Text Summarization Model based on Redundancy-Constrained Knapsack Problem

Hitoshi Nishikawa[1], *Tsutomu Hirao*[2], *Toshiro Makino*[1] *and Yoshihiro Matsuo*[1]
(1) NTT Media Intelligence Laboratories, NTT Corporation
1-1 Hikarinooka Yokosuka-shi, Kanagawa 239-0847 Japan
(2) NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
{ nishikawa.hitoshi, hirao.tsutomu, makino.toshiro, matsuo.yoshihiro }@lab.ntt.co.jp

ABSTRACT

In this paper we propose a novel text summarization model, the redundancy-constrained knapsack model. We add to the Knapsack problem a constraint to curb redundancy in the summary. We also propose a fast decoding method based on the Lagrange heuristic. Experiments based on ROUGE evaluations show that our proposals outperform a state-of-the-art text summarization model, the maximum coverage model, in finding the optimal solution. We also show that our decoding method quickly finds a good approximate solution comparable to the optimal solution of the maximum coverage model.

KEYWORDS: Text summarization, Knapsack problem, Maximum coverage problem, Lagrange heuristics.

1 Introduction

Many text summarization studies in recent years formulate text summarization as the maximum coverage problem (Filatova and Hatzivassiloglou, 2004; Yih et al., 2007; Takamura and Okumura, 2009; Gillick and Favre, 2009; Nishikawa et al., 2010; Higashinaka et al., 2010). The maximum coverage model, based on the maximum coverage problem, generates a summary by selecting sentences to cover as many information units (such as unigrams and bigrams) as possible. Takamura and Okumura (2009) and Gillick and Favre (2009) demonstrated that the maximum coverage problem offers great performance as a text summarization model. Unfortunately, its potential is hindered by the fact that it is NP-hard (Khuller et al., 1999). There is little hope that a polynomial time algorithm for the problem exists.

Another theoretical framework for text summarization, the knapsack problem, avoids trying to cover unigrams or bigrams, and instead emphasizes the selection of important sentences under the constraint of summary length. The knapsack problem can be solved by a dynamic programming algorithm in pseudo-polynomial time (Korte and Vygen, 2008). However, the knapsack model, a text summarization model based on the knapsack problem, scores each sentence independently. While it can easily maximizes the sum of their scores, it threatens to generate redundant summaries unlike the maximum coverage model.

To tackle this trade-off between summary quality and decoding speed, we propose a novel text summarization model, the redundancy-constrained knapsack model. Starting with the advantage of the knapsack model, it uses dynamic programming to achieve optimization in pseudo-polynomial time. We add to it a constraint that curbs summary redundancy. Although this constraint can suppress summary redundancy, finding the optimal solution again becomes a challenge.

To ensure that our proposed model can find good approximate solutions, we turn to the Lagrange heuristic (Haddadi, 1997). This is an algorithm that finds a feasible solution from the relaxed, infeasible solution induced by Lagrange relaxation. It is known to be effective in finding good approximate solutions for the set covering problem (Haddadi, 1997).

We present the novelty and contribution of this paper as follows:

- In this paper we define a novel objective function and decoding algorithm for multi-document summarization. The model and algorithm presented in this paper are new in the context of automatic summarization research.
- Our proposal, the redundancy-constrained knapsack model, outperforms the maximum coverage model on the ROUGE (Lin, 2004) evaluation.
- The approximate solution of our proposed model, found by our proposed decoding method, is comparable with the optimal solution of the maximum coverage model. We also show that this approximate solution is found far faster than the optimal solution of the maximum coverage model.

This paper is organized as follows. In Section 2, we describe related work. In Section 3, we elaborate our proposed model. In Section 4, we explain the algorithm that finds a good approximate solution for our proposed model. In Section 5, we show results of experiments conducted to evaluate our proposal. In Section 6 we conclude this paper.

2 Related Work

The text summarization model based on the maximum coverage problem was proposed by Filatova and Hatzivassiloglou (2004). They solved their model by a greedy algorithm (Khuller et al., 1999). Yih et al. (2007) solved the model by a stack decoder. Takamura and Okumura (2009) and Gillick and Favre (2009) formulated the model as Integer Linear Programming (ILP) and solved the model using a branch-and-bound method.

The maximum coverage model has a trade-off between its performance and decoding speed. Although simple decoding algorithm like the greedy algorithm and the stack decoder can find an approximate solution quickly, in many cases it is far from optimal. The ILP-based approach can find the optimal solution but it spends too long in doing so. In contrast to the maximum coverage model, our proposed decoding algorithm uses the Lagrange heuristic to quickly find a good approximate solution comparable to the optimal solution of the maximum coverage model.

McDonald (2007) showed that the text summarization model based on the knapsack problem can be solved by dynamic programming in pseudo-polynomial time. We leverage this knowledge to develop a novel algorithm that can find good approximate solutions for our proposed model.

3 Redundancy-Constrained Knapsack Model

In this section we elaborate our proposed text summarization model, the redundancy-constrained knapsack model. We first introduce the maximum coverage model and show its relationship with the knapsack model. We then explain the redundancy-constrained knapsack model and a variant that includes the Lagrange multipliers.

We consider there are *n* input sentences containing *m* unique information units, such as unigrams and bigrams. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a binary vector whose element x_i is a decision variable indicating whether sentence *i* is contained in the summary. If sentence *i* is contained in the summary, $x_i = 1$. Let $\mathbf{z} = (z_1, \ldots, z_m)$ be a binary vector whose element z_j is a decision variable indicating whether information unit *j* is contained in the summary. If information unit *j* is contained in the summary, $z_j = 1$. Let $\mathbf{w} = (w_1, \ldots, w_m)$ be a vector whose element w_j indicates the importance of information unit *j*. Let $\mathbf{A}$ be a matrix whose element a_{ji} indicates the number of information units, *j*, contained in sentence *i*. If sentence *i* contains two information units *j*, $a_{ji} = 2$. Let $\mathbf{l} = (l_1, \ldots, l_n)$ be a vector whose element l_i indicates the length of sentence *i*. Let *K* be the maximum summary length desired.

The maximum coverage model can be formulated as follows:

$$\max_{\mathbf{z}} \quad \mathbf{w}^{\top}\mathbf{z} \tag{1}$$

$$s.t. \quad \mathbf{Ax} \geq \mathbf{z} \tag{2}$$

$$\mathbf{x} \in \{0,1\}^n \tag{3}$$

$$\mathbf{z} \in \{0,1\}^m \tag{4}$$

$$\mathbf{l}^{\top}\mathbf{x} \leq K \tag{5}$$

As mentioned above, the maximum coverage model selects sentences to cover as many information units as possible. If the summary contains information units 3 and 4, the value of the objective function is the sum of w_3 and w_4. To maximize the objective function, the summary has to cover as many information units with high w values as possible.

Next, we describe the knapsack model. If constraint (2) is $\mathbf{Ax} = \mathbf{z}$ and constraint (4) is $\mathbf{z} \in \{\mathbb{N}^0\}^{\mathrm{m}}$, which is an m-dimensional vector whose elements are the natural numbers including 0, the model is the knapsack model. The knapsack model can be solved by dynamic programming in pseudo-polynomial time O(nK). However, due to the change of constraint (4) which prevents redundancy in the summary, the summary generated by the knapsack model is likely to be redundant. We suppress this redundancy through the addition of a constraint.

We describe our novel proposal, the redundancy-constrained knapsack model, below.

$$\max_{\mathbf{z}} \quad \mathbf{w}^{\mathsf{T}}\mathbf{z} \tag{6}$$

$$s.t. \quad \mathbf{Ax} = \mathbf{z} \tag{7}$$

$$\mathbf{x} \in \{0,1\}^n \tag{8}$$

$$\mathbf{z} \in \{z_j | \mathbb{N}^0 \cap [0, r_j]\}^m \tag{9}$$

$$\mathbf{l}^{\mathsf{T}}\mathbf{x} \leq K \tag{10}$$

$r_j \in \mathbf{r}$ in constraint (9) is an integer more than or equal to 0, and is the upper bound of z_j, the number of information units, j, contained in the summary. That is, in the redundancy-constrained knapsack model, constraint (9) limits z_j to lie in the range 0 to r_j. Thus redundancy in the summary can be reduced by vector $\mathbf{r}$. Although the model originally can be solved easily, constraint (9) explodes the search space so fining the optimal solution under redundancy constraint (9) is difficult[1].

To make the model tractable, we draw on Lagrangian relaxation. We add Lagrange multipliers to objective function (6) and relax constraint (9).

$$\max_{\mathbf{z}} \quad \mathbf{w}^{\mathsf{T}}\mathbf{z} + \boldsymbol{\lambda}^{\mathsf{T}}(\mathbf{r} - \mathbf{z}) \tag{11}$$

$$s.t. \quad \mathbf{Ax} = \mathbf{z} \tag{12}$$

$$\mathbf{x} \in \{0,1\}^n \tag{13}$$

$$\mathbf{z} \in \{\mathbb{N}^0\}^m \tag{14}$$

$$\mathbf{l}^{\mathsf{T}}\mathbf{x} \leq K \tag{15}$$

Non-negative Lagrange multipliers $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ impose a penalty on objective function (11) when constraint (9) is violated. If the summary contains more than r_j information units, j, its importance w_j is reduced by Lagrange multiplier λ_j. Therefore, the number of information units, j, contained in the summary will decrease when the model is solved again by dynamic programming and the redundancy in the summary will be reduced (we detail our algorithm in the

[1] The redundancy-constrained knapsack problem can also be solved in pseudo-polynomial time. However its runtime is $\mathrm{O}(nk \prod_{j=1}^{m} r_j)$, which is in effect exponential time.

next section). The Lagrange multipliers $\boldsymbol{\lambda}$ are calculated by solving the Lagrange dual problem of $L(\boldsymbol{\lambda}) = \min_{\boldsymbol{\lambda}}\{\max_{\mathbf{z}} \mathbf{w}^{\top}\mathbf{z} + \boldsymbol{\lambda}^{\top}(\mathbf{r} - \mathbf{z})\}$ using the subgradient method. Constraint (9) is an inequality constraint, so an optimal solution on the model can't be found unlike dependency parsing (Koo and Collins, 2010) and statistical machine translation (Chang and Collins, 2011), but an approximate solution can, however, be found by the decoding algorithm proposed below.

4 Decoding with Lagrange heuristic

We propose the following algorithm to find an approximate solution on objective function (11) in Algorithm 1. We outline our decoding algorithm below.

(1). Let all Lagrange multipliers λ_j be 0.
(2). Iterate following steps T times.
 A) Find the optimal solution on objective function (11) by dynamic programming.
 B) If the solution by (A) satisfies all constraints, return the solution. If not, use the heuristic to find a feasible solution from the optimal solution by (A).
 C) If solution (B) exceeds the lower bound, update the lower bound.
 D) Update the Lagrange multipliers.
(3). Output the solution corresponding to the lower bound.

```
input $\mathbf{A}, K, \mathbf{l}, m, n, \mathbf{w}$
input $\alpha, \mathbf{r}$
initialize $\boldsymbol{\lambda} = \mathbf{0}, \mathbf{s} = \mathbf{0}, \mathbf{x} = \mathbf{0}, \mathbf{z} = \mathbf{0}$
initialize $b_l = -\infty, b_u = +\infty, \mathbf{x}_l = \mathbf{0}$
for $t = 1 \ldots T$
    $\mathbf{s} = sentence(\mathbf{A}, \boldsymbol{\lambda}, m, n, \mathbf{w})$
    $\mathbf{x} = dpkp(K, \mathbf{l}, n, \mathbf{s})$
    if $score(\mathbf{A}, m, n, \mathbf{x}, \mathbf{w}) \leq b_u$
        $b_u = score(\mathbf{A}, m, n, \mathbf{x}, \mathbf{w})$
    $\mathbf{z} = count(\mathbf{A}, m, n, \mathbf{x})$
    if $\mathbf{z}$ violates $\mathbf{r}$
        $\mathbf{x} = heuristic(\mathbf{A}, K, \mathbf{l}, m, n, \mathbf{w})$
        if $score(\mathbf{A}, m, n, \mathbf{x}, \mathbf{w}) \geq b_l$
            $b_l = score(\mathbf{A}, m, n, \mathbf{x}, \mathbf{w})$
            $\mathbf{x}_l = \mathbf{x}$
        $\boldsymbol{\lambda} = update(\alpha, b_l, b_u, \boldsymbol{\lambda}, m, \mathbf{r}, \mathbf{z})$
    else
        return $\mathbf{x}$

return $\mathbf{x}_l$
```

Algorithm 1: An iterative decoding algorithm with Lagrange heuristic. α is a parameter that controls the step size of λ. $\mathbf{s}$ is a vector whose element, s_i, indicates the score of sentence i. The score is calculated by function *sentence*. Function *dpkp* implements the dynamic programming algorithm for the knapsack problem in Algorithm 2. b_l and b_u indicate the lower bound and upper bound of the objective function, respectively, and are also used to decide the step size of λ. Function *score* calculates the score of summary $\mathbf{x}$. Function *count* counts the information units contained in summary $\mathbf{x}$, which is indicated by vector $\mathbf{z}$. $\mathbf{x}_l$ preserves the solution corresponding to the lower bound b_l.

This iterative algorithm based on the Lagrange heuristics (Haddadi, 1997) can find a feasible solution at each iteration. If the algorithm doesn't converge in T iterations, the algorithm returns the most recent lower bound, which is the best feasible solution. If convergence is achieved, the solution is feasible. We show a dynamic programming algorithm to solve the knapsack problem in Algorithm 2. The Lagrange multipliers are updated by the following formula (Korte and

Vygen, 2008):

$$\lambda_j \leftarrow \max\left(\lambda_j + \alpha \frac{b_u - b_l}{\|\mathbf{d}\|^2}(z_j - r_j), 0\right) \tag{16}$$

where α is a parameter that controls the step size of λ_j; b_u and b_l are the lower and upper bounds; $\mathbf{d}$ is a subgradient of the Lagrange dual problem. This formula is based on the following search strategy:

(1). If the gap between the upper and lower bounds is large, λ_j should be updated substantially.
(2). λ_j should be updated in proportion to the gap between z_j and r_j.

Our heuristic, which recovers a feasible solution from the infeasible solution, is implemented as a greedy algorithm. We outline it below:

(1). Remove iteratively a sentence from the summary until the summary satisfies the redundancy constraint. The sentence whose score divided by its length is the least among the sentences that have information units violating the redundancy constraint is removed.
(2). If the summary satisfies the constraint, remove the sentences contained in the summary and its length from the original problem, generate a sub-problem, and then solve this sub-problem by the greedy method (Khuller et al., 1999).

```
input $K$, $\mathbf{l}$, $n$, $\mathbf{s}$
initialize $\mathbf{x} = \mathbf{0}$
for $j = 0 \ldots K$
        $T[0][j] = 0$
for $i = 1 \ldots n$
        for $j = 0 \ldots K$
                $T[i][j] = T[i - 1][j]$
                $U[i][j] = 0$
        for $j = \mathbf{l}[i] \ldots K$
                if $T[i - 1][j - \mathbf{l}[i]] + \mathbf{s}[i] \geq T[i][j]$
                        $T[i][j] = T[i - 1][j - \mathbf{l}[i]] + \mathbf{s}[i]$
                        $U[i][j] = 1$
$j = K$
for $i = n \ldots 1$
        if $U[i][j] = 1$
                $x_i = 1$
                $j = j - \mathbf{l}[i]$

return $\mathbf{x}$
```

Algorithm 2: A dynamic programming algorithm for the knapsack problem. The algorithm fills out two dimensional arrays *T* and *U*. *T*[*i*][*j*] preserves the maximum score achieved at the time of *i* and *j*. *U*[*i*][*j*] remembers whether sentence *i* is added to achieve the maximum score at the time of *i* and *j*. After filling out *T* and *U*, the best solution can be found by backtracking *U*.

5 Experiment

We evaluate our proposed method in terms of two criteria.

(1). **ROUGE**: We evaluate the quality of summaries produced from ROUGE (Lin, 2004).
(2). **Time**: We measure the time taken to generate the summaries of 30 input document sets.

We compare the following four methods:

(1). **Redundancy-constrained knapsack model (RCKM)**: Our proposed method. Find the optimal solution of Equation (11) using lp_solve[2] solver.

(2). **Redundancy-constrained knapsack model with the Lagrange heuristic (RCKM-LH)**: Our proposed method. Find the approximate solution of Equation (11) by our proposed algorithm shown in Algorithm 1. We evaluate the proposed algorithm with 10 iterations (T = 10) and 100 (T = 100) iterations.

(3). **Maximum coverage model (MCM)**: Baseline. Find the optimal solution using lp_solve.

(4). **Knapsack model (KM)**: Baseline. Find the optimal solution using the algorithm shown in Algorithm 2.

5.1 Data

We use the TSC-3 corpus (Hirao et al., 2004) for evaluation. It is an evaluation corpus for multi-document summarization and was used in Text Summarization Challenge 3[3]. It contains 30 Japanese news article sets, 352 articles and 3587 sentences. Each set has three reference summaries. Detailed information of the corpus is shown in (Hirao et al, 2004).

5.2 Parameter settings

We set the three essential parameters as follows:

- α: We set α as the inverse of the number of times that Lagrange multipliers have been updated.
- **r**: The allowed redundancy r_j can be set for each information unit j. We set $r_j = \lfloor \sqrt{\mathrm{tf}_j} \rfloor$ where tf_j is the number of information units, j, contained in the input document set and $\lfloor \quad \rfloor$ is the floor function.
- **w**: we simply set j as a content word, and weight w_j based on tf-idf (Filatova and Hatzivassiloglou, 2004; Clarke and Lapata, 2007), $\mathrm{tf}_j \log\left(\frac{N}{\mathrm{df}_j}\right)$. N and df_j are the total number of documents and the number of documents containing word j in the corpus, respectively. They are calculated from the Mainichi Shimbun corpora[4] 2003 and 2004.

α is used only by RCKM-LH. **r** is used by RCKM and RCKM-LH. **w** is used by all methods. Although **r** and **w** are can be estimated in a more sophisticated fashion such as the supervised approach, in this paper we simply estimate these parameters from just the input documents, i.e. the unsupervised approach. The use of the supervised approach is a future topic.

5.3 Results and Discussions

We show the results of the ROUGE evaluation in Table 1. Our proposed method, RCKM, yielded the top score. The differences between RCKM and other methods are significant[5] according to the Wilcoxon signed-rank test (Wilcoxon, 1945). The differences between KM and other methods are also significant. One reason for the success of the proposal is that the references usually contain some redundant information units. Interestingly, reference summaries contain two or more instances of the same word. In Figure 1, we show the frequency distribution of content word occurrence. Obviously, some of words occur more than once in the document. The study of

[2] http://lpsolve.sourceforge.net/

[3] http://lr-www.pi.titech.ac.jp/tsc/tsc3-en.html

[4] http://mainichi.jp/

[5] $P < 0.01$

text coherence evaluation leverages this repetition to capture the coherence (Barzilay and Lapata, 2005); to make a text coherent, sometimes the same words are used in two successive sentences. In the context of automatic text summarization research, this repetition is referred to as Lexical Chain and can be leveraged to find important sentences (Barzilay and Elhadad, 1997). While MCM considers these repetitions as redundant information, RCKM can permit some redundancy in the summary. In view of this, redundancy parameter **r** can be estimated from the aspect of text coherence.

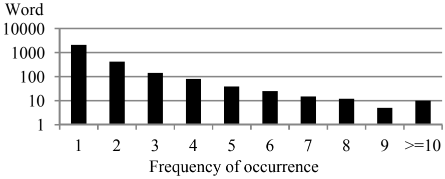

Figure 1: Frequency distribution of content word occurrence in the references. The horizontal axis indicates the frequency of content word occurrence in one reference; the vertical axis indicates the number of words. For example, there are 2093 words that occur once in one reference; there are 10 words that occur more than 9 times in one reference. This graph shows that some words occur more than once in one reference.

We also show the time spent for decoding in Table 1. MCM decoding took more than one week. KM can be quickly decoded by dynamic programming. The solver can decode RCKM far faster than MCM. RCKM-LH solves the dynamic programming iteratively. Hence the time is roughly proportional to the number of iterations.

	ROUGE-1	ROUGE-2	Time (sec.)
RCKM	**0.493**	**0.238**	2642.4
RCKM-LH (10)	0.454	0.217	72.4
RCKM-LH (100)	0.466	0.223	649.8
MCM	0.459	0.218	924349.3
KM	0.443	0.204	8.1

Table 1: ROUGE evaluation results and time taken to summarize 30 input document sets.

6 Conclusion

Our proposed model, the redundancy-constrained knapsack model, improves the quality of summaries significantly compared to a state-of-the-art system, the maximum coverage model. Our model can be decoded by the Lagrange heuristic, and the algorithm proposed here can quickly find approximate solutions of good quality.

Immediate future work is to estimate redundancy parameter **r** from large corpora. Although there are a lot of studies on estimating the weight of units, the allowed redundancy for each word has received less attention. We also plan to test our proposal on other corpora and evaluation criteria.

Acknowledgements

We would like to sincerely thank the reviewers for their comments.

References

Barzilay, R. and Elhadad, M. (1997). Using Lexical Chains for Text Summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS)*. Pages 10—17.

Barzilay, R. and Lapata, M. (2005). Modeling Local Coherence: An Entity-Based Approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 141—148.

Chang, Y.-W. and Collins, M. (2011). Exact Decoding of Phrase-Based Translation Models through Lagrangian Relaxation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26—37.

Clarke, J. and Lapata, M. (2007). Modelling Compression with Discourse Constraints. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and on Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1—11.

Filatova, E. and Hatzivassiloglou, V. (2004). A formal model for information selection in multi-sentence text extraction. In *Proceedings of Coling 2004*, pages 397–403.

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

Haddadi, S. (1997). Simple Lagrangian heuristic for the set covering problem. *European Journal of Operational Research*, 97:200–204.

Higashinaka, R., Minami, Y., Nishikawa, H., Dohsaka, K., Meguro, T., Kobashikawa, S., Masataki, H., Yoshioka, O., Takahashi, S. and Kikui, G. 2010. Improving hmm-based extractive summarization for multi-domain contact center dialogues. In *Proceedings of the IEEE Workshop on Spoken Language Technology*.

Hirao, T., Fukushima, T., Okumura, M., Nobata, C., and Nanba, H. (2004) Corpus and Evaluation Measures for Multiple Document Summarization with Multiple Sources. In *Proceedings of the 20th International Conference on Computational Linguistics (Coling)*, pages 535—541.

Khuller, S., Moss, A. and Naor, J. (1999). The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45.

Koo, T. and Collins, M. (2010). Efficient third order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1—11.

Korte, B. and Vygen, J. (2008). *Combinatorial Optimization*. Springer-Verlag, third edition.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. In *ECIR'07: Proceedings of the 29th European conference on IR research*, pages 557–564.

Nishikawa, H., Hasegawa, T., Matsuo, Y. and Kikui, G. (2010). Opinion summarization with integer linear programming formulation for sentence extraction and ordering. In *Coling 2010:Posters*, pages 910–918.

Takamura, H. and Okumura, M. (2009). Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European*

Chapter of the ACL (EACL), pages 781–789.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin,* 1(6):80—83.

Yih, W.-t., Goodman, J., Vanderwende, L. and Suzuki, H. (2007). Multi-document summarization by maximizing informative content-words. In *IJCAI'07: Proceedings of the 20th international joint conference on Artificial intelligence*, pages 1776–1782.