He JW, Jiang WJ, Chen GB *et al.* Enhancing N-gram based metrics with semantics for better evaluation of abstractive text summarization. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 37(5): 1118–1133 Sept. 2022. DOI 10.1007/s11390-022-2125-6

Enhancing N-Gram Based Metrics with Semantics for Better Evaluation of Abstractive Text Summarization

Jia-Wei He (何嘉伟), Wen-Jun Jiang* (姜文君), *Senior Member, CCF*, Guo-Bang Chen (陈国榜)
Yu-Quan Le (乐雨泉), and Xiao-Fei Ding (丁晓菲)

College of Information Science and Electronic Engineering, Hunan University, Changsha 410082, China

E-mail: {hejiawei, jiangwenjun, gbchen, leyuquan}@hnu.edu.cn; ding_xiaofei@outlook.com

Received December 31, 2021; accepted September 30, 2022.

Abstract Text summarization is an important task in natural language processing and it has been applied in many applications. Recently, abstractive summarization has attracted many attentions. However, the traditional evaluation metrics that consider little semantic information, are unsuitable for evaluating the quality of deep learning based abstractive summarization models, since these models may generate new words that do not exist in the original text. Moreover, the out-of-vocabulary (OOV) problem that affects the evaluation results, has not been well solved yet. To address these issues, we propose a novel model called ENMS, to enhance existing N-gram based evaluation metrics with semantics. To be specific, we present two types of methods: N-gram based Semantic Matching (NSM for short), and N-gram based Semantic Similarity (NSS for short), to improve several widely-used evaluation metrics including ROUGE (Recall-Oriented Understudy for Gisting Evaluation), BLEU (Bilingual Evaluation Understudy), etc. NSM and NSS work in different ways. The former calculates the matching degree directly, while the latter mainly improves the similarity measurement. Moreover we propose an N-gram representation mechanism to explore the vector representation of N-grams (including skip-grams). It serves as the basis of our ENMS model, in which we exploit some simple but effective integration methods to solve the OOV problem efficiently. Experimental results over the TAC AESOP dataset show that the metrics improved by our methods are well correlated with human judgements and can be used to better evaluate abstractive summarization methods.

Keywords summarization evaluation, abstractive summarization, hard matching, semantic information

1 Introduction

With the rapid development of online applications, a large amount of semantic data emerges everyday. It leads to information overload for online users. To facilitate online users, it is necessary to summarize the large amount of semantic texts[1–5]. It further leads to the necessity of the evaluation for text summarization techniques. In this paper, we strive to improve the evaluation metrics of text summarization, especially the N-gram based metrics for abstractive summarization.

Abstractive summarization has attracted many attentions recently. Examples include RNN (or LTSM)-based model[6–8], attention-based model[9–11] and so on. Different from extractive summarization models that directly extract some words from the original text, abstractive summarization models can generate new words that are not contained in the original text. Although the new words may have similar meanings to the original words, it causes the difficulty of evaluating the summary quality with existing metrics.

We take the following two sentences as an example. 1) "He always gets to school early." 2) "He often arrives at classroom early." Existing hard matching based evaluation metrics will treat these similar words, i.e., "always" and "often", "gets to" and "arrives at" as different, and give them a zero similarity score, which

Regular Paper

This work was supported by the National Natural Science Foundation of China under Grant Nos. 62172149, 61632009, 62172159, and 62172372, the Natural Science Foundation of Hunan Province of China under Grant No. 2021JJ30137, and the Open Project of ZHEJIANG LAB under Grant No. 2019KE0AB02.

*Corresponding Author

is unreasonable. We deem those evaluation methods, e.g., ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [12] and BLEU (Bilingual Evaluation Understudy) [13], as hard matching, since they compare the generated summary with the reference summary in the string level, neglecting the semantic information. Therefore, it is unsuitable for evaluating abstractive summarization models, especially models based on neural networks, with traditional hard matching metrics.

Many researches have been conducted on the evaluation of automatic text summarization. There are also continuous efforts to improve the automatic summarization evaluation measures, e.g., the automatically evaluating summaries of peers (AESOP) task in TAC [14]. Among those evaluation methods, ROUGE [12] is the first, well-accepted and most widely-used one, for its strong correlation with human assessment and its simplicity of computation. Meanwhile, BLEU [13], a precision-based metric which is used to evaluate machine translation initially, can also serve as a text summarization evaluation metric [15].

Both ROUGE [12] and BLEU [13] are typical N-gram based evaluation metrics. There is an important concept in this branch, i.e., N-gram[①], which represents a contiguous sequence of N items from a given text. N-grams are widely used in language models which are based on statistics or neural networks, and most NLP tasks including machine translation, text summarization, and so on [16–19]. When $N = 2$, it is called bi-gram; when $N = 3$, it is called tri-gram; when $N = 4$, it is called four-gram; and so on. In general, a larger N means a higher accuracy the evaluation model achieves; however, it also results in a higher complexity. Bi-gram and tri-gram are two commonly-used N-grams. In this paper, we try to enhance N-gram based evaluation metrics for abstractive summarization.

Most of neural network based models treat a word as a vector [20]. Many well-studied language models such as Word2vec [21] and GloVe [22], can obtain high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. Therefore, instead of hard matching, it is more natural to utilize word embedding in the process of matching [23]. Ng and Abrecht [24] proposed ROUGE-WE, a metric making use of word embedding when computing ROUGE. ShafieiBavani *et al.* [25] proposed GROUGE which adopts a graph-based algorithm into ROUGE to capture the semantic similarities of summaries. Shao *et al.* [26] proposed WESM, a metric which can measure the semantic similarity of documents based on Word Mover's Distance (WMD). Owing to the success of word embedding, in the above researches, researchers attempted to evaluate the quality of summaries in the semantic level. Nevertheless, the semantic information of N-gram cannot be represented by word embedding directly. Therefore, it may take much calculation cost to represent the semantic information of N-grams or sentences with complex algorithms in the above researches [24–26].

Our Motivation. Inspired by the above observations, our motivations are threefold. 1) A better way to evaluate abstractive summarization: existing hard matching metrics usually calculate similarity scores based on matching N-grams in the string level, which goes against the abstractive summarization because they generate new words. Unlike hard matching, we strive to study some alternative metrics to evaluate summaries in the semantic level, and measure the similarity more effectively and reasonably. 2) A more accurate and efficient mechanism to represent the semantic information of N-gram: we try to explore a way to represent the semantic information of N-gram directly, avoiding the cost of using additional algorithms. 3) A more extensive verification for improved metrics based on N-gram: while existing studies only conduct experiments with their improved metrics based on bi-gram, we try to study more commonly-used scenarios where bi-gram, tri-gram, and four-gram are being considered.

In summary, the abstractive summarization models use semantic information to generate summaries, but the traditional evaluation metrics usually compare the generated summary with the reference summary in the string level, neglecting the semantic information and leading to unreasonable evaluation results. To this end, we propose an Ehanced N-gram Metric by Semantic model (ENMS for short) to enhance existing N-gram based evaluation metrics with semantics. Our work takes N-gram semantic information into account, to overcome the shortcomings of existing hard matching metrics. Our main contributions are threefold.

1) As the basis of our evaluation model, we first present an N-gram vector representation mechanism, in which we provide several simple but effective integration methods to solve the out-of-vocabulary (OOV) problem (Subsection 4.1).

2) Based on the above N-gram vector representation

[①] The "N" in N-gram actually is a variable. But, in order to keep consistent with the term used in the literature, N-gram is used instead of N-gram in the paper.

mechanism, we propose a novel model called ENMS (Subsection 4.2) for evaluating abstractive summarization methods, by considering the semantic information. Our model includes two methods of N-gram based Semantic Matching (NSM) and N-gram based Semantic Similarity (NSS). They work in different ways, that is, NSM matches N-grams directly, while NSS mainly improves the similarity measurement.

3) We conduct extensive experiments with a widely-used benchmark dataset (Section 5). The experimental results demonstrate that metrics improved by our methods are well correlated with human judgments and can be used to better evaluate abstractive summarization methods.

The reminder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the problem we solve. Section 4 introduces our ENMS model in details. Section 5 describes the experiments and analyses. Finally, Section 6 concludes this paper and suggests the future work.

2 Related Work

In this section, we briefly review the related work. Automatic text summarization is a fundamental task in natural language processing [27]. Automatic text summarizer generates a short summary according to original long text, and the generated summary should consist of the most relevant information that exists in the original document [28]. There are two types of automatic text summarization: extractive and abstractive. Extractive summarizer selects a few relevant sentences from the original document to generate a summary [29]. Meanwhile, abstractive summarization produces an abstractive summary like human-beings [30], which may include new words and phrases different from the ones occurring in the source document [31].

2.1 Abstractive Text Summarization

Ding *et al.* [31] integrated the sentiment-attention and the semantic-attention mechanisms with the encoder-decoder framework, and achieved better performance for generating product review summaries. Gerani *et al.* [32] presented abstractive summarization of product reviews using Discourse Structure. Liu *et al.* [33] proposed a method to generate English Wikipedia articles as a multi-document summarization of source documents. They used extractive summarization to coarsely identify salient information and a neural abstractive model to generate the article. Tan *et al.* [8] proposed a coarse-to-fine approach to solve the headline generation task, improving the performance of neural sentence summarization models.

Tan *et al.* [34] proposed a graph-based attentional neural model combined with a contextual input encoder and an output decoder, so as to address the saliency factor of summarization. Nallapati *et al.* [6] proposed an encoder-decoder RNN model with attention to generate abstractive summaries. Most recently, Chu and Liu [35] proposed the MeanSum model, which makes a breakthrough and can generate summaries with the encoder-decoder framework in an unsupervised way. Cachola *et al.* [36] proposed TLDR, an extreme summarization of scientific documents. It can generate scientific papers and peer review comments. Zhang *et al.* [37] proposed PEGASUS, a pre-training method with extracted gap-sentences for abstractive summarization. Kouris *et al.* [38] combined sequence-to-sequence neural-based text summarization with structure and semantics to deal with the problem of out-of-vocabulary or rare words. To produce coherent summaries for a long article and respect the facts in the article or commonsense knowledge, Gunel *et al.* [39] extended the transformer encoder-decoder architecture that integrates entity-level knowledge in the attention calculations and encodes a longer term dependency.

Although many novel abstractive summarization models have been proposed, the main evaluation metrics are rarely updated, which motivates our work in this paper.

2.2 Evaluation Metrics

In order to measure the quality of the generated summaries, it is necessary to build up proper metrics. There are many studies on the evaluation metrics of automatic text summarization [23, 40].

Some studies exploit human-generated benchmarks, e.g., ROUGE [12] and BLEU [13]. ROUGE includes a large number of evaluation metrics. Among them, ROUGE-1, ROUGE-2, and ROUGE-SU4 have been reported to have a strong correlation with human assessments [41]. They become the most widely used metrics to evaluate summaries. BLEU, a precision-based metric which is used to evaluate machine translation initially [14], is also validated to serve as a metric for evaluating text summarization [15]. Moreover, the Pyramid metric is a semi-automatic evaluation method [42], which has been introduced as one of the principal metrics for evaluating summaries in the TAC conference since DUC 2005. In addition, the ROUGE-BE metric was proposed to address some shortcomings of ROUGE[43], which uses very small units of content.

Some other studies try to evaluate the quality of summaries without human-generated benchmarks. Torres-Moreno *et al.*[44] proposed a content-based method for the evaluation of text summarization systems without human models. Shao *et al.*[26] proposed the WESM (word-embedding similarity measure) metric, which also can evaluate summary by making use of the original text directly. Cabrera-Diego and Torres-Moreno[45] proposed SummTriver, which is a trivergent model to evaluate summaries automatically without human references. Radev *et al.*[46] presented a series of experiments to demonstrate the validity of relative utility (RU) as a measure for evaluating extractive summarization. Shafieibavani *et al.*[47] proposed a graph-theoretic summary evaluation for ROUGE. Cohan and Goharian[48] proposed a new metric SERA (summarization evaluation by relevance analysis) that consistently achieves high correlations with manual scores in evaluation of scientific article summarization.

While the above studies are mainly working on the string level, there are some researches exploiting word-embedding[21, 22, 49, 50] to evaluate text summarization on the semantic level. For example, Ng and Abrecht[24] proposed ROUGE-WE, an automatic metric using word-embedding to compute ROUGE, improving the correlations with human assessments. ShafieiBavani *et al.*[25] proposed GROUGE which adopts a graph-based algorithm into ROUGE to capture the semantic similarities of summaries. Passonneau *et al.*[51] proposed an evaluating method combining Pyramid with word-embedding.

Our work is closely related to the above studies (i.e., [21, 22, 24, 25, 49–51]) in that we also exploit the technique of word embeddings, and we also focus on addressing the issues of evaluating abstractive summarization models. Our differences lie in three aspects: 1) we focus on improving N-gram based evaluation metrics, with the vector representation of N-grams, while existing studies improve in the granularity of word embedding; 2) we widely validate the effectiveness of solving the OOV problem with our N-gram based representation in bi-gram, tri-gram and even four-gram, while existing studies validate bi-gram only; and 3) our work avoids extra calculation cost of representing N-grams.

3 Problem Description

In this section, we define the basic concepts and the problem that we will solve. Notations used in this paper are described in Table 1.

Table 1. Notations

Symbol	Description
c	Candidate summary
r	Reference summary
R	Reference summary set
g_n	N-gram of length n

3.1 System Settings

The task of text summarization evaluation is to measure the similarity of a candidate summary with the standard reference summaries. According to existing work in literature[12, 16], we provide our description of the main concepts below.

Definition 1 (Candidate Summary). *A candidate summary is a summary that needs to be evaluated, which is often generated by algorithms.*

Definition 2 (Reference Summary). *A reference summary is a summary serving as the standard reference in the evaluation, which is often written by human.*

Definition 3 (Reference Summary Set). *A reference summary set contains some reference summaries (at least one) that depict different aspects of the same topic. Each candidate summary needs a reference summary set as the standards for quality evaluation.*

Definition 4 (N-Gram). *An N-gram is a contiguous sequence of N items (e.g., words) in a given sentence.*

3.2 Problem of Evaluating N-Gram Based Text Summarization

Given a candidate summary c, and a set of reference summaries R, the main goal of the text summarization evaluation is to assign c a score to measure its similarity with the standard references in R:

$$score(c) = f_{g_n}(c, R),$$

where $f_{g_n}(\cdot,\cdot)$ is the N-gram based evaluation metric that can be implemented with different methods. Generally, a good metric should generate a score that is highly correlated with human judgement. We strive to define such metrics that are suitable for abstractive summarization models.

3.3 Solution Overview

In this paper, we aim to improve the N-gram based summary evaluation metrics by considering the semantic information. We first map an N-gram to a vector representation that contains the overall meanings of the

N-gram, by training with corpus. Based on the N-gram vector representation, we propose two methods to construct the $f_{g_n}(\cdot,\cdot)$ function, so as to assign the candidate summary a similarity score.

3.4 Preliminary

We here review several widely-used hard matching evaluation methods, which are the basis of understanding our proposed ENMS model.

ROUGE-N is an N-gram recall between a candidate summary and a set of reference summaries [12]. It is calculated as in (1):

$$\mathit{ROUGE\text{-}N}(c) = \frac{\sum_{r\in R}\sum_{g_n\in r} \mathit{Count}_{\rm match}(g_n)}{\sum_{r\in R}\sum_{g_n\in r} \mathit{Count}(g_n)}, \quad (1)$$

where R represents all reference summaries, r is a summary belonging to R, and g_n is an N-gram belonging to r. $\mathit{Count}_{\rm match}(g_n)$ is the maximum number of N-grams co-occurring in both of a candidate summary and a set of reference summaries, and $\mathit{Count}(g_n)$ is the number of N-grams occurring in a reference summary. In many cases, ROUGE-N shows strong correlation with human judgement when $N = 2$.

BLEU calculates the percentage of N-grams in the candidate summary that also occur in the reference summaries, and calculates the weighted average of N-grams of different lengths that can be matched in the reference summaries [13]. It is calculated as:

$$\begin{aligned} \mathit{BLEU} &= \mathit{BP} \times \exp(\textstyle\sum_{n=1}^{N} \lambda_n \log p_n), \\ p_n &= \frac{\sum_{g_n\in c} \mathit{Count}_{\rm match}(g_n)}{\sum_{g_n\in c} \mathit{Count}(g_n)}. \end{aligned} \quad (2)$$

In (2), p_n is N-gram precision, in which c is a candidate summary, g_n is an N-gram belonging to c, and $\mathit{Count}(g_n)$ counts the number of N-grams occurring in a candidate summary. Positive weights $\lambda_n > 0$ and $\sum \lambda_i = 1$. BLEU considers all the N-grams of the length from 1 to N. BP is a penalty applied to the machine translation task. It can be set to 1 for evaluating the text summarization task.

ROUGE-SU4 uses another concept of skip-gram, which can be any pair of words extracted in the original sentences, allowing for arbitrary gaps (gaps are not longer than 4 in ROUGE-SU4) [12]. It can be calculated as:

$$\begin{aligned} &\mathit{ROUGE\text{-}SU4}(c,r) \\ &= \frac{\mathit{SKIP}_{\rm match}(c,r) + \mathit{WORD}_{\rm match}(c,r)}{C(l,2)+l}, \end{aligned}$$

where c is a candidate summary, r is a reference summary of length l, and $\mathit{SKIP}_{\rm match}(c,r)$ returns the number of skip-grams that can be matched in c and r, while $\mathit{WORD}_{\rm match}(c,r)$ returns the number of words that can be matched in c and r, and $C(l,2)$ is a combination function that any two items are selected from l items.

ROUGE-L takes a summary sentence as a sequence of words, and calculates the longest common subsequence (LCS for short) of two summary sentences. Intuitively, the longer the LCS of two summaries is, the more similar the two summaries are [13]. Given two summaries c and r, ROUGE-L can be calculated as in (3):

$$\mathit{ROUGE\text{-}L}(c,r) = \frac{\mathit{LCS}(c,r)}{l}, \quad (3)$$

where $\mathit{LCS}(c,r)$ returns the length of the LCS between c and r, and l is the total number of words in the reference summary r.

4 ENMS: Details

In this section, we introduce the details of the ENMS model. We first expatiate our N-gram vector representation mechanism. Then, we provide the details of the proposed ENMS model for better evaluating abstractive summarization. It includes two methods: N-gram based Semantic Matching (NSM for short), and N-gram based Semantic Similarity (NSS for short).

4.1 N-Gram Vector Representation Mechanism

In most of NLP tasks, the real-valued vector representation of a word, which is usually trained using a large corpus, is used to capture the fine-grained semantic information. Inspired by word vectors, our ENMS model uses the high-quality vector representation of N-grams to capture the semantic information contained in N-grams, i.e., we improve the semantic level from words to N-grams. However, previous studies of word vectors cannot be used to train N-gram based vector representation directly. To address this issue, we take a similar way as in [52] and we treat each N-gram as a normal word and assign it with a unique vector. Then, we maximize the following conditional probabilities:

$$\sum_{\boldsymbol{w}_{ng}\in S(C)} \log p(\boldsymbol{w}_{ng}|\mathit{Context}(\boldsymbol{w}_{ng})),$$

where C is a corpus, $S(C)$ returns a set that contains all words and N-grams belonging to C, and $\boldsymbol{w}_{ng}$ is a word or an N-gram in this set. $\mathit{Context}(\boldsymbol{w}_{ng})$ returns all words and N-grams contained in the context of $\boldsymbol{w}_{ng}$. After that, we can obtain a pre-trained vector representation of N-gram, called PTV for short. Please note that PTV also contains the word embedding of all words.

Nevertheless, there are some N-grams that we cannot find their vector representations in PTV. This is the so-called out-of-vocabulary(OOV) problem. No matter how large corpus we use to train PTV, the OOV problem cannot be avoided. In addition, skip-grams that are used by ROUGE-SU4, rarely appear in the PTV. That is, the OOV problem is more serious for skip-gram based metrics.

Integration Methods. To solve the OOV problem, we need some efforts to integrate individual word embedding of words that are contained in an N-gram (skip-gram) together, and use the integrated word embedding to represent the N-gram. We introduce four integration methods as follows.

- The simple multiplicative makes the individual vectors multiply together to produce the vector for an N-gram. It is proposed by Mitchell and Lapata[53] and is used in ROUGE-WE[24]. For example, if an N-gram $\boldsymbol{g} = \boldsymbol{w}_1\boldsymbol{w}_2$, the word embedding $\boldsymbol{w}_1 = (x_1, x_2, \cdots, x_d)$, $\boldsymbol{w}_2 = (y_1, y_2, \cdots, y_d)$, then the vector representation of $\boldsymbol{g}$ is:

$$\boldsymbol{V}_g = (x_1y_1, x_2y_2, \cdots, x_dy_d).$$

- The midpoint approach treats the midpoint of individual vectors as the semantic vector of the N-gram. For example, if the N-gram $\boldsymbol{g} = \boldsymbol{w}_1\boldsymbol{w}_2\cdots\boldsymbol{w}_n$, and $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n$ are the word embeddings of $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n$ respectively, then we have:

$$\boldsymbol{V}_g = \frac{\boldsymbol{v}_1 + \boldsymbol{v}_2 + \cdots + \boldsymbol{v}_n}{n}.$$

- The catenation approach catenates the individual vectors to a long vector, and uses this long vector as the semantic vector of the N-gram. For example, if the N-gram $\boldsymbol{g} = \boldsymbol{w}_1\boldsymbol{w}_2$, and the word embeddings of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are $\boldsymbol{v}_1 = (x_1, x_2, \cdots, x_d)$, $\boldsymbol{v}_2 = (y_1, y_2, \cdots, y_d)$, respectively, then we have:

$$\boldsymbol{V}_g = (x_1, x_2, \cdots, x_d, y_1, y_2, \cdots, y_d).$$

- The attention average takes the importance of the word into account. We here use the TF-IDF value as the attention weight of the word (note that this attention weight can be replaced with other more powerful attention methods). For example, if the N-gram $\boldsymbol{g} = \boldsymbol{w}_1\boldsymbol{w}_2\cdots\boldsymbol{w}_n$, and $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n$ are the word embeddings of $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n$, respectively, then we have:

$$\boldsymbol{V}_g = TF\text{-}IDF_{\boldsymbol{w}_1} \times \boldsymbol{v}_1 + \cdots + TF\text{-}IDF_{\boldsymbol{w}_n} \times \boldsymbol{v}_n,$$

where $TF\text{-}IDF_{\boldsymbol{w}_n}$ is the TF-IDF value of word $\boldsymbol{w}_n$, and it can be calculated according to the dataset that needs to be processed.

Fig.1 shows the framework of our N-gram vector representation mechanism, which has three steps. First, we search the vector representation of the N-gram in PTV. Second, we check whether the OOV problem exists. Third, if the N-gram is an OOV term, we search the word embedding in PTV for every word contained in the N-gram, and choose an integration approach to integrate it. This mechanism can weaken the impact of the OOV problem in our ENMS model.

4.2 Improving Metrics with ENMS

We propose two possible methods to improve some existing hard matching metrics: N-gram based Semantic Matching (NSM) and N-gram based Semantic Similarity (NSS). The former method calculates the matching degree directly while the latter mainly improves the similarity measurement. All the original metrics and improved ones are shown in Table 2, in which Hard represents the existing hard matching metrics.

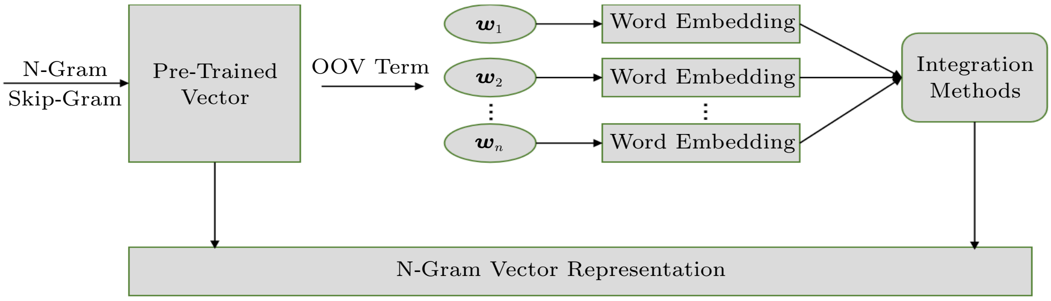


Fig.1. N-gram vector representation mechanism.

Table 2. Existing Metrics and Improved Ones by NSM and NSS

Hard	NSM Improved	NSS Improved
ROUGE-N	NSM-RN	NSS-RN
BLEU	NSM-BL	NSS-BL
ROUGE-SU4	NSM-RSU4	NSS-RSU4
ROUGE-L	S-RL	S-RL

4.2.1 N-Gram Based Semantic Matching Method (NSM)

In the hard matching, evaluation metrics do not consider the semantic similarity between two N-grams if they are different in the string level. As an improvement, we explore the vector representation of N-gram to match the N-grams between sentences. It only takes some minor changes to upgrade existing N-gram based evaluation metrics with NSM. Taking ROUGE-N as an example, we can calculate NSM-RN as in (4).

$$NSM\text{-}RN(c) = \frac{\sum\limits_{r \in R} \sum\limits_{g_n \in c} NSM(g_n, NG(r), \alpha)}{\sum\limits_{r \in R} COUNT(r)}, \tag{4}$$

where R represents all reference summaries, r is a summary belonging to R, c represents a candidate summary, and g_n is an N-gram belonging to c. $NG(r)$ returns all N-grams occurring in r, and $COUNT(r)$ returns the number of N-grams occurring in r. As for the N-gram semantic matching function $NSM(g_n, NG(r), \alpha)$, we provide its detailed description in Algorithm 1.

Algorithm 1. N-Gram Based Semantic Matching

```
Input: N-gram /*an N-gram to be matched*/
       ref-N-grams /*all N-grams belonging to a reference summary*/
       α /*a similarity parameter*/
Output: 1 or 0 /*indicates if match or not*/
N-gram-vec ⟵ get_embedding(N-gram)
max-sim ⟵ −∞
for r-gram in ref-N-grams do
    r-vec ⟵ get_embedding(r-gram)
    sim ⟵ cos(N-gram-vec, r-vec)
    if max-sim < sim then
        max-sim ⟵ sim
        if max-sim > α then
            return 1
        else
            return 0
```

The BLEU metric can be modified with *NSM-BL* as in (5).

$$\begin{aligned} NSM\text{-}BL &= BP \times \exp\left(\textstyle\sum_{n=1}^{N} \lambda_n \log NSMP_n\right), \\ NSMP_n &= \frac{\sum\limits_{r \in R} \sum\limits_{g_n \in c} NSM(g_n, NG(r), \alpha)}{COUNT(c)}, \end{aligned} \tag{5}$$

where BP is a penalty to make the length of machine translation close to that of the source text. In summary evaluation, we can set $BP = 1$. We use N-grams up to the maximum length N and make the positive weights λ_n summing to 1. For instance, if the N-gram of any length n is equally important, we can use uniform distributed weights, that is, $\lambda_n = 1/N$.

ROUGE-SU4 can be improved with NSM-RSU4 as:

$$\begin{aligned} &NSM\text{-}RSU4(c) \\ &= \left(\sum_{r \in R} \sum_{g_{sk} \in c} NSM(g_{sk}, SKG(r), \alpha) + \right. \\ &\left. \sum_{r \in R} \sum_{\boldsymbol{w} \in c} NSM(\boldsymbol{w}, WORD(r), \alpha)\right) \Big/ \\ &(C(l, 2) + l), \end{aligned}$$

where c is a candidate summary, r is a reference summary of length l, g_{sk} is a skip-gram belonging to c, $SKG(r)$ returns all skip-grams occurring in r, $\boldsymbol{w}$ is a word belonging to c, $WORD(r)$ returns all word occurring in r, and $C(l, 2)$ is the combination function that any two items are selected from l items. Note that, since a word can be taken as a 1-gram, $NSM(\boldsymbol{w}, WORD(r), \alpha)$ also works with words.

Algorithm 1 illustrates our N-gram semantic matching process. The similarity parameter α falls in the range of 0 to 1. The computational complexity is $O(KD)$, where K is the number of g_n that belongs to *ref-N-grams*, and D is the dimension of the vector representation of N-grams.

Case Study of NSM Method. Again, let us look at the two sentences. 1) "He always gets to school early." 2) "He often arrives at classroom early." We treat sentence 1 as the candidate summary and sentence 2 as the reference summary. The NSM calculation process of these two summaries is shown in Fig.2. The similarity parameter α is 0.6. Each rounded rectangle represents a bi-gram, the arrow lines connect the most similar bi-grams, two connected green ones (the first pair and the last pair) are semantic matched bi-gram (for which Algorithm 1 returns 1) and two connected blue ones (the three pairs in the middle) fail to match (for which Algorithm 1 returns 0). We can see that these two sentences contain similar meanings, and thus the score of NSM-R2 is better than that of ROUGE-2. Note that $NSM\text{-}R2(c)$ can be simplified with NSM-R2 if the candidate summary c is determined.

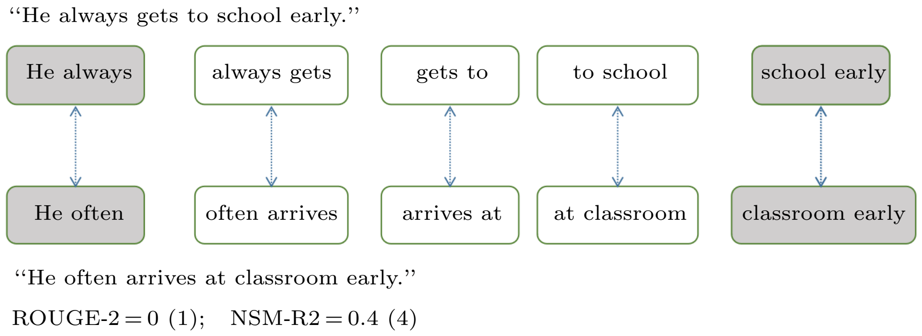


Fig.2. Illustration of calculating NSM-RN between two sentences, where we use NSM-R2.

4.2.2 *N-Gram Based Semantic Similarity Method (NSS)*

In this part, we evaluate the candidate summary in another way. We do not match N-grams between the candidate summary and the reference summary in NSS. Instead, we accumulate the similarities of all semantically similar words, so as to upgrade existing N-gram based evaluation metrics to the NSS metrics. Again, we use ROUGE-N as an example to illustrate our improvements. The improved metric NSS-RN can be calculated as in (6):

$$NSS\text{-}RN(c) = \frac{\sum\limits_{r\ \in\ R}\ \sum\limits_{g_n \in\ c} NSS(g_n, NG(r), \alpha)}{\sum\limits_{r\ \in\ refs} COUNT(r)}, \quad (6)$$

where $NSS(g_n, NG(r), \alpha)$ returns the total similarity between g_n and a reference summary r. We provide its detailed description in Algorithm 2.

Algorithm 2. N-Gram Based Semantic Similarity

```
Input: N-gram /*an N-gram to be matched*/
       ref-N-grams /*all N-grams belonging to a reference summary*/
       $\alpha$ /*the similarity parameter*/
Output: tolsim /*total similarity of soft matched N-grams*/
N-gram-vec $\longleftarrow$ get_embedding(N-gram)
max-sim $\longleftarrow -\infty$
max-count $\longleftarrow 0$
for r-gram in ref-N-grams do
    r-vec $\longleftarrow$ get_embedding(r-gram)
    sim $\longleftarrow \cos$(N-gram-vec, r-vec)
    if max-sim < sim then
        max-sim $\longleftarrow$ sim
        max-count $\longleftarrow$ get_count(r-gram)
    end if
end for
tolsim $\longleftarrow 0$
if max-sim > $\alpha$ then
    tolsim $\longleftarrow$ max-sim $\times$ max-count
end if
return tolsim = 0
```

The BLEU metric can be modified with NSS-BL in a similar way. It is calculated as in (7).

$$\begin{aligned} NSS\text{-}BL &= BP \times \exp\left(\sum_{n=1}^{N} w_n \log NSSP_n\right), \\ NSSP_n &= \frac{\sum\limits_{r\ \in\ R}\ \sum\limits_{g_n \in\ c} NSS(g_n, NG(r), \alpha)}{COUNT(c)}. \end{aligned} \quad (7)$$

Moreover, NSS-RSU4 can be calculated as:

$$\begin{aligned} & NSS\text{-}RSU4(c) \\ = & \left(\sum_{r\ \in\ R}\ \sum_{g_{sk} \in\ c} NSS(g_{sk}, SKG(r), \alpha) + \right. \\ & \left. \sum_{r\ \in\ R}\ \sum_{w \in\ c} NSS(w, WORD(r), \alpha)\right) \\ & /(C(l,2)+l). \end{aligned}$$

The computational complexity of Algorithm 2 is also $O(KD)$, where K is the number of the N-grams belonging to *ref-N-grams*, and D is the dimension of the vector representation of the N-gram.

Again, we use Fig.3 to show the comparison of the two similar sentences, where we set the similarity parameter α to 0.6. The arrow lines connect the most similar bi-grams, the decimal in the left of the line is the similarity between two bi-grams, and the one in the right of the line is the NSS similarity value obtained with Algorithm 2. We can see that NSS-R2 is more reasonable than ROUGE-2, indicating the feasibility of our NSS method.

4.2.3 *Improvements of ROUGE-L*

N-gram based evaluation metrics can be improved directly by NSM or NSS. However, some other widely-used metrics such as ROUGE-L, cannot be directly treated. Therefore, we propose an evaluation metric

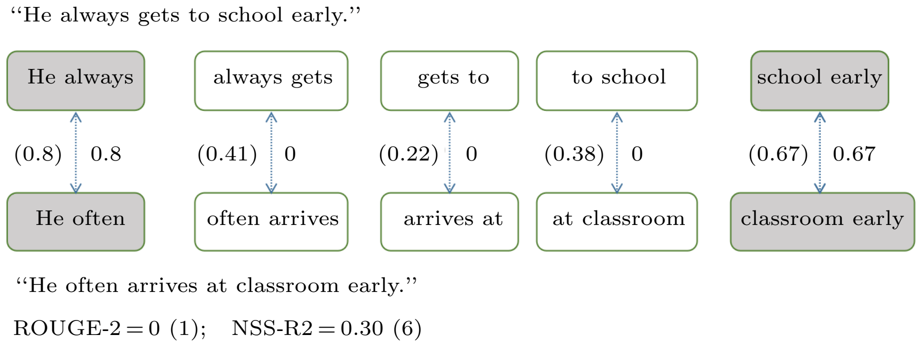


Fig.3. Illustration of calculating NSS-RN between two sentences, where we use NSS-R2.

S-RL, which improves ROUGE-L by considering the semantic information. It can be calculated as in (8).

$$S\text{-}RL(c) = \frac{\sum_{i=1}^{l_s} LSS_{\cup}(s_i, c)}{l}, \quad (8)$$

where l_s is the number of sentences in the reference summary r, s_i is the i-th sentence in the reference summary, c is the candidate summary, and l is the number of words in reference summary r. $LSS(\cdot,\cdot)$ is the longest semantically similar subsequence between two sentences. $LSS_{\cup}(s_i, c)$ is the LSS score of the union longest semantically similar subsequence between reference sentence s_i and candidate summary c. For example, if the candidate summary c contains two sentences $\{c_1, c_2\}$, and $LSS(s_i, c_1) = w_1w_2$, $LSS(s_i, c_2) = w_1w_4$, then, the union longest semantically similar subsequence of s_i, c_1, and c_2 is $w_1w_2w_4$.

We define the semantically similar subsequence as follows: given two sequences X and Y, Z is called the similar subsequence of X and Y, if Z is the subsequence of X, and there exists Y's subsequence T satisfying that all the elements at the corresponding positions between Z and T are semantically similar.

To obtain the longest semantically similar subsequence, we construct a recursive equation as follows:

$$LSS(X_i, Y_j) = \begin{cases} LSS(X_{i-1}, Y_{j-1}), \\ \quad \text{if} \cos(x_i, y_j) > \alpha, \\ \max(LSS(X_i, Y_{j-1}), LSS(X_{i-1}, Y_j)), \\ \quad \text{otherwise}. \end{cases} \quad (9)$$

In (9), $\cos(x_i, y_j)$ measures the semantically similarity between the i-th word of X and the j-th word of Y, and α is the similarity parameter. Before calculating the semantical similarity, we should map the word-to-word embedding. We can solve this recursive equation easily with dynamic programming.

Fig.4 shows the comparison of calculating ROUGE-L and our improvements S-RL with the example sentences, where we set the similarity parameter α to 0.6. The arrow lines connect the most similar bi-grams, the decimal in the left of the line is the similarity between two bi-grams, and the one in the right of the line is the NSS similarity value obtained with Algorithm 2. The results validate that S-RL is more reasonable.

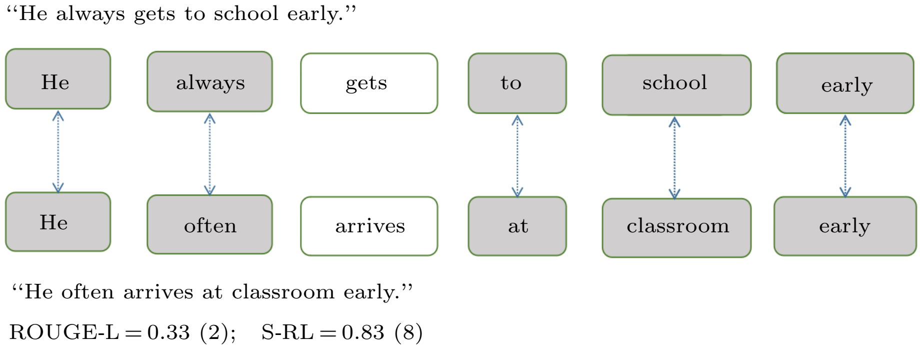

"He always gets to school early."
He
always
gets
to
school
early
He
often
arrives
at
classroom
early
"He often arrives at classroom early."
ROUGE-L = 0.33 (2); S-RL = 0.83 (8)

Fig.4. Illustration of comparing ROUGE-L with our improved S-RL.

5 Experimental Evaluation

5.1 Dataset and Evaluation Method

Dataset. We conduct extensive experiments with the benchmark dataset, TAC 2011 AESOP Task data[12]. The AESOP (Automatically Evaluating Summaries Of Peers) task is a subtask of TAC 2011 Summarization track. It aims to create an automatic scoring metric for summaries, which would correlate highly with one or more of three manual methods of evaluating summaries[54]. These three manual methods include the pyramid method[55], the overall readability[24], and the overall responsiveness[24]. Among them, pyramid is a semi-automated measure that evaluates the content of the generated summaries. Overall readability reflects the fluency and the readability of the summary. Overall responsiveness measures how well a summary adheres to the information requested, as well as the linguistic quality of the generated summaries[24], which is based on both the content and the readability.

The AESOP 2011 dataset[12] has 44 topics, and each topic has 10 documents. For each topic document set, there are four human summaries and 51 automatic summaries, which aim to summarize the content of all the documents in their topic. The data was produced by eight human summarizers who wrote reference summaries, and 51 automatic summarizers. Similar to other studies[24, 25], we treat these human summaries as the reference summary, and treat the automatic summaries as the candidate summary.

Evaluation Method. In order to evaluate the performance of our methods, we calculate the Pearson correlation scores between our improved metrics and three manual methods. We validate the performance.

- The summary level calculates the Pearson correlation scores on all summaries between our evaluation scores and manual methods scores. Note that the final evaluation score is the average of scores between each candidate summary and the corresponding four reference summaries.
- The summarizer level calculates the Pearson correlation scores on all summarizers, where each summarizer's final score is the average of all the summary-level scores that he/she has provided. For example, summarizer 1 produces eight summaries, and then we will treat the average of these eight summaries' scores (calculated by our metrics or manual methods) as the score of summarizer 1.

Other Settings. In our experiments, we implement our ENMS model by using *N-gram2vec*[2] to generate the vector representation of N-gram. We use the embedding of 300 dimensions in dense representations. We train N-gram embedding on corpus wiki2010[3].

We keep the stop words in the dataset and transfer them to the stem. The similarity parameter $\alpha = 0.6$ is the default setting, if not specified. In addition, we choose the midpoint approach as the default integration approach to solve the OOV problem.

5.2 Experimental Results and Analysis

In this subsection, we first compare our methods with the hard matching methods in the summary level. Then, we check the correlation scores with human judgements in the summarizer level. Finally, we provide the qualitative analysis about this dataset to show its latent constructions.

5.2.1 Summary-Level Comparison

We conduct correlation evaluation on 24 aspects of eight different metrics with respect to three human judgements, including ROUGE-2, ROUGE-3, ROUGE-4, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and ROUGE-SU4. Fig.5 shows all comparison results of the proposed NSM and NSS and existing hard matching methods for ROUGE and BLEU. Note that we use N to represent the N-gram, e.g., $N = 2$ represents the bigram. Figs.5(a)–5(c) show the results of ROUGE-based metrics. We can see that the proposed NSM and NSS methods have better Pearson scores for pyramid, readability and responsiveness, which indicates that the improved ROUGE methods have better correlation with human judgements.

Figs.5(d)–5(f) show that the improved BLEU-based metrics have better Pearson scores on the readability. However, they have lower scores on pyramid and responsiveness with respect to the bi-gram and the trigram. We analyze the reason and find that: the main difference between ROUGE and BLEU is that BLEU is a precision-based metric that treats the number of N-grams belonging to generated summaries as the denominator (as shown in (7) and (8)). Thus, it is difficult to measure how well a generated summary adheres to the information contained in reference summaries. Both pyramid and responsiveness are content-based evaluation (responsiveness considers both the content and the

[2] https://github.com/zhezhaoa/N-gram2vec, August 2022.

[3] http://nlp.stanford.edu/data/WestburyLab.wikicorp.201004.txt.bz2, August 2022.

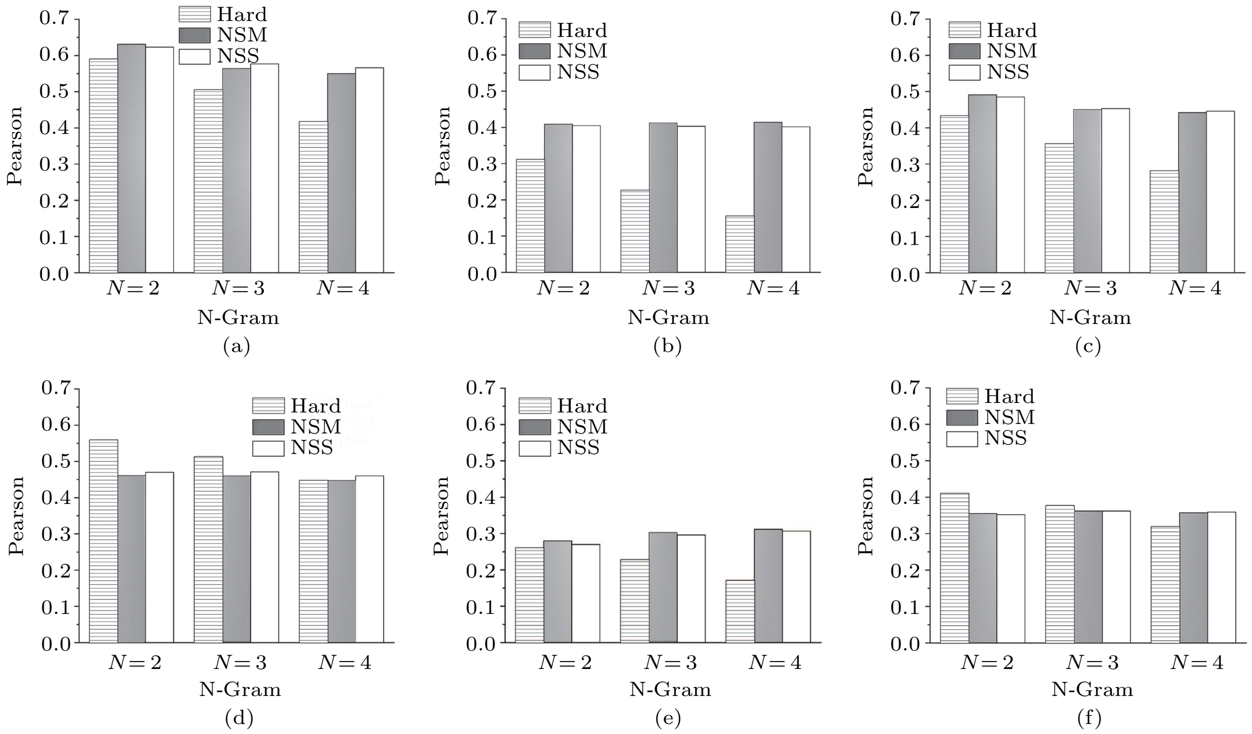


Fig.5. Comparison results on ROUGE and BLEU. (a) ROUGE pyramid. (b) ROUGE readability. (c) ROUGE responsiveness. (d) BLEU pyramid. (e) BLEU readability. (f) BLEU responsiveness.

readability). That is why BLEU has a lower correlation with pyramid and responsiveness.

We show the comparison results of S-RL, NSM-RSU4 and NSS-RSU4 over ROUGE-L and ROUGE-SU4 in Table 3. We can see that our S-RL beats ROUGE-L with a slight advantage with respect to three human judgements of pyramid, readability and responsiveness. However, our improved metrics NSM-RSU4 and NSS-RSU4 are inferior to the hard matching ROUGE-SU4 for pyramid and responsiveness. There are two possible reasons: 1) compared with the N-gram, the skip-gram is more likely to be out of vocabulary; 2) the skip-gram usually consists of some semantically unrelated words, so that the semantic information contained in the vector representation of the skip-gram is not obvious.

Table 3. Comparison Results on Improved Metrics

Method	Pyramid	Readability	Responsiveness
ROUGE-L	0.624	0.391	0.495
S-RL	**0.629**	**0.392**	**0.500**
ROUGE-SU4	**0.643**	0.368	**0.486**
NSM-RSU4	0.598	**0.409**	0.472
NSS-RSU4	0.598	**0.409**	0.471

Note: The first two rows compare ROUGE-L with our S-RL, while the last three rows compare ROURSE-SU4 with our NSM-RSU4 and NSS-RSU4. The maximum values among ROUGE metrics and our improved ones are in bold.

Meanwhile, Table 3 shows that all the NSM and NSS improved metrics have better performance with respect to the readability. It indicates that our methods are more appropriate than hard matching as for measuring the readability of the candidate summary. In Fig.5(b) and Fig.5(e), we also find that the correlation scores between our improved ROUGE and BLEU and readability increase when N becomes large. However, it is opposite for the hard matching ROUGE and BLEU. We analyze the reason and find that the more the items an N-gram has, the more the syntax information is implicitly contained in it. But for hard matching ROUGE and BLEU, the N-gram is hard to match when N is large, and even there are similar semantics between two N-grams. Therefore, our NSM and NSS improved metrics can overcome this shortcoming and utilize the advantage for large N, which makes our methods achieve better correlation with human judgements with respect to the readability.

Integrating the comparison in Fig. 5 and Table 3, 62.5% of our correlation evaluation achieves better performance in all the 24 aspects (i.e., eight metrics with respect to three human judgements). It implies that a combination of semantics with N-gram based metrics is a good choice for evaluating abstractive text summarization.

5.2.2 Summarizer-Level Comparison

In this subsection, we compare our NSM-R2 with the hard matching method ROUGE-2 and other two most recent related studies of ROUGE-WE-2 [24] and GROUGE-2 [25]. Note that both ROUGE-WE and GROUGE are methods improved from ROUGE by making use of word embedding.

Table 4 shows that our NSM-R2 achieves the best performance with respect to the readability. It is a little inferior to the state-of-the-art method GROUGE-2 with respect to pyramid and responsiveness, i.e., 0.005 on pyramid and 0.007 on responsiveness. Since our other proposed methods have similar performance to NSM-R2 (a slight increase or decrease against the state-of-the-art results), we do not display their results for the summarizer-level comparison.

Table 4. Results on Summarizer-Level Comparison

Method	Pyramid	Readability	Responsiveness
ROUGE-2	0.961	0.752	0.942
ROUGE-WE-2	0.977	0.782	0.953
GROUGE-2	**0.979**	0.787	**0.954**
NSM-R2	0.974	**0.791**	0.947

Note: The maximum value in each column is in bold.

Both the results of the summary-level comparison and the summarizer-level comparison validate that our evaluation methods are superior to existing methods with respect to readability. However, as for pyramid and responsiveness, our methods show insignificant advantages. Possible reasons may be as follows. 1) Most of summarizers are the extractive models so that they generate words from the original text and never generate semantically similar words such as "college" with "university", "good" with "well". 2) In this dataset, the hard matching method ROUGE-2 already achieves quite high correlation scores, i.e., 0.961 on pyramid and 0.942 on responsiveness. Therefore, it is very difficult to make even a little improvement given such as a high basis. 3) The number of automatic summarizers is too small in this dataset, i.e., there are only 51 automatic summarizers, and it may be far from enough to say that these high correlation scores in the summarizer level are reliable.

The above three reasons weaken the performance of all semantic-based evaluation methods. It also makes the advantages of our N-gram vector based methods not be shown clearly with respect to word embedding based methods (i.e., ROUGE-WE-2, GROUGE-2). Even so, our NSM-R2 is still better than the hard matching method ROUGE-2 and shows well correlation with human judgements. In order to further analyze the above results and find the reasons, we make a quantitative analysis of this dataset as in the following.

Quantitative Analysis. For every candidate summary in this dataset, we check how many semantically similar words it has with respect to its reference summary. Supposing that we have a word w in the candidate summary, we use the following method to find its semantically similar words: 1) calculating the cosine similarity between w and every word in its reference summary, and finding a word $w_{\max}$ that belongs to the reference summary and has the greatest similarity to w; 2) checking whether w and $w_{\max}$ are the same word, if not, we treat them as semantically similar words. We do this for every word in a candidate summary, and calculate the ratio of semantically similar words contained in the candidate summary (SSW ratio for short).

Table 5 shows the statistics in all 2 244 candidate summaries. The number of candidate summaries that contain no more than 10% semantically similar words is up to 1 898, taking a percentage of 84.6% in the dataset; the number of candidate summaries that contain more than 20% semantically similar words is only 98, taking a percentage of 4.3% in the dataset. That is why the correlation score of hard matching methods is pretty well and the advantages of our NSM and NSS methods are not obvious with respect to pyramid and responsiveness in this benchmark dataset. We believe that the proposed N-gram vector based methods will show much better performance when large datasets with more abstractive summarizers are constructed in the future.

Table 5. Results on Summarizer-Level Comparison

SSW Ratio	Count
0%–5%	981
5%–10%	917
10%–15%	223
15%–20%	25
>20%	98

5.3 Impact of Parameter α and Integration Methods

Algorithm 1 and Algorithm 2 are involved with a similarity parameter α, which serves as a threshold to determine whether two N-grams are semantically similar. Moreover, in Subsection 4.1, we provide four integration methods to solve the OOV problem. Here we study the impact of the similarity parameter and the integration methods on the performance for our proposed metrics. In this part of experiments, we use a representative metric NSM-R2 to analyze the performance. For each integration method we test, we calculate the correlation score with three manual methods.

Fig. 6 shows that the attention average approach achieves the best performance with respect to pyramid and responsiveness. The midpoint approach is close to the attention average approach and the simple multiplicative gets the worst performance compared with the other three approaches. It may be because that the attention average approach gives a larger attention weight to the important words, which is more similar to humans' reading. However, the simple multiplicative approach is much different from the original embedding of words, which loses some semantic information. As for the catenation approach, we can see from Fig.6(b) that it has no prominent performance with respect to pyramid and responsiveness, but achieves the best performance with respect to readability when α is small. The reason may be that the catenation approach keeps the order and the original embedding of all words, and thus remains more syntax information.

Meanwhile, it can be seen that all integration methods achieve their best performance when $\alpha = 0.6$ with respect to pyramid and responsiveness, and when $\alpha = 0.4$ with respect to readability. The best value of similarity parameter α is related to the properties of N-gram embedding we use (e.g., the dimension, and the training corpus). It is a bit strange that the best value of α with respect to readability is different with that of pyramid and responsiveness. In addition, when α turns bigger, the performance with respect to readability decreases. The main reasons could be the followings. When two N-grams are matched in a small similarity parameter α, they may share similar syntax information, instead of similar semantic information. Therefore, NSM and NSS pay more attention to the syntax information when the similarity parameter α is small.

6 Conclusions

In this paper, we enhanced the N-gram based metrics to better evaluate the performance of abstractive summarization models, by considering semantic information. Two types of semantic-based evaluation methods including N-gram based Semantic Matching (NSM) and N-gram based Semantic Similarity (NSS) were proposed to evaluate the quality of generated summaries considering the semantic information, which overcomes the unreasonable evaluation of existing hard matching metrics. NSM and NSS work in different ways: NSM matches N-grams directly, while NSS mainly focuses on improving the similarity measurement. Extensive experiments on the AESOP dataset validated that our methods are well correlated with human judgements, indicating that they can evaluate abstractive summarization models in a more fair way. We also proposed an N-gram representation mechanism to solve the OOV problem and serve as the basis of our methods.

Due to the lack of the proper dataset for abstractive summarization, we only tested the performance with a relatively small dataset. We would like to test our methods with other large datasets in future. Furthermore, we mainly improved the evaluation metrics from the semantic aspect in the current version. In future work, we are interested in improving the evaluation metrics in more aspects like readability. We would also like to apply our work in more application scenarios like review-based recommendation[56] and so on.

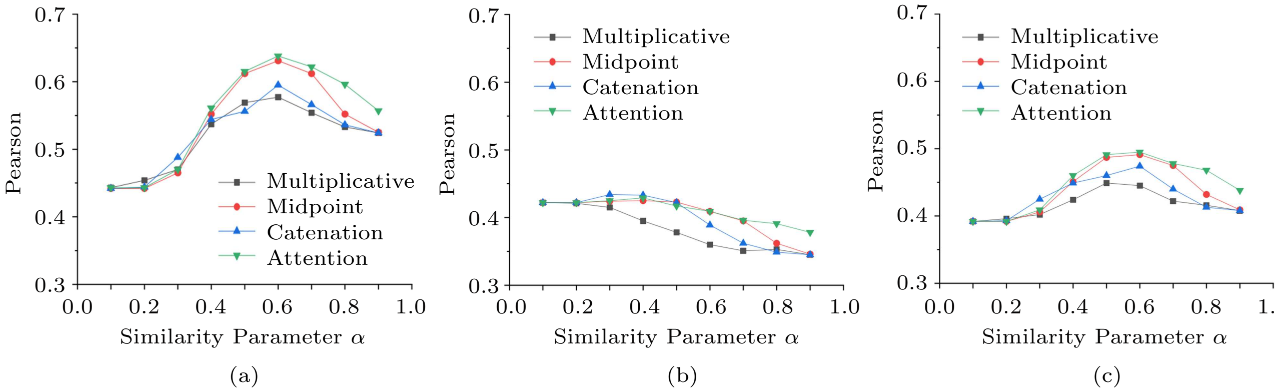


Fig.6. Impact of parameter α and the integration methods. (a) Pyramid. (b) Readability. (c) Responsiveness.

References

[1] Marujo L, Ribeiro R, Gershman A, De Matos D M, Neto J P, Carbonell J. Event-based summarization using a centrality-as-relevance model. *Knowledge and Information Systems*, 2017, 50(3): 945-968. DOI: 10.1007/s10115-016-0966-4.

[2] Qumsiyeh R, Ng Y K. Enhancing web search by using query-based clusters and multi-document summaries. *Knowledge and Information Systems*, 2016, 47(2): 355-380. DOI: 10.1007/s10115-015-0852-5.

[3] Verberne S, Krahmer E, Wubben S, van den Bosch A. Query-based summarization of discussion threads. *Natural Language Engineering*, 2020, 26(1): 3-29. DOI: 10.1017/S1351324919000123.

[4] Vougiouklis P, Elsahar H, Kaffee L A, Gravier C, Laforest F, Hare J, Simperl E. Neural Wikipedian: Generating textual summaries from knowledge base triples. *Journal of Web Semantics*, 2018, 52-53: 1-15. DOI: 10.1016/j.websem.2018.07.002.

[5] Wan X J, Luo F L, Sun X, Huang S F, Yao J E. Cross-language document summarization via extraction and ranking of multiple summaries. *Knowledge and Information Systems*, 2019, 58(2): 481-499. DOI: 10.1007/s10115-018-1152-7.

[6] Nallapati R, Zhou B W, dos Santos C N, Gulçehre Ç, Xiang B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proc. the 20th SIGNLL Conference on Computational Natural Language Learning*, Aug. 2016, pp.280-290. DOI: 10.18653/v1/K16-1028.

[7] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks. In *Proc. the 27th Int. Conference on Neural Information Processing Systems*, Dec. 2014, pp.3104-3112.

[8] Tan J W, Wan X J, Xiao J G. From neural sentence summarization to headline generation: A coarse-to-fine approach. In *Proc. the 26th Int. Joint Conference on Artificial Intelligence*, Aug. 2017, pp.4109-4115.

[9] Chopra S, Auli M, Rush A M. Abstractive sentence summarization with attentive recurrent neural networks. In *Proc. the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2016, pp.93-98. DOI: 10.18653/v1/N16-1012.

[10] Rush A M, Chopra S, Weston J. A neural attention model for abstractive sentence summarization. In *Proc. the 2015 Conference on Empirical Methods in Natural Language Processing*, Sept. 2015, pp.379-389. DOI: 10.18653/v1/D15-1044.

[11] Le Y Q, Wang Z J, Quan Z, He J W, Yao B. ACV-tree: A new method for sentence similarity modeling. In *Proc. the 27th Int. Joint Conference on Artificial Intelligence,* Jul. 2018, pp.4137-4143. DOI: 10.24963/ijcai.2018/575.

[12] Lin C Y. ROUGE: A package for automatic evaluation of summaries. In *Proc. the Workshop on Text Summarization Branches Out*, Jul. 2004, pp.74-81.

[13] Papineni K, Roukos S, Ward T, Zhu W J. BLEU: A method for automatic evaluation of machine translation. In *Proc. the 40th Annual Meeting on Association for Computational Linguistics*, Jul. 2002, pp.311-318. DOI: 10.3115/1073083.1073135.

[14] Dang H T, Owczarzak K. Overview of the TAC 2011 summarization track: Guided task and AESOP task. In *Proc. the 2011 Text Analysis Conference*, Nov. 2011.

[15] Pastra K, Saggion H. Colouring summaries BLEU. In *Proc. the 2003 EACL Workshop on Evaluation Initiatives in Natural Language Processing*, Apr. 2003, pp.35-42. DOI: 10.3115/1641396.1641402.

[16] Clement R, Sharp D. Ngram and Bayesian classification of documents for topic and authorship. *Literary and Linguistic Computing*, 2003, 18(4): 423-447. DOI: 10.1093/llc/18.4.423.

[17] Tang D Y, Wei F R, Yang N, Zhou M, Liu T, Qin B. Learning sentiment specific word embedding for Twitter sentiment classification. In *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics*, Jun. 2014, pp.1555-1565. DOI: 10.3115/v1/P14-1146.

[18] Farahani M, Gharachorloo M, Manthouri M. Leveraging ParsBERT and pretrained mT5 for Persian abstractive text summarization. In *Proc. the 26th Int. Computer Conference, Computer Society of Iran,* Mar. 2021. DOI: 10.1109/CSICC52343.2021.9420563.

[19] Huang C L, Jiang W J, Wu J, Wang G J. Personalized review recommendation based on users' aspect sentiment. *ACM Transactions on Internet Technology*, 2020, 20(4): Article No. 42. DOI: 10.1145/3414841.

[20] Calzavara S, Rabitti A, Bugliesi M. Semantics-based analysis of content security policy deployment. *ACM Transactions on the Web*, 2018, 12(2): Article No. 10. DOI: 10.1145/3149408.

[21] Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In *Proc. the Annual Conference on Neural Information Processing Systems*, Dec. 2013, pp.3111-3119.

[22] Pennington J, Socher R, Manning C. GloVe: Global vectors for word representation. In *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing*, Oct. 2014, pp.1532-1543. DOI: 10.3115/v1/D14-1162.

[23] Leung K W T, Jiang D, Lee D L, Ng W. Constructing maintainable semantic relation network from ambiguous concepts in web content. *ACM Transactions on Internet Technology*, 2016, 16(1): Article No. 6. DOI: 10.1145/2814568.

[24] Ng J P, Abrecht V. Better summarization evaluation with word embeddings for rouge. In *Proc. the 2015 Conference on Empirical Methods in Natural Language Processing*, Sept. 2014, pp.1925-1930. DOI: 10.18653/v1/D15-1222.

[25] ShafieiBavani E, Ebrahimi M, Wang R, Chen F. A semantically motivated approach to compute ROUGE scores. arXiv:1710.07441, 2017. https://arxiv.org/abs/1710.07441, Jul. 2022.

[26] Shao L Q, Zhang H, Jia M, Wang J. Efficient and effective single-document summarizations and a word-embedding measurement of quality. In *Proc. the 9th International Conference on Knowledge Discovery and Information Retrieval,* Nov. 2017, pp.114-122. DOI: DOI: 10.5220/0006581301140122.

[27] Gambhir M, Gupta V. Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review*, 2017, 47(1): 1-66. DOI: 10.1007/s10462-016-9475-9.

[28] Jiang W J, Chen J, Ding X F, Wu J, He J W, Wang G J. Review summary generation in online systems: Frameworks for supervised and unsupervised scenarios. *ACM Transactions on the Web*, 2021, 15(3): Article No. 13. DOI: 10.1145/3448015.

[29] Lin H, Bilmes J. Multi-document summarization via budgeted maximization of submodular functions. In *Proc. the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Jun. 2010, pp.912-920.

[30] Wang L, Raghavan H, Castelli V, Florian R, Cardie C. A sentence compression based framework to query-focused multi-document summarization. arXiv:1606.07548, 2016. https://arxiv.org/abs/1606.07548, Jul. 2022.

[31] Ding X F, Jiang W J, He J W. Generating expert's review from the crowds': Integrating a multi-attention mechanism with encoder-decoder framework. In *Proc. the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, Oct. 2018, pp.954-961. DOI: 10.1109/SmartWorld.2018.00170.

[32] Gerani S, Mehdad Y, Carenini G, Ng R T, Nejat B. Abstractive summarization of product reviews using discourse structure. In *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing*, Oct. 2014, pp.1602-1613. DOI: 10.3115/v1/D14-1168.

[33] Liu P, Saleh M, Pot E, Goodrich B, Sepassi R, Kaiser L, Shazeer N. Generating Wikipedia by summarizing long sequences. In *Proc. the 2018 International Conference on Learning Representations,* April 30-May 3, 2018.

[34] Tan J W, Wan X J, Xiao J G. Abstractive document summarization with a graph-based attentional neural model. In *Proc. the 55th Annual Meeting of the Association for Computational Linguistics*, July 30-August 4, 2017, pp.1171-1181. DOI: 10.18653/v1/P17-1108.

[35] Chu E, Liu P. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proc. the 2019 International Conference on Machine Learning*, Jun. 2019, pp.1223-1232.

[36] Cachola I, Lo K, Cohan A, Weld D. TLDR: Extreme summarization of scientific documents. In *Proc. the 2020 Conference on Empirical Methods in Natural Language Processing*, Nov. 2020, pp.4766-4777. DOI: 10.18653/v1/2020.findings-emnlp.428.

[37] Zhang J Q, Zhao Y, Saleh M, Liu P J. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proc. the 37th International Conference on Machine Learning*, Jul. 2020, pp.11328–11339.

[38] Kouris P, Alexandridis G, Stafylopatis A. Abstractive text summarization: Enhancing sequence-to-sequence models using word sense disambiguation and semantic content generalization. *Computational Linguistics*, 2021, 47(4): 813-859. DOI: 10.1162/coli_a_00417.

[39] Gunel B, Zhu C G, Zeng M, Huang X D. Mind the facts: Knowledge-boosted coherent abstractive text summarization. arXiv:2006.15435, 2020. https://arxiv.org/abs/2006.15435, Jul. 2022.

[40] Jones K S. Automatic language and information processing: Rethinking evaluation. *Natural Language Engineering*, 2001, 7(1): 29-46. DOI: 10.1017/S1351324901002583.

[41] Lin C Y. Looking for a few good metrics: ROUGE and its evaluation. In *Proc. the 4th NTCIR Workshop Meeting*, June 2004.

[42] Passonneau R J, Nenkova A, Mckeown K, Sigelman S. Applying the pyramid method in DUC 2005. In *Proc. the 2005 Workshop of the Document Understanding Conference*, Oct. 2005. DOI: https://doi.org/10.7916/D8TX3PVD.

[43] Hovy E H, Lin C Y, Zhou L, Fukumoto J. Automated summarization evaluation with basic elements. In *Proc. the 5th Int. Conference on Language Resources and Evaluation*, May 2006, pp.899-902.

[44] Torres-Moreno J M, Saggion H, Da Cunha I, SanJuan E, Velázquez-Morales P. Summary evaluation with and without references. *Polibits*, 2010, 42: 13-19. DOI: 10.17562/PB-42-2.

[45] Cabrera-Diego L A, Torres-Moreno J M. SummTriver: A new trivergent model to evaluate summaries automatically without human references. *Data Knowledge Engineering*, 2018, 113: 184-197. DOI: 10.1016/j.datak.2017.09.001.

[46] Radev D R, Tam D, Erkan G. Single-document and multi-document summary evaluation using relative utility. Technical Report, University of Michigan, 2007. https://www.eecs.umich.edu/techreports/cse/2007/CSE-TR-538-07.pdf, Jul. 2022.

[47] Shafieibavani E, Ebrahimi M, Wong R, Chen F. A graph-theoretic summary evaluation for ROUGE. In *Proc. the 2018 Conference on Empirical Methods in Natural Language Processing*, October 31-November 4, 2018, pp.899-902. DOI: 10.18653/v1/D18-1085.

[48] Cohan A, Goharian N. Revisiting summarization evaluation for scientific articles. In *Proc. the 10th International Conference on Language Resources and Evaluation*, May 2016, pp.806-813.

[49] Bengio Y, Ducharme R, Vincent P, Janvin C. A neural probabilistic language model. *Journal of Machine Learning Research*, 2003, 3: 1137-1155.

[50] Wieting J, Bansal M, Gimpel K, Livescu K. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 2015, 3: 345-358. DOI: 10.1162/tacl_a_00143.

[51] Passonneau R J, Chen E, Guo W, Perin D. Automated pyramid scoring of summaries using distributional semantics. In *Proc. the 51st Annual Meeting of the Association for Computational Linguistics*, Aug. 2013, pp.143-147.

[52] Zhao Z, Liu T, Li S, Li B, Du X Y. Ngram2vec: Learning improved word representations from Ngram co-occurrence statistics. In *Proc. the 2017 Conference on Empirical Methods in Natural Language Processing*, Sept. 2017, pp.244-253. DOI: 10.18653/v1/D17-1023.

[53] Mitchell J, Lapata M. Vector-based models of semantic composition. In *Proc. the 46th Annual Meeting of the Association for Computational Linguistics*, Jun. 2008, pp.236-244.

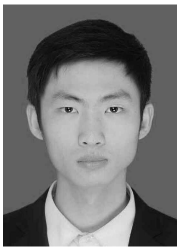
[54] Kumar N, Srinathan K, Varma V. Using unsupervised system with least linguistic features for TACAESOP task. In *Proc. the 4th Text Analysis Conference*, Nov. 2011.

[55] Passonneau R J, Chen E, Guo W W, Perin D. Automated pyramid scoring of summaries using distributional semantics. In *Proc. the 51st Annual Meeting of the Association for Computational Linguistics* (*ACL*), Aug. 2013, pp.143-147.

[56] Xia P, Jiang W, Wu J, Xiao S, Wang G. Exploiting temporal dynamics in product reviews for dynamic sentiment prediction at the aspect level. *ACM Transactions on Knowledge Discovery from Data*, 2021, 15(4): Article No. 68. DOI: 10.1145/3441451.

Jia-Wei He received his M.S. degree in computer science from Hunan University, Changsha, in 2018. Currently, he is an algorithm engineer in Hunan Jiuligong Supply Chain Co., Ltd. His main research interests include data mining and text analysis.

Wen-Jun Jiang received her Bachelor's degree in computer science from Hunan University, Changsha, in 2004, her Master's and Ph.D. degrees in computer software and theory from the Huazhong University of Science and Technology, Wuhan, and the Central South University, Changsha, in 2007 and 2014, respectively. Currently, she is a professor and Ph.D. supervisor of Hunan University, Changsha, and a senior member of CCF. Her main research interests include data mining and social network analysis.



Guo-Bang Chen received his M.S. degree in computer science from Hunan University, Changsha, in 2022. His main research interests include machine learning and social network analysis.

Yu-Quan Le received his M.S. degree in computer science from Hunan University, Changsha, in 2018. He is currently a Ph.D. candidate in computer science from Hunan University, Changsha. His main research interests include machine learning and data mining.

Xiao-Fei Ding received his M.S. degree in computer science from Hunan University, Changsha, in 2018. Currently, he is an algorithm engineer in Tencent. His research interests include recommendation system and natural language processing.