatorial optimization algorithms such as dynamic programming or minimum cut can exploit the inner structure of problems and achieve significant speedup over ILP.

System	R-2	R-SU4	LQ	sec.
Graph Cut	11.74	14.54	6.5	0.056
Graph Cut w/o C0	12.05	14.71	5.4	0.053
ILP	11.86	14.62	6.6	5.5
ILP (Non Neg.)	11.82	14.60	6.6	5.2
ILP (2-step)	11.72	14.49	6.5	1.1
Sent Extractive	11.06	13.93	7.1	0.13
ICSI-1	11.0	13.4	-	0.38†
ICSI-2	11.1	14.3	-	-
BGK'11	11.70	14.38	6.5†	-
WL'12	11.37	14.47	-	-
AM'13	12.30$^+$	15.18$^+$	4.2†	0.41†

Table 3: Experimental results on TAC2008 dataset. Columns 2-5 are scores of ROUGE-2, ROUGE-SU4, linguistic quality, and speed (seconds per topic). ROUGE-2 and ROUGE-SU4 scores are multiplied by 100. All the experiments are conducted on the platform Intel Core i5-2500 CPU 3.30GHz. † numbers are not directly comparable due to different annotations or platforms. $^+$ extra resources are used.

Four example summaries produced by our system are shown below. Words in gray are not selected in the summary.

[2] For fair comparison, we only recode the running time for decoding. Other time costs like feature extraction, IO operations are excluded.

[3] www.gurobi.com

India's space agency is ready to send a man to space within seven years if the government gives the nod, while preparations have lready begun for the launch of an unmanned lunar mission, a top official said. India will launch more missions to the moon if its maiden unmanned spacecraft Chandrayaan-1, slated to be launched by 2008, is successful a top space fficial said Tuesday. The United States, the European Space Agency, China, Japan and India are all planning lunar missions during the ext decade.India is "a step ahead" of China in satellite technology and can surpass Beijing in space research by tapping the talent of its huge pool of young scientists, India's space research chief said Monday. The space agencies of India and France signed an agreement on Friday to cooperate in launching a satellite in four years that will help make climate predictions more accurate. The Indian Space Research Organization (ISRO) has short-listed experiments from five nations including the United States, Britain and Germany, for a slot on India's unmanned moon mission Chandrayaan-1 to be undertaken by 2006-2007, the Press Trust of India (PTI) reported Monday. A three-member Afghan delegation is in Bangalore seeking help to set up a high-tech telemedicine facility in 10 Afghan cities linked via Indian satellites, Indo-Asian News Service reported Saturday.

A woman was killed in Mississippi when a tree crashed on her car, becoming the 11th fatality blamed on the powerful Hurricane Katrina that slammed the US Gulf coast after pounding Florida, local TV reported Monday. The bill for the Hurricane Katrina disaster effort has so far reached 2.87 billion dollars, federal officials said on Tuesday. The official death toll from Hurricane Katrina has risen to 118 people in and around the swamped city of New Orleans, officials said Thursday. The Foreign Ministry on Friday reported the first confirmed death of a Guatemalan due to Hurricane Katrina in the United States. The Ugandan government has pledged 200,000 US dollars toward relief and rebuilding efforts in the aftermath of Hurricane Katrina, local press reported on Friday. Swiss Reinsurance Co., the world's second largest reinsurance company on Monday doubled to 40 billion US dollars its initial estimate of the global insured losses caused by Hurricane Katrina in the United States.

The A380 'superjumbo', which will be presented to the world in a lavish ceremony in southern France on Tuesday, will be profitable from 2008, its maker Airbus told the French financial newspaper La Tribune. The A380 will take over from the Boeing 747 as the biggest jet in the skies. An association of residents living near Paris's Charles-de-Gaulle airport on Wednesday denounced the noise pollution generated by the giant Airbus A380, after the new airliner's maiden flight. One problem that Airbus is encountering with its new A380 is that the craft pushes the envelope on the maximum size of a commercial airplane. With a whisper more than a roar, the largest passenger airliner ever built, the Airbus 380, took off on its maiden flight Wednesday.

"When she came in, she was in good spirits," a prison staffer told the New York Daily News. Martha Stewart, the American celebrity homemaker who had her own cooking and home improvement TV show, reported to a federal prison in Alderson, West Virginia, on Friday to serve a five-month sentence for lying about a stock sale. Home fashion guru Martha Stewart said on Friday that she has adjusted to prison life and is keeping busy behind bars since reporting a week ago to a federal penal camp in West Virginia, where she is serving a five-month sentence for lying about a stock sale. The lawyer said he did not know what she is writing, but Stewart has suggested since her conviction that she might write a book about her recent experience with the legal system. Walter Dellinger, the lawyer leading the appeal, said on NBC's "Today" that Stewart is exploring "innovative ways to do microwave cooking" The lawyer said he did not know with her fellow inmates. As Martha Stewart arrives at the red-brick federal prison in Alderson, W. Va., on Friday to begin a five-month sentence, the company she founded is focused both on life without her and on life once she returns.

In most cases, the removed phrases do not hurt the readability of the summaries. The errors are mainly caused by the break of sub-clauses or main clauses that are separated by commas, for example, the fourth sentence in the last summary, *The lawyer said he did not know what she is writing*. The compressed sentence is grammatically correct, but semantically incomplete. Other errors are due to the lack of verb, subject, or object, or incorrect removal of PP, such as the last sentence of the last summary.

6 Conclusion

In this paper, we propose a fast decoding algorithm for compressive summarization using graph cuts. Our idea is to approximate the original ILP problem using supermodular binary quadratic programming (SBQP) problem. Under the assumption that scores of concepts are non-negative, we eliminate subtree constraints and grammar constraints, and relax the length constraint and non-supermodular part of the problem step by step. Our experimental results showed that the graph cut based method achieved competitive performance compared to ILP, while about 100 times faster.

There are several possibilities for further research involving our graph cut algorithms. One idea is to apply it to the language model based compression method (Clarke and Lapata, 2008). The other is to adapt it to social media text summarization task, where text is much more noisy. As graph cut is a general method, applying it to other binary structured learning tasks is also an interesting direction.

Acknowledgments

We'd like to thank three anonymous reviewers for their valuable comments. This work is partly supported by NSF award IIS-0845484 and DARPA under Contract No. FA8750-13-2-0041. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

A. Aker and R. Gaizauskas. 2009. Summary generation for toponym-referenced images using object type language models. In *Proceedings of RANLP*.

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

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

A. Billionnet and M. Minoux. 1985. Maximizing a supermodular pseudoboolean function: A polynomial algorithm for supermodular cubic functions. *Discrete Applied Mathematics*, 12(1):1 – 11.

Endre Boros and Peter L. Hammer. 2002. Pseudoboolean optimization. *Discrete Applied Mathematics*, 123(1C3):155 – 225.

Ronald Brandow, Karl Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5):675 – 685.

Yllias Chali and Sadid A. Hasan. 2012. On the effectiveness of using sentence compression models for query-focused multi-document summarization. In *COLING*, pages 457–474.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *J. Artif. Intell. Res. (JAIR)*, 31:399–429.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. 2008. Efficient projections onto the L1-ball for learning in high dimensions. In *Proceedings of ICML*, pages 272–279.

Jack Edmonds and Richard M. Karp. 1972. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, April.

H. P. Edmundson. 1969. New methods in automatic extracting. *J. ACM*, 16(2):264–285, April.

Daniel Freedman and Petros Drineas. 2005. Energy minimization via graph cuts: Settling what is possible. In *CVPR (2)*, pages 939–946.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18, June.

Dan Gillick, Benoit Favre, and Dilek Hakkani-Tur. 2008. The ICSI summarization system at tac 2008. In *Proceedings of the Text Understanding Conference*.

Matthew R. Gormley and Jason Eisner. 2013. Nonconvex global optimization for latent-variable models. In *Proceedings of ACL*, pages 444–454, August.

Vladimir Kolmogorov and Ramin Zabih. 2004. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81.

Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proceedings of EMNLP 2010*, pages 1288–1298, October.

Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. 2013a. Document summarization via guided sentence compression. In *Proceedings of EMNLP (to appear)*, October.

Chen Li, Xian Qian, and Yang Liu. 2013b. Using supervised bigram-based ILP for extractive summarization. In *Proceedings of ACL*, pages 1004–1013, August.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of ACL*, pages 510–520, June.

Fei Liu and Yang Liu. 2009. From extractive to abstractive meeting summaries: Can it be done by sentence compression? In *Proceedings of ACL-IJCNLP 2009*, pages 261–264, August.

André F. T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 1–9.

Kiyohito Nagano, Yoshinobu Kawahara, and Kazuyuki Aihara. 2011. Size-constrained submodular minimization through minimum norm base. In *ICML*, pages 977–984.

Xian Qian and Yang Liu. 2013. Branch and bound algorithm for dependency parsing with non-local features. *TACL*, 1:105–151.

Dragomir R. Radev. 2001. Experiments in single and multidocument summarization using mead. In *In First Document Understanding Conference*.

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