Proceedings of CCIS2019

On Improving Text Generation Via Integrating Text Coherence

Lisi Ai, Baoli Gao*, Jianbing Zheng, Ming Gao

East China Normal University, Shanghai 200062, China
irisinsh@163.com, gao-bao-li@foxmail.com, zhengjb@js.chinamobile.com, mgao@dase.ecnu.edu.cn

Abstract: Automatic text generation techniques with either extractive-based or generative-based methods are becoming increasingly popular and widely used in industry. In contrast to existing extractive-based text generation approaches that ignore the text coherence, our proposed approach integrates both keyword coverage and text coherence into our optimization framework. In this paper, we employ the semantics-based coherence and syntax-based coherence metrics to evaluate the text coherence. Extensive experiments on a real corpus demonstrate that our method outperforms baselines overall regrading ROUGE and Human evaluation metrics. Our model provides new insights on how to utilize coherence measures to arrange the sentences extracted by keyword covering method. The proposed method has been deployed on a real system to help generate coherent text.

Keywords: Automatic text generation; Text extraction; Integer programming; Coherence

1 Introduction

Automatic text generation aims at generating the natural language texts by computer, which is a hot topic in both academia and industry. It has been widely used in many applications such as Question-Answer system, news report, online promotion, and so on.

To date, there are many existing methods in this area and most of them have already worked in the practice. The NeATS system [1] is a multi-document summarization system that attempts to extract relevant or interesting portions from a set of documents about some topic. The Newsblaster system [2] is a successful system from Columbia university, which is a news tracking tool to summarize the important news every day. Although great achievements have been made in text generation, it is still a challenged task due to the following factors:

(1) Some existing methods have a high-level complexity and it is difficult to apply to large-scale corpus.

(2) Although these methods can extract important and non-redundancy information from the given documents to some extent, it is difficult to satisfy the readability, which can be measured by text coherence. The coherence is that the extracted sentences should be connected to one another coherently so that it becomes easy to read.

978-1-7281-3863-3/19/$31.00 ©2019 IEEE

To address the coherence problem, we propose a coherent text generation method based on integer programming by incorporating text coherence and keyword coverage into objective function. Our method considers coherence when extracting sentences, rather than presenting them in coherent order after extracting sentences. In summary, the main contributions of this paper are as follows:

(1) Based on topic model and entity coherence, we propose semantics-based coherence and syntax-based coherence metrics to measure the text coherence.

(2) We develop a Coherence-Based Text Generation algorithm which can further improve the readability of the generated text via integrating the text coherence and keyword coverage into the objective function.

(3) Extensive experiments on real data illustrate our method is efficient and outperforms baseline methods overall.

2 Related work

In the past few years, automatic text generation technology has developed rapidly, which can be divided into text-to-text generation [3-12], meaning-to-text generation [13], data-to-text generation [14-16], and image-to-text generation [17] according to the data types of input. Our method belongs to text-to-text generation and focus on the text coherence.

Text-to-text generation is a well-studied problem. Carbonell [18] presents a Maximal Marginal Relevance (MMR) metric which strives to reduce redundancy. Goldstein et al. [19] improve the MMR in relevance and diversity. Besides, document summarization methods based on Integer Linear Programming [20] and submodular [21] have also been proposed. TSP heuristics methods [22] can also be used to generate summarization.

The above methods are able to satisfy importance and non-redundancy, but only few works have considered coherence while generating the text. Local coherence has been extensively studied within the modeling framework put forward by Centering Theory [23]. One of the main assumptions underlying Centering is that a text segment which contains a single entity is perceived to be more coherent than a segment in which multiple entities are discussed. This theory formalizes this intuition by introducing constraints on the distribution of discourse entities in coherent text. This kind of method has been adopted in many studies. Barzilay et al. [24] publish a model to calculate the coherence score of text, which lies in what entities it contains and how their roles change. But it can only be used to measure coherence. AbuJbara and Radev [25] preprocess the citation sentences to filter out irrelevant sentences or sentences fragments.

Different from previous work on text coherence task, our methods do not require preprocess the sentence and have higher efficiency. What's more, our methods improve the readability of generated text by incorporating coherence cost into objective function.

3 Our approach

3.1 Problem formulation

The research problem of automatic coherent text generation via text extraction is defined as follows. A summary of key notations in this work is presented in Table I.

We are given a documents set $D = \{d_i\}_{i=1}^{n}$ with same topic, where d_i is a document in D. All sentences belongs to D are defined as $S = \left\{s_i \middle| s_i \in \bigcup_{j=1}^{n} d_j\right\}_{i=1}^{N}$. Our task is to find the ordered sentences collection $C \subset S$, which can improve the coherence of the sentences as much as possible while ensuring information importance and non-redundancy.

For any given D with same topic, the model should be able to generate a text with higher coherence.

Table I A summary of key notations in this work.

Notation	Explanation
$\mathbf{D}$	A documents set with same topic
$\mathbf{K}$	Keywords set
$\mathbf{S}$	All sentences belongs to D
$\mathbf{C}$	Ordered sentences collection to form text
$\mathbf{N}$	The number of sentences in S
$\mathbf{C_{ij}}$	The coherence between $\mathbf{s_i}$ and $\mathbf{s_j}$
$\mathbf{s_j}$	j-th sentence in S
$\mathbf{len_j}$	The length of $\mathbf{s_j}$
$\mathbf{nk_j}$	The number of keywords in $\mathbf{s_j}$
$\mathbf{x_j}$	$x_j = 1$ indicates $\mathbf{s_j}$ is extracted
$\mathbf{y_{ij}}$	$y_{ij} = 1$ indicates $\mathbf{s_i}$ is followed by $\mathbf{s_j}$
$\mathbf{b_{ij}}$	$b_{ij} = 1$ indicates $k_j \in \mathbf{s_i}$
$\mathbf{y_{s,i}}$	$y_{s,i} = 1$ indicates start token is followed by $\mathbf{s_i}$
$\mathbf{y_{i,e}}$	$y_{i,e} = 1$ indicates $\mathbf{s_i}$ is followed by end token
$\mathbf{e_k}$	k-th entity
$\mathbf{M_{ij}}$	Semantic similarity between $\mathbf{s_i}$ and $\mathbf{s_j}$
$\mathbf{E_{ij}}$	syntactic coherence between $\mathbf{s_i}$ and $\mathbf{s_j}$

3.2 Method of coherence-based text generation

In the following sections, we describe the proposed framework build on corpus with multiple topic. In general, there are three parts in our framework, the third part is the focus of paper. Details are shown in Algorithm 1.

(1) **Text Clustering**: Because the corpus may consist of multiple topics, we employ Latent Dirichlet Allocation (LDA) to predict distributions of topics in corpus and employ K-means method to divide documents into several clusters in terms of the topic distributions of documents. Finally, we obtain several documents set with low confusion as candidate set.

Algorithm 1: Coherence-Based Text Generation

Input: Corpus with multiple topics.
Output: Ordered sentences collection *C*
1 Initialize $\boldsymbol{C} = \emptyset$
Text Clustering
2 Use LDA topic vector to represent documents.
3 Employ K-means to obtain **D** with same topic.
Keywords Selection
4 Use IF-IDF score to select keywords and obtain **K.**
Text Generation
5 Use Integer Programming to determine which sentence is selected and the order of sentences. The ordered sentences form sentences collection *C*.
Return *C*

(2) **Keywords Selection**: For each candidate set, the documents set is D defined in section 3.1. Then we select several keywords to form the keywords set $K = \{k_i\}_{i=1}^{m}$ using IF-IDF score, where k_i is a keyword of documents in D. The keywords cover the important information, and thus the generated text should contain all keywords. Until now, the data is ready to generate a text.

(3) **Text generation**: This is the module to model documents set D and keywords set K to learn sentences extractive pattern which can cover all keywords from K and improve the coherence of the sentences as much as possible while ensuring information importance and non-redundancy. We model this problem as Integer Programming.

The objective function is

$$\max \ \sum_{j=1}^{N} -\frac{len_j}{nk_j} x_j + \sum_{i,j=1}^{N} C_{ij} y_{ij},$$

where $x_j = 1$ represents sentence s_j is contained in $\boldsymbol{C}$, $y_{ij} = 1$ represents sentence s_i is followed by s_j, and C_{ij} represents the coherence between $\mathbf{s_i}$ and $\mathbf{s_j}$. There are two parts in objective function. The first half guarantees the extraction of important sentences, the second half guarantees the coherence of the sentences, and both guarantee the non-redundancy.

The constraints of integer programming are as follows.

$$\sum_{i=1}^{N} x_i b_{ij} \geq 1, j = 1,2,\cdots,m \quad (1)$$

$$\sum_{i=1}^{N} y_{s,i} = 1 \quad (2)$$

$$\sum_{i=1}^{N} y_{i,e} = 1 \quad (3)$$

$$\left(\sum_{i=1}^{N} y_{ij} + y_{s,j}\right) - \left(\sum_{i=1}^{N} y_{ji} + y_{j,e}\right) = 0, j = 1,\cdots,N \quad (4)$$

$$\left(\sum_{i=1}^{N} y_{ij} + y_{s,j}\right) - x_j = 0, j = 1,\cdots,N \quad (5)$$

$$x_j \in \{0,1\}, y_{ij} \in \{0,1\}, b_{ij} \in \{0,1\} \quad (6)$$

$b_{ij} = 1$ represents that sentence s_i contains keyword k_j. Constraint (1) guarantees that the extracted sentences can cover all given keywords. (2) and (3) show text with one start token and one end token. (4) and (5) show that the sentence is extracted at most once and all extracted sentences have context (start or end token also belong to context).

According to the objective function and six constraints, we can write a program to solve this problem by using GLPK for java, where x_i indicates which sentences we should extract, and y_{ij} tells us the order of the sentences. We can obtain ordered sentences collection C according to x_i and y_{ij}.

We explore two different implementations for text coherence C_{ij} as follows: 1) Considering the semantic similarity of sentences as C_{ij}. 2) Considering the syntactic of sentences to measure coherence C_{ij}. We will present the details of these two implementations in Section 3.3 and Section 3.4.

3.3 Semantics-based coherence

The assumption of semantics-based coherence is that putting sentences with similar topics in adjacent position will generate text with stronger coherence.

Semantic similarity matrix. We define $M = (M_{ij})$ as semantic similarity matrix where M_{ij} represents the similarity between sentence s_i and s_j. The calculation method of M_{ij} is described as follows.

Firstly, we train a LDA model with k topics. And then, we use topic vector $\mathbf{v_i} = (z_{i1}, z_{i2}, \cdots, z_{ik})$ to represent the sentence s_i where z_{ij} indicates the probability that the sentence s_i belongs to the j-th topic. Finally, we calculate the cosine similarity of any two sentences. The formula is as follows.

$$M_{ij} = \frac{< \mathbf{v_i}, \mathbf{v_j} >}{|\mathbf{v_i}| \times |\mathbf{v_j}|}$$

Therefore, in text generation model via integrating semantics-based text coherence, C_{ij} in Section 3.2 is equal to M_{ij}.

3.4 Syntax-based coherence

Compared with model in Section 3.3, the main difference is that we consider sentences coherence from syntactic linkages. The main assumptions are as follows: 1) The entities that play an important role in coherence are the subject and object. 2) It is more coherent to mention the same entity between adjacent sentences.

Syntactic coherence matrix. We define $E = (E_{ij})$ as syntactic coherence matrix where E_{ij} represents the coherence between sentence s_i and s_j. The calculation method of E_{ij} is described as follows.

Firstly, we employ dependency parsing (DP) to extract entities of sentences in D and mark them as S (subject), O (object) or X (neither subject nor object).

Secondly, for every entity, we calculate the average count of role transitions in adjacent sentences in whole documents set D. The formula is

$$P_{e_k}(a, b) = \frac{\#e_k(a,b)}{N-n}, a, b \in \{S, O, X\},$$

where $\#e_k(a, b)$ represents the count of role transitions from role *a* in current sentence to role *b* in next sentence, N is the number of sentences and n is the number of documents.

Finally, we calculate the syntactic coherence for any two sentences. The formula is as follows.

$$E_{ij} = \sum_{e_k \in s_i \cap s_j} \sum_{a,b \in \{S,O,X\}} P_{e_k}(a, b)$$

Therefore, in text generation model via integrating syntax-based text coherence, C_{ij} in Section 3.2 is equal to E_{ij}.

4 Empirical study

4.1 Data set description

The corpus is crawled from news Website. It contains 1703 documents, including four categories of catering, makeup, decoration and medicine. The number of documents in each category are shown in Table II.

Table II Statistics of the corpus.

Category	The number of documents
catering	529
makeup	245
decoration	649
medicine	280
Total	1703

4.2 Baselines

We consider the following baselines for comparison:

JaccardCoh. The coherence metric is Jaccard similarity,

$$J(s_i, s_j) = \frac{|\{w|w \in s_i\} \cap \{w|w \in s_j\}|}{|\{w|w \in s_i\} \cup \{w|w \in s_j\}|}$$

where w is a word in a sentence. The model is similar with Section 3.2, but C_{ij} should be equal to $J(s_i, s_j)$.

4.3 Evaluation methodology

For the evaluation metrics, we adopt ROUGE [26] and Human Evaluation.

ROUGE. We adopt ROUGE 1, ROUGE 2, ROUGE S4 and ROUGE SU4, abbreviated as R1, R2, RS4 and RSU4. They evaluate algorithm by calculating the number of matches of 4-grams. The larger value is, the better performance achieves.

Human Evaluation. To compare the coherence and readability, we ask volunteers to assign a score to each generated text. The score is an integer or decimal between 1 (very poor) and 5 (very good).

4.4 Parameter settings

In *Text Clustering* phase, the number of topic in LDA is 4, the parameter of K-means algorithm is 4. In *Keywords Selection* phase, the number of keywords equal to twice the number of documents.

4.5 Experimental results analysis

We propose two model called LDACoh and SynCoh based on semantics of sentences and syntax of sentences, respectively. In the following subsections, we will analyze the effectiveness of our model in detail.

Performance Comparison. To illustrate the reliability of human evaluation scores, we apply the Kendall concordance coefficient(W) to measure whether the volunteers agree in ranking texts. Kendall concordance coefficient is a statistic used to measure the ordinal association between two measured quantities. It is a measure of rank correlation. We rank the texts according to the score from volunteers. In the end, we obtain W = 0.58, which indicates a relatively high agreement. Therefore, the human evaluation scores are reliable.

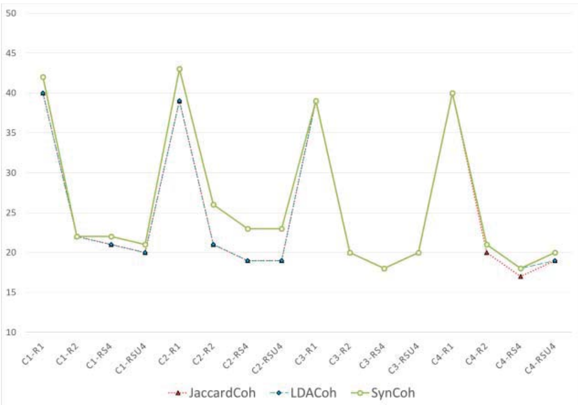

Figure 1 The ROUGE evaluation of different methods

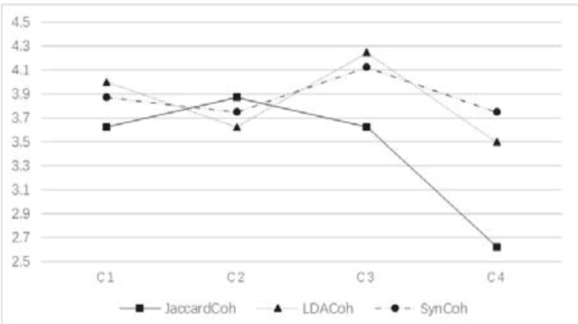

Figure 2 Human evaluation on coherence

We present evaluation results over different metrics in Figure 1 and Figure 2 respectively. Note that C1, C2, C3, C4 represent the category of catering, makeup, decoration and medicine, respectively. We summarize our observations as follows: 1) In terms of ROUGE, LDACoh perform best, SynCoh is second and JaccardCoh is worst. 2) In terms of Human evaluation, our models perform better than baseline except C2. It can be seen that the performance of JaccardCoh is very unstable, so performance well in C2 may be accidental.

In summary, our models perform better than baseline overall. Our model can generate more coherent text which is easy to read.

Utility of Coherence measures in objective function.

To demonstrate the effectiveness of integrating text coherence, we compare our models with its variant that removes the modeling of coherence, the second half in objective function in Section 3.2. The variant is named **IntegerPro**. The results are shown in Table III. We also do Kendall test and report the results indicated that the human scores are reliable.

Table III Comparison for our models with its variant.

	Evaluation metrics			
	ROUGE			
	C1	C2	C3	C4
IntegerPro	36.50	21.50	22.00	18.50
LDACoh	25.75	24.50	**24.25**	24.50
SynCoh	26.75	**28.75**	**24.25**	**24.75**
	Human Evaluation			
	C1	C2	C3	C4
IntegerPro	2.75	3.00	3.25	3.00
LDACoh	**4.00**	3.63	**4.25**	3.50
SynCoh	3.88	**3.75**	4.13	**3.75**

Note that the ROUGE is the mean of R1, R2, RS4 and RSU4. From Table III, we find that in terms of Human evaluation, our models outperform IntegerPro. In terms of ROUGE, our models perform better than IntegerPro except C1. But It can be seen that the variance of performance of Integer Pro is very big, so performance well in C1 may be accidental.

Overall, our models perform better. It indicates that our proposed methods of integrating coherence metric into objective function is rather effective and stable to generate more coherent and readable text.

5 Conclusions

In this paper, we propose an automatic text generation framework build on corpus which is crawled from news. After *Text Clustering* and *Keywords Selection* phase, we can obtain documents set and keywords set which are used to generate text. And then we develop an Integer Programming with coherent measures in objective function. Extensive experiments show our methods outperform baseline methods overall. The proposed methods have been deployed onto ***Trueland***, a company aimed to smart marketing.

Acknowledgements

This work has been supported by the National Key Research and Development Program of China under grant 2016YFB1000905, and the National Natural Science Foundation of China under Grant No. U1811264, 61877018, 61672234, 61672384, and the Shanghai Agriculture Applied

Technology Development Program, China (Grant No. T20170303).

References

[1] Lin C Y, Hovy E. From single to multi-document summarization[C]//Proceedings of the 40th annual meeting of the association for computational linguistics. 2002.

[2] Evans D K, Klavans J L, McKeown K R. Columbia newsblaster: Multilingual news summarization on the web[J]. Demonstration Papers at HLT-NAACL 2004, 2004.

[3] Li S, Ouyang Y, Wang W, et al. Multi-document summarization using support vector regression. Proceedings of DUC.2007.

[4] Knight K, Marcu D. Statistics-based summarization-step one: Sentence compression[J]. AAAI/IAAI, 2000, 2000: 703-710.

[5] Clarke J, Lapata M. Global inference for sentence compression: An integer linear programming approach[J]. Journal of Artificial Intelligence Research, 2008, 31: 399-429.

[6] Filippova K. Multi-sentence compression: Finding shortest paths in word graphs[C]//Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 2010: 322-330.

[7] Thadani K, McKeown K. Supervised sentence fusion with single-stage inference[C]//Proceedings of the Sixth International Joint Conference on Natural Language Processing. 2013: 1410-1418.

[8] Fujita A, Inui K, Matsumoto Y. Exploiting lexical conceptual structure for paraphrase generation [C]//International Conference on Natural Language Processing. Springer, Berlin, Heidelberg, 2005: 908-919.

[9] Duboue P A, Chu-Carroll J. Answering the question you wish they had asked: The impact of paraphrasing for question answering. Association for Computational Linguistics. 2006: 33-36

[10] Blei D M, Ng A, Jordan M. Latent Dirichlet allocation Journal of Machine Learning Research (3)[J]. 2003.

[11] Mihalcea R, Tarau P. Textrank: Bringing order into text[C]//Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.

[12] Chen Y, Ma Y, Mao X, et al. Multi-Task Learning for Abstractive and Extractive Summarization[J]. Data Science and Engineering, 2019, 4(1): 14-23.

[13] Zhang Y, Krieger H U. Large-scale corpus-driven PCFG approximation of an HPSG[C]//Proceedings of the 12th international conference on parsing technologies. Association for Computational Linguistics, 2011: 198-208.

[14] Sripada S, Reiter E, Davy I. SumTime-Mousam: Configurable marine weather forecast generator[J]. Expert Update, 2003, 6(3): 4-10.

[15] Kukich K. Design of a knowledge-based report generator[C]//Proceedings of the 21st annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 1983: 145-150.\

[16] Portet F, Reiter E, Gatt A, et al. Automatic generation of textual summaries from neonatal intensive care data[J]. Artificial Intelligence, 2009, 173(7-8): 789-816.

[17] Karpathy A, Fei-Fei L. Deep visual-semantic alignments for generating image descriptions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3128-3137.

[18] Carbonell, Jaime and Goldstein, Jade: The use of MMR, diversity-based reranking for reordering documents and producing summaries. International ACM SIGIR Conference on Research and Development in Information Retrieval.1998: 335–336.

[19] Goldstein J, Mittal V, Carbonell J, et al. Multi-document summarization by sentence extraction[C]//Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization. Association for Computational Linguistics, 2000: 40-48.

[20] Gillick D, Favre B. A scalable global model for summarization[C]//Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing. Association for Computational Linguistics, 2009: 10-18.

[21] Ai L, Li N, Zheng J, et al. Automatic Text Generation via Text Extraction Based on Submodular[C]//Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data. Springer, Cham, 2017: 237-246.

[22] Huang W, Yu J X. Investigating TSP heuristics for location-based services[J]. Data Science and Engineering, 2017, 2(1): 71-93.

[23] Grosz B J, Weinstein S, Joshi A K. Centering: A framework for modeling the local coherence of discourse[J]. Computational linguistics, 1995, 21(2): 203-225.

[24] Barzilay R, Lapata M. Modeling local coherence: An entity-based approach[J]. Computational Linguistics, 2008, 34(1): 1-34.

[25] Abu-Jbara, Amjad and Radev, Dragomir Coherent citation-based summarization of scientific papers In:Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume. 1, 2011: 19–24

[26] Lin C Y. Rouge: A package for automatic evaluation of summaries. Text summarization branches out: Proceedings of the ACL-04 workshop. 2004: 74–81.