PKUTM participation in TAC2011

Huiying Li
Institute of Computer Science and Technology,
Peking University
lihuiying@pku.edu.cn

Yue Hu
Institute of Computer Science and Technology,
Peking University
ayuejuice@gmail.com

Zeyuan Li *
Beijing University of Posts and Telecommunications
zyli5313@gmail.com

Xiaojun Wan
Institute of Computer Science and Technology,
Peking University
wanxiaojun@pku.edu.cn

Jianguo Xiao
Institute of Computer Science and Technology,
Peking University
xiaojianguo@pku.edu.cn

Abstract

In this paper, we present details of the participation of PKUTM in the Guided Summarization and AESOP tracks at TAC 2011. For the Guided Summarization task, we develop two extractive summarization systems based on manifold ranking and Integer Linear Programming (ILP) respectively. In our first system, we score the sentences by linearly combining manifold ranking scores and scores based on other surface features. In our second system, we extend the traditional ILP approach to Tolerated ILP (T-ILP), where certain degree of redundancy is tolerated in the optimization process. We also introduce Wikipedia as domain knowledge into the concept weighting step of the T-ILP framework. For post-processing, we revise our sentence ordering scheme proposed in our last year's participation. In the AESOP track, we submit for the no model task. We combine the Rouge scores by training regression models on the TAC 2010's summarization evaluation data and apply the regression model to predict the TAC 2011's system performance.

Part I Guided Summarization Track

1 Introduction

Document Understanding Conference (DUC) and later Text Analysis Conference (TAC) have organized various summarization tracks by providing benchmark datasets and conducting automatic and manual evaluation. The TAC 2010 Guided Summarization Task aims to encourage summarization systems to make a deeper linguistic analysis of the source documents to generate short fluent multi-document summaries. For a given topic, all the documents are separated into two document sets Set A and Set B. Systems are required to generate an initial summary for documents in Set A, and a update summary for documents in Set B with the assumption that the documents in Set A have been read.

In this year PKUTM participates in the Guided Summarization Task and submitted two runs generated by two different extractive systems. For both systems we follow the preprocess, sentence selection, postprocess pipeline. In the sentence selection step, we use a manifold-ranking based approach and an Integer Linear Programming (ILP) base approach respectively.

Our first system, employing manifold-ranking for sentence scoring and selection, is developed on the basis of PKUTM's 2010 summarization system. This year we introduce several additional features and combine them with the manifold ranking score to produce the final sentence saliency score. We also experiment with more sentence selection strategies in this year's system. Moreover, we use HKLID as probability distribution measure between initial and update language model for update summary generation.

The second system is based on the ILP framework, which achieves good results in previous summarization tracks. We enhance the ILP model by

*This work was performed when the third author was an intern at Institute of Computer Science and Technology of Peking University proposing a variant of ILP: Tolerated ILP. Tolerated ILP differs from original ILP in that it allows concepts to be selected multiple times into the summary. We also automatically introduce Wikipedia articles as domain knowledge to assist concept weighting in our systems. The rest of the Guided Summarization part of this paper is organized as follows. We introduce related work in section 2. Detailed description of our system implementation is in section 3. We show the experiment results in section 4 and conclude our work in section 5.

2 Related Work

Extractive summarization treats summarization as a sentence ranking problem and involves assigning salience scores to some unit (in our case sentence is the selection unit) of the documents. Manifold ranking is a sentence ranking approach based on semi-supervised learning. The manifold ranking process naturally makes full use of both the relationships among all the sentences in the documents and the relationships between the given topic and the sentences. The ranking score is obtained for each sentence in the manifold-ranking process to denote the biased information richness of the sentence (Wan et al., 2007). Extended version of manifold-ranking include Manifold Ranking with Sink Points(MRSP) and Topic Guided Manifold Ranking with Sink Points (TMSP)(Du et al., 2010). MRSP and it topic focused version TMSP all based on the enhanced model that sink points are introduced into the ranking graph, where a sink point does not spread any ranking score to its neighbors.

Recent years, Integer Linear Programming has been introduced for summarization sentence selection (Gillick et al., 2009; Takamura and Okumura, 2009). The ILP approach model sentence selection as the well-known set-cover problem, where a summary is the set of sentences that best covers the relevant concepts in the document set(Gillick et al., 2009). The ILP has been used in TAC summarization tracks and achieved fairly good results. Much work has been done in exploring concept representation and weighting within the ILP framework. Concepts are usually represented by low order ngrams and named entities. Concept weighting scheme includes simple metrics such as document frequency(Gillick et al., 2008), term frequency and tf*idf, as well as complicated approaches involving in-depth analysis like Labeled LDA and RankNet scoring(Jin et al., 2010).

3 Our Approach

The system architecture is shown in Figure 1. Following the preprocessing, sentence selection, post-processing pipeline we develop two extractive summarization systems. In the sentence selection part, we use the manifold-ranking based approach and the Integer Linear Programming (ILP) base approach for two systems respectively.

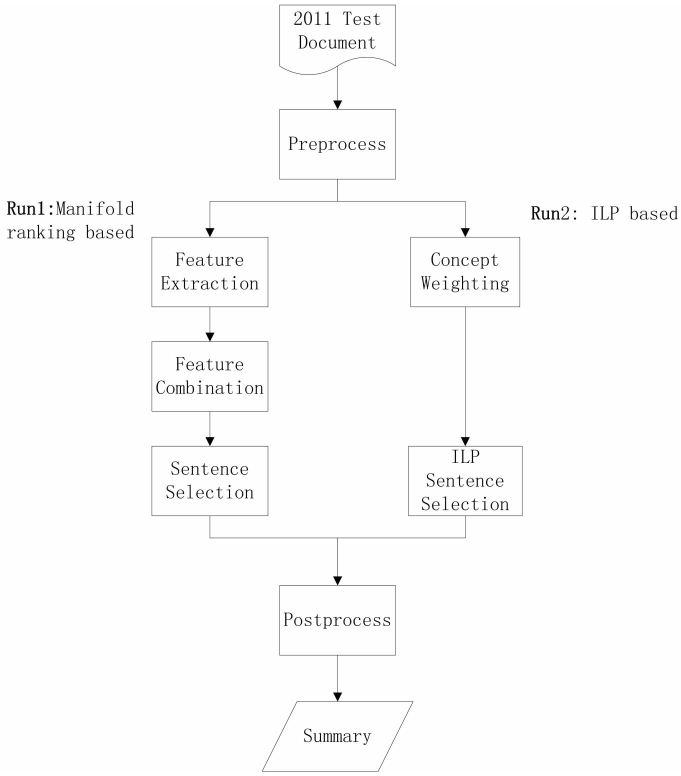

Figure 1: System Overview

3.1 Preprocessing

3.1.1 Sentence Segmentation

Sentence segmentation has been an important procedure of processing documents. Bad segmentation will not only lead to incorrect judgment of the information coverage of the sentence but also greatly harm the readability of the resulted summary.

In previous Guided Summarization track TAC requires the participants to develop their own sentence segmentation module. This year TAC provides clean documents which contain segmented sentences with paragraph structure maintained. We make use of the clean documents and remove the paragraph tags to obtain the set of sentences and feed them to the next step.

3.1.2 Named entity recognition

Named entities have been proven to be strong indicators of important information. We perform the NER step to extract named entities as concepts and also provide clues for sentence selection.

We use Stanford NER tool to extract the three types of named entities: person, location and organization. Besides, we also use regex expression to match and annotate time as a kind of named entity in this step.

3.1.3 Data cleaning

In this step we concern the integrity and appropriateness of sentences and filter out those that we consider to be inept. We apply the following criteria to filter sentences.

- Unclear reference. Following (Gillick et. al, 2009), we eliminate sentences including unclear references that do not make sense in isolation. We use the LingPipe coreference resolution tool to identify unclear references in sentences: sentences with unresolved output by LingPipe are marked as containing unclear references.

- said clause. We check if a sentence contains said clause by going through the text and see if a said, says, told, tells word and quotation marks appear simultaneously.

- Punctuation mismatch. We check the completeness of paired punctuations such as quotation marks and parentheses.

- Bad sentence boundary. We check whether each sentence ends up with end sentence punctuations.

- Question sentences and exclamatory sentences. We achieve this by simply checking if each sentence ends up with question mark and exclamation mark.

- Short sentence. We eliminate sentences containing no more than three words.

Sentences with the above conditions are marked as sentences we considered inappropriate to be included in the final summary. For the ILP based system, we delete the substandard sentences right before the ILP sentence selection procedure. In the manifold-ranking based framework, however, we retain the sentences until the final sentence selection step, i.e. these sentences will be involved in the manifold ranking sentence scoring process. We handle the inept sentences in this fashion because we think these sentences also contribute to the content of the topic, but just improperly formed for summary sentences.

3.2 Manifold-ranking based approach

This year we employ manifold ranking (Wan et al., 2007) algorithm based approach for our first submission. Additional to our previous version at TAC2010 Guided Summarization Track, we revise our system by combining several other surface features and using a greedy sentence selection scheme.

3.2.1 Manifold-ranking algorithm

The manifold-ranking method is a universal ranking algorithm and it is initially used to rank data points along their underlying manifold structure. In this summarization system we apply manifold-ranking of sentences to topic-focused multi-document summarization. The manifold-ranking based summarization approach consists of two steps: (1) the manifold-ranking score is computed for each sentence in the manifold-ranking process where the score denotes the biased information richness of a sentence; (2) based on the manifold-ranking scores, the diversity penalty is imposed on each sentence and the overall ranking score of each sentence is obtained to reflect both the biased information richness and the information novelty of the sentence (Jia et al., 2010). The details of manifold ranking algorithm can be found in (Wan et al., 2007).

3.2.2 Sentence scoring

We calculate the score of each sentence by combining the following features extracted from the document set.

- Manifold-raking score. The manifold ranking score is obtained by applying the manifold ranking process to sentences. We made changes to the original algorithm proposed in (Wan et al., 2007) and the detailed implementation can be found in our report (Jia et al., 2010) at TAC2010.
- Sentence position.
- Sum of tf*idf weights. We calculate the tf*idf scores of words in the sentence and sum them up.
- Named entity coverage. For each sentence, we count the number of different name entities and divide it by the total number of name entities contained in the document.

To assign each sentence a saliency score we manually combine the four features with weights we acquire empirically from experiments.

3.2.3 Sentence selection

In our last year participation, we select the sentence iteratively and at each iteration we select sentences with highest score after diversity penalty is imposed. This year we have experimented with several selection policies: (i) ILP selection on sentence level, which reduce to basic 0-1 backpack problem in this case, (ii) greedy strategy to choose sentence with highest score at each iteration, and (iii) greedy strategy to select the highest $\frac{score}{sentencelength}$ ratio each iteration. We choose the second approach as our final sentence selection algorithm.

3.2.4 Updated guided summarization

The update guided summarization requires system to provide summaries of a second document set of the topic under the assumption that user has already read the documents in the first document set. The systems should be able to recognize new information in the second document set and avoid content overlap with previous documents. In this system, we employ Hybrid Kullback-Leibler Information Divergence (HKLID) to measure the content difference between the first and second document set. Sentences are scored according to Equation 1

$$s_{update} = s_{initial} - \lambda \cdot \overline{HKLID}(LM_{initial}, LM_{update}) \quad (1)$$

where $s_{initial}$ denotes the score obtained by regardless of the update requirement, as described in the previous sections; $\overline{HKLID}(LM_{initial}, LM_{update})$ denotes the average HKLID between the update sentence and each sentence in the initial summary generated in previous steps; λ is used to adjust the weight of HKLID. HKLID has been adopted by (Varma and et al., 2010) to model the difference of language models in their participation in TAC2010.

3.3 Tolerated Integer Linear Programming (T-ILP) based Approach

We adopt the ILP framework for our second extractive summarization system. We modified the original ILP approach to tolerate multiple presences of concepts in the summary. We also make use of Wikipedia articles as domain knowledge to improve concept weighting.

3.3.1 Tolerated ILP

The ILP approach for summarization proposed by (Gillick et al., 2008) addresses the sentence selection process as a global optimization problem. In the ILP model, each sentence is considered to consist of a set of concepts, where each concept presents an information unit and is assigned with a weight. The ILP approach aims to maximize the number of concepts covered by a selection of sentences (Gillick et al., 2008). The formulation of ILP is presented as follows:

$$\begin{aligned} maximize & \\ & \sum_i w_i \cdot z_i^c \\ s.t. & \\ & \sum_j z_j^s \cdot |s_j| \leq L \\ & \sum_j z_j^s \cdot I(i,j) \geq z_j^c, \forall i \\ & z_i^c, z_j^s \in \{0,1\}, \forall i,j \end{aligned} \quad (2)$$

where

w_i	the weight of a concept
s_j	sentence to be selected in the document set
$\|s_j\|$	number of words in s_j
z_i^c	the variable that indicates whether concept i is included in the summary
z_j^s	the variable that indicates whether sentence j is included in the summary
$I(i,j)$	the indicator variable that indicate whether concept i appears in sentence j
L	the length limit of the output summary

The length limit of 100 words is addressed by the first constraint. And the second constraint indicates that if a concept appears in summary, at least one sentence containing the concept should be selected. Redundancy is avoided by considering each concept only once in the object function.

Considering each concept only once naturally maximizes the diversity of concepts included in the output. Good summaries, however, may focus on a topic coherently and therefore a concept may appear more than one time in the text. The original ILP model, which gives second appearance of a concept zero weight, could optimize the information coverage but at the same time harm the consistency and coherency of the output summary. Based on this observation we extend the original ILP approach to allow a concept to be counted multiple times with weight while attend to redundancy. We achieve this by modifying the object function and constraints of the ILP formulation to:

$$\begin{aligned} & maximize \\ & \qquad \sum_i w_i(k) \cdot z_{i,k}^c \\ & s.t. \\ & \qquad \sum_j z_j^s \cdot |s_j| \leq L \\ & \qquad \sum_j z_j^s \cdot C(i,j) \geq \sum_k z_{i,k}^c, \forall i \\ & \qquad z_{i,k}^c, z_j^s \in \{0,1\}, \forall i,j \end{aligned} \tag{3}$$

where $w_i(k)$ is the weight function denotes the weight of concept c_i's k^{th} appearance in the summary, and $z_{i,k}^c$ is the indicator variable that denotes concept c_i's k^{th} appearance. The indicator function Ii,j in the original ILP is set to $C(i,j)$, which indicates the number of presences of concept c_i in sentence s_j, correspondently.

We introduce $w_i(k)$ as the weight for c_i's k^{th} presence respectively to avoid the redundancy, by decreasing $w_i(k)$ as k increases, i.e. later presence deserves lower weight. We simply set the weight function as exponential decline as in Equation 4

$$w_i(k) = \alpha^{k-1} \tag{4}$$

where $\alpha < 1$ is the penalty factor for multiple presence.

In our implementation, we designate the concept set to be the union set of unigrams and bigrams in the document set together with named entities we extracted in previous step. We removed stopwords and did stemming before putting unigrams and bigrams to the concept pool. The simple term frequency is chosen as the primary concept weighting scheme. The penalty factor α is set empirically in our experiment. We solve the optimization problem using the IBM CPLEX optimizer. Note that without any restrictions, k could be arbitrarily large and $w_i(k)$ is fed to CPLEX in a piecewise function manner. In this fashion the optimizing process takes long hour to finish and for the efficiency consideration we simply restrict that $k \in 1, 2, 3$, *i.e.* we tolerate up to three times of each concept's presences.

Moreover, special attention should be addressed on replicated content in the Tolerated ILP case. Since Tolerated ILP favors concepts with high weights, if (i) certain concepts within a sentence is assigned with very high weight that its $w_i(k) > 1$ still weights more than other concepts in the concept pool, and (ii) the high-weight concepts appear in several sentences paraphrasing or even identical to each other, these sentences will be all selected into the output summary by T-ILP due to its tolerance to multiple presences of concepts. To prevent this erroneous result, we adopt Longest Common Sequence (LCS) ratio measure sentence content overlap. LCS ratio is defined as the length of the longest common sequence of two sentence divided by the shorter length of the two sentence, where the unit of comparison is a word. To incorporate this restriction into the optimizing framework, we added an additional constraint to the T-ILP formulation:

$$
\begin{aligned}
s.t.\quad & \sum_j z^s_j \cdot |s_j| \le L \\
& \sum_j z^s_j \cdot C(i,j) \ge \sum_k z^c_{i,k}, \forall i \\
& z^s_m + z^s_n \le 1, \forall LCSR(s_m, s_n) > \delta \\
& z^c_{i,k}, z^s_j \in \{0,1\}, \forall i,j
\end{aligned} \tag{5}
$$

where z^s_m,z^s_n are indicator variables for sentences s_m, s_n's presences in summary. The third constraint implies that for any two sentences s_m, s_n whose Longest Common Sequence Ratio (LCSR) is over a threshold δ, only one of them can be selected into the summary. In our experiment we set the threshold δ empirically.

3.3.2 Wikipedia domain knowledge

Wikipedia is a rich knowledge resource for various linguistic computing tasks for its abundant entries containing high quality information about real world entities or events. Wikipedia articles are usually written in introductive and synthesized manner that the articles themselves can be viewed as summaries. It is natural that summarization system may borrow some concepts from corresponding Wikipedia article of the topic.

Varma and et al. (2010) have introduced Wikipedia into their system to build domain model. Their work, however, involves manual selection of Wikipedia articles. We explore the possibility of search for and disambiguate Wikipedia articles automatically.

In our experiment we mine the Wikipedia in the following steps:

1. We fed the topic title into the Wikipedia search engine and downloaded the the top ten returned wikipages;

2. The downloaded HTML pages are cleaned to plain texts;

3. We employed standard cosine similarity measure between the whole docset A and the cleaned wiki-articles. The article with highest similarity is assigned to the topic as domain knowledge. After this step, each topic is assigned with zero or one Wikipedia article;

4. We would like to discover the important concepts in the domain knowledge: compute term weights in Wikipedia articles by standard tf*idf scheme and retrieve terms whose weights are higher than a threshold after max normalization. We call the set of important terms domain dictionary. The threshold is tuned empirically in our experiment;

5. For each concept in test documents' concept pool, if the concept is also contained in domain dictionary, we reweight the concept to graphic, where-graphic is a reward factor we set empirically in our experiment.

3.3.3 Update Guided Summarization

In the Tolerated ILP frame-work, diversity and content overlap preclusion can be achieved by imposing penalty on concepts appearing in the first document set. In practice we simply reweight each concept by Equation 6.

$$
w_i(k)_{update} = \gamma \cdot w_i(k) \tag{6}
$$

3.4 Postprocessing

We reorder the selected sentences in this step. We slightly modify our sentence ordering algorithm in last year's participation (Jia et. al, 2010). We take the number of named entities in sentences into account in this year's sentence ordering scheme as showed in Equation 7.

$$
I(s_i, s_j, d) = \begin{cases} 2 \times sign(id(m_{i,d}) - id(m_{j,d})), \\ if(s_i \in d \vee s_j \in d) \wedge sim_{i,d} > \tau \wedge sim_{j,d} > \tau \\ sign(id(m_{i,d}) - id(m_{j,d})), \\ if(s_i \notin d \wedge s_j \notin d) \wedge sim_{i,d} > \tau \wedge sim_{j,d} > \tau \\ 0, otherwise \end{cases} \tag{7}
$$

$$
sign(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \end{cases} \tag{8}
$$

$$
TextOrder(s_i, s_j) = sign(\sum_d I(s_i, s_j, d)) \tag{9}
$$

Additional to our previous formula presented above, we incorporate the number of named entities in the following fashion to arrive at the final order score as in Equation 10

$$FinalOrder(s_i, s_j) = \begin{cases} 2sign(2TextOrder \\ +sign(\#NE(s_j) - \#NE(s_i))), \\ \qquad if(s_i \in d \vee s_j \in d) \\ sign(TextOrder+ \\ sign(\#NE(s_j) - \#NE(s_i))), \\ \qquad otherwise \end{cases} \tag{10}$$

4 Experiment Results

NIST assessors wrote 4 model summaries for each document set. All submitted systems are evaluated both automatically and manually, including ROUGE-2, ROUGE-SU4, Pyramid, Linguistic Quality and Overall Responsiveness. We submit two runs for our systems employing the two approaches described in the previous section respectively. Table 1 and Table 2 show our performance of our systems in initial and update summaries. In the tables 'LQ' denotes Linguistic Quality, 'Pyr' denotes Pyramid and 'Overall' denotes Overall Responsiveness.

	R-2	R-SU4	Pyr	LQ	Overall
Model	0.115	0.154	0.782	4.869	4.818
PKUTM1	0.102	0.141	0.418	3.136	2.977
PKUTM2	0.115	0.150	0.477	3.432	3.136
Best	0.134	0.165	0.477	3.75	3.159

Table 1: Initial Summarization Results

	R-2	R-SU4	Pyr	LQ	Overall
Model	0.0961	0.134	0.661	4.898	4.669
PKUTM1	0.0709	0.114	0.264	3.023	2.432
PKUTM2	0.0816	0.119	0.313	3.273	2.477
Best	0.0959	0.131	0.353	3.818	2.591

Table 2: Update Summarization Results

There are 25 teams and totally 50 runs submitted in the Guided Summarization Track for both initial and update tasks. Our system PKUTM2 achieves 2nd place in the initial summarization and 11th in the update summarization while PKUTM1 achieves 13th and 15th in initial and update summarization respectively, according to the overall responsiveness metric. PKUTM2 achieves 1st place in initial summarization by Pyramid metric. Generally speaking, PKUTM2, which employs the Tolerated-ILP approach, is superior to the manifold ranking based PKUTM1.

5 Conclusion

The two systems are PKUTM's second participation in TAC Guided Summarization Track. We carried on our previous system based on manifold ranking and further developed an improved version of the ILP-based approach, namely Tolerated-ILP. From the evaluation results we see our T-ILP based system reaches good results among all teams.

Part II Automatically Evaluating Summaries of Peers(AESOP)

1 Introduction

In the AESOP track participants are required to evaluate the quality of machine generated summaries automatically. The AESOP track includes *all peer* task and *no model* task, where the former task aims at discriminating between system summaries and human written model summaries while the latter task aims at evaluating the qualities of all system summaries. In our participation in the AESOP, we submitted our results to *no model* task.

2 Related Work

Lots of work has been done in automatic evaluation of machine generated summaries. Lin (2004) proposed ROUGE metrics, a method that evaluates machine generated summaries by counting their overlap with manual model summaries. ROUGEs are adopted by NIST as the automatic evaluation metrics for summarization track and has become the de facto standard automatic summary evaluation metrics for summarization systems to compare their results when human evaluation is not available. Developed after ROUGEs, Basic Elements (BE) chunks the sentences into basic information elements (Basic Elements) as the unit of overlap comparison and counting (Hovy et al., 2006). Conroy and Dang (2008) later proposed ROSE, a automatic content evaluation model that combines multiple ROUGE scores using canonical correlation. ROSE has been used in the previous TAC AESOP tracks and achieved competitive results.

3 Our Approach

Since this is our first trial to this track, we simply follow the ROSE approaches and submitted only for no model task. We use TAC 2010 Guided Summarization data as training set to obtain a combination of ROUGE scores that maximize the correlation with human responsiveness. For implementation, we use MATLAB to solve the regression problem. We use robustfir(), lsqnonneg() and canoncorr() functions and regress to overall responsiveness and pyramid scores. The resulted regression model is then used to predict the scores of peer summaries of TAC 2011 Guided Summarization track.

4 Experiment Results

We submitted four runs to the no model task. Our results as well the best system results by pearson's correlation metric for both initial and update summarizations are shown in Table 3 and Table 4.

	Pyramid	Readability	Responsiveness
PKUTM1	0.968	0.767	0.936
PKUTM2	0.947	0.762	0.916
PKUTM3	0.962	0.755	0.943
PKUTM4	0.962	0.767	0.930
Best	0.981	0.819	0.954

Table 3: Initial Summarization Results

	Pyramid	Readability	Responsiveness
PKUTM1	0.696	0.380	0.672
PKUTM2	0.820	0.529	0.819
PKUTM3	0.904	0.662	0.919
PKUTM4	0.794	0.485	0.782
Best	0.911	0.742	0.927

Table 4: Update Summarization Results

5 Conclustion

This is our first trial in TAC AESOP track. We follow the ROSE metrics and combined the ROUGE scores by regression and generate results correlate with human evaluations.

Acknowledgments

This work was supported by NSFC (60873155), Beijing Nova Program (2008B03) and NCET (NCET-08-0006).

References

John M. Conroy and Hoa Trang Dang. 2008. Mind the gap: Dangers of divorcing evaluations of summary content from linguistic quality. In *proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*.

Pan Du, Jin Zhang, Jiafeng Guo, and Xueqi Cheng. 2010. Tmsp: Topic guided manifold ranking with sink points for guided summarization. In *Proceedings of Text Analysis Conference 2010*.

Dan Gillick, Benoit Favre, and et al. 2008. The icsi summarization system at tac 2008. In *Proceedings of Text Analysis Conference 2008*.

Dan Gillick, Benoit Favre, and et al. 2009. The icsi/utd summarization system at tac 2009. In *Proceedings of Text Analysis Conference 2009*.

Eduard Hovy, Chin-Yew Lin, L. Zhou, and Junichi Fukumoto. 2006. Automated summarization evaluation with basic elements. In *Proceedings of the International Conference on Language Resource and Evaluation (LREC 2006)*.

Houping Jia, Xiaojiang Huang, Tengfei Ma, Xiaojun Wan, and Jianguo Xiao. 2010. Pkutm participation at tac 2010 rte and summarization track. In *Proceedings of Text Analysis Conference 2010*.

Feng Jin, Minlie Huang, and Xiaoyan Zhu. 2010. The thu summarization systems at text analysis conference 2010. In *Proceedings of Text Analysis Conference 2010*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of sumaries. In *Proceedings of Workshop on Text Summarization of ACL 2004*.

Hiroya Takamura and Manabu Okumura. 2009. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of Proceedings of EACL 2009*.

Vasudeva Varma and et al. 2010. Iiit hyderbad in guided summarization and knowledge base population. In *Proceedings of Text Analysis Conference 2010*.

X. Wan, J. Yang, and J. Xiao. 2007. Manifold-ranking based topic-focused muilti-document summarization. In *Proceedings of Proceedings of IJCAI 2007*.