Review Summary Generation in Online Systems: Frameworks for Supervised and Unsupervised Scenarios

WENJUN JIANG*, JING CHEN, XIAOFEI DING, Hunan University, China
JIE WU, Temple University, USA
JIAWEI HE, Hunan University, China
GUOJUN WANG, Guangzhou University, China

In online systems, including e-commerce platforms, many users resort to the reviews or comments generated by previous consumers for decision making, while their time is limited to deal with many reviews. Therefore, a review summary, which contains all important features in user-generated reviews, is expected. In this paper, we study "how to generate a comprehensive review summary from a large number of user-generated reviews." This can be implemented by text summarization, which mainly has two types of extractive and abstractive approaches. Both of these approaches can deal with both supervised and unsupervised scenarios, but the former may generate redundant and incoherent summaries, while the latter can avoid redundancy but usually can only deal with short sequences. Moreover, both approaches may neglect the sentiment information. To address the above issues, we propose comprehensive **Review Summary Generation** frameworks to deal with the supervised and unsupervised scenarios. We design two different preprocess models of re-ranking and selecting to identify the important sentences while keeping users' sentiment in the original reviews. These sentences can be further used to generate review summaries with text summarization methods. Experimental results in seven real world datasets (Idebate, Rotten Tomatoes Amazon, Yelp and three unlabelled product review datasets in Amazon) demonstrate that our work performs well in review summary generation. Moreover, the re-ranking and selecting models show different characteristics.

CCS Concepts: • **Computing methodologies** → **Natural language generation**; • **Information systems** → **Online shopping**;

Additional Key Words and Phrases: user-generated review, review summary generation, text summarization, supervised and unsupervised scenarios

ACM Reference Format:
Wenjun Jiang*, Jing Chen, Xiaofei Ding, Jie Wu, Jiawei He, and Guojun Wang. 2021. Review Summary Generation in Online Systems: Frameworks for Supervised and Unsupervised Scenarios. *ACM Trans. Web* 1, 1, Article 1 (January 2021), 33 pages. https://doi.org/10.1145/3448015

* Wenjun Jiang is the Corresponding Author.
The preliminary results of this manuscript were accepted to the 15th IEEE International Conference on Ubiquitous Intelligence and Computing (IEEE UIC 2018), entitled with "Generating Expert's review from the Crowds': Integrating A Multi-Attention Mechanism with Encoder-Decoder Framework." It exploits multi-attention mechanism with Encoder-decoder framework to generate expert's review, and it mainly focuses on the supervised scenarios.
Authors' addresses: Wenjun Jiang*, Jing Chen, Xiaofei Ding, Hunan University, Changsha, China, jiangwenjun@hnu.edu.cn,jessicachan@hnu.edu.cn,ding_xiaofei@outlook.com; Jie Wu, Temple University, Philadelphia, USA, jiewu@temple.edu; Jiawei He, Hunan University, Changsha, China, j452577895@163.com; Guojun Wang, Guangzhou University, School of Computer Science and Educational Software, Guangzhou, 510006, China, csgjwang@gzhu.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Association for Computing Machinery.
1559-1131/2021/1-ART1 $15.00
https://doi.org/10.1145/3448015

1 INTRODUCTION

Online reviews usually contain users' true feelings and opinions about products or services, which provide a valuable reference for other users' decision making as well as for improving the quality of products/services. Online websites (e.g., e-commerce or social networking platforms) have numerous reviews and each review contains multiple sentences. However, users don't have enough time and patience to read too many reviews. Therefore, exploring useful information in user-generated reviews and generating a comprehensive review summary will benefit customers greatly. We call this *Review Summary Generation*. In this paper, we strive to propose review summary generation frameworks to solve both supervised scenarios (i.e., standard references are available) and unsupervised scenarios (i.e., standard references are unavailable).

In general, the task can be implemented by the text summarization technique [62], which generates a short summary for the original long text, while keeping most of the important information. This has two types of approaches to mitigate the information overload: the extractive approach and the abstractive approach. The former produces summaries by selecting important sentences from the original text [34]; it can deal with both supervised and unsupervised scenarios, but it may generate incoherent summaries with redundant information because the sentences selected from the original text are often in different positions. Meanwhile, the latter approach is close to the way humans write summaries (e.g., generating new phrases) [60]. However, it mainly deals with short texts and only supervised scenarios until the most recent MeanSum model [11], which makes a breakthrough and can generate summaries with Encoder-decoder framework in an unsupervised way.

Important sentences are the essential basis of a good summary. Existing work (e.g. [18, 38]) usually measures the similarity between sentences and selects the sentences that have the highest similarity with the standard references. Since they rely on the standard references and they use hard matching at the string level, they are unsuitable for the unsupervised scenarios and some supervised scenarios where similar meanings are expressed with different words.

Supervised summarization method (e.g., the encoder-decoder framework) can be used in several applications, including machine translation [54], image captioning [35, 40], recommendation system [16, 26, 33, 45], social networks [27, 58, 63], and abstractive summarization [49] for short texts. However, it is difficult to deal with long text sequences. In order to make the encoder-decoder framework work well in more complicated real world scenarios, it is necessary to design a comprehensive preprocess strategy. Existing research on text summarization usually produces a summary using extractive approaches [14]. Some research tries to employ abstractive approaches [59], but usually neglects the sentiment information of individual reviews.

Furthermore, in many real world scenarios (e.g., the e-commerce platforms) there is a lack of human written summaries used as a gold standard. Thus, the datasets in those scenarios cannot be used to train the supervised model. Hence, it is necessary to consider an unsupervised review summary generation method.

Our motivations. Based on the above analysis, our motivations are threefold: (1) Effectively identify important sentences and reviews that cover the most useful information and aspects. (2) Attempt to apply a supervised method (encoder-decoder framework) in more complicated scenarios. (3) Use an unsupervised review summary generation method to deal with more scenarios for items (e.g., popular e-commerce products) that usually have a lot of reviews.

Keeping the tasks of review summary generation and the technique issues of text summarization in mind, we propose frameworks that exploit the strengths of extractive and abstractive summarization approaches. We conduct review summary generation with two steps: pre-processing and summary generation. We propose two different pre-process models of re-ranking and selecting;

meanwhile, we propose two summary generation methods for supervised and unsupervised scenarios. For the pre-processing: we (1) *identify important sentences*, by measuring the semantic importance of sentences in original reviews and re-ranking these sentences according to their importance; we (2) *select a review subset based on aspect-level analysis* by analyzing the aspects in each original review and selecting the reviews that cover more aspects with fewer sentences (i.e., high coverage and efficiency). For summary generation: we (1) *proposing two attention mechnisms with Encoder-decoder framework* to generate the review summary for supervised scenarios, and we (2) *eliminate the redundancy* of the selected review set or sentences to generate a summary and *calculate the aspect weights* for the unsupervised scenarios. Our contributions are summarized as follows:

1. *We develop two comprehensive pre-process strategies to identify important sentences or reviews: Re-ranking model for sentences and Selecting model for review subsets.* The re-ranking model is used to re-rank the sentences of the reviews by their semantic similarity and user's sentiment. The selecting model is used to select a subset of reviews covering as many aspects as possible. Note that our selecting model can adaptively determine the size of review subsets.
2. *We apply the encoder-decoder generation model to the review summary generation task for the supervised scenarios.* Our solution combines extractive and abstractive approaches. Both the re-ranking and the selecting model can serve as the input for our encoder-decoder generation model if the dataset has the reference.
3. *We propose an unsupervised method to deal with unsupervised scenarios that have no human written summaries as standard references.* The input of our unsupervised methods are the sentence or review subset of the re-ranking and selecting models. We filter the sentences that contain the same aspects and weight the aspects from a global view. This can keep the summary concise, non-redundant, and authoritative.
4. *We conduct extensive comparative experiments in seven real world data sets: Idebate, Rotten Tomatoes, Amazon, Yelp and three unlabelled product review datasets in Amazon.* The first four datasets have human written summaries and the other three do not. The results indicate that our method (for both supervised and unsupervised scenarios) performs better than other baselines.

We organize the paper as follows. Section 2 describes related work. Section 3 formulates the problem we solve in this paper. We describe our re-ranking model in Section 4.1 and our selecting model in Section 4.2. Section 5 provides the review summary generation model in detail. We present the experiments and analyses in Sections 6 and 7. Finally we conclude this paper in Section 8.

2 RELATED WORK

We briefly review the literature in this section and point out how it connects with or differs from our work.

Research on sentence/review ranking. This involves re-ranking the original text and selecting important sentences within the text to generate a summary. TextRank [38] and LexRank [18] re-rank words or sentences through the PageRank algorithm [43], which uses the literal similarity of two sentences (nodes) to define the edge in a graph. Radev et al. [47] construct a pseudo-sentence of the texts called centroids according to the scores of all of the sentences. Many unsupervised approaches are based on high sentence frequency [21] and the basic of high information frequency is semantic similarity. Matt [29] uses the Word Mover¡¯s Distance (WMD) based on word embeddings to measure the similarity between two text documents. Our work is mainly related to Sanjeev et al. [5], who provide a new sentence embedding method that achieves a better performance than other approaches on many textual similarity tasks.

Research on review selection. Some review selection approaches consider the product's aspects to identify a subset of the most helpful non-redundant reviews. Nguyen et al. [42] propose to

select an efficient set of reviews that covers the set of tips (micro-reviews), given that there is a collection of reviews and a collection of tips. Lappas et al. [31] propose the selection of a set that represents the majority opinion on each feature. Lappas et al. [30] formalize the review selection task as a combinatorial optimization problem and design some effective algorithms. In the above works, the aspects (features) are pre-defined artificially and they may be not suitable for items with different categories. Our work is close to [42], however, it relies on the micro-reviews which may not be available in many scenarios. To address this issue, we propose using aspect extraction methods to get aspects instead of micro-reviews. In addition, our work has no restriction on the number of selected reviews.

Research on aspect extraction. Hu et al. [25] propose to extract product aspects through association mining, and opinion terms by augmenting a seed opinion set using synonyms and antonyms in WordNet [39]. Yoshihiko et al. [52] use an Aspect-based Sentiment Analysis model to extract opinion phrases from reviews. In addition, syntactic relations are further exploited for aspect/opinion extraction [46]. Although the above models are unsupervised, they heavily depend on predefined rules for extraction; they are also restricted to specific types of part-of-speech (POS) tags for product aspects and opinions. He et al. [24] present a neural approach with the aim of discovering coherent aspects. Zhou et al. [65] present an opinion mining system for Chinese microblogs called CMiner and they use the aspect extraction technique to get the opinion.

Research on review sentiment analysis. A sentiment analysis task identifies the sentiment polarity of each sentence. Angelidis et al. [4] predict text sentiment based on weak supervision and they predict the sentiment of text segments based on the supervised neural model [3]. Nikolaos et al. [44] also adopt a weak supervised learning model on multi-aspect sentiment analysis. Many unsupervised methods adopt lexical knowledge for sentiment detection [37]. Emitza et al. [23] use a lexical sentiment extraction tool to get the sentiment score of each APP review. Bollegala et al. [8] adopt sentiment sensitive embeddings to deal with cross-domain sentiment classification. Kotzias et al. [28] propose an approach to predict labels for sentences given labels for reviews, using a convolutional neural network to infer sentence similarity. Many works extract negative or positive opinions from user-generated reviews through sentiment analysis techniques [14, 23]. Zhao et al. [64] propose a novel deep learning framework for product review sentiment classification that exploits widely available ratings as weak supervision signals. Xia et al. [61] propose the aspect-based sentiment dynamic prediction models, which can dynamically capture and exploit the change patterns of users and items at the aspect level, with uniform or non-uniform time intervals. Kim et al. [50] provide a survey on aspect-level sentiment analysis. Our work employs sentiment analysis techniques to keep each review's sentiment at the aspect level.

Research on supervised review summary generation. Alexandra Balahur et al. [7] present a feature-driven opinion summarization method for customer reviews on the web by identifying general features, product specific features, and feature attributes. Gu et al. [22] incorporate copying into neural network-based Seq2Seq learning and they propose CopyNet with an encoder-decoder structure. It can choose subsequences in the input sequence and put them at proper places in the output sequence. Abigail et al. [51] present a hybrid pointer generator architecture with coverage to generate summary in supervised scenarios. Different from them, we focus on summary generation for both supervised and unsupervised scenarios, which includes the preprocessing module, supervised generation summary model, and unsupervised generation summary model respectively.

Research on unsupervised review summary generation. Yu et al. [62] propose a phrase-based summarization algorithm for the task of product review summarization. Amplayo et al. [1] propose the Condense-Abstact (CA) framework for opinion summarization that eliminates the work of pre-selecting salient content. Different from them, our unsupervised summarization focus

Table 1. Symbol Definitions

symbol	definition
R	the set of reviews for an item
S	the set of selected sentences
R^*	the subset of selected reviews
r	a review in R
w	a word in a sentence
y	review summary
e_w	word embedding
v_s	sentence embedding
q	the similarity score
A	the set of aspects for a type of items
a	an aspect

on the aspect-based extraction and aims to select the reviews with high coverage and few sentences. Elsahar et al. [17] propose a self-supervised setup that considers an individual document as a target summary for a set of similar documents. Coavoux et al. [12] present an unsupervised opinion summarization method, based on language modelling and aspect-based clustering. What's more, Yoshihiko et al. present an abstractive opinion summarization framework, which uses a spect-based sentiment analysis model to extract opinion phrases from reviews [53]. With the development of neural networks, a lot of research shows the application of the encoder-decoder model to generate the summary. Chu et al. [11] propose an end-to-end, neural model architecture (MeanSum) to perform unsupervised abstractive summarization. Similarly, Brazinskas [9] proposes an abstractive opinion summarization framework which can force the novelty to be minimal, and produce a text reflecting consensus opinions. Amplayo et al. [2] propose a summarization model that introduces explicit denoising, partial copy and discrimination modules. These works focuses on the abstractive summary generation, while our unsupervised summarization task focuses on high quality review selection and extraction to generate summary.

Different from existing works, our work: (1) tries to propose general review summary generation frameworks for both supervised and unsupervised scenarios, (2) tries to promote the encoder-decoder framework in more complex scenarios, and (3) tries to deal with e-commerce products which have many reviews.

3 PROBLEM DEFINITION

In this section, we describe the problem we solve and our solution overview. The notations used in this paper are shown in Table 1.

3.1 The Review Summary Generation Problem

Given a set of reviews R of some item (e.g., a product, a movie, etc.), the tasks of review summary generation are (1) to select some important reviews R^* or some important sentences S and (2) to generate a comprehensive **review summary** y according to the selected sentences S or reviews R^* as follows:

$$y = f(S \ or \ R^*) \tag{1}$$

where f is our proposed frameworks.

3.2 Solution Overview

We propose frameworks to generate summaries from user-generated reviews that can deal with both supervised and unsupervised scenarios. Fig. 1 shows the overview of our solution. It has two main steps: pre-processing and summary generation. The former is to get the important sentences or the optimal subset of reviews by the re-ranking or the selecting models, respectively. The latter generates summaries for the supervised and unsupervised scenarios. The main four modules are as follows.

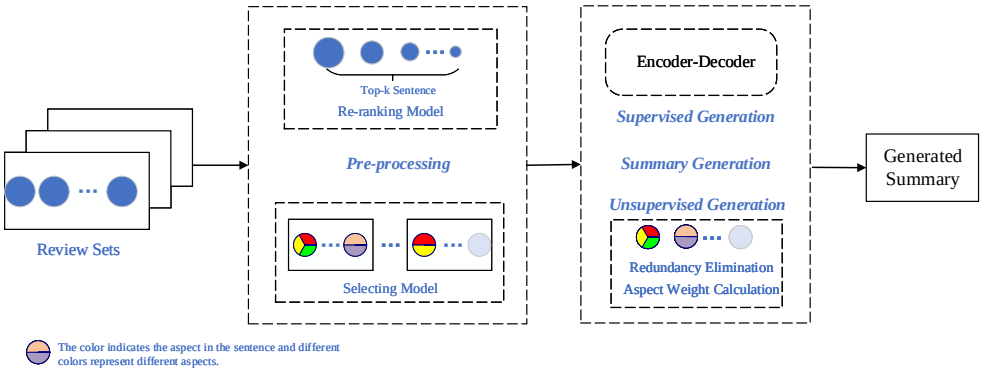


Fig. 1. The frameworks of review summary generation

(1) **Re-ranking model**: This module consists of similarity measurement and sentiment analysis. The former utilizes the semantic similarity to measure the importance of sentences (SIM-attention), while the latter utilizes user's sentiment (SA-attention).

(2) **Selecting model**: The selecting module utilizes a greedy algorithm (EMC-RS) to select the reviews that cover as many aspects as possible with as few sentences as possible. Note that the length of the selected review set is self-adaptive in our method.

(3) **Supervised summary generation**: This module adopts a supervised method to generate review summaries. We integrate two attention mechanisms (i.e., SIM-attention and SA-attention) with the encoder-decoder framework.

(4) **Unsupervised summary generation**: This module adopts an unsupervised method to generate review summaries. We conduct review selection, redundancy elimination, and aspect weight calculation to make the summary more concise.

4 PRE-PROCESSING

In this section, we describe the details of two pre-processing models, i.e., the re-ranking model and the selecting model. In traditional sequence-to-sequence models, the length of a sequence is often limited as input. Once the length is too long, it is hard to train the model and the accuracy declines. Meanwhile, in unsupervised scenarios, it is challenging to generate a summary directly from so many reviews. Therefore, we propose two pre-process strategies.

4.1 The Re-ranking Model

The re-ranking model is used to rearrange sentences of the original review set and select top-k sentences that include more important information. It has four steps: (1) sentence embedding, (2) similarity measurement, (3) sentiment analysis, and (4) sim-sa-ranking.

Table 2. An example review, its three semantic levels, and aspects with sentiment.

Product: Mobile Phone
Example Review: I hate this phone. I was drawn to it by it's appearance and size but what a horrible disappointment.
Sentence Level: "I hate this phone", "I was drawn to it by it's appearance and size but what a horrible disappointment."
Phrase Level: "I", "this phone", "it's appearance", "size", "what a horrible disappointment"
Word Level: "I", "phone", "appearance", "size"
Aspect: "$phone^-$(negative)", "$appearance^-$", "$size^-$"

4.1.1 Sentence Embedding. To deal with the source text in reviews, we use a sentence embedding method [5] to map the input sentence to a vector representation v_s, i.e., $(v_1, ..., v_k)$. The implementation of sentence embedding is based on the word vector. To get a high quality vector representation of the word, we use Wikipedia [1] as an external corpus for the first-stage training. We apply the Word2vec tool [2] as the training model, to guarantee that preseted word-embedding is available for further process.

The sentence embedding v_s for all sentences of a review is calculated as follows:

$$v_s = \frac{1}{|s|} \sum_{w \in s} \frac{t}{t + p(w)} e_w, \tag{2}$$

where $p(w)$ is the unigram probability of word w in the entire corpus and t is a scalar. Each word w in the vocabulary has a vector e_w using pre-trained embeddings learned from Wikipedia.

4.1.2 Similarity Measurement. To select important sentences from reviews, we adopt an improved TextRank algorithm. Traditional TextRank [38] uses hand-matching to calculate the similarity, which fails to identify the real relation between two sentences. Therefore, we improve the original TextRank to measure the importance of sentences based on semantic similarity and we select top-k sentences. We design three methods to calculate the semantic similarity in three levels, (i.e., sentence level, phrase level, and word level). We use the different levels to represent the review. Table 2 shows the sentence level, the phrase level, and the word level semantics.

Sentence-level: The semantic similarity of two sentences s_i and s_j is defined as follows:

$$Similarity(s_i, s_j) = cos(v_i, v_j), \tag{3}$$

where v_i is the sentence embedding of the sentence s_i.

Phrase-level: We calculate the semantic similarity of **phrases** with Algorithm 1. First, the function *GetChunk* (lines 2-3) extracts noun phrases from the input. For each noun phrase n_i in $S1$, we calculate the semantic similarity with each noun phrase k_j in S_2 using the function *GetSimilarity*. The phrase embedding, similar to the sentence embedding, maps the noun phrase to a vector representation. *GetSimilarity* calculates the semantic similarity between phrases by cosine similarity of their phrase vectors. We keep the highest $Score_i$ as the similarity score of the noun phrase n_i (lines 3-7). Each noun phrase gets a score and only the scores above ε are kept in the set D_1. Finally, we use the average score to define the semantic similarity of two sentences (lines 8-10).

The function *GetChunk* can be implemented by the Python natural language processing toolkit *spaCy* [3]. For example, the *noun_ chunks* from the natural language processing library of spacy can extract noun phrases.

[1]https://www.wikipedia.org
[2]https://github.com/stanfordnlp/GloVe
[3]https://github.com/explosion/spaCy

Algorithm 1 Similarity Calculation

Input: Sentence s_1, Sentence s_2
Output: Similarity score q_{12}

1: Let $S1, S2$ be the sets of phrase in $s1$ and $s2$, and $D1$ be the set of similarity scores.
2: $S1 \longleftarrow GetChunk(s_1)$
3: $S2 \longleftarrow GetChunk(s_2)$
4: **for** each n_i in $S1$ **do**
5: Let P_i be the set of similarity scores between the *noun* in $S1$ and $S2$
6: **for** each k_j in $S2$ **do**
7: $score_{ij} \longleftarrow GetSimilarity(n_i, k_j)$
8: $P_i \longleftarrow P_i \cup \{score_{ij}\}$
9: $Score_i \longleftarrow Max(P_i)$
10: **if** $Score_i > \varepsilon$ **then**
11: $D_1 \longleftarrow D_1 \cup \{Score_i\}$
12: $q_{12} \longleftarrow \sum_{d_i \in D_1} d_i / |D_1|$
13: **return** q_{12}

Word-level:We can also calculate the semantic similarity of **words** with Algorithm 1. The only difference from that of *Phrase-level* is that here we extract words (Nouns and Adjectives) from the input using the function *GetChunk* (lines 2-3).

In our model, the calculated similarity is defined as the weight q_{ij} associated to the edge connecting vertices V_i and V_j. The score of a vertex is defined as follows:

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{q_{ji}}{\sum_{V_k \in Out(V_j)} q_{jk}} WS(V_j), \tag{4}$$

where d is a damping factor in [0,1], usually set to be 0.85. $In(V_i)$ is the set of vertices that point to V_i and $Out(V_i)$ is the set of vertices that V_i points to.

Iteratively, the importance score associated with each vertex is calculated by Eq. 4 until a given threshold is achieved. In the process, sentences are sorted in descending order by score. The top-k sentences are selected for summary generation.

In addition, the importance scores are used to define Sim-Attention, as follows:

$$\gamma_i = softmax(WS(V_i)), \tag{5}$$

where $softmax$ is employed to normalize the weights and it is defined as $softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$.

4.1.3 Sentiment Analysis. This is review summary generation based on users' reviews is a special task of text summarization. Unlike documents such as a news story, broadcast shows or scientific article, user-generated reviews consist of users' opinions and sentiments. However, traditional text summarization models do not take the sentiment of users into account. Hence, to generate a professional review that also preserves the sentiment of the text, Sa-Attention is proposed to select the sentences that contain users' sentiments about a product.

Given the sentence s_i, we utilize SentiWordNet to determine the sentiment polarity δ_i. SentiWordNet [6] is a publicly available lexical resource for sentiment analysis and assigns to each word three sentiment scores: positivity, negativity, and objectivity. Sentiment lexicon plays a key role in unsupervised sentiment analysis technologies. It can match the sentiment words of a sentence and give a sentiment polarity. To preserve the sentiment of the text, the sentences including more

positive or negative words have to be selected. Hence, the sentiment polarity δ_i of sentence s_i is calculated by:

$$\delta_i = softmax\left(\frac{Count_{pos}(s_i) + Count_{neg}(s_i)}{Count(s_i)}\right) \tag{6}$$

where $Count_{pos}(s_i)$, $Count_{neg}(s_i)$ and $Count(s_i)$ are the counts of positive words, negative words and all words, respectively.

4.1.4 SIM-SA-Ranking. After acquiring the importance score and sentiment polarity of each sentence, our re-ranking model needs to select the important sentences that preserve as much sentiment as possible, so the original reviews are re-ranked by the attention value η:

$$\eta_i = \gamma_i \delta_i, \tag{7}$$

where γ_i, δ_i and η_i are Sim-Attention, Sa-Attention, and Sim-Sa-Attention scores of sentence i, respectively.

Finally, the sentences of reviews are sorted with their attention scores and the top-k sentences with the highest attention scores are selected as the input for our generation model.

4.1.5 The Time Complexity Analysis. In terms of time complexity, Algorithm 1 takes the time complexity of $O(|N|^2)$, where $|N|$ is the number of nouns in product reviews. The re-ranking model depends on the number of iterations and sentences. The worst-case time complexity is $O(k_1|R|^2|r|^2|N|^2)$, where $|R|$ and $|r|$ are the number of original reviews and maximum sentences for each review respectively, and k_1 is the number of iterations until convergence. As the number of review sentences increases, the time cost of the re-ranking model increases significantly, and the model converges more slowly.

4.2 The Aspect-based Selecting Model

In daily life, people tend to browse valuable reviews before purchasing a product. However, the number of product reviews is too large. Moreover, valuable comments may have different meanings for different person [61]. Taking the cellphone for instance, some people may pay more attention to the performance, while some others may pay more attention to the appearance. This indicates that different people care about different aspects. Therefore, we propose an aspect-based review selection model in this paper in order to select the reviews that cover more aspects with fewer sentences. It has three steps: (1) aspect extraction, (2) aspect analysis, and (3) review subset selection.

4.2.1 Aspects Extraction. We first use the TextRank algorithm [38] to extract keywords of original reviews. Mihalcea et al. [38] prove that the TextRank algorithm succeeds in identifying the most important words in text based on information exclusively drawn from the text itself. Then we keep the nouns of these keywords as the aspects (i.e., $\{a_1, a_2, ..., a_k\}$) of the product.

Based on the key nouns aspects, we further consider the aspect polarity. In real life, there are two possible sentiment polarities (i.e., non-negative and negative) about an aspect a. To better distinguish the fine-grained aspect information, we take the same aspect with different polarities as two different aspects. We denote the aspect of positive (i.e., non-negative) sentiment on a as a^+, and the negative one as a^-. Given the collection of n reviews of one product (denoted as $R = \{r_1, ..., r_n\}$), we use $A = \{a_1{}^+, a_1{}^-, ..., a_k{}^+, a_k{}^-\}$ to denote all the aspects of the product. Table 2 shows an example review and its aspect with sentiment.

4.2.2 Aspect Analysis. We conduct aspect analysis on the coverage and efficiency of reviews.

To determine whether a review covers an aspect, we define a mapping function between review sentences and aspects. Each review r is composed by a set of $|r|$ sentences $r = \{s_1, ...s_{|r|}\}$. All

sentences of all reviews R about one product are defined as μ_R. The mapping function is defined as follows:

$$F(s, a^p) = \begin{cases} 1 & if\ s\ cover\ a^p \\ 0 & otherwise \end{cases} \tag{8}$$

In Eq. 8, $s \in \mu_R$, $a^p \in A$ and $p \in \{+, -\}$. We consider that each noun in the review may be an aspect of the product. If the noun in sentence s and the aspect a in A have high a semantic similarity, we calculate sentiment polarity of a sentence, using the *TextBlob of Python library* to obtain the sentiment value. If the value ≥ 0, then the aspect is denoted as a^+, otherwise, the aspect is denoted as a^-. Finally, we can say that sentence s covers the aspect a^p (i.e., a^+ or a^-). Note that the semantic similarity between two words is calculated by the cosine similarity of their word vectors.

Based on the above mapping function, we can define aspect coverage and efficiency of a review. We use the definitions of coverage and efficiency by [42]. For each review r, the set of aspects A_r that are covered by at least one sentence in r is defined as follows:

$$A_r = \{a \in A : \exists s \in r, F(s, a) = 1\} \tag{9}$$

The aspect coverage of review r, i.e.,$Cov(r)$, is defined as the number of aspects covers by the review r, i.e.,$\mid A_r \mid$. Moreover, the coverage of the collection of reviews R^* selected from R is defined as:

$$Cov(R^*) = \mid \cup_{r \in R^*} A_r \mid \tag{10}$$

The efficiency of the review r is defined as the fraction of sentences in r that covers at least one aspect. Formally:

$$Eff(r) = \frac{\mid \{s \in r : \exists a \in A, F(s, a) = 1\} \mid}{\mid r \mid} \tag{11}$$

Our selecting model is designed to find the review set with high coverage and high efficiency.

4.2.3 Review Selection. We propose the aspect-based review selection problem and provide the EMC-RS algorithm to solve it.

The Problem. Given a set of reviews R, a set of aspects A, the mapping function F, and parameters α and β, the goal of review selection is to select a subset $R^* \subseteq R$ such that the aspect coverage $Cov(R^*)$ is maximized, while the efficiency of any review in R^* is at least α.

The hardness. The review selection problem of maximizing coverage with efficiency constraint can be simplified to maximize the coverage, while keeping each selected review have a minimal efficiency of α. It has been proved to be NP-hard in [57].

The submodular property. Suppose R_1 and R_2 are two review sets: $R_1 \subseteq R_2$, and a review r: $r \notin R_2$. First, we have $Cov(R_1) \leq Cov(R_2)$, indicating the monotonicity of the coverage function. Next, the marginal gain of adding r to R_1 is $Cov(R_1 \cup \{r\}) - Cov(R_1)$. Since R_2 contains everything in R_1 and even more others, it is self-evident that the marginal gain of adding r to R_2 is not larger than the marginal gain of adding r to R_1. That is, $Cov(R_1 \cup \{r\}) - Cov(R_1) \geq Cov(R_2 \cup \{r\}) - Cov(R_2)$. This proves that the coverage function is submodular, which has also been proved in [57]. According to [41], based on the submodularity and monotonicity, the problem of maximizing the coverage has a greedy solution with an approximation ratio (e-1)/e.

The EMC-RS Algorithm. Our review selection problem is to maximize the aspect coverage by the efficiency constraint and our goal is to select the review r^* that has the highest gain-to-cost ratio at each iteration.

We use Eq. 12 to define the gain of selecting a review r, i.e., $gain(r)$, which is related to the coverage $Cov(r)$. When a review r is added to the selected Review Set R^*, the coverage $Cov(R^*)$ may be increased and the gain may be increased accordingly.

$$gain(r) = Cov(R^* \cup r) - Cov(R^*) \tag{12}$$

Algorithm 2 EMC-RS: The aspect-based EMC-review selection algorithm

Input: Set of reviews R and aspect set A; Efficiency function Eff; parameters α, β
Output: A set of reviews $R^* \subseteq R$

1: $R^* \longleftarrow \varnothing$
2: **while** $Cov(R^*) <| A |$ **do**
3: Initialize two dictionaries *gain* and *cost* to record the gains and costs of reviews
4: **for** r_i in R **do**
5: **if** $Eff(r_i) < \alpha$ **then**
6: $R \longleftarrow R \setminus r_i$
7: **else**
8: $gain(r_i) \longleftarrow Cov(R^* \cup r_i) - Cov(R^*)$
9: $cost(r_i) \longleftarrow \beta(1 - Eff(r_i)) + (1 - \beta)$
10: **if** $max(gain(r_i) = 0)$ **then**
11: **return** R^*
12: $r^* \longleftarrow arg\ max\ gain(r_i)/cost(r_i)$
13: $R \longleftarrow R \setminus r^*$
14: $R^* \longleftarrow R^* \cup r^*$
15: **return** R^*

The cost of selecting a review r, $cost(r)$, is defined as follows:

$$cost(r) = \beta(1 - Eff(r)) + (1 - \beta) \tag{13}$$

When $\beta = 1$, the effect of the efficiency on the review selection is maximized. When $\beta = 0$, the review selection is not affected by the efficiency of the reviews, only by the coverage. That is, when $cost(r) = 1$, gain-to-cost ratio is only related to $gain(r)$.

We utilize a greedy algorithm to deal with the review selection problem, i.e., the EMC-RS algorithm in Algorithm 2. At each iteration, if the efficiency of a review r is lower than α, it will be removed from R (lines 5-6). Otherwise, we calculate the gain $gain(r)$ and the cost $cost(r)$ for r (lines 8-9). The gain represents the increase in coverage when adding a new review r to the subset R^*, and the cost reflects its inefficiency. At each iteration, the review r^* that has the highest **gain-to-cost** ratio is added to R^* (line 12). We end the process when there is no more gain (lines 10-11).

It is worth noting that our EMC-RS algorithm is similar to that in [42], which mainly selects a fixed number of reviews (e.g., 10) with the text length ≤ 140 for each (they call it a micro-review), and it aims to select reviews that cover as many micro-reviews as possible with the fewest sentences. Our differences lie in three aspects. (1) We conduct fine-grained aspect-level analysis instead of relying on micro-reviews. (2) We strive to get a self-adaptive size of the set $R^* \subseteq R$. That is, the size of R^* can be different for different items, while [42] selects a fixed number of reviews. (3) Last but not least, our focus in this paper is review summary generation. The selected reviews by EMC-RS algorithm will be further used to generate a summary.

4.2.4 The Worst-case Time Complexity Analysis. The time complexity of Algorithm 2 depends on the number of reviews and aspects. Line 2 is the cut-off condition for EMC-RS, taking the maximum time of $O(|A|)$. Lines 4 to 14 calculate the *gain* and *cost* for each review, taking the time complexity of $O(|r||R|)$. The worst-case time complexity of Algorithm 2 is $O(|A||r||R|)$.

It is worth noting that the selecting model has additional cost of extracting aspects in product reviews. It takes time complexity of $O(k_2|N|^2)$, where k_2 is the number of iteration until convergence. Therefore, the selecting model has a worst time complexity of $O(k_2|N|^2 + |A||r||R|)$, where $|N|$

is the number of nouns in product reviews, $|R|$ and $|r|$ are the number of original reviews and maximum sentences for each review, respectively.

4.3 Summary for Pre-processing

In this section, we introduce the two pre-processing methods. The re-ranking model identifies important sentences and the selecting model selects a review subset that covers more aspects maintaining efficiency. The resulting sentences and review subset will be used to generate the review summary.

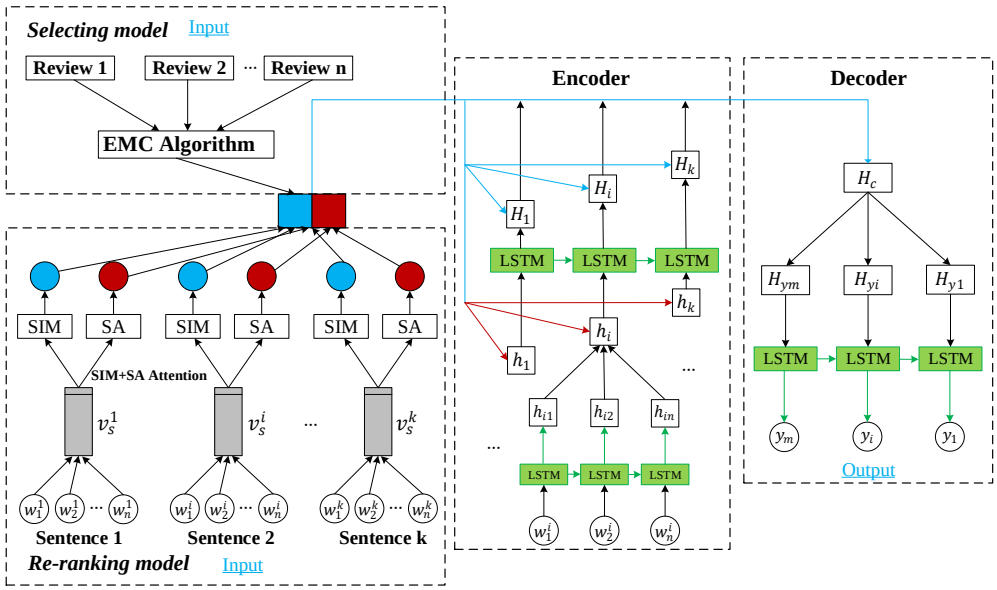


Fig. 2. The supervised summary generation with Encoder-Decoder framework

5 REVIEW SUMMARY GENERATION

In this section, we introduce the details of the proposed review summary generation model. We adopt two generation methods for the supervised and unsupervised scenarios. For the scenarios with human written summaries, we use the supervised generation method that takes the advantage of the encoder-decoder framework. For the scenarios without human written summaries, we propose an unsupervised method of aspect-based summary generation.

5.1 Supervised Review Summary Generation

For supervised scenarios, we propose to integrate two attention mechanisms with encoder-decoder framework, i.e., the similarity attention and the sentiment attention. Figure 2 illustrates the overview of our supervised method which is based on the encoder-decoder framework. The left part illustrates the input of the model, for which we proposed two pre-processing models. The right part shows the details of our supervised summary generation model, consisting of the encoder module and the decoder module. Details are as follows.

Encoder: The encoder module takes either the important sentences selected by the re-ranking model, or the reviews with high aspect-coverage selected by the selecting model, as the input. The input is further mapped into a semantic vector H_c, which contains all the semantic information. Different from traditional encoders, we adopt two encoders [32] consisting of a word encoder and

Algorithm 3 RE-RG: Redundancy Elimination to Generate Review Summary

Input: Aspects A; Reveiw sets R^* by selecting or re-ranking models
Output: Summary y

```
1:  Initialize a new Aspect set: A* ⟵ ∅
2:  y ⟵ ∅
3:  for r in R* do
4:      Initialize Sentences Set S_e
5:      Se ⟵ Segment review r by sentence
6:      for s in Se do
7:          a_i^p ⟵ Get the aspect and aspect polarity of s
8:          if a_i^p ∉ A* then
9:              A* ⟵ a_i^p ∪ A*
10:             y ⟵ y + s
11: return Summary y
```

a sentence encoder. The former is used to encode a sequence of words and the latter is used to encode a sequence of sentences.

Decoder with Attentions: The decoder module is utilized to generate a summarized review according to the merged representation H_c. Generally, different words and sentences make different contributions to the generation of the review summary. In order to capture the important sentences and words with more informative semantics, we adopt the multiple attentions mechanism [55] to decode the context information. Suppose the decoder generates a word at step t, the attention mechanism exploits the sentence-level and the word-level attentions to form the context vector c_t. The details can be found in our conference paper [15].

5.2 Unsupervised Review Summary Generation

For unsupervised scenarios, we conduct review selection, redundancy elimination, and aspects weight calculation. Figure 3 shows the unsupervised framework.

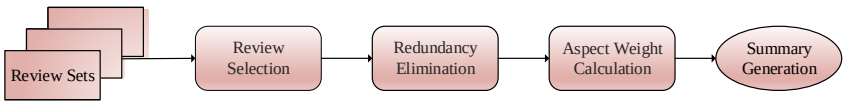


Fig. 3. The unsupervised framework of aspect-based summary generation

The first step is to select the review set that covers more aspects with fewer sentences. The first selected review has the highest gain-to-cost ratio. The second selected review has the largest gain-to-cost ratio among the remaining reviews, so on and so forth. We find that the selected review set may suffer from redundancy. Taking Table 9 for instance, the Top-3 reviews are selected by EMC-RS about phone "Nokia". The aspects of "battery", "music", "bluetooth", and "screen" are repeatedly described. It makes the review summary redundant and not concise at all. Therefore, we need to eliminate the redundancy.

The second step is to filter the sentences in which the redundant aspects are located, so as to keep the review summary concise. Algorithm 3 (lines 3-10) describes how to eliminate redundancy and generate the review summary. We segment each selected review by sentences in Line 5. If the sentence contains a new aspect sentiment description about the product, which means the sentence can contribute to the summary, we will keep it in the final review summary (Lines 6-10). Finally,

Algorithm 4 Aspect Weight Calculation

Input: Set of reviews R; Aspects A
Output: Aspect Weights: $\{w_{a_1^p}, ..., w_{a_i^p}, ..., w_{a_k^p}\}$

1: Initialize a dictionary C to record the count of each aspect
2: **for** r in R **do**
3: Sentences Set: $Se \longleftarrow Get\ sentences\ of\ review\ r$
4: **for** s in Se **do**
5: $a_i^p \longleftarrow Get\ the\ aspect\ and\ aspect\ polarity\ of\ s$
6: $C(a_i{}^p) \longleftarrow C(a_i{}^p) + 1$
7: $w_{a_i^p} \longleftarrow C(a_i{}^p)/\sum_{i\in[1,k]} C(a_i{}^p)$
8: **return** $\{w_{a_1^p}, ..., w_{a_i^p}, ..., w_{a_k^p}\}$

we keep all the valuable sentences of the product to generate a review summary (Line 11). It takes the time complexity of $O(|R^*||r||A^*|)$ in the worst case, where $|R^*|$ is the size of the selected review set, $|r|$ is the number of maximum sentences in a review, and $|A^*|$ is the size of aspects covered by the summary.

Although the summary covers many aspects efficiently, it is not clear how the sentiment distributes in different aspects. Therefore, we need to calculate the aspect weights in all original reviews to demonstrate the overall sentiment distribution, which leads to the third step, as follows.

The third step is to calculate the aspect weights, which will appear in the review summary, indicating the importance of each aspect in the whole original reviews. Algorithm 4 describes the process, taking the time complexity of $O(|R||r||A|)$. First, we traverse the original reviews and count the number of occurrences of each aspect $a_i^p \in A$ (Lines 2-6). Next, we normalize all the aspect counts by Eq. 14 (Line 7). Finally, we can obtain the summary and aspect weights (Line 8).

$$w_{a_i^p} = \frac{C(a_i^p)}{\sum_{i=1}^{|A|} C(a_i^p)}, \tag{14}$$

where $w_{a_i^p}$ is the weight of aspect a_i^p and $C(a_i^p)$ is the number of occurrences of a_i^p. $|A|$ is the number of aspects in A.

6 EXPERIMENTAL SETTINGS

In this section, we present the implementation of our model and the experimental settings in details. The code and data is available online[4].

6.1 Dataset

We use seven real world datasets which contain four supervised datasets (i.e., Idebate [59], RottenTomatoes [59], Amazon [9]) and Yelp [11], and three unsupervised datasets (i.e., three product reviews in AmazonReview[5]).

6.1.1 Four Datasets for Supervised Scenarios. **Idebate** is an argumentation dataset from idebate.org. The website is a Wikipedia-style site for collecting professional and debated arguments on controversial issues. The arguments of each debate are divided into different "for" and "against" points. Each point contains a central claim, which is built by editors to sum up the corresponding arguments and is regarded as the gold standard. The Idebate dataset includes 676 idebates with 2,259 claims.

[4]https://github.com/Dingxiaofei2017/expertreview
[5]https://nijianmo.github.io/amazon

RottenTomatoes is a famous American film review aggregation website which includes professional reviews and user reviews. A sentence critic consensus is constructed by an editor to summarize the opinions of the professional critics for each movie. The dataset includes 246,164 critics and their opinion consensuses for 3,731 movies. Each movie has around 66 reviews on average. The opinion consensus is considered as a summary of the gold standard.

Amazon is a product reviews dataset which is used by Brazinskas et al. [9]. It contains two different categories ¨C "Movies and TV" and "Electronics" and consist of 780 products and 73,040 reviews. It has the reference summaries provided by Brazinskas et al. [9].

Yelp review dataset contains reviews of businesses. The public yelp corpus of restaurant reviews are provided by Chu and Liu [11] which consists of 1,337 businesses and 129,840 reviews for testing. The businesses were then filtered to those with at least 50 reviews.

6.1.2 Three Datasets for Unsupervised Scenarios. **AmazonReview** includes several datasets of product reviews [36]. We select three different categories: "Movies and TV", "Electronics" and "Cell phones" and they have 50,052, 63,001 and 10,429 products respectively. In "Movies and TV", "Electronics" and "Cell phones", each product has 34, 27, and 19 reviews on average, respectively, and each review has around 8, 6, and 5 sentences on average, respectively. For each category, we divide the products into three sub-categories for review summarization tasks: the hot products with a large number (e.g., >300), ordinary products with a medium number (e.g., [30,300]) and unpopular products with a small number (e.g., <30) of reviews. Figure 4 shows the percentage of unpopular, ordinary, and hot products in three datasets. It is worth noting that, the new datasets have no human written summaries, which is typical in unsupervised scenarios.

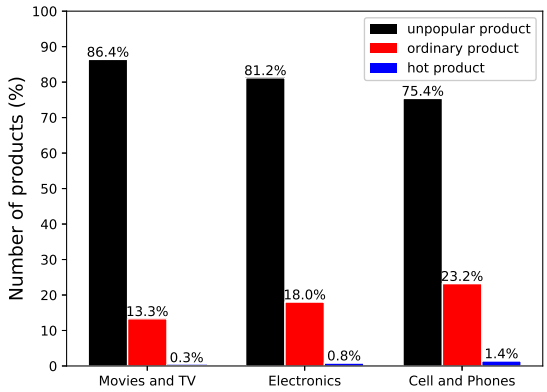


Fig. 4. Percentage of unpopular, ordinary, and hot products

6.2 Evaluation Metrics

For the supervised evaluation, we adopt the widely-used automatic evaluation metric ROUGE [19]. Following previous work, we report the scores from Rouge-1, Rouge-2 and Rouge-L, which are respectively calculated using the matches of unigrams, bigrams, and longest common subsequences, with the ground truth summaries. We obtain the ROUGE scores using the *pyrouge package*[6]. We also evaluate with the METEOR metric [13], both in exact match mode (rewarding only exact

[6]https://pypi.python.org/pypi/pyrouge/0.1.3

matches between words) and full mode (which additionally rewards matching stems, synonyms and paraphrases)[7].

In general, the purpose of the summarization task is to condense many reviews to a summary with the expectation that the content of the summary is consistent with the original reviews. For the unsupervised evaluation, we introduce the metric of a word overlap (WO) score [11] which uses the ROUGE-1 score between the summary and each review and then averages these scores. Eric Chu et al. [11] validate that WO is a reasonable metric for guiding model development in unsupervised scenarios. They also validate that WO is correlated with ROUGE-1 and ROUGE-L, with statistically significant pearson coefficients of 0.797 and 0.728, respectively. This is calculated by Eq. 15.

$$\mathrm{WO} = \frac{1}{|R|} \sum_{j=1}^{|R|} \mathrm{ROUGE}\left(y, r_j\right), \tag{15}$$

where $|R|$ is the number of product reviews. r_j is the j-th review of the product and y is the generated review summary.

To measure the quality of the generated summary of several product reviews, we define two new metrics, aspect coverage of the summary (COV for short) and aspect density of sentences (ADS for short). COV is calculated by $\frac{|A_s|}{|A|}$ and ADS is calculated by $\frac{|A_s|}{|s|}$, where $|s|$ is the number of sentences in the summary, $|A_s|$ is the number of aspects covered by the summary, and $|A|$ is the number of aspects contained in the original reviews. Note that the same aspect is only considered once.

6.3 Implementation

Data Pre-processing. All datasets are lowercased and tokenized and all movie titles in Rotten-Tomatoes are replaced with a generic label. To get adjectives, noun phrases, and key words, we use Python toolkit *spaCy* which provides implementations of many basic NLP tasks. We keep the most frequent 30,000 words in our vocabulary and replace other words with '<unk>'. We also remove the reviews or sentences that have less than 5 words.

Model Implementation. For the re-ranking model, we use a Python toolkit TextRank [8] to caculate similarity and use the SentiWordNet [9] to generate the sentiment polarity of sentences. Based on empirical studies, the parameter ε is set to 0.4 for the similarity calculation. α is set to 0.5 and β is set to 0.9 in the EMC-RS algorithm. We will test the parameter sensitivity in the experiments.

In the review generation stage, we use an improved encoder-decoder framework with attention mechanisms [56]. The preprocess of the re-ranking model and the selecting model will generate a re-ranked text S or selected reviews R^*. We feed the re-ranked S or selected reviews R^* to the improved generation model, so as to generate an expert review. For the word-level encoder, we use four hidden layers of bidirectional GRU, while for sentence-level encoder we use three layers of GRU. For the decoder we use four hidden layers of GRU with 25 timesteps. Each hidden layer has 256 units. Note that the word-embedding has been initialized in the preprocess stage. The dimension of word-embedding is 300. The batch size is set to be 8 texts. The training step reaches 100,000, which approximates to 44 epochs on Idebate data and 26 epochs on RottenTomatoes. The improved encoder-decoder model is implemented in tensorflow. We run the model on a Tesla-P100 GPU card, which takes about 24 hours for every 40 epochs.

[7]http://www.cs.cmu.edu/ alavie/METEOR/
[8]https://github.com/davidadamojr/TextRank
[9]http://sentiwordnet.isti.cnr.it

6.4 Comparative Methods

We compare the performance of the component (i.e., the re-ranking model and the selecting model), the supervised model, and the unsupervised model with some baselines and state-of-the-art methods.

For evaluating the performance of two preprocess strategies of re-ranking and selecting, we choose some state-of-the-art methods and some variations of our model.

(1) **TextRank** [38] is a graph-based method inspired by the PageRank algorithm, which computes sentence importance based on the eigenvector centrality in a graph representation of sentences.

(2) **Opinosis** [20] is a novel graph-based summarization method which generates concise abstractive summaries of highly redundant opinions.

(3) **EMC-RS-Lead-3** selects the top-3 reviews using the EMC-RS Algorithm.

(4) **Sa-Lead-3** selects top-3 sentences using the sentiment polarity.

(5) **Sim-Lead-3** selects top-3 sentences using semantic importance scores.

(6) **Sim-Sa-Lead-3** selects top-3 sentences using both the semantic importance scores and the sentiment polarity (i.e., our re-ranking model)..

In addition, to verify the effectiveness of the supervised generation model, we also compare our approach with various variants of seq2seq models.

(1) **NN-ABS** [49] is a neural network based model with local attention modeling for abstractive sentence summarization.

(2) **Doc-Distraction** [10] is a state-of-the-art neural abstractive document summarization model.

(3) **Att-lead-n** is a simple RNN-based attentional sentence summarization model which generates review summary according to the top-n reviews.

(4) **Doc-h** takes all reviews as input. It first encodes sentences to sentence vectors and then encodes sentence vectors to a document vector.

(5) **Doc-h-att** adds sentence-level attention to **Doc-h**.

(6) **Doc-hh-att** adds both sentence-level and word-level attention to **Doc-h**.

(7) **EMC-RS-Lead-3-ED** selects the top-3 reviews using our selecting model and conducts supervised generation with encoder-decoder.

(8) **Sim-Sa-Lead-3-ED** selects the top-3 sentences using the re-ranking model and conducts supervised generation with encoder-decoder.

Finally, to verify the effectiveness of the unsupervised method in supervised and unsupervised scenarios, we compare our approach with several recent models.

(1) **Centroid-based** model [48] is an extractive method for unsupervised summarization task. It uses a centroid-based method that exploits the compositional capabilities of word embeddings.

(2) **MeanSum** [11] is an end to end neural model for unsupervised multi-document abstractive summarization.

(3) **CopyCat** [9] is an abstractive summarizer of opinions, which does not use any summaries in training and is trained end-to-end on a large collection of reviews.

(4) **Sim-Sa-Lead-3-RE** selects top-3 sentences using the re-ranking model and eliminates redundancy with our RE-RG algorithm to generate the summary (i.e., re-ranking + RE-RG algorithm).

(5) **EMC-RS** selects a review subset using our EMC-RS algorithm.

(6) **EMC-RS-RE** is our unsupervised method which selects a review set using EMC-RS algorithm and then generates the summary using RE-RG algorithm.

7 EXPERIMENTAL RESULTS

In this section, we analyze the performance of our work in supervised and unsupervised scenarios. Table 4 and 5 show the results in supervised scenarios which are based on RottenTomatoes and Idebate, and Amazon and Yelp, respectively. Table 6 shows the results in unsupervised scenarios which is based on three datasets (e.g., "Cell Phone", "Electronics", and "Movies and TV") of AmazonReview. We also analyze parameter sensitivity and conduct case study.

7.1 Performance in Supervised Scenarios

We first test the effects of model components, then we compare our work with several other methods.

7.1.1 The Effects of Model Components. In supervised scenarios, we explore three fine-grained similarity calculation methods for the re-ranking model and compare **TextRank**, **Opinosis**, **EMC-RS-Lead-3**, **Sa-Lead-3**, **Sim-Lead-3**, and **Sim-Sa-Lead-3** to test the effects of pre-processing.

The effects of similarity methods. We calculate similarity with three levels of granularity: sentence-level, phrase-level, and word-level. In order to verify which one is more accurate, we utilize our re-ranking model with each of them to get top-3 sentences as summaries. The results on RottenTomatoes are shown in Table 3. It shows that the rouge-1 score of **Lead-3(Word-level)** is 40% higher than that of **Lead-3(Sentence-level)** and 26% higher than that of **Lead-3(Phrase-level)**. This indicates that the *word-level* similarity is more effective than others. It is worth noting that **Lead-3(N&A)** only keeps nouns and adjectives and its performance is close to that of **Lead-3(word-level)**. This indicates the importance of nouns and adjectives. We use **Lead-3(word-level)** in the following experiments.

Table 3. Performance of different similarity levels in RottenTomatoes.

	Rouge-1	Rouge-2	Rouge-L
Lead-3(Sentence-level)	18.34	6.22	15.70
Lead-3(Phrase-level)	20.39	7.33	17.43
Lead-3(N&A)	24.12	**8.76**	18.53
Lead-3(Word-level)	**25.84**	8.42	**18.90**

The effects of pre-processing. We compare the effects of pre-processing methods in Table 4 (rows 1-6). We can see that the Rouge-1 score of **Sim-Lead-3** (row 5) in RottenTomatoes is 21% higher than that of *TextRank* (row 1) and 15% higher than that of *Opinosis* (row 2). Similarly, the Rouge-1 score in Idebate of **Sim-Lead-3** is 20% higher than that of *TextRank* and 23% higher than that of *Opinosis*. The performance of **Sim-Sa-Lead-3** (row 6) is even better than that of **Sim-Lead-3**, e.g., Rouge-1 in RottenTomatoes improves by 3%. This indicates that the sentiment polarities are able to improve the performance. However, the Rouge-1 score of **Sa-Lead-3** is much lower than **Sim-Lead-3**, indicating that similarity takes more effect. In summary, considering both sentiment polarity and semantic similarity can improve the performance of pre-processing. Moreover, the selecting model (**EMC-RS-Lead-3**) also performs better than *TextRank*, *Opinosis*, and **Sa-Lead-3**, but a bit worse than **Sim-Lead-3** and **Sim-Sa-Lead-3**. We study this deeply in more AmazonReview datasets later, which show its advantages in unsupervised scenarios. Moreover, comparing **EMC-RS-Lead-3** (row 3) with *Doc-h*, *Doc-h-att* and *Doc-hh-att*, it indicates that the selecting model performs better than those supervised baselines.

We also calculate the sim-to-summary scores of top-k (Lead-k) reviews and utilize the mean value of scores to evaluate these ranking methods. The sim-to-summary score represents the semantic similarity between a review and the human-written summary. As shown in Figure 5, the improvements of our re-ranking method over **TexRank** is about 15%, indicating its effectiveness.

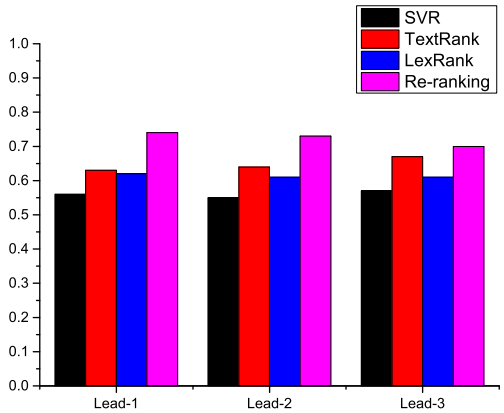


Fig. 5. Evaluation of Ranking Methods

7.1.2 Comparative Studies. We conduct two groups of experiments for comparative studies. The first group is on the movie review dataset RottonTomatoes and the argumentation dataset Idebate. The second group is on the product review datasets, Amazon and Yelp.

Comparative studies on RottenTomatos and Idebate. For the first group comparison, we compare the supervised models (**Sim-Sa-Lead-3-ED, EMC-RS-Lead-3-ED**) with state-of-the-art methods (i.e., **NN-ABS**, **Doc-Distraction** and **Att-lead-n**) to verify the effect on *Rouge scores* and *METEOR*. The results are displayed in Table 4. Note that EMC-RS-Lead-3 and EMC-RS-Lead-3-ED have no results in Idebate since there is no aspect imformation in the Idebate. Idebate describes the professional and debated arguments on controversial issues. The arguments of each debate are divided into "for" and "against". The EMC-Lead-3 and EMC-Lead-3-ED are based on the aspect-based selecting model which skills in the aspect-based extractive summarization for items. Therefore, these methods don't work on the Idebate dataset.

We compare the performance of summary generation methods and show the results in rows 7-14 of Table 4. The description of the methods can be found in Section 6.4. The results indicate that our methods, i.e., **Sim-Sa-Lead-3-ED** and **EMC-RS-Lead-3-ED** (rows 13-14) perform much better than others, e.g., the Rouge-1 scores in RottenTomatoes of **Sim-Sa-Lead-3-ED** is 97% higher and **EMC-RS-Lead-3-ED** is 87% higher than *Doc-Distraction*. Comparing the results of **Sim-Sa-Lead-3-ED** with **EMC-RS-Lead-3-ED**, **Sim-Sa-Lead-3-ED** is 5% higher. This indicates that, on the review abstractive summarization task, the re-ranking model can be better for the neural abstractive summarization models. Comparing the results of *Doc-h*, *Doc-h-att* and *Doc-hh-att* (rows 10-12), we find that it is difficult for neural seq2seq models to process document-level datasets and discover important reviews from the original reviews. This indicates that our pre-processings with encoder-decoder model can improve the performance.

Comparative studies on Amazon and Yelp. For the second group comparison, we compare the unsupervised models (**EMC-RS-RE**) with state-of-the-art methods (i.e., **CopyCat** and **MeanSum**) to verify the effect on *Rouge scores*. The results are displayed in Table 5. It shows that **EMC-RS-RE** performs better than others on Rouge-1 and Rouge-l in Amazon. To specific, the Rouge-1 scores in Amazon of **EMC-RS-RE** is 17% higher than that of **MeanSum** and 13% higher than that of **CopyCat**. The Rouge-l score in Amazon of **EMC-RS-RE** is improved by 35% than that of **CopyCat**. While the Rouge-2 in Amazon of **EMC-RS-RE** decreases 6% than that of **CopyCat**. However, **EMC-RS-RE** performs worse on Rouge-1 and Rouge-2 in Yelp. Only Rouge-l of **EMC-RS-RE** is higher than others. We analyze the reason and find that the reviews in Yelp dataset mainly describe the food, which contain only a few aspects, and there are only a few reviews (e.g., 8) for each

Table 4. Comparison on **RottenTomatoes** and **Idebate** (supervised scenarios)

			RottenTomatoes				Idebate			
			Rouge-1	Rouge-2	Rouge-L	METEOR	Rouge-1	Rouge-2	Rouge-L	METEOR
Unsupervised methods	1.	TextRank	21.44	6.20	14	6.50	20.33	5.92	13.25	6.03
	2.	Opinosis	22.34	6.50	15.78	7.21	19.81	6.32	14.37	7.23
	3.	**EMC-RS-Lead-3**	24.95	8.50	17.95	7.51	-	-	-	-
	4.	**Sa-Lead-3**	17.09	5.32	12.78	6.02	16.63	5.01	11.32	5.65
	5.	**Sim-Lead-3**	25.84	8.42	18.90	9.21	24.35	7.98	17.20	8.66
	6.	**Sim-Sa-Lead-3**	**26.84**	**9.56**	**19.32**	**10.11**	**25.84**	**8.56**	**18.35**	**9.42**
Supervised Methods	7.	NN-ABS	25.82	7.03	23.07	8.01	24.32	6.89	22.67	7.87
	8.	Doc-Distraction	14.53	4.49	13.83	5.11	13.24	3.98	12.89	4.54
	9.	Att-lead-n	26.57	6.84	23.13	7.56	25.63	7.51	21.39	8.21
	10.	Doc-h	15.91	2.04	14.32	2.56	14.21	2.14	12.82	2.45
	11.	Doc-h-att	17.12	3.98	15.46	4.08	14.12	2.85	13.64	3.69
	12.	Doc-hh-att	18.53	4.33	16.78	5.03	16.24	3.37	14.43	4.01
	13.	**EMC-RS-Lead-3-ED**	27.22	8.28	24.02	9.57	-	-	-	-
	14.	**Sim-Sa-Lead-3-ED**	**28.71**	9.12	**25.14**	**11.32**	**26.71**	8.32	**23.14**	**9.83**

"supervised scenarios" indicates whether the datasets have standand reference summaries.

Table 5. Comparison on Amazon and Yelp datasets (supervised scenarios)

Method	Amazon			Yelp		
	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
MeanSum	28.46	3.66	15.57	29.20	4.70	18.15
CopyCat	29.47	**5.26**	18.09	**32.00**	**5.81**	20.16
EMC-RS-RE	**33.33**	4.92	**24.35**	27.07	3.14	**20.76**

"supervised scenarios" indicates whether the datasets have standand reference summaries.

product. Our unsupervised model is more suitable for dealing with e-commerce products that have a large number of reviews. It tries to select the high quality reviews that covering more aspects with fewer sentences, which can remove redundancy from plenty of reviews. However, when there are only a few reviews and a few aspects, the advantages of EMC-RS-RE cannot be fully taken.

Moreover, we conduct some human evaluation on Amazon and Yelp to check the quality of summaries (Appendix A). The results validate that the review summary generated by our unsupervised model is less redundant, and contains more detailed product descriptions. Moreover, the summary generated by our unsupervised model is close to the original reviews.

7.2 Performance in Unsupervised Scenarios

In this subsection, we check the performance of our work in unsupervised scenarios with three AmazonReview datasets. We first check the effects of review quantity and redundancy elimination, then we compare the performance of different methods. Note that we also tested with the RottenTomatoes dataset, which shows similar trends. Hence, we do not display those results.

We compare three sets of models on the metrics of *WO*, *ADS* and *COV*: 1) **Centroid-based** method [48], which is a state-of-art extractive summarization method; 2) **Sim-Sa-Lead-3**, which selects top-3 sentences by our re-ranking model and **Sim-Sa-Lead-3-RE**, that generates the summary by our RE-RG algorithm based on Sim-Sa-Lead-3; 3) **EMC-RS** which selects a review subset by EMC-RS algorithm and **EMC-RS-RE** that generates review summary by our RE-RG algorithm based on EMC-RS. Table 6 shows the comparison results. There are three values for each metric, corresponding to the results of unpopular, ordinary, and hot products.

The effects of review quantity. We test the effects of review quantity on the summary quality by checking the performance on the unpopular, ordinary, and hot products, as shown in Table 6. We can see that the performance of the first set models, i.e., **Centroid-based** method, keeps relatively stable among different review quantities on all the three metrics of *WO*, *COV*, and *ADS*. For the second set of models, i.e., **Sim-Sa-Lead-3** and **Sim-Sa-Lead-3-RE**, the performance has a steady decline on *WO* with the number of reviews; e.g., the value of *WO* for **Sim-Sa-Lead-3** in ordinary products increases 18% compared to that of unpopular products. The value of *COV* first increases and remains keeps stable with the number of reviews; e.g., the value of *COV* on "Cell Phones" with ordinary products increases by 93% compared to that of unpopular products. The value of *ADS* for **Sim-Sa-Lead-3-RE** first decreases and then increases with the number of reviews; e.g., the value of *ADS* decreases 6% on "Cell Phones". For the third set of models, i.e., **EMC-RS** and **EMC-RS-RE**) the value of *WO* eventually decreases a little with the number of reviews, e.g., **EMC-RS** decreases 8% in "Cell Phones". In contrast, performance keeps a steady growth on *COV* with the number of reviews, e.g., **EMC-RS-RE** improving *COV* by 69% on "Cell Phones". Similarly, the value of *ADS* also increases, which improves by 1.08 times. In a word, the performance of the selecting model on the three metrics is relatively stable. We want to emphasize that our unsupervised model can handle the scenarios with a large number of product reviews and keep a better performance on all the three metrics of *WO*, *COV*, and *ADS*.

The effects of the redundancy elimination algorithm. We further compare the two methods within the second and the third set of models, so as to check the effects of our redundancy elimination algorithm, as shown in Table 6. Comparing with **Sim-Sa-Lead-3** in "Cell Phones", **Sim-Sa-Lead-3-RE** improves 68% on *ADS*, and the other two metrics remain almost unchanged. Similarly, compared with **EMC-RS**, **EMC-RS-RE** increases in *ADS* by 1.84 times. This demonstrates that our Redundancy Elimination algorithm (RE-RG algorithm) can improve the aspect density for each sentence (*ADS*) of the generated review summary and make the summary more efficient. Meanwhile, **EMC-RS** reaches slightly higher than **EMC-RS-RE** on *WO*, indicating that redundancy elimination may slightly reduce the consistency with the original review.

Overall Comparison. We compare the performance of the three sets of models in three datasets as shown in Table 6. We can see that our unsupervised method (**EMC-RS-RE**) performs the best on three metrics, indicating that our model can generate summaries with high efficiency and high consistency with the original reviews. For example, on the "Electronics" dataset, **EMC-RS-RE** improves on *WO* by 28% over that of the **Centroid-based** method, improves 2.54 times on *COV*, and improves 1.19 times on *ADS*. Moreover, compared to **Sim-Sa-Lead-3, EMC-RS** achieves better performance on three metrics, improving 1.38 times on *COV* and around 37% and 23% on *WO* and *ADS*, respectively.

In summary, compared with the re-ranking model, the sentences selected by the selecting model can cover more aspects with fewer sentences and they are also more consistent with original reviews. In addition, the selecting model is more efficient than the re-ranking model when there is a large amount of reviews. Recall that the time complexity of the re-ranking model is $O(k_1|N|^2 + |R||r||A|)$, while that of the selecting model is $O(k_2|R|^2|r|^2|N|^2)$, where $|R|$, $|r|$, $|A|$, and $|N|$ are the number of original reviews, sentences for each review, aspects, and nouns in the original reviews, respectively; k_1 and k_2 are the number of iteration for the re-ranking model and aspect extraction in the selecting model, respectively.

7.3 Parameters Sensitivity Analysis

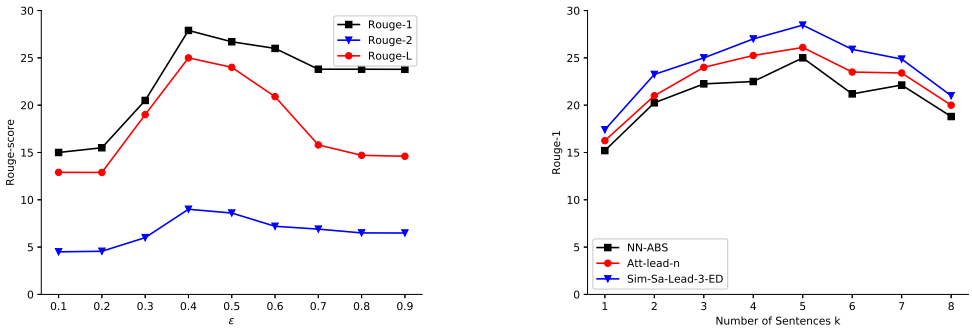
To enhance the flexibility of the proposed model, some parameters are established. In this section, we vary parameter ε to investigate their effects on the re-ranking model performance and parameters

Table 6. Comparison results on **AmazonReviews** (unsupervised scenarios)

Method	Cell phones			Electronics			Movies and TV		
	WO	COV	ADS	WO	COV	ADS	WO	COV	ADS
Centroid-based	34.14/30.30/29.38	0.13/0.21/0.18	0.54/0.60/0.56	29.31/25.85/29.97	0.13/0.21/0.19	0.59/0.50/0.53	27.16/24.45/26.27	0.14/0.20/0.26	0.50/0.57/0.36
Sim-Sa-Lead-3	33.09/27.98/25.78	0.14/0.27/0.25	0.31/0.37/0.35	24.83/23.74/22.72	0.17/0.26/0.25	0.38/0.39/0.31	26.47/22.50/22.21	0.14/0.33/0.30	0.39/0.38/0.46
Sim-Sa-Lead-3-RE	28.97/27.95/25.56	0.14/0.27/0.25	0.52/0.47/0.49	22.64/23.19/25.65	0.17/0.26/0.25	0.57/0.40/0.42	24.00/22.50/22.60	0.14/0.33/0.30	0.69/0.40/0.50
EMC-RS	**41.26**/37.70/37.86	**0.39/0.62/0.66**	0.45/0.54/0.94	**37.39/32.42**/36.27	**0.46/0.62/0.73**	0.46/0.48/0.98	**34.65/29.32/33.05**	**0.46/0.64/0.74**	0.42/0.46/0.97
EMC-RS-RE	38.62/**38.89/37.90**	**0.39/0.62/0.66**	**1.28/1.29/1.51**	36.39/32.40/**36.46**	**0.46/0.62/0.73**	**1.29/1.33/1.59**	32.27/29.00/32.93	**0.46/0.64/0.74**	**1.38/1.49/2.03**

Three values for each metric correspond to the results of unpopular, ordinary and hot products.

α and β to investigate its effect on the selecting model. We also investigate the performance of the supervised method with different numbers of sentences in the dataset of RottenTomatoes.



(a) Rouge scores of different threshold ε

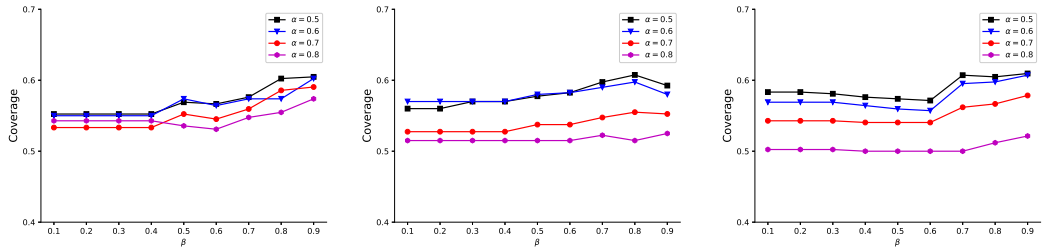
(b) Rouge-1 scores of top-k sentences

Fig. 6. The settings of threshold ε and the input sentence on RottenTomatoes

The effect of the threshold ε on the Similarity Measurement. We check the effect of the threshold ε by calculating the Rouge scores of our re-ranking model (Sim-Sa-Lead-3), varying ε in [0.1,0.9]. The results are shown in Fig. 6(a). It shows that with the increase of ε, all Rouge scores first increase and then decrease slowly. Rouge scores reach their highest when ε = 0.4, and it decrease slowly when ε >0.7.

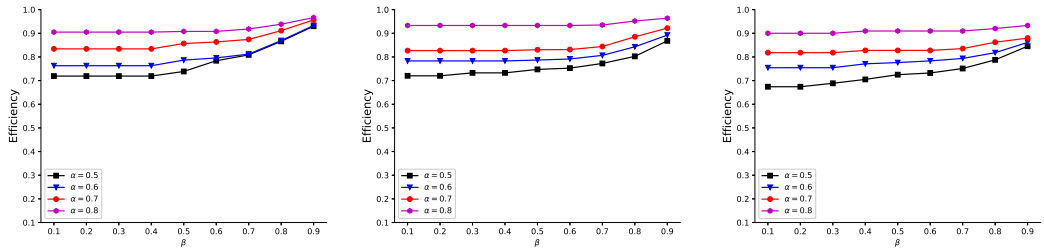
The effect of the top-k sentences. We test NN-ABS, Att-lead-n, and our supervised generation method (**Sim-Sa-Lead-k-ED**) with different numbers of top-k sentences on RottenTomatoes, varying k in [1,8]. The results are shown in Fig. 6(b). As k increases from 1 to 5, the Rouge-1 scores of all three methods increase. After that, the performance decreases.

The effect of the threshold α and β on the selecting model. α and β are used to control the coverage and efficiency of the selected review set. We check their effect as shown in Figure 7 and Figure 8. It shows that when α increases, the coverage of the selected review set decreases. Figure 7 shows that generally, the coverage is optimal when $\alpha = 0.5$ on "Electronics" and "Movies and TV", and it is optimal when $\alpha = 0.6$ on "Cell Phones". We use the constraint of $Eff \geq \alpha$ to filter out reviews with low efficiency. To keep the efficiency of the selecting model and the length of our generated summary, α cannot be too small. The results for efficiency is shown in Figure 8. It shows that the efficiency increases when β and α increases. The efficiency keeps a steady increase when β increases, for $\alpha \in [0.5, 0.8]$. In summary, a lower α is able to keep the coverage of generated summaries and a higher β is able to keep the efficiency. Therefore, α is set to 0.5 and β is set to 0.9 in the selecting model, for all datasets we use in this paper.



(a) Cell phones (b) Electronics (c) Movies and TV

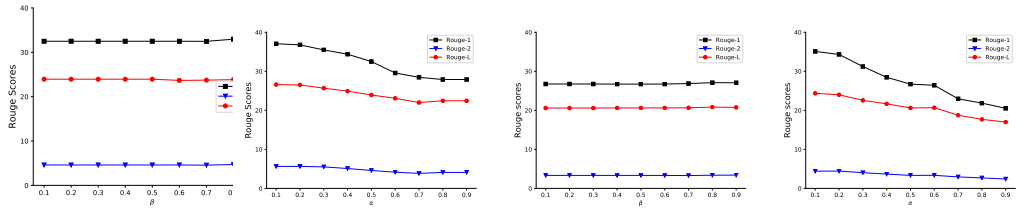
Fig. 7. Varying β and α for Coverage



(a) Cell Phones (b) Electronics (c) Movies and TV

Fig. 8. Varying β and α for Efficiency

We first check the effect of the thresholds α and β by calculating the Rouge scores of our unsupervised model on the labeled Amazon and Yelp datasets, varying β and α in [0.1,0.9]. The results are shown in Figure 9. Figure 9(a) and Figure 9(c) show that when $\alpha = 0.5$ and β increases, Rouge scores almost keep steady. Figure 9(b) and 9(d) show that when we set β to 0.5, with the increases of α, the Rouge scores decreases. It indicates that the datasets are not sensitive to β. Meanwhile, a larger α filters more reviews out of the candidate sets, which indirectly decreases the Rouge scores since more information may be removed.

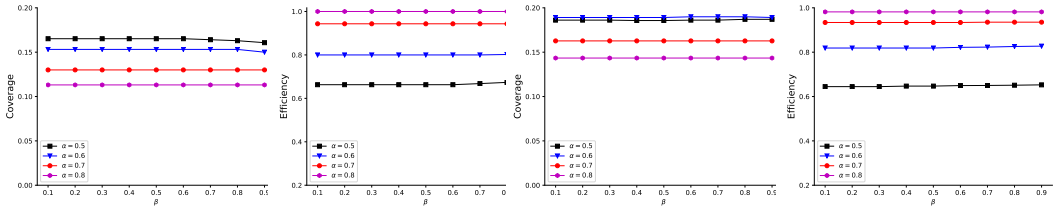


(a) α=0.5 (Amazon) (b) β=0.5 (Amazon) (c) α=0.5 (Yelp) (d) β=0.5 (Yelp)

Fig. 9. Varying α and β for Rouge scores

We also check the coverage and efficiency of the generated summary when varying α and β. Figure 10(a) and 10(c) show that when α increases, the coverage of the selected review set decreases. This is because that a larger α filters more reviews out. In addition, the aspect coverage keeps steady when keeping α unchanged and varying β. Meanwhile, Figure 10(b) and 10(d) show that as α increases, the efficiency increases. However, as β increases, the efficiency almost keeps steady. We

analyze the reason and find that there are only a few reviews (e.g., 8) for each product in the labeled Amazon and Yelp datasets and each review covers several aspects. It makes most of them be taken as high quality reviews. In fact, they are exact high quality reviews from the human's perspective. Actually, our unsupervised model is more suitable for the scenarios with a large number of reviews and having large redundancy. It tries to select the reviews that covering more aspects with fewer sentences and it can remove redundancy from many of reviews. However, as the labeled datasets have less redundancy, the filtering function of our model takes less effect.



(a) Coverage in Amazon (b) Efficiency in Amazon (c) Coverage in Yelp (d) Efficiency in Yelp

Fig. 10. Vary α and β for Coverage and Efficiency

7.4 Case Study

In this section, we conduct case studies on supervised and unsupervised scenarios.

7.4.1 Case Study on Supervised Scenarios. For supervised scenarios, we show the Top-5 reviews selected by the proposed re-ranking model in Table 7 and Table 8. Table 7 is for SIM-ranking which selects by similarity score. Table 8 is for SIM-SA-ranking which selects by both similarity score and sentiment polarity. In the SA-Attention, SentiWordNet is used to generate a sentiment polarity of each word within the reviews. The sentiment words in reviews are highlighted in red and the summary is in blue. We compare them with the human-written summary. We adopt the sim-to-summary score to evaluate the performance, which represents the semantic similarity between a review and the human-written summary. It is calculated as follows:

$$score = cos(v_{review}, v_{summary}), \tag{16}$$

where v_{review} and $v_{summary}$ are the sentence embedding of a review and summary, respectively.

Table 7. Top-5 reviews re-ranked by Sim-Lead-5 about movie "Crazy Heart" and their sim-to-summary scores

Movie: *Crazy Heart*
Top-5 reviews by SIM-RANK:
1: *Jeff Bridges realizing one of his most memorable film characters in years with a performance you won't quickly forget.*
2: *This performance reminds us that Bridges is that rare actor who has never had to make that apology. Crazy Heart lets him be every bit as grand as we'd hope him to be.*
3: *Without an ounce of doubt, Jeff Bridges' portrayal of Bad Blake is superb.*
4: *Crazy Heart, written and directed by Scott Cooper, is a small movie perfectly scaled to the big performance at its center.*
5: *It's a bit too easy, a bit too familiar, and maybe even a bit too much fun. But the easy magic Bridges brings to the screen makes it all work.*
Summary: *Thanks to a captivating performance from Jeff Bridges, Crazy Heart transcends its overly familiar origins and finds new meaning in an old story.*
SIM-to-SUMMARY score 1: 0.73 2: 0.67 3: 0.76 4: 0.65 5: 0.84

The selected top-5 reviews in Tables 7 and 8 get high sim-to-summary scores, indicating that our re-ranking model can offer better intermediate results for the input of the generation model. It

Table 8. Top-5 reviews re-ranked by Sim-Sa-Lead-5 about "Crazy Heart" and their sim-to-summary scores

Movie: *Crazy Heart*
Top-5 reviews by SIM-SA-RANK:
1: *It's a bit too easy, a bit too too familiar, and maybe even a bit too much too fun. But the easy magic Bridges brings to the screen makes it all work.*
2: *A too wonderfully easy, too confident and muscular performance from Jeff Bridges - so easy, confident and muscular that it doesn't look like acting at all - saves this movie from being pure sentimental mush.*
3: *Without an ounce of doubt, Jeff Bridges' portrayal of Bad Blake is too superb.*
4: *Jeff Bridges realizing one of his most memorable film characters in years with a performance you won't quickly forget*
5: *This performance reminds us that Bridges is that rare actor who has never had to make that apology. Crazy Heart lets him be every bit as too grand as we'd hope him to be.*
Summary: *Thanks to a captivating performance from Jeff Bridges, Crazy Heart transcends its overly familiar origins and finds new meaning in an old story.*
SIM-to-SUMMARY score 1: 0.84 2: 0.78 3: 0.76 4: 0.73 5: 0.67

can also be applied to any dataset that includes user's sentiment. Moreover, reviews 1, 3, 4, 5 in Table 8 are also in Table 7, which are ranked 5, 3, 1, 2, and review 2 in Table 8 is a new one. The sim-to-summary scores in Table 8 are higer and they are ranked in decreasing order. This indicates that SIM-SA-RANK can re-rank reviews more reasonably.

The reasons are as follows. Reviews are used to express users' feelings. Hence, the sentences that contain more sentiment words are better at expressing the user's feeling and they should be selected. To quantify the sentiment of a review, we count the number of sentiment words in a review. It is the simplest yet the most effective way to evaluate a review's sentiment and it is able to expedite our preprocessing. Taking the first review in Table 8 for instance, it contains three positive sentiment words, which better expresses the user's feeling than the first review in Table 7 which has no sentiment words. Hence, comparing with SIM-RANK, the sim-to-summary score of SIM-SA-RANK is better.

7.4.2 Case Study on Unsupervised Scenarios. We conduct a case study in unsupervised scenarios with the review set of the phone "Nokia". The selected top-3 reviews by EMC-RS are shown in Table 9 and the generated summary and aspect weights are shown in Table 10. Note that our selecting model can adaptively determine the size of review subsets, while we have only presented the selected Top-3 reviews. We use different colors to indicate the aspect polarity, red for positive and blue for negative.

For review selection, we can see that in Table 9 the first selected review ($Top-R1$) covers 5 aspects and contains two sentences which have high gain-to-cost ratios. The second review contributes to the aspects of "program", "camera", and "keyboard", but the first sentence ($Top-R2, S1$) doesn't contain any new aspects. The third review contributes to the aspect of "size" (S2) and the other sentences make no contribution.

For summary generation, we can see that in Table 10 the generated summary is concise and covers all aspects that appear in the Top-3 reviews. Moreover, the weights of aspects in the overall product reviews can reflect the overall sentiment of users on different aspects. Finally, we summarize the ratio of the aspect sentiment, which can more directly display the proportion of negative and non-negative sentiment on each aspect. For example, Table 10 shows that 7 % of reviews describe the positive "battery" aspect in the whole reviews, among which, 75% of reviews consider the phone's "battery" as positive and 25% as negative.

7.5 Summary of Experiments

We propose review summary generation frameworks for supervised and unsupervised scenarios. They achieve good results on different datasets. The main findings of the experiments are as follows.

Table 9. Top-3 reviews selected by EMC-RS about phone "Nokia"

CellPhone: Nokia
Top-R1: ***(S1:)*** *I have been using the Nokia E71 as a replacement for my Treo 680, and I must say it is a lot fancier, and with a very improved battery life, with a much better bluetooth handling, as well as music management, email, and possibility to chat and voip that is completely unbelievable with the previous cellular.* ***(S2:)*** *I must say that the thing I miss the most is the task management, since it doesn't even allow me to input tasks or todo's directly from the home screen and aside from that, it is a very good cell phone, and worth the $350 I paid for.*
Top-R2: ***(S1:)*** *I had trouble setting this up as it kept dropping my wifi signal.* ***(S2:)*** *I was also surprised to find that Nokia's Ovi map program still does not cover Israel.* ***(S3:)*** *I am returning it and getting an Android Motorola Flipout Unlocked GSM Quad-Band Android Phone with Bluetooth, Camera, QWERTY Keyboard and Wi-Fi - Unlocked Phone - US Warranty - Black with Google maps, which has covered Israel for years.*
Top-R3: ***(S1:)*** *Each phone does its own job and this one does a good job on music even with no headphones attached.* ***(S2:)*** *Pro phone size and speakers on this phone are awesome (for a phone) display Nokia.* ***(S3:)*** *Computer Software can even import Itunes music decent camera tethering.* ***(S4:)*** *Nokia headphones and CONS touch screen is laggy.* ***(S5:)*** *All plastic feels cheaply made certain software hard to use or unusable, for some reason certain features that I have used on other Symbian Nokia phones just don't seem to work properly on this handset.* ***(S6:)*** *If your seeking this phone for business go elsewhere, but if you're just looking for entertainment this is a great phone.*

Table 10. The summary of "Nokia" phone by EMC-RS-RE based on Top-3 reviews

CellPhone: Nokia
The Summary of the Selected Top-3 reviews:
1. (Top-R1,S1) *I have been using the Nokia E71 as a replacement for my Treo 680, and I must say it is a lot fancier, and with a very improved battery* ***(0.07)*** *life, with a much better bluetooth* ***(0.61)*** *handling, as well as music management* ***(0.05)****, email, and possibility to chat and voip that is completely unbelievable with the previous cellular.*
2. (Top-R1,S2) *I must say that the thing I miss the most is the task management, since it doesn't even allow me to input tasks or todo's directly from the home screen* ***(0.03)*** *and aside from that, it is a very good cell phone* ***(0.78)****, and worth the $350 I paid for.*
3. (Top-R2,S2) *I was also surprised to find that Nokia's Ovi map program* ***(0.02)*** *still does not cover Israel.*
4. (Top-R2,S3) *I am returning it and getting an Android Motorola Flipout Unlocked GSM Quad-Band Android Phone* ***(0.78)*** *with Bluetooth* ***(0.61)****, Camera* ***(0.11)****, QWERTY Keyboard* ***(0.04)*** *and Wi-Fi - Unlocked Phone - US Warranty - Black with Google maps, which has covered Israel for years.*
5. (Top-R3,S2) *Pro phone size* ***(0.15)*** *and speakers on this phone* ***(0.78)*** *are awesome (for a phone) display Nokia.*
Aspect sentiment ratio (positive : negative):
phone(30:1), camera(1:0), music(1:0), computer(1:0), battery(3:1), program(1:0), bluetooth(1:0), keyboard(1:0), size(1:0), quality(1:0), screen(2:1)

The sentiment polarities of reviews is important. User-generated reviews are different from other text documents in that they reflect users' personal sentiment. When people read reviews, they usually care more on what aspects are described and which sentiment polarities are expressed towards the aspects. Therefore, the re-ranking model and selecting model with sentiment analysis show good performance.

For supervised scenarios, the re-ranking-ED works better than the selecting model. This is because it is difficult to conduct aspects extraction on some review datasets without clear aspects. A supervised generation model adopting a deep learning framework is able to find some latent aspects of reviews automatically. This model is better at dealing with complex review datasets containing humman written summaries.

For unsupervised scenarios, the unsupervised method (EMC-RS-RE) is faster and better at dealing with many reviews of popular e-commerce products. It is able to generate review summaries that can cover more aspects with high efficiency. In addition, the selecting model is faster than the re-ranking model in dealing with unlabelled datasets.

8 CONCLUSION

In this paper, we design two pre-processing models, the re-ranking model and the selecting model, and propose comprehensive Review Summary Generation frameworks to deal with the supervised and unsupervised scenarios. For the pre-processing: the re-ranking model is used to re-rank the

sentences of the reviews by their semantic similarity and the user's sentiment and the selecting model is used to select the review subset covering more aspects with fewer sentences. For summary generation: we apply the Encoder-decoder model to generate the review summary for supervised scenarios, and eliminate the redundancy of the selected review set or sentences to generate the summary for unsupervised scenarios. Experiments in real data sets demonstrate the advantages of our work. In the current work, we focus on generating review summaries to preserve all of the most important information in original reviews. In future work, we are interested to improve the readability of generated review summaries.

9 ACKNOWLEDGMENTS

This research was supported by NSFC grant 61632009, the Science and Technology Program of Changsha City kq2004017, Open project of Zhejiang Lab 2019KE0AB02, Guangdong Provincinal NSF Grant 2017A030308006, and in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1618398, CNS 1651947, and CNS 1564128.

REFERENCES

[1] Reinald Kim Amplayo and Mirella Lapata. 2019. Informative and Controllable Opinion Summarization. *arXiv preprint arXiv:1909.02322* (2019).

[2] Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised Opinion Summarization with Noising and Denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1934–1945.

[3] Stefanos Angelidis and Mirella Lapata. 2018. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics* 6 (2018), 17–31.

[4] Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. *arXiv preprint arXiv:1808.08858* (2018).

[5] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR)*.

[6] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.. In *International Conference on Language Resources and Evaluation, Lrec 2010, 17-23 May 2010, Valletta, Malta*. 83–90.

[7] Alexandra Balahur and Andrés Montoyo. 2008. *Multilingual Feature-Driven Opinion Extraction and Summarization from Customer Reviews*. International Conference on Application of Natural Language to Information Systems. Springer, Berlin, Heidelberg. 345–346 pages.

[8] D. Bollegala, T. Mu, and J. Y. Goulermas. 2016. Cross-Domain Sentiment Classification Using Sentiment Sensitive Embeddings. *IEEE Transactions on Knowledge and Data Engineering* 28, 2 (Feb 2016), 398–410. https://doi.org/10.1109/TKDE.2015.2475761

[9] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised Opinion Summarization as Copycat-Review Generation. In *Proceedings of Association for Computational Linguistics (ACL)*. 5151–5169.

[10] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for document summarization. *arXiv preprint arXiv:1610.08462* (2016).

[11] Eric Chu and Peter Liu. 2019. MeanSum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*. 1223–1232.

[12] Maximin Coavoux, Hady Elsahar, and Matthias Gallé. 2019. Unsupervised Aspect-Based Multi-Document Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Association for Computational Linguistics, 42–47.

[13] Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *The Workshop on Statistical Machine Translation*. 376–380.

[14] Giuseppe Di Fabbrizio, Amanda Stent, and Robert J Gaizauskas. 2014. A Hybrid Approach to Multi-document Summarization of Opinions in Reviews.. In *INLG*. 54–63.

[15] Xiaofei Ding, Wenjun Jiang, and Jiawei He. 2018. Generating Expert's Review from the Crowds': Integrating a Multi-Attention Mechanism with Encoder-Decoder Framework. In *the 15th IEEE International Conference on Ubiquitous Intelligence and Computing (IEEE UIC)*. 954–961.

[16] Yunqi Dong and Wenjun Jiang. 2019. Brand purchase prediction based on time-evolving user behaviors in e-commerce. *Concurrency and Computation: Practice and Experience* 31, 1 (2019), e4882.

[17] Hady Elsahar, Maximin Coavoux, Matthias Gallé, and Jos Rozen. 2020. Self-Supervised and Controlled Multi-Document Opinion Summarization. *arXiv preprint arXiv:2004.14754* (2020).

[18] Erkan, Radev, and R Dragomir. 2004. LexRank: graph-based lexical centrality as salience in text summarization. *Journal of Qiqihar Junior Teachers College* 22 (2004), 2004.

[19] Carlos Flick. 2004. ROUGE: A Package for Automatic Evaluation of summaries. In *The Workshop on Text Summarization Branches Out*. 10.

[20] Kavita Ganesan, Cheng Xiang Zhai, and Jiawei Han. 2010. Opinosis:A Graph-based Approach to Abstractive Summarization of Highly Redundant Opinions. In *COLING 2010*. 340–348.

[21] Daniel Gillick, Benoit Favre, and Dilek Hakkani-Tür. 2008. The ICSI Summarization System at TAC 2008.. In *Tac*.

[22] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393* (2016).

[23] Emitza Guzman and Walid Maalej. 2014. How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd international requirements engineering conference (RE). IEEE*. 153–162.

[24] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 388–397.

[25] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–177.

[26] Chunli Huang, Wenjun Jiang, Jie Wu, and Guojun Wang. October, 2020. Personalized Review Recommendation based on Users' Aspect Sentiment. *ACM Trans, Internet Technol (TOIT)* 20, 4 (October, 2020), 1533–5399. https://doi.org/10.1145/3414841

[27] Wenjun Jiang, Guojun Wang, Md Zakirul Alam Bhuiyan, and Jie Wu. 2016. Understanding graph-based trust evaluation in online social networks: Methodologies and challenges. *ACM Computing Surveys (CSUR)* 49, 1 (2016), 10.

[28] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 597–606.

[29] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*. 957–966.

[30] Theodoros Lappas, Mark Crovella, and Evimaria Terzi. 2012. Selecting a characteristic set of reviews. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 832–840.

[31] Theodoros Lappas and Dimitrios Gunopulos. 2010. Efficient Confident Search in Large Review Corpora. In *European Conference on Machine Learning and Knowledge Discovery in Databases*. 195–210.

[32] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057* (2015).

[33] Xueqi Li, Wenjun Jiang, Weiguang Chen, Jie Wu, and Guojun Wang. 2019. HAES: A New Hybrid Approach for Movie Recommendation with Elastic Serendipity. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 1503–1512.

[34] Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: the 2010 Conference of the North American Chapter of the Association for Computational Linguistics*. 912–920.

[35] Peng Liu, Yue Ding, and Tingting Fu. 2019. Optimal throwboxes assignment for big data multicast in vdtns. *Wireless Networks* (2019), 1–11.

[36] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *ACM SIGIR*. ACM, 43–52.

[37] Prem Melville, Wojciech Gryc, and Richard D Lawrence. 2009. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *SIGKDD*. ACM, 1275–1284.

[38] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A Meeting of Sigdat, A Special Interest Group of the Acl, Held in Conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*. 404–411.

[39] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[40] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. 3 (2014), 2204–2212.

[41] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical programming* 14, 1 (1978), 265–294.

[42] Thanh-Son Nguyen, Hady W Lauw, and Panayiotis Tsaparas. 2013. Using micro-reviews to select an efficient set of reviews. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 1067–1076.

[43] L Page. 1998. The PageRank citation ranking : Bringing order to the web. *Stanford Digital Libraries Working Paper* 9, 1 (1998), 1–14.

[44] Nikolaos Pappas and Andrei Popescu-Belis. 2017. Explicit document modeling through weighted multiple-instance learning. *Journal of Artificial Intelligence Research* 58 (2017), 591–626.

[45] Liu Peng, Wang Chaoyu, Hu Jia, Fu Tingting, Cheng Nan, Zhang Ning, and Shen Xuemin. 2020. Joint Route Selection and Charging Discharging Scheduling of EVs in V2G Energy Network. *IEEE Transactions on Vehicular Technology* (2020).

[46] Ana Maria Popescu and Orena Etzioni. 2005. Extracting Product Features and Opinions from Reviews. In *Hlt/emnlp on Interactive Demonstrations*. 32–33.

[47] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40, 6 (2004), 919–938.

[48] Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*. 12–21.

[49] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (2015).

[50] K. Schouten and F. Frasincar. 2016. Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, 3 (March 2016), 813–830. https://doi.org/10.1109/TKDE.2015.2485209

[51] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* (2017).

[52] Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. OpinionDigest: A Simple Framework for Opinion Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5789–5798.

[53] Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. OpinionDigest: A Simple Framework for Opinion Summarization. *arXiv preprint arXiv:2005.01901* (2020).

[54] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[55] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. From Neural Sentence Summarization to Headline Generation: A Coarse-to-Fine Approach.. In *IJCAI*. 4109–4115.

[56] Jiwei Tan, Xiaojun Wan, Jianguo Xiao, Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. From Neural Sentence Summarization to Headline Generation: A Coarse-to-Fine Approach. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*. 4109–4115.

[57] Panayiotis Tsaparas, Alexandros Ntoulas, and Evimaria Terzi. 2011. Selecting a comprehensive set of reviews. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–176.

[58] Jingjing Wang, Wenjun Jiang, Kenli Li, and Keqin Li. 2021. Reducing Cumulative Errors of Incremental CP Decomposition in Dynamic Online Social Networks. *ACM Trans. Knowl. Discov. Data*, Article 1 (2021), 32 pages. https://doi.org/10.1145/3441645

[59] Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. *arXiv preprint arXiv:1606.02785* (2016).

[60] Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2016. A sentence compression based framework to query-focused multi-document summarization. *arXiv preprint arXiv:1606.07548* (2016).

[61] Peike Xia, Wenjun Jiang, Jie Wu, Surong Xiao, and Guojun Wang. 2021. Exploiting Temporal Dynamics in Product Reviews for Dynamic Sentiment Prediction at the Aspect Level. *ACM Trans. Knowl. Discov. Data*, Article 1 (2021), 28 pages. https://doi.org/10.1145/3441451

[62] Naitong Yu, Minlie Huang, Yuanyuan Shi, and Xiaoyan Zhu. 2016. Product Review Summarization by Exploiting Phrase Properties. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 1113–1124.

[63] Jifeng Zhang, Wenjun Jiang, Jie Wu, and Guojun Wang. 2021. Predict Activity Attendance in Event-based Social Network: From the Organizer's View. *ACM Transactions on the WEB (TWEB)*, Article 1 (2021), 25 pages. https://doi.org/10.1145/3440134

[64] W. Zhao, Z. Guan, L. Chen, X. He, D. Cai, B. Wang, and Q. Wang. 2018. Weakly-Supervised Deep Embedding for Product Review Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 30, 1 (Jan 2018), 185–197. https://doi.org/10.1109/TKDE.2017.2756658

[65] X. Zhou, X. Wan, and J. Xiao. 2016. CMiner: Opinion Extraction and Summarization for Chinese Microblogs. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (July 2016), 1650–1663. https://doi.org/10.1109/TKDE.2016.2541148

A APPENDIX: HUMAN EVALUATION

We perform human evaluation using best-worst scaling as in [9]. We use their test dataset which includes 50 businesses from the human-annotated Yelp test set and 30 test products from the Amazon set. We recruited 3 workers to evaluate each summaries generated by our unsupervised model, CopyCat, and human annotators (i.e., gold summary). It is worth noting that there is no large datasets of Amazon and Yelp with human-annotated summaries for training our supervised model. Therefore, we cannot conduct human evaluation on supervised model. The reviews and summaries were presented to the workers in random order and were judged using Best-Worst Scaling.

We use the following metrics to perform the human evaluation. It includes: (1) Fluency: the summary sentences should be grammatically correct, easy to read and understand; (2) Coherence: the summary should be well structured and well organized; (3) Non-redundancy: there should be no unnecessary repetition in the summary; (4) Opinion consensus: the summary should reflect common opinions expressed in the reviews; (5) Consistency: the review summary should be consistent with the original reviews; (6) Richness: the summary should contain more detailed product descriptions; (7) Overall: based on your own criteria (judgment) please select the best and the worst summary of the reviews.

For every criterion, a system's score is computed as the percentage of times it was selected as best minus the percentage of times it was selected as worst. The scores range from -1 (unanimously worst) to +1 (unanimously best). The results are shown in Table 11.

Table 11. Human evaluation results in terms of the Best-Worst scaling

Method	Amazon							Yelp						
	Fluency	Coherence	Non Red.	Opinion Cons.	Consistency	Richness	Overall	Fluency	Coherence	Non Red.	Opinion Cons.	Consistency	Richness	Overall
CopyCat	0.06	-0.067	0.02	-0.047	-0.14	-0.127	-0.067	0.033	-0.056	-0.111	-0.067	-0.122	-0.233	-0.056
EMC-RS-RE	-0.133	-0.113	0.02	0.007	**0.093**	**0.087**	0.013	-0.178	-0.044	**0.067**	-0.011	**0.111**	**0.189**	**0.044**
Gold	**0.073**	**0.093**	**0.047**	**0.04**	0.047	0.04	**0.053**	**0.144**	**0.1**	0.044	**0.078**	0.011	0.044	0.011

We can see that the review summaries generated by three models are not very different on the seven metrics. Moreover, the difference in Overall between our unsupervised model and gold summaries is not statistically significant. The summary generated by our unsupervised model is better than CopyCat on Non Redundancy, Consistency, Opinion Consensus, Richness and Overall. Compared with CopyCat, our unsupervised model performs slightly worse in Fluency and Coherence. This is because that our unsupervised model generates summaries based on aspect sentiment extraction. The more aspects contained in a review sentence, the more we think it is valuable, while ignoring the fluency of the summary itself and the relevance between sentences. The human evaluation results show that the review summary generated by our unsupervised model is less redundant, and contains more detailed product descriptions. Moreover, the review summary generated by our unsupervised model is more close to the original reviews.

B APPENDIX: A CASE STUDY ON AMAZON

In order to display the detailed process of our unsupervised model (EMC-RS-RE) on summary generation with different parameters, we conduct a case study using the labeled Amazon dataset [9]. Each product contains a few (e.g., 8) processed reviews. Tables 12 and 13 show the details of summary generation with the unsupervised model (We use different colors to indicate the aspect polarity, red for positive and blue for negative). We find that the $Top-1$ review has the highest gain/cost when $\alpha = 0.5$, $\beta \in [0.1, 0.8]$. What's more, the selected review contains more aspects, e.g., price, quality, and so on. Similarly, the second selected review also contributes to several

new aspects. The final summary generated by EMC-RS-RE is more concise and non-redundant. Compared to the summary generated by Copycat, the summary by our unsupervised model contains more aspects and more information. However, the generated summary by our model is longer than that generated by Copycat. Finally, we also summarize the weight of the aspect sentiment, which can more directly display the proportion of negative and non-negative sentiment on each aspect. Since the original review sets contain only a small number of aspects, and these aspects are concentrated in a few reviews, our unsupervised model tends to extract these reviews.

Table 12 shows the process when α increases and $\beta = 0.5$. We can see that as α varies from 0.1 to 0.5, the Rouge scores keep steady and the generated summaries are the same. Moreover, the Rouge scores are the highest when $\alpha = 0.6$ or 0.7. In EMC-RS-RE, α is used to filter out inefficient review. We can see that as α increases, the selected summary is further filtered.

Table 13 shows that with the increase of β, i.e., [0.1,0.8], the generated summary by EMC-RS-RE is the same, except when $\beta = 0.9$. This is because the dataset contains only a few reviews and each review is of high quality. It indicates that the results are not sensitive to β in the labeled Amazon dataset. Those findings are consistent with the parameter sensitivity analysis in Section 7.3.

Table 12. Case study on the labeled Amazon dataset by EMC-RS-RE ($\beta = 0.5$, varying α)

Aspect sentiment ratio (positive : negative):
problems(1:0), disc(2:0), dvd(3:2), price(2:0), quality(2:0), color(1:0), gold(3:0)

The Generated Summary by Copycat: *It's a great product. I have had no problems with it and the price is right. I would recommend this product to anyone who wants a good quality product.*

EMC-RS-RE: $\alpha = 0.1, 0.2, 0.3, 0.4, 0.5$, $\beta = 0.5$

The Summary of the Selected Top-3 reviews:
1. (Top-R1) *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. I have used these over the years for many different projects and the quality* ***(0.125)*** *is there and so is the price* ***(0.125)****. I have had trouble with some other brand named dvd's, but not with hp.*
2. (Top-R2) *I have had a ton a problems* ***(0.0625)*** *with these discs* ***(0.125)****. After about 30 minutes of a dvd* ***(0.125)****, it begins to get choppy and become unviewable. Looking at the burn side of the disc* ***(0.125)****, there is a area where you can see the burning stopped and i guess picked again. Do not recommend.*
3. (Top-R3) *Vendor describes the product as being gold* ***(0.1875)*** *in color* ***(0.0625)****. It is not it is silver. I know that hp no longer manufactures the gold* ***(0.1875)*** *was hoping this vendor had some gold version of dvd* ***(0.125)*** *+ r in it inventory. They need change the picture and description to silver instead of gold* ***(0.1875)****.*
The Generated Summary by EMC-RE-RS: *Yes, hp dvd's are dvd's for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. I have had a ton a problems* ***(0.0625)*** *with these discs* ***(0.125)****. After about 30 minutes of a dvd* ***(0.125)****, it begins to get choppy and become unviewable. Vendor describes the product as being gold* ***(0.1875)*** *in color* ***(0.0625)****.*
Rouge-1: 38.18, Rouge-2: 3.7, Rouge-L: 30.06

EMC-RS-RE: $\alpha = 0.6, 0.7$, $\beta = 0.5$

The Summary of the Selected Top-3 reviews:
1. (Top-R1) *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. I have used these over the years for many different projects and the quality* ***(0.125)*** *is there and so is the price* ***(0.125)****. I have had trouble with some other brand named dvd's, but not with hp.*
2. (Top-R2) *Vendor describes the product as being gold* ***(0.1875)*** *in color* ***(0.0625)****. It is not it is silver. I know that hp no longer manufactures the gold* ***(0.1875)*** *was hoping this vendor had some gold version of dvd* ***(0.125)*** *+ r in it inventory. They need change the picture and description to silver instead of gold* ***(0.1875)****.*
The Generated Summary by EMC-RE-RS: *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. Vendor describes the product as being gold* ***(0.1875)*** *in color* ***(0.0625)****.*
Rouge-1: 45.45, Rouge-2: 7.4, Rouge-L: 26.15

EMC-RS-RE: $\alpha = 0.8$, $\beta = 0.5$

The Summary of the Selected Top-3 reviews:
1. (Top-R1) *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. I have used these over the years for many different projects and the quality* ***(0.125)*** *is there and so is the price* ***(0.125)****. I have had trouble with some other brand named dvd's, but not with hp.*
The Generated Summary by EMC-RE-RS: *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****.*
Rouge-1: 25.45, Rouge-2: 1.8, Rouge-L: 18.0

Table 13. Case study on the labeled Amazon dataset by EMC-RS-RE (α = 0.5, varying β)

Aspect sentiment ratio (positive : negative): *problems(1:0), disc(2:0), dvd(3:2), price(2:0), quality(2:0), color(1:0), gold(3:0)*
The Generated Summary by Copycat: *It's a great product. I have had no problems with it and the price is right. I would recommend this product to anyone who wants a good quality product.*
EMC-RS-RE: $\alpha = 0.5$, $\beta = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$
The Summary of the Selected Top-3 reviews: ***1. (Top-R1)*** *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. I have used these over the years for many different projects and the quality* ***(0.125)*** *is there and so is the price* ***(0.125)****. I have had trouble with some other brand named dvd's, but not with hp.* ***2. (Top-R2)*** *I have had a ton a problems* ***(0.0625)*** *with these discs* ***(0.125)****. After about 30 minutes of a dvd* ***(0.125)****, it begins to get choppy and become unviewable. Looking at the burn side of the disc* ***(0.125)****, there is a area where you can see the burning stopped and i guess picked again. Do not recommend.* ***3. (Top-R3)*** *Vendor describes the product as being gold* ***(0.1875)*** *in color* ***(0.0625)****. It is not it is silver. I know that hp no longer manufactures the gold* ***(0.1875)*** *was hoping this vendor had some gold version of dvd* ***(0.125)*** *+ r in it inventory. They need change the picture and description to silver instead of gold* ***(0.1875)****.* **The Generated Summary by EMC-RE-RS**: *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. I have had a ton a problems* ***(0.0625)*** *with these discs* ***(0.125)****. After about 30 minutes of a dvd* ***(0.125)****, it begins to get choppy and become unviewable. Vendor describes the product as being gold* ***(0.1875)*** *in color* ***(0.0625)****.* **Rouge-1: 38.18, Rouge-2: 3.7, Rouge-L: 30.06**
EMC-RS-RE: $\alpha = 0.5$, $\beta = 0.9$
The Summary of the Selected Top-3 reviews: ***1. (Top-R1)*** *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. I have used these over the years for many different projects and the quality* ***(0.125)*** *is there and so is the price* ***(0.125)****. I have had trouble with some other brand named dvd's, but not with hp.* ***2. (Top-R2)*** *Vendor describes the product as being gold* ***(0.1875)*** *in color* ***(0.0625)****. It is not it is silver. I know that hp no longer manufactures the gold* ***(0.1875)*** *was hoping this vendor had some gold version of dvd* ***(0.125)*** *+ r in it inventory. They need change the picture and description to silver instead of gold* ***(0.1875)****.* **The Generated Summary by EMC-RE-RS**: *Yes, hp dvd's* ***(0.1875)*** *are dvd's* ***(0.1875)*** *for the better. Better price* ***(0.125)****. Better quality* ***(0.125)****. Vendor describes the product as being gold* ***(0.1875)*** *in color* ***(0.0625)****.* **Rouge-1: 45.45, Rouge-2: 7.4, Rouge-L: 26.15**