Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 16, No. 4, December 2016, pp. 293-298
http://dx.doi.org/10.5391/IJFIS.2016.16.4.293

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Document Summarization via Convex-Concave Programming

Minyoung Kim
Department of Electronics & IT Media Engineering, Seoul National University of Science & Technology, Seoul, Korea

Abstract

Document summarization is an important task in various areas where the goal is to select a few the most descriptive sentences from a given document as a succinct summary. Even without training data of human labeled summaries, there has been several interesting existing work in the literature that yields reasonable performance. In this paper, within the same unsupervised learning setup, we propose a more principled learning framework for the document summarization task. Specifically we formulate an optimization problem that expresses the requirements of both faithful preservation of the document contents and the summary length constraint. We circumvent the difficult integer programming originating from binary sentence selection via continuous relaxation and the low entropy penalization. We also suggest an efficient convex-concave optimization solver algorithm that guarantees to improve the original objective at every iteration. For several document datasets, we demonstrate that the proposed learning algorithm significantly outperforms the existing approaches.

Keywords: Document summarization, Natural language processing, Text mining, Optimization

1. Introduction

Document summarization is the task of automatic generation of a succinct summary or gist from a document or multiple related documents. While it has a relatively long (more than 50 years) history of research in natural language processing (NLP), text mining, and related fields, the document summarization has received unprecedented attention recently due to the enormous amount of text, news, blogs, or web pages data that need to be processed efficiently. For instance, in search engines today like Google, the top-ranked web pages relevant to a user query are shown with their titles, links, and the informative summaries (called snippets) of web pages in the most descriptive 20 to 30 words. That is, more accurate and efficient document summarization algorithms are highly demanding.

As there are several variants of document summarization, here we clarify the exact problem definition that we are going to deal with in this paper. First, there are two different tasks of summarization depending on the output forms: the *extractive* summary selects sentences from a document (i.e., a summary consists of copies of sentences from a document), while the *abstractive* summary aims to produce rephrased summary by understanding the key idea of the document in different words from those in the document. Of course, the latter task is more challenging and a long-term goal, and in this paper we focus on the extractive summary task.

Received: Nov. 21, 2016
Revised : Dec. 12, 2016
Accepted: Dec. 13, 2016

Correspondence to: Minyoung Kim
(mikim@seoultech.ac.kr)

Also, the summary task can be either query-based or general. The former aims to generate a summary that are relevant to specific user queries, while the latter outputs a gist of a document in a general sense. In this paper we deal with the general extractive summary task.

Moreover, in the learning/optimization point of view, the summarization task can be categorized into either supervised or unsupervised. It is based on whether one is given an offline (training) data of pairs of summaries and documents. Despite the fact that collecting human supervised data (manual summaries) is very expansive, in current research it is often yet difficult to exploit the supervised data effectively. Several proposed supervised summarization algorithms often underperform unsupervised methods that do not require manually supervised summary data. For the very reason that collecting human summary supervision is expensive, in this paper we deal with the unsupervised learning setup.

Perhaps, the pioneering research in (unsupervised) document summarization is the work by Luhn [1], in which he introduced the so-called *significance factor* of a sentence. The significance factor is typically derived from the number of occurrences of the significant words in the sentence where the word significance is defined by weighted word occurrence scores or other statistical indexes [2]. Then the top ranked sentences in terms of the significance factor are selected as a summary sentence. The approach is quite simple, but so effective that it is comparable to even current state-of-the-arts.

More sophisticated summarization approaches have been proposed later on. Enumerating all the related work is impossible, and here we list a few important previous approaches. The probabilistic approaches in general aim to represent the statistical process of generating words in a sentence where the distributions are differentiated depending on the importance of a sentence. While the Naive Bayes model [3] assumes word-wise independency for computational simplicity, the restriction is relaxed to yield more flexible models by incorporating sequential dependency (i.e., word ordering in sentences): the hidden Markov models [4] or conditional log-linear models [5]. The discriminative approaches like neural network training [6] often results in more accurate summarization due to the rich representational capacity in possibly deep network architectures. However, these approaches are mostly based on supervised learning, suffering from the cost of collecting lots of labeled data (i.e., summaries manually given by humans). Even more recently, the sub-modular optimization techniques have been proposed [7] that greedily finds the most descriptive sentences.

In this paper we propose a fairly simple but very efficient algorithm for document summarization. We formulate a reasonable optimization problem that expresses the requirements of both faithful preservation of the document contents and the summary length constraint. The difficulty of integer programming originating from binary selection variables, is circumvented by real value relaxation and the low entropy penalization. The relaxed/approximated problem becomes an instance of a tractable convex-concave optimization, and we provide an efficient solution method that iteratively upper-bounds and solves the original problem. Tested on some real-world news document data, the proposed approach is shown to produce more plausible summaries than existing/baseline methods.

The rest of the paper is organized as follows. After introducing formal notations and discussing some baseline document summarization methods in Section 2, our main approach of convex-concave formulation of the problem is described in Section 3. The empirical results of the proposed method are provided in Section 4, followed by concluding remarks.

2. Baselines for Document Summarization

In this section we introduce some notations that are used throughout the paper, and describe several baseline unsupervised approaches that can yield reasonable document summarization results.

A document is comprised of words from the vocabulary set $\mathcal{V}$ of size V. In text mining and natural language processing, it is typical to have a vector representation for a document (or sentence), and the most popular one is the so-called *term-frequency* (tf) vector that counts the frequency of each word that occurs in a document (or sentence). Formally, one denotes the tf vector by $[tf_1, tf_2, \ldots, tf_V]^\top$, a V-dim vector where $tf_k =$ the number of times the term t_k occurs in the document (or sentence).

While the tf representation captures salient features about a document/sentence, it cares about the frequencies of words, treating every word equally important. This is counter-intuitive in that certain stop words (e.g., articles *a* or *the*) usually appear the most frequently, thus are considered the most significant. To avoid this drawback, one needs to discount the importance of such stop words by multiplying the so-called *inverse document frequencies* (idf) vector $[idf_1, idf_2, \ldots, idf_V] \in \mathbb{R}^V$ where $idf_k = \log(\frac{n}{df_k})$ and $df_k =$ the number of training documents that contain the word t_k among a set of n documents. This results in the tf-idf vector that has the $tf_k \cdot idf_k$ as the k-th entry. In the tf-idf representation, relatively unique terms (low df_k) are highly focused, while ubiquitous terms (high df_k) like stop words are effectively ignored.

Now we discuss several reasonable baseline approaches for document summarization. Although these approaches look simple, they often quite successful in producing plausible summaries of a document. In the extractive summary task, the goal is to select (at most) b sentences from a document that best describes the whole document. The first method is simply select the first b sentences from a document. The rationale is that often authors/writers tend to put their main themes or ideas at the beginning of a document. This approach is denoted by `First-`b.

The second approach is the significance factor method introduced in [1]. Specifically, we evaluate the tf-idf vector for the whole document to assign the importance score to each word. Then we evaluate the significance score for each sentence as the sum of the importance scores of the constituting words. Then we select b highest scored sentences as a summary. We denote this classical but quite successful approach by `Luhn58`.

For the third baseline, one can come up with a fairly reasonable probabilistic (e.g., Gaussian) density model for representing the sentence generation process. More formally, for a document comprised of m sentences, we let a_i be the feature vector (e.g., tf or tf-idf) for the sentence i ($i = 1, \dots, m$). One can then consider an underlying Gaussian density model $P(a) = \mathcal{N}(a; \mu, \Sigma)$ from which the sentence feature vectors a_i's are generated independently. Under this modeling assumption, the model parameters μ and Σ can be identified by maximum likelihood estimation (To avoid overfitting the Gaussian covariance might be restricted to be diagonal or isotropic), which simply results in sample mean and covariance (i.e., $\mu = \frac{1}{m}\sum_{i=1}^{m} a_i$ and $\Sigma = \frac{1}{m'}\sum_{i=1}^{m}(a_i - \mu)(a_i - \mu)^\top$, where m' can be either m (biased) or $m - 1$ (unbiased)). Once the model is estimated, the summary can be formed by selecting b sentences that have the highest likelihood scores $P(a^i)$ among $i = 1, \dots, m$. This approach is denoted by `Gaussian`.

3. Our Approach

In this section we propose our document summarization formulation. For a document comprised of m sentences, we assign the binary variable x_i for each sentence i ($i = 1, \dots, m$), where $x_i = 1$ (or 0) indicates that the sentence i is selected (or not) as a summary. The selection vector x is thus m-dimensional binary vector we should choose. The summary of the document is then represented as a feature vector contingent on x, denoted by $\Phi(x)$. If we use the term-frequency representation for each sentence, for instance, $\Phi(x)$ is the sum of the tf-vectors for the sentences selected as a summary (i.e., those with $x_i = 1$). Naturally, selecting *all* sentences as summary, that is, having the summary feature $\Phi(e)$ where e is the m-dim vector with entries all 1, captures whole contents of the document. Of course, we usually have a length limit for a summary, say we are allowed to choose at most b sentences as a summary. Then the goal is to make the summary features as close as the full-document features $\Phi(e)$. This can be formulated as the following optimization:

$$\begin{aligned} &\min_x \, ||\Phi(e) - \Phi(x)||^2 \qquad (1) \\ &\text{s.t. } e^\top x \le b, \;\; x \in \{0,1\}^m. \end{aligned}$$

In (1) the constraint $e^\top x \le b$ encodes the summary length limit discussed before. It is also worth noting that while we employ the number of sentences as a summary length limit, one can incorporate more general budget constraint. Defining c as the m-dim vector of cost where c_i is a cost of selecting the sentence i and letting b as a budget limit in general, the constraint $c^\top x \le b$ can be quite expressive. For instance, by having c_i be the number of words in the sentence i and b be the word count limit for a summary, we obviously restrict the number of words in a summary by $c^\top x \le b$. Although in (1) we rather use $c = e$ to constrain the number of sentences (b) in a summary, our subsequent derivations apply straightforwardly to general situations with little modification.

However, the problem of (1), as can be reduced to the famous knapsack problem, is NP hard. We will propose a series of relaxation and approximation methods to yield a tractable optimization problem, followed by an efficient solution method. First the integer-valued x is relaxed to real valued in the interval $[0, 1]$. That is, instead of all-or-nothing hard selection of sentences, we do a sort of soft selection: x_i close to 1 (0) means the sentence i is more likely to be selected, and vice versa. However, one needs to enforce the selection variables to have strong confidence in selection decision, namely having them close to 0 or 1, not around 0.5 which can incur ambiguity in final sentence selection stage. For this purpose we add the regularization term to the objective to penalize large entropy for each x_i value. More specifically, our relaxed approximate optimization is formulated as follows.

$$\min_x \, ||\Phi(e) - \Phi(x)||^2 + \lambda \sum_{i=1}^{m} H(x_i), \qquad (2)$$

$$\text{s.t. } e^\top x \leq b, \;\; 0 \leq x \leq e,$$

where the inequalities $0 \leq x \leq e$ in the constrains are elementwise. In the objective $H(x_i) = -x_i \log(x_i) - (1-x_i)\log(1-x_i)$ is the entropy of the selection confidence for the sentence i, which takes a small value for x close to either 0 or 1, and vice versa. The two terms in the objective are balanced by the trade-off parameter λ (≥ 0).

To be more concrete, notice that the first term in the objective is convex quadratic in x. If we use the tf vector representation (The same applies to any term weighting schemes such as tf-idf representation since we can define a_i to be the product of the tf vector and the term weighting vector (e.g., idf)), by letting a_i be the tf vector of the sentence i, we have the tf-vector of the whole summary sentences as: $\Phi(x) = \sum_{i=1}^n x_i a_i$. Introducing $A = [a_1, \ldots, a_m]$, the $(V \times m)$ matrix with a_i's in columns, allows more succinct notation $\Phi(x) = Ax$. Thus the first term can be written as $||Ae - Ax||^2$, which is obviously convex quadratic in x.

Furthermore, while tf vectors contain word counts in a standard tf-based treatment, for the issue of scale matching between Ae and Ax (the former usually larger than the latter due to the budget constraint), in practice it is often more effective to use the *normalized* tf vectors. In essence a tf vector is divided by the number of whole words to represent relative frequencies instead (i.e., the entries summed up to 1). To incorporate normalized features, we first define a_i as a normalized tf (or tf-idf) vector, then define $\Phi(x) = \frac{1}{\sum_{i=1}^m x_i} \sum_{i=1}^m x_i a_i$. This obviously makes the entries in $\Phi(x)$ summed up to 1. Here the denominator can be replaced by b considering the budget constraint to be tight (i.e., $e^\top x = b$). There is no harm in this replacement since adding more sentences to a summary does not usually increases the objective (cost) of the feature mismatch. Similarly, the normalized tf vector for the entire document is simply the average of a_i's, namely dividing the sum of a_i's by m. In summary, we replace the first term in the objective of (2) by $||\frac{1}{m}Ae - \frac{1}{b}Ax||^2$ where A has normalized sentence-wise features as columns. The budget constraint is also changed to equality $e^\top x = b$.

Now, we discuss how to solve the optimization problem. Although the constraints are all linear, (2) is overall non-convex due to the second entropy term which is concave in x. The objective is of the form of convex plus concave, and this type of optimization problem is often called the *convex-concave programming*. We take advantage of the iterative linearization technique for convex-concave programming [8]. Defining $f(x) = ||\frac{1}{m}Ae - \frac{1}{b}Ax||^2$ and $g(x) = \lambda\big(\sum_{i=1}^m x_i \log(x_i) + (1-x_i)\log(1-x_i)\big)$, our convex-concave optimization can be written as:

$$\min \; f(x) - g(x) \;\; \text{s.t. } e^\top x = b, \;\; 0 \leq x \leq e. \tag{3}$$

Note that $f(x)$ and $g(x)$ are both convex (but the objective is their difference, hence non-convex), and also the constraints are linear inequalities. The non-convexity originates from the subtraction of $g(x)$, and our optimization strategy is iteratively approximating and solving the problem by linearizing $g(x)$ around the previous iterate. We make it formalized below.

After iteration k where we have the iterate $x^{(k)}$, we approximate $(f(x) - g(x))$ as a convex function. Since $f(x)$ is already convex, the concave $-g(x)$ is convexified. As the best convex approximate of a concave function is affine (linear), we define the convexified objective as:

$$h(x) = f(x) - g(x^{(k)}) - \nabla g(x^{(k)})^\top (x - x^{(k)}). \tag{4}$$

Here we used the first-order Taylor approximation for $g(x)$ around $x^{(k)}$, and the approximate objective $h(x)$ is obviously convex. Furthermore, $h(x)$ is global upper bound of the original objective $(f(x) - g(x))$ due to the convexity of $g(x)$. That is,

$$h(x) \geq f(x) - g(x), \;\; \forall x. \tag{5}$$

Now, we have the convexified approximate problem:

$$\min \; h(x) \;\; \text{s.t. } e^\top x = b, \;\; 0 \leq x \leq e, \tag{6}$$

which can be solved by any off-the-shelf convex optimization solver (e.g., the interior point method [9]). We denote the optimal solution of (6) by $x^{(k+1)}$. There are a couple of important things to note. First, since $x^{(k)}$ is the optimal solution for the approximate optimization in the previous iteration, $x^{(k)}$ is feasible (i.e., satisfying the inequality constraints). Secondly, as $h(x^{(k)}) = f(x^{(k)}) - g(x^{(k)})$ from (4), $h(x^{(k)})$ cannot be smaller than $h(x^{(k+1)})$, the optimal value of (6). Combining it with (5), we have the following relations:

$$f(x^{(k)}) - g(x^{(k)}) = h(x^{(k)}) \geq \; h(x^{(k+1)}), \tag{7}$$

$$\geq f(x^{(k+1)}) - g(x^{(k+1)}). \tag{8}$$

Since both $x^{(k)}$ and $x^{(k+1)}$ are feasible, (8) implies that we make improvement in the original objective value at every iteration. This guarantees that the iterations eventually converge to a local optimum of (3). We summarize the overall

Algorithm 1. Document summarization via convex-concave programming

```
Input: $A$ = Normalized sentence features (e.g., tf-idf) for a document, $b$ = the number of sentences to be selected as a summary.
Output: $b$ sentences as a summary.
Randomly choose $x^{(0)}$ from the feasible set $\{x : e^\top x = b,\ 0 \le x \le e\}$ as an initial iterate.
Repeat for $k = 0, 1, \ldots$ until convergence:
  Solve the convex optimization:
    $x^{(k+1)} = \arg\min_x ||\frac{1}{m}Ae - \frac{1}{b}Ax||^2 - \lambda \sum_{i=1}^{m} \log \frac{x_i^{(k)}}{1-x_i^{(k)}} x_i$ s.t. $e^\top x = b,\ 0 \le x \le e$.
The final solution denoted by $x^{\text{opt}}$.
Take $b$ sentences corresponding to the $b$ largest $x_i^{\text{opt}}$'s among $i = 1, \ldots, m$.
```

convex-concave solution algorithm in Algorithm 1, where we use $[\nabla g(x)]_i = \lambda \log \frac{x_i}{1-x_i}$ for $i = 1, \ldots, m$. In addition, once the optimization is done, the final summary of b sentences is constructed by collecting sentences corresponding to top-b scorers in x_i's among $i = 1, \ldots, m$.

4. Empirical Evaluations

We test the proposed document summarization algorithm on two text datasets with human summarization manually given. The brief descriptions of the datasets are summarized below.

- *State Union Dataset*: This is the collection of about 200 documents from US presidents' speeches. The vocabulary size is around $23,000$, among which we randomly select 20 documents for summarization. We manually select the most descriptive sentences that best describe the key themes of the documents.
- *Culture Dataset*: The dataset is comprised of about 700 Internet articles on historic cultures such as architectures, celebrities, paintings, and so on. Each document is of length about 2-3 Kbyte, and the vocabulary size is around $22,000$. For each of the randomly chosen 40 documents, the key summary sentences are selected manually.

As a performance metric, we use the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) criterion [10] that is the most popular measure in the document summary literature. The ROUGE basically measures how many terms are overlapped between the retrieved summary and the human selected, and the ROUGE-1 measure we are going to use in this section is specifically defined as follows.

$$\text{ROUGE-1} = \frac{\sum_{t \in S} \min(\#(t,X), \#(t,S))}{\sum_{t \in S} \#(t,S)}, \qquad (9)$$

where S is the reference summary (i.e., the set of sentences selected by human), and X is the set retrieved by a summarization algorithm. Also, $\#(t,C)$ indicates the counts of the term t in the sentence set C. Note that the denominator is the sum of occurrences of all terms in the reference summary, and obviously the larger ROUGE-1 score is the better.

Table 1. ROUGE-1 scores on the state union dataset

Methods	State union data	Culture data
`CVXCAV`	0.5690	0.4966
`First-b`	0.1715	0.1580
`Luhn58`	0.4845	0.4534
`Gaussian`	0.4277	0.3793

For the above two datasets, we compare the baseline approaches (`First-b`, `Luhn58`, and `Gaussian`) as described in Section 2, with our convex-concave optimization approach (denoted by `CVXCAV`). The summary budget constraint b is set to 3, and we choose empirically and fix the balancing trade-off parameter $\lambda = 0.01$. The ROUGE-1 scores are depicted in Table 1. Our convex-concave optimization approach performs the best for both datasets, which can be mainly attributed to our principled formulation of the ultimate goal of faithful representation of the entire document term distribution within the given budget constraint. Moreover, the convex-concave approximate optimization method appears to be quite effective in finding viable relaxed solutions.

5. Conclusion

In this paper we have proposed a novel optimization problem formulation for the unsupervised document summarization tasks. The objective function deals with the word distribution mismatch between the whole document and the summary, while the entropy penalizing term encourages the strong confidence in sentence selection. The proposed convex-concave optimization approach is not only guaranteed to converge theoretically, but also performs well in practice as shown on several real-world datasets. Extensions of the approach to the supervised learning setup and the abstractive summarization tasks remain as future work.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This study was supported by Seoul National University of Science & Technology.

References

[1] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159-165, 1958. http://dx.doi.org/10.1147/rd.22.0159

[2] C. Y. Lin and E. Hovy, "The automated acquisition of topic signatures for text summarization," in *Proceedings of the 18th Conference on Computational Linguistics*, Saarbrucken, Germany, 2000, pp. 495-501. http://dx.doi.org/10.3115/990820.990892

[3] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development In Information Retrieval*, Seattle, WA, 1995, pp. 68-73. http://dx.doi.org/10.1145/215206.215333

[4] J. M. Conroy and D. P. O'leary, "Text summarization via hidden Markov models," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, LA, 2001, pp. 406-407. http://dx.doi.org/10.1145/383952.384042

[5] M. Osborne, "Using maximum entropy for sentence extraction," in *Proceedings of the ACL-02 Workshop on Automatic Summarization*, Philadelphia, PA, 2002, pp. 1-8. http://dx.doi.org/10.3115/1118162.1118163

[6] K. M. Svore, L. Vanderwende, and C. J. C. Burges, "Enhancing single-document summarization by combining RankNet and third-party sources," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech, 2007, pp. 448-457.

[7] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, 2011, pp. 510-520.

[8] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Computation*, vol. 15, no. 4, pp. 915-936, 2003. http://dx.doi.org/10.1162/08997660360581958

[9] Y. Ye, *Interior point algorithms: theory and analysis*. New York: John Wiley & Sons, 1997.

[10] C. Y. Lin and E. Hovy, "Automatic evaluation of summaries using N-gram co-occurrence statistics," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003, pp. 71-78. http://dx.doi.org/10.3115/1073445.1073465

Minyoung Kim received his BS and MS degrees both in Computer Science and Engineering from Seoul National University, South Korea. He earned a PhD degree in Computer Science from Rutgers University in 2008. From 2009 to 2010 he was a postdoctoral researcher at the Robotics Institute of Carnegie Mellon University. He is currently an associate professor in Department of Electronics and IT Media Engineering at Seoul National University of Science and Technology in Korea. His primary research interest is machine learning and computer vision. His research focus includes graphical models, motion estimation/tracking, discriminative models/learning, kernel methods, and dimensionality reduction.
E-mail: mikim@seoultech.ac.kr