Oracle Summaries of Compressive Summarization

Tsutomu Hirao and **Masaaki Nishino** and **Masaaki Nagata**
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
{hirao.tsutomu,nishino.masaaki,nagata.masaaki}@lab.ntt.co.jp

Abstract

This paper derives an Integer Linear Programming (ILP) formulation to obtain an oracle summary of the *compressive summarization* paradigm in terms of ROUGE. The oracle summary is essential to reveal the upper bound performance of the paradigm. Experimental results on the DUC dataset showed that ROUGE scores of compressive oracles are significantly higher than those of extractive oracles and state-of-the-art summarization systems. These results reveal that compressive summarization is a promising paradigm and encourage us to continue with the research to produce informative summaries.

1 Introduction

Compressive summarization, a joint model integrating sentence extraction and sentence compression within a unified framework, has been attracting attention in recent years (Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Almeida and Martins, 2013; Qian and Liu, 2013; Kikuchi et al., 2014; Yao et al., 2015). Since compressive summarization methods can use a sub-sentence as an atomic unit, they can pack more information into summaries than extractive methods, which employ sentences as atomic units. Thus, compressive summarization is essential when we want to produce summaries under tight length constraints. There are two approaches to compress entire document(s) to be grammatical; one is trimming the phrase structure trees (Berg-Kirkpatrick et al., 2011) and the other is trimming the dependency trees obtained from the document(s) (Martins and Smith, 2009; Almeida and Martins, 2013; Qian and Liu, 2013; Kikuchi et al., 2014; Yao et al., 2015). This paper focuses on the latter approach because recently it has been receiving much attention.

To measure the performance of compressive summarization methods, ROUGE (Lin, 2004), an automatic evaluation metric, is widely used. ROUGE evaluates a system summary by exploiting a set of human-made reference summaries to give a score in the range [0,1]. When n-gram occurrences of the system summary agree with those in a set of reference summaries, the value is 1. However, system summaries cannot achieve ROUGE=1 since summarization systems cannot reproduce reference summaries in most cases. In other words, the maximum ROUGE score that can be achieved by compressive summarization is unclear. As a result, researchers cannot know how much room for further improvement is left. Thus, it is beneficial to reveal the upper bound summary that achieves the maximum ROUGE score and can be produced by the systems. The upper bound summary is known as the oracle summary. To obtain the oracle summary on extractive summarization paradigms, several approaches have been proposed. Sipos et al. (2012) utilized a greedy algorithm, and Kubina et al. (2013) utilized exhaustive search based on heuristics. However, their oracle summaries do not always retain the optimal (maximum) ROUGE score. Recently, Hirao et al. (2017) derived an Integer Linear Programming (ILP) formulation to obtain the optimal oracle summary. Their oracle summary can help researchers to comprehend the strict limitation of the extractive summarization paradigm. However, their method cannot be applied to obtain compressive oracle summaries.

To reveal the ultimate limitation of the compressive summarization paradigm, we propose an ILP formulation to obtain a compressive oracle summary that maximizes the ROUGE score. We con-

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 275–280
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-2043

ducted experimental evaluation on the Document Understanding Conference (DUC) 2004 dataset. The result demonstrated that ROUGE scores of compressive oracle summaries completely outperformed those of extractive oracle summaries and those of state-of-the-art summarization methods. This indicates that compressive summarization is a promising paradigm for leveraging research resources.

2 Definition of Compressive Oracle Summaries

Before defining compressive oracle summary, we briefly describe ROUGE$_n$. Given K reference summaries $\boldsymbol{R}=\{R_1,\dots,R_K\}$ and a system summary S. Let $G=\{g_1^n,\dots,g_M^n\}$ be the set of all n-grams appearing in reference summaries. Let $|G|=M$. ROUGE$_n$ is defined as follows:

$$\text{ROUGE}_n(\boldsymbol{R},S) = \frac{\sum_{k=1}^{K}\sum_{j=1}^{M}\min\{N(g_j^n,R_k),N(g_j^n,S)\}}{\sum_{k=1}^{K}\sum_{j=1}^{M}N(g_j^n,R_k)} \tag{1}$$

g_j^n represents the j-th n-gram appearing in reference summaries. $N(g_j^n,R_k)$ and $N(g_j^n,S)$ are the number of occurrences of n-gram g_j^n in R_k and S, respectively. Thus, compressive oracle summaries are defined as follows:

$$\begin{aligned} O = &\mathop{\arg\max}_{S\subseteq T} \text{ROUGE}_n(\boldsymbol{R},S) \\ &\text{s.t.}\quad \ell(S)\le L_{\max}. \end{aligned} \tag{2}$$

T is the set of all valid word subsequences[1] obtained from sentences contained in the input document(s), and $L_{\max}$ is the length limitation of the oracle summary. $\ell(S)$ indicates the number of words in the summary. Neither approximation nor exact algorithms are known for solving this problem.

3 ILP Formulation to Obtain the Compressive Oracle Summary

3.1 Dependency Structure of a Sentence

In this paper, we follow the dependency tree trimming approach proposed by Filippova et al. (2008; 2013). They proposed rules that transform a tree that represents dependency relation between words into a tree that represents dependency relation between chunks (consisting of a word or word sequence). Since we can trim their dependency trees without loss of grammatical consistency, Thus, we employ the trees in our compressive summarization framework. Figure 1 shows examples.

3.2 ILP Formulation

$$\begin{aligned}
&\text{maximize} && \sum_{k=1}^{K}\sum_{j=1}^{M} z_{k,j} && (3)\\
&\text{s.t.} && \sum_{i=1}^{|D|}\sum_{u=1}^{E_i} \ell_{i,u}b_{i,u} \le L_{\max} && (4)\\
&\forall j: && \sum_{\tau\in\mathcal{T}(g_j^n)} m_\tau \ge z_{k,j} && (5)\\
&\forall j: && N(g_j^n,R_k)\ge z_{k,j} && (6)\\
&\forall i,u: && b_{i,\text{parent}(i,u)} \ge b_{i,u} && (7)\\
&\forall i,v,q\in V_i(w_{i,v}): && b_{i,q}\ge m_{i,v} && (8)\\
&\forall i,v,p\in U_i(w_{i,v}): && b_{i,p}\le 1-m_{i,v} && (9)\\
&\forall i,u: && m_{i,v}\in\{0,1\} && (10)\\
&\forall i,v: && b_{i,u}\in\{0,1\} && (11)\\
&\forall k,j: && z_{k,j}\in\mathbb{Z}_+. && (12)
\end{aligned}$$

Since the denominator of equation (1) is constant for a given set of reference summaries, we can find an oracle summary by maximizing the numerator of equation (1). Equation (3) is the objective function that corresponds to maximization of the numerator of equation (1). $z_{k,j}$ is the count of the j-th n-gram that is contained in both the k-th reference summary and the oracle summary. Equation (4) ensures that the length of the oracle summary is less than $L_{\max}$. $b_{i,u}$ is a binary decision variable indicating whether u-th chunk in i-th sentence is contained in an oracle summary or not. $\ell_{i,u}$ indicates the number of the words in u-th chunk in the i-th sentence. D is a set of sentences and E_i is the number of chunks in the i-th sentence. Equations (5) and (6) represent min operation in equation (1). $w_{i,v}$ is the v-th possible word sequence whose length is n and that is contained in the i-th sentence, and $m_{i,v}$ is a binary decision variable indicating whether $w_{i,v}$ is contained in the oracle summary or not. $\mathcal{T}(g_j^n)$ is a set of tuples consisting of indices (i,v) whose word sequence corresponds to g_j^n, *i.e.*, $\mathcal{T}(g_j^n)=\{(i,v)|w_{i,v}=g_j^n\}$. Thus, $z_{k,j}=\min\{N(g_j^n,R_k),N(g_j^n,S)\}$. Equation (7) ensures that an oracle summary consists

[1] Word subsequences can be regarded as grammatical sentences. We regard rooted subtrees of dependency trees as valid word subsequences. For details, see Section 3.1.

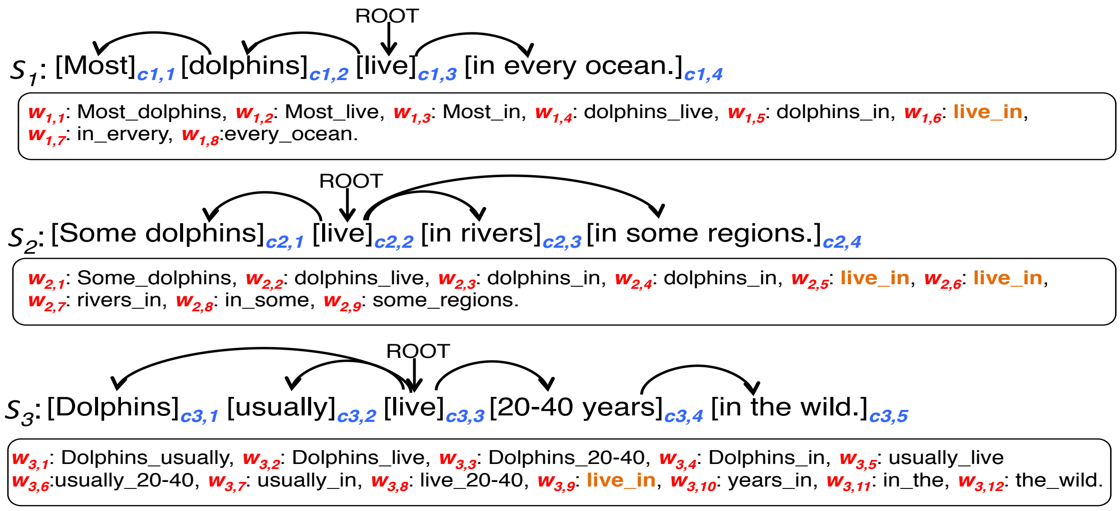
ROOT
S1: [Most]c1,1 [dolphins]c1,2 [live]c1,3 [in every ocean.]c1,4
w1,1: Most_dolphins, w1,2: Most_live, w1,3: Most_in, w1,4: dolphins_live, w1,5: dolphins_in, w1,6: live_in,
w1,7: in_ervery, w1,8:every_ocean.
ROOT
S2: [Some dolphins]c2,1 [live]c2,2 [in rivers]c2,3 [in some regions.]c2,4
w2,1: Some_dolphins, w2,2: dolphins_live, w2,3: dolphins_in, w2,4: dolphins_in, w2,5: live_in, w2,6: live_in,
w2,7: rivers_in, w2,8: in_some, w2,9: some_regions.
ROOT
S3: [Dolphins]c3,1 [usually]c3,2 [live]c3,3 [20-40 years]c3,4 [in the wild.]c3,5
w3,1: Dolphins_usually, w3,2: Dolphins_live, w3,3: Dolphins_20-40, w3,4: Dolphins_in, w3,5: usually_live
w3,6:usually_20-40, w3,7: usually_in, w3,8: live_20-40, w3,9: live_in, w3,10: years_in, w3,11: in_the, w3,12: the_wild.

Figure 1: Examples of trees that represent dependency relations between chunks, and word sequences (whose length is 2). Chunks are enclosed in square brackets. Note that we disregard word sequences that are generated by destroying the structure of chunks such as "live_every" in S_1, "dolphins_in" in S_2, "live_wild" in S_3.

of a set of rooted subtrees of the sentences in the entire document(s). Function parent (i, u) returns the index of the parent chunk of the u-th chunk in the dependency tree obtained from the i-th sentence. Equations (8) and (9) represent the dependency relation between n-grams and chunks. When we include $w_{i,v}$ in the oracle summary, we have to include all chunks that contain the words in $w_{i,v}$. In addition, when the above chunks have gap(s), we have to drop chunk(s) within the gap(s). Here, $V_i(w_{i,v})$ is a set of indices of chunks that includes words in $w_{i,v}$, and $U_i(w_{i,v})$ is a set of indices of chunks within the gap(s), defined as $\{h| \min(V_i(w_{i,v})) < h < \max(V_i(w_{i,v}))\}$ and $h \notin V_i(w_{i,v})$.

We give an example to show how chunks and word sequences are related. When we pack a bigram "live_in" in an oracle summary, there are four candidates in the source document (Fig. 1). Word subsequences, $w_{1,6}$,$w_{2,5}$,$w_{2,6}$ and $w_{3.9}$ match "live_in". Thus, $\mathcal{T}$(live_in) $= \{(1, 6), (2, 5), (2, 6), (3, 9)\}$. Here, when we want to pack $w_{2,6}$ into the oracle summary, we have to pack both chunks $c_{2,2}$ and $c_{2,4}$ ($b_{2,2} = b_{2,4} = 1$) because $U_2(w_{2,6}) = \{2, 4\}$. Then, we have to drop chunk $c_{2,3}$($b_{2,3} = 0$) because $c_{2,3}$ is within the gap between chunks $c_{2,2}$ and $c_{2,4}$ ($V_2(w_{2,6}) = 3$). Similarly, when we pack $w_{3,9}$ into an oracle summary, we have to pack both chunks $c_{3,3}$ and $c_{3,5}$ and drop chunk $c_{3,4}$. However, this compression is not allowed since there is no dependency relationship between $c_{3,3}$ and $c_{3,5}$.

After solving the ILP problem, we can obtain compressive oracle summaries by collecting chunks according to $b_{i,u}$=1.

4 Experiments

To investigate the potential limitation of the compressive summarization paradigm, we compare ROUGE scores of compressive oracle summaries with those of extractive oracle summaries and those obtained from state-of-the-art summarization systems. Extractive oracle summaries are obtained by solving the ILP formulation proposed by (Hirao et al., 2017). System summaries are extracted from a public repository[2] (Hong et al., 2014).

4.1 Settings

We conducted experimental evaluation on the DUC-2004 dataset for multiple document summarization evaluation, a widely used benchmark test set for generic multiple document summarization tasks. The dataset consists of 50 topics, each of which contains 10 newspaper articles. To obtain oracle summaries based on the ILP formulation described in section 3.2, first, we applied the Stanford parser (de Marneffe et al., 2006) to all

[2] http://www.cis.upenn.edu/~nlp/corpora/sumrepo.html

Method		Metric			
		n=1	n=2	n=1+2	Sent.
Ext.	n=1	**42.6**	13.1	24.1	5.34
	n=2	36.6	**16.9**	24.3	5.06
	n=1+2	40.9	16.1	**25.4**	5.24
Comp.	n=1	**50.9**	13.8	27.7	10.6
	n=2	40.9	**21.3**	28.6	7.82
	n=1+2	47.9	19.7	**30.3**	8.48
	RegSum	33.1	10.2	18.8	4.9
	ICSISumm	31.0	10.3	18.0	4.2

Table 1: ROUGE scores and the number of sentences of extractive and compressive oracle summaries and those obtained from state-of-the-art summarization systems, RegSum and ICSISumm. n= 1 corresponds to ROUGE$_1$, n=2 corresponds to ROUGE$_2$, n=1+2 corresponds to ROUGE-SU0. "Sent." indicates the average number of sentences in the summaries.

sentences in the dataset to obtain dependency relations between words, and then we transformed them into trees that represent the dependency relations between chunks by applying Filippova's rules (Filippova and Strube, 2008; Filippova and Altun, 2013). To solve the ILP problem, we utilized `CPLEX version 12.5.1.0`.

We obtained and evaluated oracle summaries based on three variants of ROUGE, ROUGE$_1$, ROUGE$_2$ and ROUGE-SU0, with the following conditions[3]: (1) ROUGE$_1$, utilizing unigrams excluding stopwords (2) ROUGE$_2$, utilizing bigrams with stopwords, and (3) ROUGE-SU0, which is an extension of ROUGE$_n$, utilizing unigram and bigram (excluding skip-bigram) statistics.

4.2 Results and Discussion

Table 1 shows ROUGE scores of compressive and extractive oracle summaries and those of RegSum (Hong and Nenkova, 2014) that achieved the best ROUGE$_1$ and ICSISumm (Gillick and Favre, 2009; Gillick et al., 2009) that achieved the best ROUGE$_2$ on the DUC-2004 dataset, respectively.

We compare ROUGE scores of compressive oracle summaries with extractive oracle summaries. The best scores are obtained when we use the same ROUGE variant for both computation and evaluation (see bolded scores in Table 1). There are large differences between the best scores of extractive method and compressive method. The differences are 8.3 points, 4.4 points and 4.9 points for ROUGE$_1$, ROUGE$_2$, ROUGE-SU0, respectively. As one of the reasons for the above results, compressive oracle summaries have a much larger number of (sub-)sentences than extractive oracle summaries for the same length limitation. This is an advantage of compressive summarization over extractive summarization.

Method		Score
Ext.	n=1	4.55
	n=2	4.58
Comp.	n=1	3.88
	n=2	4.07

Table 2: Readability evaluation by human subjects

However, we have to note that compressive oracle summaries optimized to ROUGE$_1$ may not be desirable since they are produced by compressing sentences by ignoring contexts. In fact, they obtained remarkable gain for ROUGE$_1$ score (8.3 points), while they obtained modest gains in ROUGE$_2$ and ROUGE-SU0 (0.7 and 3.6 points, respectively). This may suggest that the resultant summaries overfit to the unigrams in the reference summaries.

We compare ROUGE scores of compressive oracle summaries with those of system summaries, ROUGE scores of compressive oracle summaries completely outperformed those of state-of-the-art systems. The differences are in a range from 11 to 17 points.

The results demonstrated that compressive summarization is a promising approach to produce more informative summaries, and room still exists for further improvement. Thus, compressive summarization is important research topic to leverage our resources.

4.3 Readability evaluation

We conducted human evaluation to compare readability of extractive oracle summaries to that of compressive oracle summaries. We presented the oracle summaries to five human subjects and asked them to rate the summaries using an integer scale from 1 (very poor) to 5 (very good). Table 2 shows the results. Extractive oracle summaries achieved near perfect scores. Although the scores of compressive oracle summaries are inferior to those of extractive oracle summaries, they achieved good

[3] With stop words: options "-n 2 -s -m -x" are used. Without stop words: options "-n 2 -m -x" are used.

Reference:
The Wye River accord has not been implemented. As the Israeli cabinet was considering the agreement, Islamic Jihad militants exploded a car bomb in nearby Mahane Yehuda market. The cabinet suspended ratification of the agreement, demanding the Palestinian Authority take steps against terrorism. Further, after the bombing, Israeli Prime Minister Netanyahu announced the resumption of construction of a new settlement, Har Homa, in a traditionally Arab area east of Jerusalem. Israel also demands that Arafat outlaw the military wings of Islamic Jihad and Hamas. The attack injured 24 Israelis, but only the two assailants, Sughayer and Tahayneh, were killed.

Extractive oracle summary $n = 1$:
The procedure is part of the Wye River agreement negotiated last month. The radical group Islamic Jihad claimed responsibility Saturday for the market bombing and vowed more attacks to try to block the new peace accord. Most recently, Israel's Cabinet put off a vote to ratify the accord after a suicide bombing Friday in Jerusalem that killed the two assailants and injured 21 Israelis. David Bar-Illan, a top aide to Israeli Prime Minister Benjamin Netanyahu, said Sunday that Israel expects Palestinian leader Yasser Arafat to formally outlaw the military wings of Islamic Jihad and the larger militant group Hamas.

Compressive oracle summary $n = 1$:
The Israeli cabinet suspended ratification of the Wye agreement. A Prime Minister Benjamin Netanyahu said that Israel would continue to build Jewish neighborhoods throughout Jerusalem including at a site in the Arab sector of the city. Netanyahu's Cabinet delayed action on the peace accord. The radical group Islamic Jihad claimed responsibility for the bombing and vowed attacks. Implementation of the Israeli-Palestinian land-for-security accord was to have begun. David Bar-Illan said that Israel expects Palestinian Yasser Arafat to outlaw the military wings of Islamic Jihad and the Hamas. Their car-bomb blew in a Jerusalem market killing men and wounding 24 people.

Extractive oracle summary $n = 2$:
In response to the attack, the Israeli cabinet suspended ratification of the Wye agreement until there " is verification that the Palestinian authority is indeed fighting terrorism." The radical group Islamic Jihad claimed responsibility Saturday for the market bombing and vowed more attacks to try to block the new peace accord. Most recently, Israel's Cabinet put off a vote to ratify the accord after a suicide bombing Friday in Jerusalem that killed the two assailants and injured 21 Israelis. Their car-bomb blew apart two hours later in a Jerusalem market, killing both men and wounding 24 people. I'm going to Paradise. "

Compressive oracle summary $n = 2$:
The Israeli cabinet suspended ratification of the agreement. Hassan Asfour said the Palestinian Authority condemned the attack. Two people were killed. The procedure is part of the Wye River agreement. The radical group Islamic Jihad claimed responsibility for the bombing and vowed more attacks. Israel is demanding that the military wings of two radical Islamic groups be outlawed. Implementation of the land-for-security accord was to have begun. Israel's Cabinet put off a vote to ratify the accord after a bombing in Jerusalem that killed the two assailants and injured 21 Israelis. Their car-bomb blew in a Jerusalem market killing men.

Figure 2: Summaries obtained from topic:D30010

enough score, around 4. The results support that our trimming approach based on chunk is effective. We show examples of oracle summaries in Figure 2.

5 Conclusion

To reveal the ultimate limitations of the compressive summarization paradigm, this paper proposed an Integer Linear Programming (ILP) formulation to obtain compressive oracle summaries in terms of ROUGE. Evaluation results obtained from the DUC 2004 dataset demonstrated that ROUGE scores of compressive summaries are significantly superior to those of extractive oracle summaries and those of the state-of-the-art systems. These results imply that the compressive summarization paradigm is a promising direction to produce informative summaries and encourage leveraging of further resources for the research.

References

Miguel B. Almeida and André F.T. Martins. 2013. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 196–206.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 481–490.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *In Proceedings of International Conference on Language Resources and Evaluation (LREC)*. pages 449–454.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proc. of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1481–1491.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proc. of* *the 5th International Natural Language Generation Conference (INLG)*. pages 25–32.

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

Dan Gillick, Benoit Favre, Dilek Hakkani-Tur, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/UTD summarization system at TAC 2009. In *Proc. of the Text Analysis Conference (TAC)*.

Tsutomu Hirao, Masaaki Nishino, Jun Suzuki, and Masaaki Nagata. 2017. Enumeration of extractive oracle summaries. In *Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pages 386–396.

Kai Hong, John Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proc. of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. pages 1608–1616.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *Proc. of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pages 712–721.

Yuta Kikuchi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2014. Single document summarization based on nested tree structure. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 315–320.

Jeff Kubina, John Conroy, and Judith Schlesinger. 2013. ACL 2013 multiling pilot overview. In *Proc. of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*. pages 29–38.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proc. of Workshop on Text Summarization Branches Out*. pages 74–81.

Andre Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proc. of the Workshop on Integer Linear Programming for Natural Language Processing*. pages 1–9.

Xian Qian and Yang Liu. 2013. Fast joint compression and summarization via graph cuts. In *Proc. of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1492–1502.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Large-margin learning of submodular summarization models. In *Proc. of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pages 224–233.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2015. Compressive document summarization via sparse optimization. In *Proc. of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*. pages 1376–1382.