Appl Intell
DOI 10.1007/s10489-015-0747-x

A new evaluation measure using compression dissimilarity on text summarization

Tong Wang[1] · **Ping Chen**[2] · **Dan Simovici**[1]

Abstract Evaluation of automatic text summarization is a challenging task due to the difficulty of calculating similarity of two texts. In this paper, we define a new dissimilarity measure – compression dissimilarity to compute the dissimilarity between documents. Then we propose a new automatic evaluating method based on compression dissimilarity. The proposed method is a completely "black box" and does not need preprocessing steps. Experiments show that compression dissimilarity could clearly distinct automatic summaries from human summaries. Compression dissimilarity evaluating measure could evaluate an automatic summary by comparing with high-quality human summaries, or comparing with its original document. The evaluating results are highly correlated with human assessments, and the correlation between compression dissimilarity of summaries and compression dissimilarity of documents can serve as a meaningful measure to evaluate the consistency of an automatic text summarization system.

Keywords Summarization evaluation · Compression

✉ Tong Wang
tongwang0001@gmail.com

Ping Chen
Ping.Chen@umb.edu

Dan Simovici
dsim@cs.umb.edu

[1] Department of Computer Science,
University of Massachusetts Boston, Boston MA, USA

[2] Department of Computer Engineering,
University of Massachusetts Boston, Boston MA, USA

1 Introduction

As information grows rapidly every day, it is difficult for people to quickly grab the key ideas from the vast amount of information. Over the past decades, many automatic text summarization systems were proposed in several domains, which could help people to quickly get the summary of a large number of documents. However, evaluating the performance of an automatic text summarization system, and the quality of a system-generated summary (automatic summary) remains a difficult problem. Different people may have different standards to evaluate a summary. Moreover, human involvement is time-consuming and expensive. This has triggered the research in the area of automatic text summarization evaluation. Automatic evaluation helps to avoid labor-intensive and inconsistent human evaluation.

Evaluating a text summary is very difficult. Because any such process must be able to comprehend the full meaning of a document, extract the most salient and novel facts, check how many main topics are covered in the summary, and evaluate the quality of the content. There are a variety of ways to evaluate the performance of an automatic text summarization system. Generally, they can be categorized into *intrinsic* and *extrinsic* evaluation methods [1]. Extrinsic evaluation measures the quality and acceptability of the automatic summaries for a given task, while intrinsic evaluation measures the system itself [2]. Intrinsic evaluation may involve a comparison of the automatic summary with the source document to determine how many main topics and ideas are covered; or find out how similar an automatic summary is with a human-generated summary (reference summary). The intrinsic evaluations can then be divided into *content evaluation* and *text quality evaluation* [3]. Our work belongs to intrinsic evaluation together with such systems as Rouge [4] and the LSA-based measure [3].

Published online: 30 January 2016

Lossless data compression algorithm (LZW [21], DEFLATE [22]) can find data redundancy and represent data concisely without losing any information. Its losslessness and concision motivate us to use compression on automatic text summarization evaluation.

In this paper, we explore the potential of compression as an approach for summary evaluation; specifically, we define a new dissimilarity measure, compression dissimilarity, to measure the dissimilarity between documents. Our experiments show the compression dissimilarity can differentiate automatic summaries from human summaries, and the correlation between compression dissimilarity of automatic summaries with corresponding documents is able to evaluate the quality of summaries and the automatic text summarization system itself.

The advantages of using compression dissimilarity on automatic text summarization evaluation include: completely "black box", involves no additional natural language processing analysis like n-grams, parsing, stemming, converting to vector space model, no domain knowledge has to be taken into consideration, applicable to any natural language, and it is enough to only compare an automatic summary with the original document, without comparing with the ideal reference summary written by a human.

This paper is organized as below: we describe the related work about automatic text summarization evaluation and data compression in Section 2; we discuss the use of compression in automatic text summarization evaluation in Section 3; the experiment results are shown and analyzed in Section 4; finally, we conclude the paper in Section 5.

2 Related work

The most widely used automatic text summarization evaluation measure is ROUGE (Recall-Oriented Understudy of Gisting Evaluation) introduced by Lin in 2004 [4]. It automatically determines the quality of a summary by comparing it with other summaries (usually created by humans). Its basic idea is to count the number of overlapping units such as n-grams, word sequences, and word pairs between the system-generated summary and the ideal summary created by humans. The disadvantage of ROUGE is the automatic summaries have to be evaluated by comparing with human summaries.

The Pyramid method [8] is a semi-automatic evaluation method to identify a weighted inventory of SCUs (summarization content units). It is very reliable, and can perform predictions and diagnostic. However, it is a semi-automatic method and needs extra annotation work.

LSA based evaluation proposed in the paper [3] is another evaluation measure which captures the main topics of a document using LSA (Latent Semantic Analysis) [9]. In this approach the summaries are evaluated according to the similarity of the main topics of summaries and their reference documents. LSA based approach needs preprocessing steps like converting a text document into the vector space model, and using Singular Value Decomposition (SVD) for matrix decomposition.

Other measures to evaluate summarization include Cosine Similarity [12], which is based on vector space model, Unit Overlap [13], which is based on the intersection sets of words, and Longest Common Subsequence [14], which is based on the length and edit distance between sequence of words.

There are recent works relating to compression and data mining [5]. They demonstrate that the compression algorithms can identify data that contains repetitive patterns, and is allowed to decide whether a dataset is worth mining.

3 Compression on evaluating summarization

Compression has received a lot of attention in data mining. Data compression can be regarded as one of the fundamental approaches to data mining, since the goal of data mining is to "compress data by finding some structure in it" [15]. We introduce several notions related to pattern occurrence in strings and their influence in compression ratios. A document (which is a collection of strings) contains certain patterns, which allows us to apply compression to explore the underlying structure of a document, measure the dissimilarity between documents, and evaluate a text summary.

3.1 Patterns in strings and compression

Compression can be used as a tool to evaluate the potential of a data set of producing interesting results in a data mining process. The basic idea is that data that contains patterns that occur with a certain regularity will be compressed more efficiently compared to data that has no such characteristics.

An *alphabet* is a finite and non-empty set whose elements are referred to as *symbols*. Let A^* be the set of sequences on the alphabet A. We refer to these sequences as *words* or *strings*. The length of a string w is denoted by $|w|$. The null string on A is denoted by λ, and we define A^+ as $A^+ = A^* - \{\lambda\}$. The subsets of A^* are referred to as *languages* over A [6].

If $w \in A^*$ can be written as $w = utv$, where $u, v \in A^*$ and $t \in A^+$, we say that the pair (t, m) is an *occurrence* of t in w, where m is the length of u. The number of occurrences of a string x in a string w is denoted by $n_x(w)$. Clearly, we have $\sum\{n_a(w) \mid a \in A\} = |w|$ for any symbol $a \in A$. The

prevalence of x in w is the number $f_x(w) = \frac{n_x(w)\cdot|x|}{|w|}$, which gives the ratio of the characters contained in the occurrences of x relative to the total number of characters in the string.

The result of applying a compression algorithm C to a string $w \in A^*$ is denoted by $C(w)$ and the *compression ratio* is calculated as:

$$CR(w) = \frac{|C(w)|}{|w|}.$$

We shall use the binary alphabet $B = \{0, 1\}$ and the compression algorithm implemented in the package java.util.zip.

We generated random strings of bits (0s and 1s) and computed the compression ratio for strings with a variety of symbol distributions. Experiments show a string w that contains only 0s (or only 1s) achieves a very good compression ratio of $CR(w) = 0.012$ for 100 K bits and $CR(w) = 0.003$ for 500 K bits [7].

The worst compression ratio is achieved when 0s and 1s occur with equal frequencies as expected. This can also be explained by that equal frequencies implies most "uncertainty" of the structure, which leads to the highest entropy in the distribution of symbols. Conversely, a string w that contains only 0s (or only 1s) is less informative, has a certain structure and the lowest entropy. Therefore, such kind of strings could get a better compression ratio.

For strings of small length (less than 10^4 bits), the compression ratio may exceed 1 because of the overhead introduced by the algorithm. However, when the size of the random string exceeds 10^6 bits this phenomenon disappears and the compression ratio depends mainly on the prevalence of the bits and is relatively independent on the size of the string.

To see how the prevalence of bits (pattern) influences the compression ratio, we still use the binary alphabet $B = \{0, 1\}$, and denote t as a pattern, where $t \in B^+$ (e.g., $t = 001, t = 0010$). We refer to a random string w that satisfies: $f_t(w) \approx 0$; and $n_0(t) : n_1(t) = n_0(w) : n_1(w)$ as the *baseline string* of pattern t. In other words, the baseline string w has the same proportion of 0s and 1s with the pattern t, while the bits are randomly distributed. For example, for pattern 001, the distribution of bits of baseline string w should satisfy $n_0(w) = 2 \times n_1(w)$, and w does not contain 001 or only contains a minimum guaranteed number of occurrences of 001. We also refer to the compression ratio of a baseline string with pattern t as the *baseline compression ratio*.

We created a series of binary strings $\varphi_{t,m}$ which have a minimum guaranteed number m of occurrences of patterns t, where $0 \le m \le 100$. The compression baselines for files containing the patterns 01, 001,0010, and 00010 are shown in Table 1.

Specifically, we created 101 files $\varphi_{001,m}$ for the pattern 001, each containing 100K bits and we generated similar series for $t \in \{01, 0010, 00010\}$. In the case of the 001 pattern the baseline is established at 0.934, and after the prevalence exceeds 20 % the compression ratio drops dramatically. Results of the experiment for 001 are shown in Table 2.

Table 1 Baseline compression ratio for files containing a minimum guaranteed number of patterns

Pattern	Proportion of 1s (%)	Baseline
01	50	1.007
001	33	0.934
0010	25	0.844
00010	20	0.779

3.2 Dissimilarity of documents

From the previous section, we see that data contains patterns that occur with a certain regularity will be compressed more efficiently. For a human language, the alphabet V is the whole vocabulary together with punctuation, numbers or other symbols. A string w is a text document generated from V, $w \in V^*$. Text document is also the data containing certain patterns that occur with a regularity. Each text document may contain several topics, and each topic consists of specific words (patterns). This is the basic assumption for many Natural Language Processing models. For example, in Latent Dirichlet Allocation (LDA) [10, 11], a document in a corpus can be modeled as a random mixture of topics

Table 2 Pattern '001' prevalence versus the compression ratio $CR(w)$

Prevalence of '001' pattern (%)	$CR(w)$
0	0.93
10	0.97
20	0.96
30	0.92
40	0.86
50	0.80
60	0.72
70	0.62
80	0.48
90	0.31
95	0.19
100	0.01

involving specific terms. Each word is generated with a certain probability given a specific topic.

We denote x and y as two text documents respectively, xy as the text document obtained by concatenating x and y. Similarly, yx as the text document obtained by concatenating y and x, $C(x)$ as the size of the compressed document x. Even though $|xy| = |yx|$, usually $C(xy) \neq C(yx)$.

Intuitively, if two documents x and y have similar content, $C(xy)$ will be close to $C(x)$, and $C(yx)$ will be close to $C(y)$. As there is limited information gain after concatenating two similar documents, the document xy and yx are as informative as x or y respectively. Conversely, if the documents refer to different topics, xy and yx will contain more topics and information than either x or y, $C(xy)$ and $C(yx)$ will be greater than $C(x)$ or $C(y)$. In particular, $C(xx) = C(x)$, since xx contains same information with x (Actually there is an overhead after compressing a document, the equality holds only in theory). This leads to the definition of *compression dissimilarity* between two documents as

$$d(x, y) = \frac{C(xy) + C(yx)}{C(x) + C(y)}$$

It is obvious that $d(x, y) \geq 1$, since $C(xy) \geq C(x)$ and $C(yx) \geq C(y)$. The equality holds when $x = y$, which means when document x and y are identical, then $C(xy) = C(x)$ and $C(yx) = C(y)$, thus $d(x, y) = 1$. Compression dissimilarity also satisfies symmetric property since $d(x, y) = d(y, x)$.

3.3 Evaluating automatic text summarization using compression dissimilarity

We are able to apply compression dissimilarity in the evaluation of automatic text summarization in three ways.

The first way is similar to Rouge, which ranks systems by measuring how close an automatic summary is to a human summary. An automatic summary will receive a high score if its dissimilarity with a human summary is very low. Experiments in Section 4.1 shows the proposed evaluating method using compression dissimilarity between automatic summaries and human summaries are correlated with human assessment, and compression dissimilarity could differentiate automatic summaries from human summaries clearly.

The second way is to compare an automatic summary with its original document. In this approach, an automatic summary receives a high score if the dissimilarity between the document and the automatic summary is low. In most cases, we do not have human summary to compare with, comparing with the original document would be more practical. Experiments in Section 4.2 show rankings by our evaluating method using compression dissimilarity between automatic summaries and documents are correlated with rankings obtained by human assessment.

The third way is to evaluate the automatic text summarization system itself. We could evaluate systems by measuring the correlation between the dissimilarity matrix of summaries and the dissimilarity matrix of documents. The dissimilarity matrix is a $N \times N$ matrix, where N is the number of documents (summaries). The dissimilarity matrix is obtained by computing the compression dissimilarity between each pair of the N documents. Intuitively, the dissimilarity matrix of summaries and the dissimilarity matrix of documents should have a positive correlation, because the topics and information in the document should be similar to topics and information in summary. This correlation can be used as a measure on the consistency of the summary generation, which could be a potential measure to evaluate automatic text summarization systems. We define the correlation between the dissimilarity matrix of summaries and the dissimilarity matrix of documents as *consistency score*. Experiments in Section 4.3 show that consistency score has a high correlation with human assessment.

4 Experiments

We show that compression dissimilarity can differentiate human summaries from automatic summaries in Section 4.1. We use the compression dissimilarity to evaluate an automatic summary by comparing with its source document in Section 4.2. We also explore the correlation between summaries and documents, and use consistency scores to evaluate systems in Section 4.3.

4.1 Differentiating human summaries from automatic summaries in multi-documents summarization

We use the dataset in the DUC2007 [19] corpus. The main task in DUC2007 consists of 45 topics, where each topic consists of 25 documents. There are 32 participants in DUC2007 main task for multi-documents summarization. Ten NIST assessors wrote summaries for the 45 topics and each topic had 4 human summaries. Thus there are totally 36 summaries for each topic, including 32 automatic summaries and 4 human summaries.

All submitted summaries are manually evaluated, including linguistic quality and responsiveness. In these experiments we use responsiveness score to indicate the quality of a summary. NIST assessors assigned a content responsiveness score to each of the automatic summaries and human summaries. The content responsiveness score is an integer between 1 (poor) and 5 (good).

We evaluate summaries in a topic by measuring the compression dissimilarity between this summary and the 4 human summaries. If the dissimilarity between the evaluating summary and human summaries is small, it means the evaluating summary is well written. We measure the compression dissimilarity between each human summary with all other 35 summaries (except itself), and get a 4×35 dissimilarity matrix.

Since there are 4 human summaries in each topic, we use two ways to get the final dissimilarity between the evaluating summary (including automatic summaries and human summaries) and the human summary. One way is to find the mean compression dissimilarity, the other is to find the smallest compression dissimilarity. The choice of the smallest dissimilarity is motivated by the assumption that a summary is well written if it is close to any of human summaries.

We visualize the results in Fig. 1. The columns are the topics from 1 to 45, rows are the compression dissimilarity between an automatic summary and the 4 human summaries (marked by green circles), and compression dissimilarity between a human summary and the other 3 human summaries (marked by red crosses). There are 36 such compression dissimilarities in each column, including 32 automated summaries and 4 human summaries.

From Fig. 1 (left), we can see that nearly all the 4 red marks in each column lay under the 32 green marks. The mean compression dissimilarity among human summaries is around 1.7, the highest dissimilarity is 1.77, the lowest is 1.61. While the mean compression dissimilarity between human summaries and automatic summaries is around 1.75, the highest dissimilarity is 1.82, the lowest is 1.67. Figure 1 (right) gives similar results, The mean compression dissimilarity among human summaries is around 1.68, the highest dissimilarity is 1.76, the lowest is 1.58. While the mean compression dissimilarity between human summaries and automatic summaries is around 1.74, the highest dissimilarity is 1.80, the lowest is 1.60. Clearly, the dissimilarity between human summaries is smaller than the dissimilarities between automatic summaries and human summaries, for both the mean compression dissimilarity (Left) and smallest compression dissimilarity (Right).

The experiments imply that a human summary is more similar to other human summaries than to automatic summaries. If we regard human summaries as a standard, we can conclude human summaries get a better score than automatic summaries. In order to find the relationship between our method with responsiveness score assessed by humans, we normalize dissimilarity score to 1 to 5. We first find the maximum value and minimum value of the dissimilarity matrix, and then let $interval = \frac{max-min}{5}$. The dissimilarity score that locates in $[min, min+interval]$ will score 5, that locates in $[min + 4interval, max]$ will score 1. The results are shown in Fig. 2.

The left image in Fig. 2 is the responsiveness score evaluated by human, the right image is the normalized score using mean compression dissimilarity. Both images show human summaries (red crosses) have a higher score than automatic summaries (green circles). Also, we find that the correlation between compression dissimilarity and responsiveness score is -0.5261, the correlation between normalized compression score and responsiveness score is

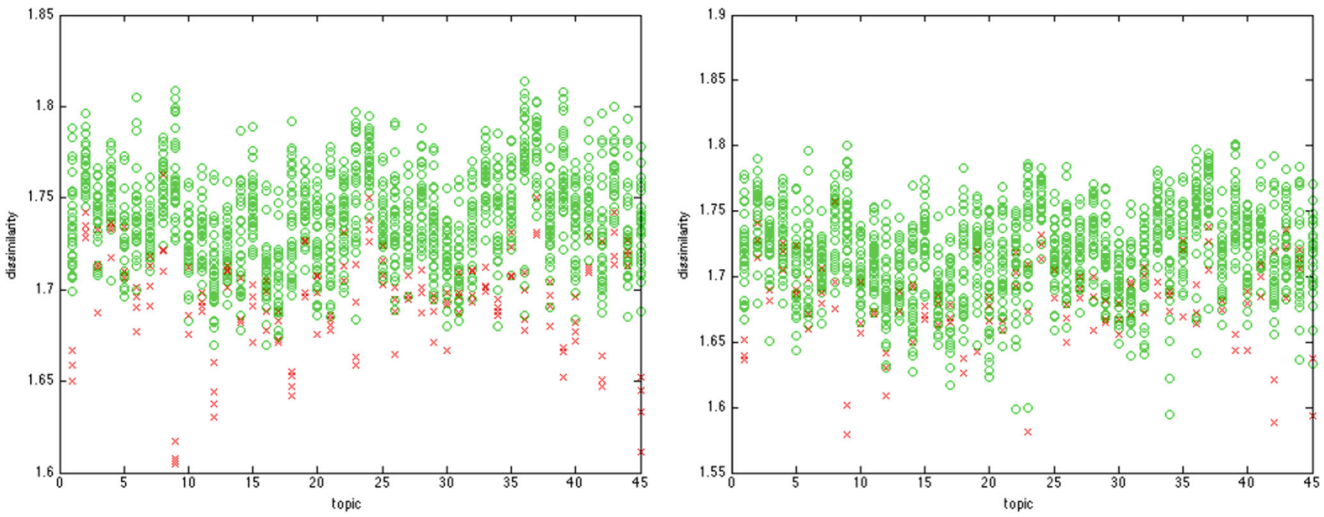

Fig. 1 Compression dissimilarity between human summaries vs compression dissimilarity between human summaries and automatic summaries. *Left*: Mean compression dissimilarity. *Right*: Smallest compression dissimilarity.The *red cross* is the mean (smallest) dissimilarity between a human summary with all other human summaries, the *green circle* is the mean (smallest) dissimilarity between an automatic summary and all human summaries

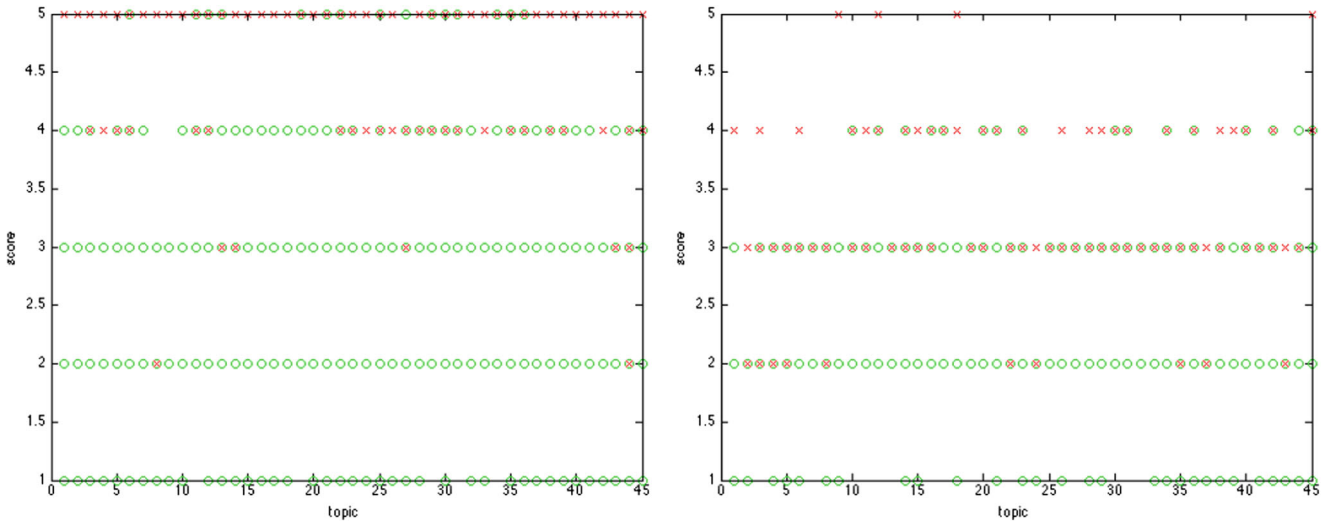

Fig. 2 Summary scores. *Left*: Responsiveness score evaluated by human manually. *Right*: Normalized score obtained by compression dissimilarity. The *red cross* is the score of human summaries, the *green circle* is the score of automatic summaries

0.4798. Therefore, evaluating a summary by comparing with human summaries through compression dissimilarity correlates with human assessment.

4.2 Evaluating summarization by comparing with source document in single-document summarization

In most cases, we do not have a human summary to compare with to evaluate an automatic summary. Thus comparing with the original source document would be a more practical choice.

In this experiment we use DUC2002 [18, 20] dataset. DUC2002 includes a single-document summarization task, in which 13 teams participated. There are totally 59 topics, we choose 5 documents in each topic. Each team's system generated a single-document summary for every document. All summaries are manually evaluated by human from 12 peer quality questions including counts of errors, mean coverage, etc. We use mean coverage as the evaluating score for summaries. Similar to other automatic evaluating methods, compression evaluating method cannot evaluate linguistic quality, it can only evaluate general coverage of a summary.

We measure compression dissimilarity between an automatic summary with its corresponding original document. If the dissimilarity between the document and the summary is small, we can say the summary is similar to the document, as there is not much information gained by concatenating the summary and the document. We measure the compression dissimilarity of all $59 \times 5 = 295$ summaries with the original documents from 13 systems. In order to compare our evaluating method with human evaluating rankings, we computed the correlation between our method with human rankings, which is -0.5561. Since the coverage of a summary should increase if the dissimilarity value decreases, our evaluating measure has a negative correlation with human rankings.

There are already some comparison work done in the review paper [3] for summary and full text similarity, and a list of correlation measures and human assessments were discussed, which facilitates the comparison between our evaluating measures with other evaluating measures. Table 3 contains the results of this comparison.

Our method does not outperform LSA — based evaluating measure and the keyword measure. Note that Rouge is not designed for comparing summaries with full text documents, so we do not list Rouge in the table. Even though the correlation of compression dissimilarity evaluating measure with human assessment is not as good as we expected, our method is more efficient and does not need any preprocessing steps. For example, LSA - based measure needs to convert text into vector space model, and use SVD for matrix decomposition to get latent semantic structure of the document. If a word combination pattern is salient, this pattern will be captured by one of the left singular vectors [16,

Table 3 Correlation between evaluation measures and human rankings

Evaluating measures	Correlation
LSA - main topic similarity	0.8599
LSA - term significance similarity	0.8557
Keywords	0.8097
Compression dissimilarity	0.5561
Cosine similarity	0.2712

17]. Keyword measure is used to find the most frequent lemmas of words in the document which do not occur in stop word list. It needs preprocessing steps like removing stopwords and lemmatizing. Cosine similarity measure also requires vector space model and term frequency counts as preprocessing steps.

Our method does not need any preprocessing steps mentioned above, the only thing we need to consider is the size of summaries and documents. Therefore, it is efficient and light-weighted comparing to other methods.

However, we should avoid the case that the file size of full text document is too big. If it is too big relative to summary, then $\frac{C(xy)+C(yx)}{C(x)+C(y)} \approx \frac{C(x)+C(x)}{C(x)} = 2$, where x is the document, y is summary. That may skew the compression dissimilarity. So the difference between document size and summary size can not be too big.

4.3 Correlation between summaries and documents

To further explore compression dissimilarity on evaluating summarization, we continue applying compression dissimilarity to explore the relationship between summaries and documents.

The dataset is from DUC2002 as in the previous section. We selected the first 10 topics and the last 10 topics, and there are 5 documents in each topic. We computed the 50×50 dissimilarity matrix for automatic summaries generated from 13 teams, human summaries, and original documents. The dissimilarity matrix is obtained by computing the compression dissimilarity between each pair of the 50 automatic summaries (human summaries, original documents). Since the matrix is symmetric, the correlation is very high if we use the original matrix. So we only choose the left triangle of the matrix without the diagonal (which is near to 1 because it is from compression dissimilarity of same documents). Then, we compute the correlation between the left triangle dissimilarity matrix of original documents and 14 left triangle dissimilarity matrices of summaries (including one matrix of human summaries and 13 matrices of automatic summaries). The resulting 14 correlations are listed in Table 4. Experiments demonstrate that there is a positive correlation between compression dissimilarity of summaries and compression dissimilarity of documents. The reason why the correlations of system 17 and 30 are very bad is that there are many summaries missing in these two systems.

We consider the correlation between compression dissimilarities of original documents and compression dissimilarity of summaries as a way to evaluate how consistent a system is. If the correlation is high, we can say this system can produce good summaries. We regard the correlation as a consistency score of a system. We measure the correlation between mean coverage assessed by human and consistency score. The results are shown in the last column of Table 4. The high correlation shows consistency score can serve as a measure to evaluate automatic text summarization systems.

Table 4 Correlation between summaries and documents

System ID	Mean coverage	First 10 topics consistency score	Last 10 topics consistency score
15	0.332	0.4919	0.4412
16	0.303	0.5000	0.4694
17	0.082	0.2704	−0.0007
18	0.323	0.5673	0.3215
19	0.389	0.5408	0.2256
21	0.370	0.5925	0.5243
23	0.335	0.2976	0.3751
25	0.290	0.3511	0.2670
27	0.383	0.4595	0.3371
28	0.380	0.5709	0.4932
29	0.361	0.6132	0.5063
30	0.057	0.0558	0.1250
31	0.360	0.6597	0.3709
Human	0.505	0.5863	0.4883
Correlation with mean coverage		0.8138	0.7787

5 Conclusion

In this paper, we define a new dissimilarity measure – compression dissimilarity to compute the dissimilarity between documents. We propose a new evaluation method based on compression dissimilarity for automatic text summarization, which is quite efficient and easy to implement. Even though it is not as accurate as Rouge or other methods like LSA based evaluating method, compression is completely "black box" and does not need any preprocessing steps like extracting n-grams, removing stop words, converting to vector space model, etc. Additionally, since our method does not require any linguistic knowledge or processing, it is reasonable to assume that it works for any language, so our method is a suitable method when NLP tools are not available or too expensive to use as required by many other summarization evaluation techniques.

References

1. Jones KS, Galliers JR (1995) Evaluating natural language processing systems: an analysis and review, vol 1083. Springer Science & Business Media
2. Hassel M (2004) Evaluation of automatic text summarization. Licentiate Thesis, Stockholm, Sweden, pp 1–75

3. Steinberger J, Jeek K (2012) Evaluation measures for text summarization. In: Computing and Informatics, vol 28.2, pp 251–275
4. Lin C-Y (2004) ROUGE: a package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop, vol 8
5. Simovici D, Pletea D, Baraty S (2013) Evaluating data minability through compression an experimental study. In: Proceedings of Data Analytics, pp 97–102
6. Simovici D, Tenney R (1999) Theory of formal languages with applications. World Scientific
7. Simovici D, Chen P, Wang T, Pletea D (2015) Compression and data mining. In: Computing, Networking and Communications (ICNC), 2015 International Conference on. IEEE, pp 551–555
8. Nenkova A, Passonneau R (2005) Evaluating content selection in summarization: the pyramid method
9. Deerwester SC, Dumais ST, Landauer TK, Furnas GW, Harshman RA (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391–407
10. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
11. Wang T, Viswanath V, Chen P Extended topic model for word dependency. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Beijing, China, pp 506–510
12. Salton G (1998) Automatic text processing. Addison-Wesley Publishing Company
13. Saggion H, Radev D, Teufel S, Lam W, Strassel SM (2002) Developing infrastructure for the evaluation of single and multi-document summarization systems in a cross-lingual environment. In: Ann Arbor, vol 1001, pp 48109–1092
14. Radev DR, Teufel S, Saggion H, Lam W, Blitzer J, Qi H, Celebi A, Liu D (2003) Evaluation challenges in large-scale multi-document summarization: the mead project. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol 1. Association for Computational Linguistics, pp 375–382
15. Mannila H (2000) Theoretical frameworks for data mining, vol 1.2. ACM SIGKDD Explorations Newsletter, pp 30–32
16. Berry MW, Dumais ST, O'Brien GW (1995) Using linear algebra for intelligent information retrieval. SIAM Rev 37.4:573–595
17. Ding CHQ (2005) A probabilistic model for latent semantic indexing. J Am Soc Inf Sci Technol 56.6:597–608
18. Document understanding conference 2002. http://www-nlpir.nist.gov/projects/duc/
19. Document understanding conference 2007. http://www-nlpir.nist.gov/projects/duc/
20. Hirao T, Sasaki Y, Isozaki H, Maeda E (2002) NTTs text summarization system for DUC-2002. In: Proceedings of the Document Understanding Conference 2002
21. Nelson MR (1989) LZW data compression. Dr. Dobb's Journal 14.10:29–36
22. Deutsch LP (1996) DEFLATE compressed data format specification version 1.3