Using Supervised Bigram-based ILP for Extractive Summarization

Chen Li, Xian Qian, and **Yang Liu**
The University of Texas at Dallas
Computer Science Department
chenli,qx,yangl@hlt.utdallas.edu

Abstract

In this paper, we propose a bigram based supervised method for extractive document summarization in the integer linear programming (ILP) framework. For each bigram, a regression model is used to estimate its frequency in the reference summary. The regression model uses a variety of indicative features and is trained discriminatively to minimize the distance between the estimated and the ground truth bigram frequency in the reference summary. During testing, the sentence selection problem is formulated as an ILP problem to maximize the bigram gains. We demonstrate that our system consistently outperforms the previous ILP method on different TAC data sets, and performs competitively compared to the best results in the TAC evaluations. We also conducted various analysis to show the impact of bigram selection, weight estimation, and ILP setup.

1 Introduction

Extractive summarization is a sentence selection problem: identifying important summary sentences from one or multiple documents. Many methods have been developed for this problem, 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 (ILP) and submodular methods. These global optimization methods have been shown to be quite powerful for extractive summarization, because they try to select important sentences and remove redundancy at the same time under the length constraint.

Gillick and Favre (Gillick and Favre, 2009) introduced the concept-based ILP for summarization. Their system achieved the best result in the TAC 09 summarization task based on the ROUGE evaluation metric. In this approach the goal is to maximize the sum of the weights of the language concepts that appear in the summary. They used bigrams as such language concepts. The association between the language concepts and sentences serves as the constraints. This ILP method is formally represented as below (see (Gillick and Favre, 2009) for more details):

$$\max \quad \sum_i w_i c_i \tag{1}$$
$$s.t. \quad s_j Occ_{ij} \leq c_i \tag{2}$$
$$\sum_j s_j Occ_{ij} \geq c_i \tag{3}$$
$$\sum_j l_j s_j \leq L \tag{4}$$
$$c_i \in \{0,1\}\ \forall i \tag{5}$$
$$s_j \in \{0,1\}\ \forall j \tag{6}$$

c_i and s_j are binary variables (shown in (5) and (6)) that indicate the presence of a concept and a sentence respectively. w_i is a concept's weight and Occ_{ij} means the occurrence of concept i in sentence j. Inequalities (2)(3) associate the sentences and concepts. They ensure that 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.

There are two important components in this concept-based ILP: one is how to select the concepts (c_i); the second is how to set up their weights (w_i). Gillick and Favre (Gillick and Favre, 2009) used bigrams as concepts, which are selected from a subset of the sentences, and their document frequency as the weight in the objective function.

In this paper, we propose to find a candidate summary such that the language concepts (e.g., bigrams) in this candidate summary and the reference summary can have the same frequency. We expect this restriction is more consistent with the

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1004–1013,
Sofia, Bulgaria, August 4-9 2013.

ROUGE evaluation metric used for summarization (Lin, 2004). In addition, in the previous concept-based ILP method, the constraints are with respect to the appearance of language concepts, hence it cannot distinguish the importance of different language concepts in the reference summary. Our method can decide not only which language concepts to use in ILP, but also the frequency of these language concepts in the candidate summary. To estimate the bigram frequency in the summary, we propose to use a supervised regression model that is discriminatively trained using a variety of features. Our experiments on several TAC summarization data sets demonstrate this proposed method outperforms the previous ILP system and often the best performing TAC system.

2 Proposed Method

2.1 Bigram Gain Maximization by ILP

We choose bigrams as the language concepts in our proposed method since they have been successfully used in previous work. In addition, we expect that the bigram oriented ILP is consistent with the ROUGE-2 measure widely used for summarization evaluation.

We start the description of our approach for the scenario where a human abstractive summary is provided, and the task is to select sentences to form an extractive summary. Then Our goal is to make the bigram frequency in this system summary as close as possible to that in the reference. For each bigram b, we define its gain:

$$\text{Gain}(b, sum) = \min\{n_{b,ref},\ n_{b,sum}\} \quad (7)$$

where $n_{b,ref}$ is the frequency of b in the reference summary, and $n_{b,sum}$ is the frequency of b in the automatic summary. The gain of a bigram is no more than its frequency in the reference summary, hence adding redundant bigrams will not increase the gain.

The total gain of an extractive summary is defined as the sum of every bigram gain in the summary:

$$\begin{aligned}\text{Gain}(sum) &= \sum_b \text{Gain}(b, sum) \\ &= \sum_b \min\{n_{b,ref}, \sum_s z(s) * n_{b,s}\} \quad (8)\end{aligned}$$

where s is a sentence in the document, $n_{b,s}$ is the frequency of b in sentence s, $z(s)$ is a binary variable, indicating whether s is selected in the summary. The goal is to find z that maximizes $\text{Gain}(sum)$ (formula (8)) under the length constraint L.

This problem can be casted as an ILP problem. First, using the fact that

$$\min\{a, x\} = 0.5(-|x - a| + x + a), \quad x, a \geq 0$$

we have

$$\begin{aligned}&\sum_b \min\{n_{b,ref}, \sum_s z(s) * n_{b,s}\} = \\ &\sum_b 0.5 * (-|n_{b,ref} - \sum_s z(s) * n_{b,s}| + \\ &n_{b,ref} + \sum_s z(s) * n_{b,s})\end{aligned}$$

Now the problem is equivalent to:

$$\begin{aligned}\max_z \quad & \sum_b (-|n_{b,ref} - \sum_s z(s) * n_{b,s}| + \\ & n_{b,ref} + \sum_s z(s) * n_{b,s}) \\ s.t. \quad & \sum_s z(s) * |S| \leq L; \quad z(s) \in \{0, 1\}\end{aligned}$$

This is equivalent to the ILP:

$$\max \quad \sum_b (\sum_s z(s) * n_{b,s} - C_b) \quad (9)$$

$$s.t. \quad \sum_s z(s) * |S| \leq L \quad (10)$$

$$z(s) \in \{0, 1\} \quad (11)$$

$$-C_b \leq n_{b,ref} - \sum_s z(s) * n_{b,s} \leq C_b \quad (12)$$

where C_b is an auxiliary variable we introduce that is equal to $|n_{b,ref} - \sum_s z(s) * n_{b,s}|$, and $n_{b,ref}$ is a constant that can be dropped from the objective function.

2.2 Regression Model for Bigram Frequency Estimation

In the previous section, we assume that $n_{b,ref}$ is at hand (reference abstractive summary is given) and propose a bigram-based optimization framework for extractive summarization. However, for the summarization task, the bigram frequency is unknown, and thus our first goal is to estimate such frequency. We propose to use a regression model for this.

Since a bigram's frequency depends on the summary length (L), we use a normalized frequency

in our method. Let $n_{b,ref} = N_{b,ref} * L$, where $N_{b,ref} = \frac{n(b,ref)}{\sum_b n(b,ref)}$ is the normalized frequency in the summary. Now the problem is to automatically estimate $N_{b,ref}$.

Since the normalized frequency $N_{b,ref}$ is a real number, we choose to use a logistic regression model to predict it:

$$N_{b,ref} = \frac{\exp\{w' f(b)\}}{\sum_j \exp\{w' f(b_j)\}} \quad (13)$$

where $f(b_j)$ is the feature vector of bigram b_j and w' is the corresponding feature weight. Since even for identical bigrams $b_i = b_j$, their feature vectors may be different ($f(b_i) \neq f(b_j)$) due to their different contexts, we sum up frequencies for identical bigrams $\{b_i | b_i = b\}$:

$$\begin{aligned} N_{b,ref} &= \sum_{i,b_i=b} N_{b_i,ref} \\ &= \frac{\sum_{i,b_i=b} \exp\{w' f(b_i)\}}{\sum_j \exp\{w' f(b_j)\}} \end{aligned} \quad (14)$$

To train this regression model using the given reference abstractive summaries, rather than trying to minimize the squared error as typically done, we propose a new objective function. Since the normalized frequency satisfies the probability constraint $\sum_b N_{b,ref} = 1$, we propose to use KL divergence to measure the distance between the estimated frequencies and the ground truth values. The objective function for training is thus to minimize the KL distance:

$$\min \sum_b \widetilde{N}_{b,ref} \log \frac{\widetilde{N}_{b,ref}}{N_{b,ref}} \quad (15)$$

where $\widetilde{N}_{b,ref}$ is the true normalized frequency of bigram b in reference summaries.

Finally, we replace $N_{b,ref}$ in Formula (15) with Eq (14) and get the objective function below:

$$\max \sum_b \widetilde{N}_{b,ref} \log \frac{\sum_{i,b_i=b} \exp\{w' f(b_i)\}}{\sum_j \exp\{w' f(b_j)\}} \quad (16)$$

This shares the same form as the contrastive estimation proposed by (Smith and Eisner, 2005). We use gradient decent method for parameter estimation, initial w is set with zero.

2.3 Features

Each bigram is represented using a set of features in the above regression model. We use two types of features: word level and sentence level features. Some of these features have been used in previous work (Aker and Gaizauskas, 2009; Brandow et al., 1995; Edmundson, 1969; Radev, 2001):

- Word Level:
  - **1. Term frequency1:** The frequency of this bigram in the given topic.
  - **2. Term frequency2:** The frequency of this bigram in the selected sentences[1].
  - **3. Stop word ratio:** Ratio of stop words in this bigram. The value can be {0, 0.5, 1}.
  - **4. Similarity with topic title:** The number of common tokens in these two strings, divided by the length of the longer string.
  - **5. Similarity with description of the topic:** Similarity of the bigram with topic description (see next data section about the given topics in the summarization task).

- Sentence Level: (information of sentence containing the bigram)
  - **6. Sentence ratio:** Number of sentences that include this bigram, divided by the total number of the selected sentences.
  - **7. Sentence similarity:** Sentence similarity with topic's query, which is the concatenation of topic title and description.
  - **8. Sentence position:** Sentence position in the document.
  - **9. Sentence length:** The number of words in the sentence.
  - **10. Paragraph starter:** Binary feature indicating whether this sentence is the beginning of a paragraph.

3 Experiments

3.1 Data

We evaluate our method using several recent TAC data sets, from 2008 to 2011. The TAC summarization task is to generate at most 100 words summaries from 10 documents for a given topic query (with a title and more detailed description). For model training, we also included two years' DUC data (2006 and 2007). When evaluating on one TAC data set, we use the other years of the TAC data plus the two DUC data sets as the training data.

[1] See next section about the sentence selection step

3.2 Summarization System

We use the same system pipeline described in (Gillick et al., 2008; McDonald, 2007). The key modules in the ICSI ILP system (Gillick et al., 2008) are briefly described below.

- Step 1: Clean documents, split text into sentences.
- Step 2: Extract bigrams from all the sentences, then select those bigrams with document frequency equal to more than 3. We call this subset as initial bigram set in the following.
- Step 3: Select relevant sentences that contain at least one bigram from the initial bigram set.
- Step 4: Feed the ILP with sentences and the bigram set to get the result.
- Step 5: Order sentences identified by ILP as the final result of summary.

The difference between the ICSI and our system is in the 4th step. In our method, we first extract all the bigrams from the selected sentences and then estimate each bigram's $N_{b,ref}$ using the regression model. Then we use the top-n bigrams with their $N_{b,ref}$ and all the selected sentences in our proposed ILP module for summary sentence selection. When training our bigram regression model, we use each of the 4 reference summaries separately, i.e., the bigram frequency is obtained from one reference summary. The same pre-selection of sentences described above is also applied in training, that is, the bigram instances used in training are from these selected sentences and the reference summary.

4 Experiment and Analysis

4.1 Experimental Results

Table 1 shows the ROUGE-2 results of our proposed system, the ICSI system, and also the best performing system in the NIST TAC evaluation. We can see that our proposed system consistently outperforms ICSI ILP system (the gain is statistically significant based on ROUGE's 95% confidence internal results). Compared to the best reported TAC result, our method has better performance on three data sets, except 2011 data. Note that the best performing system for the 2009 data is the ICSI ILP system, with an additional compression step. Our ILP method is purely extractive. Even without using compression, our approach performs better than the full ICSI system. The best performing system for the 2011 data also has some compression module. We expect that after applying sentence compression and merging, we will have even better performance, however, our focus in this paper is on the bigram-based extractive summarization.

	ICSI ILP	Proposed System	TAC Rank1 System
2008	0.1023	**0.1076**	0.1038
2009	0.1160	**0.1246**	0.1216
2010	0.1003	**0.1067**	0.0957
2011	0.1271	0.1327	**0.1344**

Table 1: ROUGE-2 summarization results.

There are several differences between the ICSI system and our proposed method. First is the bigrams (concepts) used. We use the top 100 bigrams from our bigram estimation module; whereas the ICSI system just used the initial bigram set described in Section 3.2. Second, the weights for those bigrams differ. We used the estimated value from the regression model; the ICSI system just uses the bigram's document frequency in the original text as weight. Finally, two systems use different ILP setups. To analyze which factors (or all of them) explain the performance difference, we conducted various controlled experiments for these three factors (bigrams, weights, ILP). All of the following experiments use the TAC 2009 data as the test set.

4.2 Effect of Bigram Weights

In this experiment, we vary the weighting methods for the two systems: our proposed method and the ICSI system. We use three weighting setups: the estimated bigram frequency value in our method, document frequency, or term frequency from the original text. Table 2 and 3 show the results using the top 100 bigrams from our system and the initial bigram set from the ICSI system respectively. We also evaluate using the two different ILP configurations in these experiments.

First of all, we can see that for both ILP systems, our estimated bigram weights outperform the other frequency-based weights. For the ICSI ILP system, using bigram document frequency achieves better performance than term frequency (which verified why document frequency is used in their system). In contrast, for our ILP method,

#	Weight	ILP	ROUGE-2
1	Estimated value	Proposed	**0.1246**
2		ICSI	0.1178
3	Document freq	Proposed	0.1109
4		ICSI	0.1132
5	Term freq	Proposed	0.1116
6		ICSI	0.1080

Table 2: Results using different weighting methods on the top 100 bigrams generated from our proposed system.

#	Weight	ILP	ROUGE-2
1	Estimated value	Proposed	0.1157
2		ICSI	**0.1161**
3	Document freq	Proposed	0.1101
4		ICSI	0.1160
5	Term freq	Proposed	0.1109
6		ICSI	0.1072

Table 3: Results using different weighting methods based on the initial bigram sets. The average number of bigrams is around 80 for each topic.

the bigram's term frequency is slightly more useful than its document frequency. This indicates that our estimated value is more related to bigram's term frequency in the original text. When the weight is document frequency, the ICSI's result is better than our proposed ILP; whereas when using term frequency as the weights, our ILP has better results, again suggesting term frequency fits our ILP system better. When the weight is estimated value, the results depend on the bigram set used. The ICSI's ILP performs slightly better than ours when it is equipped with the initial bigram, but our proposed ILP has much better results using our selected top100 bigrams. This shows that the size and quality of the bigrams has an impact on the ILP modules.

4.3 The Effect of Bigram Set's size

In our proposed system, we use 100 top bigrams. There are about 80 bigrams used in the ICSI ILP system. A natural question to ask is the impact of the number of bigrams and their quality on the summarization system. Table 4 shows some statistics of the bigrams. We can see that about one third of bigrams in the reference summary are in the original text (127.3 out of 321.93), verifying that people do use different words/bigram when writing abstractive summaries. We mentioned that we only use the top-N (n is 100 in previous experiments) bigrams in our summarization system. On one hand, this is to save computational cost for the ILP module. On the other hand, we see from the table that only 127 of these more than 2K bigrams are in the reference summary and are thus expected to help the summary responsiveness. Including all the bigrams would lead to huge noise.

# bigrams in ref summary	321.93
# bigrams in text and ref summary	127.3
# bigrams used in our regression model (i.e., in selected sentences)	2140.7

Table 4: Bigram statistics. The numbers are the average ones for each topic.

Fig 1 shows the bigram coverage (number of bigrams used in the system that are also in reference summaries) when we vary N selected bigrams. As expected, we can see that as n increases, there are more reference summary bigrams included in the system. There are 25 summary bigrams in the top-50 bigrams and about 38 in top-100 bigrams. Compared with the ICSI system that has around 80 bigrams in the initial bigram set and 29 in the reference summary, our estimation module has better coverage.

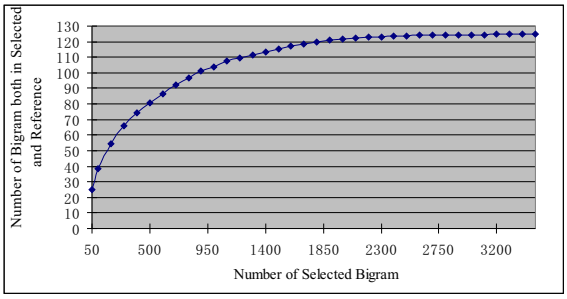

Figure 1: Coverage of bigrams (number of bigrams in reference summary) when varying the number of bigrams used in the ILP systems.

Increasing the number of bigrams used in the system will lead to better coverage, however, the incorrect bigrams also increase and have a negative impact on the system performance. To examine the best tradeoff, we conduct the experiments by choosing the different top-N bigram set for the two ILP systems, as shown in Fig 2. For both the ILP systems, we used the estimated weight value for the bigrams.

We can see that the ICSI ILP system performs better when the input bigrams have less noise (those bigrams that are not in summary). However, our proposed method is slightly more robust to this kind of noise, possibly because of the weights we use in our system – the noisy bigrams have lower weights and thus less impact on the final system performance. Overall the two systems have similar trends: performance increases at the beginning when using more bigrams, and after certain points starts degrading with too many bigrams. The optimal number of bigrams differs for the two systems, with a larger number of bigrams in our method. We also notice that the ICSI ILP system achieved a ROUGE-2 of 0.1218 when using top 60 bigrams, which is better than using the initial bigram set in their method (0.1160).

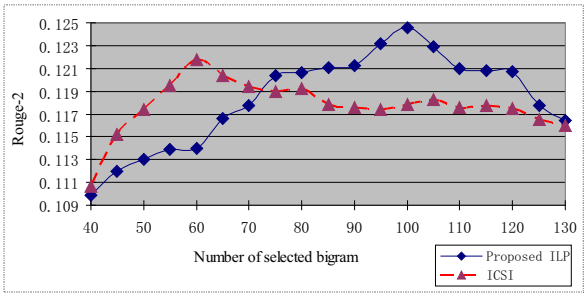

Figure 2: Summarization performance when varying the number of bigrams for two systems.

4.4 Oracle Experiments

Based on the above analysis, we can see the impact of the bigram set and their weights. The following experiments are designed to demonstrate the best system performance we can achieve if we have access to good quality bigrams and weights. Here we use the information from the reference summary.

The first is an oracle experiment, where we use all the bigrams from the reference summaries that are also in the original text. In the ICSI ILP system, the weights are the document frequency from the multiple reference summaries. In our ILP module, we use the term frequency of the bigram. The oracle results are shown in Table 5. We can see these are significantly better than the automatic systems.

From Table 5, we notice that ICSI's ILP performs marginally better than our proposed ILP. We hypothesize that one reason may be that many bigrams in the summary reference only appear once. Table 6 shows the frequency of the bigrams in the summary. Indeed 85% of bigram only appear once

ILP System	ROUGE-2
Our ILP	0.2124
ICSI ILP	0.2128

Table 5: Oracle experiment: using bigrams and their frequencies in the reference summary as weights.

and no bigrams appear more than 9 times. For the majority of the bigrams, our method and the ICSI ILP are the same. For the others, our system has slight disadvantage when using the reference term frequency. We expect the high term frequency may need to be properly smoothed/normalized.

Freq	1	2	3	4	5	6	7	8	9
Ave#	277	32	7.5	3.2	1.1	0.3	0.1	0.1	0.04

Table 6: Average number of bigrams for each term frequency in one topic's reference summary.

We also treat the oracle results as the gold standard for extractive summarization and compared how the two automatic summarization systems differ at the sentence level. This is different from the results in Table 1, which are the ROUGE results comparing to human written abstractive summaries at the n-gram level. We found that among the 188 sentences in this gold standard, our system hits 31 and ICSI only has 23. This again shows that our system has better performance, not just at the word level based on ROUGE measures, but also at the sentence level. There are on average 3 different sentences per topic between these two results.

In the second experiment, after we obtain the estimated $N_{b,ref}$ for every bigram in the selected sentences from our regression model, we only keep those bigrams that are in the reference summary, and use the estimated weights for both ILP modules. Table 7 shows the results. We can consider these as the upper bound the system can achieve if we use the automatically estimated weights for the correct bigrams. In this experiment ICSI ILP's performance still performs better than ours. This might be attributed to the fact there is less noise (all the bigrams are the correct ones) and thus the ICSI ILP system performs well. We can see that these results are worse than the previous oracle experiments, but are better than using the automatically generated bigrams, again showing the bigram and weight estimation is critical for

summarization.

#	Weight	ILP	ROUGE-2
1	Estimated value	Proposed	0.1888
2		ICSI	0.1942

Table 7: Summarization results when using the estimated weights and only keeping the bigrams that are in the reference summary.

4.5 Effect of Training Set

Since our method uses supervised learning, we conduct the experiment to show the impact of training size. In TAC's data, each topic has two sets of documents. For set A, the task is a standard summarization, and there are 4 reference summaries, each 100 words long; for set B, it is an update summarization task – the summary includes information not mentioned in the summary from set A. There are also 4 reference summaries, with 400 words in total. Table 8 shows the results on 2009 data when using the data from different years and different sets for training. We notice that when the training data only contains set A, the performance is always better than using set B or the combined set A and B. This is not surprising because of the different task definition. Therefore, for the rest of the study on data size impact, we only use data set A from the TAC data and the DUC data as the training set. In total there are about 233 topics from the two years' DUC data (06, 07) and three years' TAC data (08, 10, 11). We incrementally add 20 topics every time (from DUC06 to TAC11) and plot the learning curve, as shown in Fig 3. As expected, more training data results in better performance.

Training Set	# Topics	ROUGE-2
08 Corpus (A)	48	0.1192
08 Corpus(B)	48	0.1178
08 Corpus (A+B)	96	0.1188
10 Corpus (A)	46	0.1174
10 Corpus (B)	46	0.1167
10 Corpus (A+B)	92	0.1170
11 Corpus (A)	44	0.1157
11 Corpus (B)	44	0.1130
11 Corpus (A+B)	88	0.1140

Table 8: Summarization performance when using different training corpora.

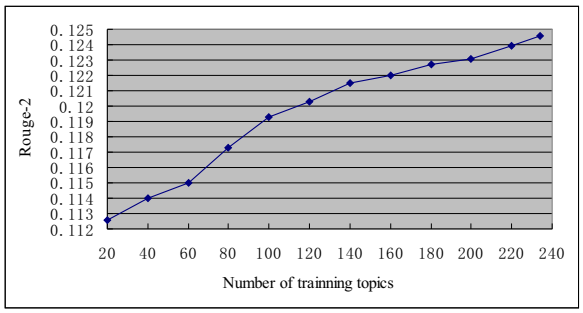

Figure 3: Learning curve

4.6 Summary of Analysis

The previous experiments have shown the impact of the three factors: the quality of the bigrams themselves, the weights used for these bigrams, and the ILP module. We found that the bigrams and their weights are critical for both the ILP setups. However, there is negligible difference between the two ILP methods.

An important part of our system is the supervised method for bigram and weight estimation. We have already seen for the previous ILP method, when using our bigrams together with the weights, better performance can be achieved. Therefore we ask the question whether this is simply because we use supervised learning, or whether our proposed regression model is the key. To answer this, we trained a simple supervised binary classifier for bigram prediction (positive means that a bigram appears in the summary) using the same set of features as used in our bigram weight estimation module, and then used their document frequency in the ICSI ILP system. The result for this method is 0.1128 on the TAC 2009 data. This is much lower than our result. We originally expected that using the supervised method may outperform the unsupervised bigram selection which only uses term frequency information. Further experiments are needed to investigate this. From this we can see that it is not just the supervised methods or using annotated data that yields the overall improved system performance, but rather our proposed regression setup for bigrams is the main reason.

5 Related Work

We briefly describe some prior work on summarization in this section. Unsupervised methods have been widely used. In particular, recently several optimization approaches have demonstrated

competitive performance for extractive summarization task. Maximum marginal relevance (MMR) (Carbonell and Goldstein, 1998) uses a greedy algorithm to find summary sentences. (McDonald, 2007) improved the MMR algorithm to dynamic programming. They used a modified objective function in order to consider whether the selected sentence is globally optimal. Sentence-level ILP was also first introduced in (McDonald, 2007), but (Gillick and Favre, 2009) revised it to concept-based ILP. (Woodsend and Lapata, 2012) utilized ILP to jointly optimize different aspects including content selection, surface realization, and rewrite rules in summarization. (Galanis et al., 2012) uses ILP to jointly maximize the importance of the sentences and their diversity in the summary. (Berg-Kirkpatrick et al., 2011) applied a similar idea to conduct the sentence compression and extraction for multiple document summarization. (Jin et al., 2010) made a comparative study on sentence/concept selection and pairwise and list ranking algorithms, and concluded ILP performed better than MMR and the diversity penalty strategy in sentence/concept selection. Other global optimization methods include submodularity (Lin and Bilmes, 2010) and graph-based approaches (Erkan and Radev, 2004; Leskovec et al., 2005; Mihalcea and Tarau, 2004). Various unsupervised probabilistic topic models have also been investigated for summarization and shown promising. For example, (Celikyilmaz and Hakkani-Tür, 2011) used it to model the hidden abstract concepts across documents as well as the correlation between these concepts to generate topically coherent and non-redundant summaries. (Darling and Song, 2011) applied it to separate the semantically important words from the low-content function words.

In contrast to these unsupervised approaches, there are also various efforts on supervised learning for summarization where a model is trained to predict whether a sentence is in the summary or not. Different features and classifiers have been explored for this task, such as Bayesian method (Kupiec et al., 1995), maximum entropy (Osborne, 2002), CRF (Galley, 2006), and recently reinforcement learning (Ryang and Abekawa, 2012). (Aker et al., 2010) used discriminative reranking on multiple candidates generated by A* search. Recently, research has also been performed to address some issues in the supervised setup, such as the class data imbalance problem (Xie and Liu, 2010).

In this paper, we propose to incorporate the supervised method into the concept-based ILP framework. Unlike previous work using sentence-based supervised learning, we use a regression model to estimate the bigrams and their weights, and use these to guide sentence selection. Compared to the direct sentence-based classification or regression methods mentioned above, our method has an advantage. When abstractive summaries are given, one needs to use that information to automatically generate reference labels (a sentence is in the summary or not) for extractive summarization. Most researchers have used the similarity between a sentence in the document and the abstractive summary for labeling. This is not a perfect process. In our method, we do not need to generate this extra label for model training since ours is based on bigrams – it is straightforward to obtain the reference frequency for bigrams by simply looking at the reference summary. We expect our approach also paves an easy way for future automatic abstractive summarization. One previous study that is most related to ours is (Conroy et al., 2011), which utilized a Naive Bayes classifier to predict the probability of a bigram, and applied ILP for the final sentence selection. They used more features than ours, whereas we use a discriminatively trained regression model and a modified ILP framework. Our proposed method performs better than their reported results in TAC 2011 data. Another study closely related to ours is (Davis et al., 2012), which leveraged Latent Semantic Analysis (LSA) to produce term weights and selected summary sentences by computing an approximate solution to the Budgeted Maximal Coverage problem.

6 Conclusion and Future Work

In this paper, we leverage the ILP method as a core component in our summarization system. Different from the previous ILP summarization approach, we propose a supervised learning method (a discriminatively trained regression model) to determine the importance of the bigrams fed to the ILP module. In addition, we revise the ILP to maximize the bigram gain (which is expected to be highly correlated with ROUGE-2 scores) rather than the concept/bigram coverage. Our proposed method yielded better results than the previous state-of-the-art ILP system on different TAC data

sets. From a series of experiments, we found that there is little difference between the two ILP modules, and that the improved system performance is attributed to the fact that our proposed supervised bigram estimation module can successfully gather the important bigram and assign them appropriate weights. There are several directions that warrant further research. We plan to consider the context of bigrams to better predict whether a bigram is in the reference summary. We will also investigate the relationship between concepts and sentences, which may help move towards abstractive summarization.

Acknowledgments

This work is partly supported by DARPA under Contract No. HR0011-12-C-0016 and FA8750-13-2-0041, and NSF IIS-0845484. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or NSF.

References

Ahmet Aker and Robert Gaizauskas. 2009. Summary generation for toponym-referenced images using object type language models. In *Proceedings of the International Conference RANLP*.

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

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

Ronald Brandow, Karl Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Inf. Process. Manage.*

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the SIGIR*.

Asli Celikyilmaz and Dilek Hakkani-Tür. 2011. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of the ACL*.

John M. Conroy, Judith D. Schlesinger, Jeff Kubina, Peter A. Rankel, and Dianne P. O'Leary. 2011. Classy 2011 at tac: Guided and multi-lingual summaries and evaluation metrics. In *Proceedings of the TAC*.

William M. Darling and Fei Song. 2011. Probabilistic document modeling for syntax removal in text summarization. In *Proceedings of the ACL*.

Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. 2012. Occams - an optimal combinatorial covering algorithm for multi-document summarization. In *Proceedings of the ICDM*.

H. P. Edmundson. 1969. New methods in automatic extracting. *J. ACM*.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*

Dimitrios Galanis, Gerasimos Lampouras, and Ion Androutsopoulos. 2012. Extractive multi-document summarization with integer linear programming and support vector regression. In *Proceedings of the COLING*.

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

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing on NAACL*.

Dan Gillick, Benoit Favre, and Dilek Hakkani-Tür. 2008. In *The ICSI Summarization System at TAC 2008*.

Feng Jin, Minlie Huang, and Xiaoyan Zhu. 2010. A comparative study on ranking and selection strategies for multi-document summarization. In *Proceedings of the COLING*.

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

Jure Leskovec, Natasa Milic-Frayling, and Marko Grobelnik. 2005. Impact of linguistic analysis on the semantic graph coverage and learning of document extracts. In *Proceedings of the AAAI*.

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

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

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the European conference on IR research*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the EMNLP*.

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

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

Seonggi Ryang and Takeshi Abekawa. 2012. Framework of automatic text summarization using reinforcement learning. In *Proceedings of the EMNLP*.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: training log-linear models on unlabeled data. In *Proceedings of the ACL*.

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

Shasha Xie and Yang Liu. 2010. Improving supervised learning for meeting summarization using sampling and regression. *Comput. Speech Lang.*