Information Sciences 560 (2021) 283–306

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Opinion subset selection via submodular maximization

Yang Zhao, Tommy W.S. Chow *

Department of Electrical Engineering, City University of Hong Kong, 83 Tat Chee Av., Kowloon Tong, Hong Kong Special Administrative Region

ARTICLE INFO

Article history:
Received 9 October 2019
Received in revised form 9 December 2020
Accepted 30 December 2020
Available online 10 February 2021

2010 MSC:
00-01
99-00

Keywords:
Subset selection
Opinion mining
Opinion summarization
Submodular function optimization
Opinion subset selection
Sentiment analysis

ABSTRACT

Current research on subset selection for opinion analysis assumes that their methods can retrieve the opinions expressed in documents from general text features. However, such relaxed conditions can hardly maintain the performance of the analysis in opinion mining, especially when given strict limitations on the subset size. In this paper, we propose a framework for opinion subset selection. This framework can select a small set of instances from original data to convey a subjective representation for opinion classification and regression. Compared with our framework, the conventional submodular based subset selection approach cannot capture the fine-grained opinion features expressed in the corpus. Specifically, we propose a monotone non-decreasing score function and a framework based on topic modeling and submodular maximization for filtering irrelevant information and selecting the subsets. Our work further introduces an opinion-sensitive algorithm for optimizing the proposed function for opinion subset construction. We perform extensive experiments and comparative analysis of different subset selection methods in this work. The experimental result shows that the proposed opinion subset selection framework can compress the original text training set and preserve the test set's classification and regression metric performance at the same time.

1. Introduction

The user-generated text increases tremendously due to the development of the social network and machine storage. Under the circumstances, subset selection for text analysis has become cumulatively crucial. Subset selection was firstly proposed in the feature selection area, which is called feature subset selection [21]. Feature subset selection attempts to select important features rather than construct or create new features for machine learning-based data analysis [11]. Afterward, lots of other researchers [9,29,14,50,15,3,25,42,35,31,18,22,28,20,48,32,16] started to extend the concept of subset selection from the feature perspective to instance perspective. They proposed many definitions of subset selection from this perspective, including instance subset selection [16], data subset selection [48], data summarization [31], or coresets [9,29,14,50,15,3]. Despite the different ways in extracting instance subsets, they all shared a similar objective of providing a robust algorithm for constructing the subset, which means to describe the characteristics of the whole data set by a small significant representative set of samples within a given instance amount limitation. There are lots of applications for this research area. Campbell and Broderick [9] proposed to use subset selection for Bayesian inference. Their work showed the possibility of applying subset selection techniques in information extraction. Other researchers applied subset selection

[*] Corresponding author.
E-mail address: eetchow@cityu.edu.hk (T.W.S. Chow).

https://doi.org/10.1016/j.ins.2020.12.083
0020-0255/© 2021 Elsevier Inc. All rights reserved.

in the machine learning area, including regression problem [50], clustering problem [3], dimension reduction [15], and compression of the convolutional neural network [14].

There are a massive number of opinion statements on news, blogs, or other kinds of text containers posted online every day. Apparently, there is a need to develop a systematic and robust extraction method to mine opinions expressed in various forms, such as positive attitude, sarcasm, or other valuable sentiments. Hence, the research on opinion or sentiment analysis emerged. According to Liu [26], the research on sentiment analysis or opinion mining started since the first work by Wiebe [49] and was officially proposed by Nasukawa and Yi [34] three years later. These two terms, sentiment analysis and opinion mining, are interchangeably used in the opinion analysis research.

On the other hand, users can hardly perceive natural language opinion statements in large amounts. Thus, it becomes an important and challenging issue for employing machine intelligence to extract the right amounts of opinions out of the large-scale opinion statements corpus. Conventional one-hot features are constructed using n-grams, which usually are word or phrase tokens from the opinion statements. These tokens contain many differences of opinions, which are represented by a polarity-based system. The classifier for separating opinions in these tokens can be misled by the massive amounts of polarity sets with different labels. Given this problem, submodular maximization was then introduced to deliver a subset representing the main opinions existing in a given corpus. Lots of attempts were made to deal with the subset selection problems with submodular maximization [25,42,31,18,20,48,32]. Most of them [25,42,31,18,32,47] focused on text summarization, but some applied submodularity to a general subset selection context [42,20,48]. Shinohara [42] used a submodular optimization approach to select sentence subsets. Kai, Iyer and Bilmes [48] performed the submodular subset selection combined with active learning. Kirchhoff and Bilmes [20] proposed a set of submodular functions in the machine translation area to perform the text selection. Their experimental results showed that adopting subsets can achieve higher training speed and can maintain the same translation performance compared with the original training data. The used subset was constructed by submodular optimization and contains only 10% to 40% of data in the original training set. Experimental studies revealed the potential performance of the subset in other NLP applications.

To further investigate the capability of subset selection in opinion mining, we propose a new method for opinion subset selection using topic modeling and submodular optimization, formally defined in Section 3. The contributions of this paper include the followings:

- We propose a framework to enhance subset selection for opinion analysis, where existing selection approaches cannot deliver subsets concerning opinion features.
- Compared with submodular subset selection methods that only employ coverage and relevance features, the proposed submodular score function in the framework can select document instances according to fine-grained opinion features for its category. Additionally, an opinion-sensitive optimization algorithm is also introduced for the framework.
- The experimental results analysis exhibits that our proposed framework can improve the performance of opinion analysis in terms of four opinion classification metrics: accuracy, f1-score, recall, and precision.

The remainder of this paper is structured as follows: Section 2 discusses the related works of our proposed opinion subset selection concept. Section 3 defines the problem of the opinion subset selection and our proposed solution. In Section 4, the results of the experiment are presented. We perform a thorough analysis on our method against six subset selection methods, including a graph-based and five submodular-based ones. Section 5 provides a conclusion and lists the possibly adaptable applications of our opinion subset selection approach.

2. Related work

In this part, we first briefly recapitulate the concept and recent developments of subset selection and its homogeneous types, such as coresets. After that, we introduce the submodular maximization and fine-grained opinion mining in natural language processing, which can be employed to investigate opinion subset selection in the following sections.

2.1. Text feature selection and coresets

Opinion mining tasks, such as document opinion classification, are challenging due to its high dimensionality and diversity of text features [26]. To solve such issues, feature selection is applied in the opinion mining areas. Song, Ni and Wang [43] offered a cluster-based method for text feature subset selection. After removing the irrelevant feature, a minimum spanning tree was built and clustered. The selected text features were then constructed from each cluster as a subset for further analysis. Rouane, Belhadef and Bouakkaz [40] proposed another cluster-based method. K-means clustering was applied to the corpus first, and then the feature subsets were selected using the Apriori Algorithm in each cluster. Cavaliere, Senatore and Loia [10] built a graph to describe the semantic relationship between the documents in the corpus. Then the clusters were built as concepts to express the features. Rather than selecting the text features, our proposed framework focuses on selecting the instances (documents). The framework uses the topic-based method to filter irrelevant documents, which is considered to be the candidate document selection. After that, the document subsets are constructed by using the greedy submodular selection. The detailed explanation of this process is in Section 3.

Researches in [9,29,14,50,15,3] proposed a concept called *coresets* for dataset subset selection. The goal of coresets is to find a weighted subset of the given data. The opinion analysis model can deliver the same or better performance with relatively low computational cost by using this subset.

Indyk et al. [29] tried to construct the coresets for determinant maximization using the greedy and local-search algorithm. They proposed a *composable* property for the coresets. It means that the union of multiple selected coresets can represent the main characteristics of the union of multiple original datasets. If considering the data instances under different categories as distinct datasets, our work heuristically matches this composable property by combining the selected subsets under each category, precisely, positive, negative, and neutral. Their work focused on determinant maximization that can be regarded as a non-monotone submodular maximization problem. In this problem, the logarithm of their objective function holds the submodularity. However, our proposed function is strictly monotone submodular, which is proved in Section 3.4. Dubey, Chatterjee, and Ahuja [14] applied the coresets concept to parameter compression in the convolutional neural network (CNN). They proposed a filter coresets construction method that is retraining free and applicable to convolutional and fully-connected layers in various CNNs. On the other hand, their work revealed the existence of parameter redundancy in most CNNs. Their experimental results showed that the coresets could reduce the size and improve the efficiency of CNN simultaneously. The goal of our proposed opinion subset selection method also falls into a similar selection condition, which is to seek high performance and efficiency concurrently. Compared with Dubey, Chatterjee, and Ahuja's method, our proposed method aims at instance subset selection for general classifiers instead of a specific one. Yan and Phillips [50] constructed the coresets for Gaussian kernel regression. Their work tested the effectiveness of existing aggregation methods for coresets construction with a proven performance lower bound. Instead of proposing our own lower bound for analysis, our method already fits into an existing lower bound, which was proved in [35] under the circumstance of using a greedy algorithm.

2.2. Subset selection and document summarization

Submodular subset selection has been widely adopted in text analysis. To obtain an optimal value of a submodular function is an NP-complete optimization problem [24]. As long as the objective function holds the submodularity, Nemhauser, Wolsey, and Fisher [35] proved that maximizing a monotone submodular function by the greedy algorithm can achieve at least $1 - \frac{1}{e}$ of the optimal solution.

The results presented in [25,31,18,32,47,46] showed the probability of applying submodular optimization to document summarization using the greedy algorithm. Extractive document summarization is to provide a summary (a collection of sentences) of a single long passage or a set of multiple documents. It is similar to subset selection, where the summary can be considered as a representation of the given multiple data instances. Lin and Bilmes [25] proposed the concept of fidelity and diversity reward for text summarization. Based on these two concepts, their work delivered convincing results for forming a multi-document summary. Morita et al. [32] used a positive reward concept to maintain the readability of the output text summary. Van Lierde and Chow [46] integrated the concept of coverage with the fuzzy hypergraph and relevance into a query-based system. These text summarization works inspire us to utilize the relevant concepts for text subset selection at the document (instance) level.

Liu [26] defined opinion document summarization based on aspect level. He introduced an enhancement that each sentence can be selected according to its aspects in the ontology. As a result, a summary was created from the aspect-split sentence sets. Wan and Wang [47] researched on representing topics using text summarization techniques. Precisely, their work is based on submodular maximization. Motivated by these two works, our proposed method splits the original set into different topics using Latent Dirichlet Allocation (LDA) [6]. Each of these topics is regarded as an aspect. After that, our proposed method is applied for further subset selection under each of these aspects.

Jaynth, Sundararaj, and Bhattacharyya [18] proposed three functions to model the opinions in the dataset: an essential submodular function, a budget-additive function, and a facility location function. Then they further extended the last two types of functions by employing the opinion polarity information. Their work used an opinion knowledge base for opinion modeling. It required an automatic information extraction step for labeling the word tokens for opinion analysis. It is worth noting that this step was prone to generating errors that pose a significant effect on the subsequent analysis. Our proposed method addresses this issue by an opinion submodular function which employs a fine-grained opinion word embedding. It does not require the labeling step, and thus it can offer higher performance and lower cost. The description of the method will be illustrated in Section 3.

2.3. Fine-grained opinion mining

Attempts [27,4,2,1,45] were made on the exploration of fine-grained opinion mining. Liu, Joty, and Meng [27] brought the Recurrent Neural Network (RNN) to the fine-grained opinion mining. They adopted the pre-trained word embedding for the initialization of RNN input and then fine-tuned the embedding with task-specific linguistic features, such as part-of-speech tags. Their results showed that the deep framework could provide a significant increase in fine-grained opinion mining. Balikas, Moura and Amini [4] also applied RNN to fine-grained opinion mining. The additional text features were incorporated as the same layer of Liu's work. However, rather than only fine-tuned to a specific task, their work proposed a multitask learning model. Angelidis and Lapata [1] applied CNN to fine-grained opinion mining tasks. This work could predict the polarity score of a document by extracting the opinion in a series of clauses rather than just word tokens. These three works showed the capability of deep models to construct the fine-grained opinion embeddings.

Aragón et al. [2] proposed a feature representation method called the bag of sub-emotions. Their work demonstrated that the opinion mining performance could be improved by integrating the fine-grained emotions into the feature representation. Tang et al. [45] proposed a topic modeling-based learning method for fine-grained opinion mining. This work gave explicit evidence that the topics underlying the corpus could be employed for opinion mining. Inspired by their work, we utilize LDA as the topic modeling approach, which is integrated into our proposed framework for opinion subset selection.

However, all the mentioned works modeled opinions either using the conventional polarity system or dividing the fine-grained aspects for a particular task or domain. Instead, Poria et al. [39] divided the instinctive positive and negative based opinion polarity into four different parts and mapped them to a three-dimensional space. In this space, the polarities are positioned like an hourglass following the Gaussian opinion assumption proposed in their work. This 16 chunks opinion system was the foundation of the SenticNet knowledge base [8] for fine-grained opinion recognition. Starting from SenticNet 5, the word embedding in SenticNet was trained using a deep long short-term memory (LSTM) network. In our proposed work, we integrate the fine-grained opinion detection capability offered by SenticNet 5 into our proposed opinion function described in Section 3.4.3 and opinion-sensitive algorithm illustrated in Section 3.5. The proposed submodular function and optimization algorithm can help the system perceive the opinion expressed in each document precisely when performing subset selection.

3. Problem formulation and methodology

3.1. Opinion subset selection

There are two primary objectives of selecting a subset of document instances from a given corpus. First, it is to retrieve a subset representing the main opinion in both subjectivity and polarity of a corpus. Second, due to the property of repetition in natural language, we need to reduce the redundancy existed in all large-scale text datasets. To simplify the terminology, we call and define the proposed opinion subset selection (OSS) as

> *Selecting a subset of document instances from the original data to convey subjective impressions that delivering the main opinion characteristics for opinion classification and regression.*

To accomplish the OSS task, a subset from the original set can be constructed by maximizing the score of three different properties in documents: diversity, relevance, and opinion. The selected subset should cover the semantic information in the original dataset as much as possible. The details can be elucidated as follows:

Text Diversity: This property enables the subset to provide a diversified description of the corpus from different views. As the nature of the text, the document subset can describe an object from various aspects. For example, different users can hold distinct feelings for different advantages of the same product. This property encourages the subset to cover such opinions as much as possible.

Text Relevance: The chosen subset naturally tries to depict the corresponding category by representation analogy. The higher the relevance, the better the result. However, if the relevance is too strong, the result may suffer from the homogenization problem, where the selected document instances may have similar text features. The previous property can balance the selected subset between strong information relevance and multiplicity.

Text Opinion: The opinion correlations between a document subset and a given corpus should be encouraged. The precise opinion mapping can provide the selected subset of the capability to have a strong sentiment linkage. The opinion can be fine-grained first and then represented by using a knowledge base. The subset with a higher opinion score should fulfill our goal of opinion selection.

3.2. System framework

We propose a framework to perform opinion subset selection and analyze the subset validity by using opinion classification metrics. The workflow of this framework is depicted in Fig. 1. First, basic text preprocessing steps are employed, including standard text normalization, contraction expansion, special character removal, stopword removal, and tokenization. After that, the term frequency and inverse document frequency (TFIDF) feature extraction method is applied to transform the tokens into a one-hot encoding feature matrix for the next step. In the following topic modeling step, we utilize LDA to model the generated TFIDF feature matrix. LDA produces several topic vectors. Each vector is represented by a collection of word features to form the topic matrix from the original feature matrix.

The corpus does not consist of tags to indicate the topic of each document instance. Hence the distance between each instance feature vector and each topic vector is calculated for assigning the document into the corresponding topic. Under each topic, the documents are ranked first and then filtered by a given portion. This step is called the candidate document selection. In the penultimate step, candidate document instances are fed into the greedy submodular selection module to

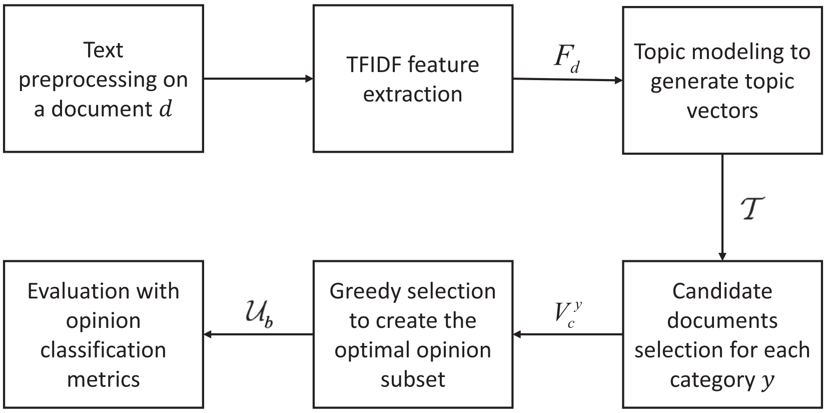

Text preprocessing on a document d
TFIDF feature extraction
F_d
Topic modeling to generate topic vectors
$\mathcal{T}$
Candidate documents selection for each category y
V_c^y
Greedy selection to create the optimal opinion subset
$\mathcal{U}_b$
Evaluation with opinion classification metrics

Fig. 1. The flow chart of the proposed opinion subset selection framework. The mechanism of each step are shown, where d is a given document, F_d represents TFIDF features of words in $d, \mathscr{T}$ is a set of topic vectors, V_c^y is a candidate document set for category y, and $\mathscr{U}_b$ is the optimal output opinion document subset.

form an output subset. In the final step, the classification metrics based on different selected subsets and the original set are compared for subset validity checks.

In the context of opinion classification, a training set contains different document instances, and their categories are given. Under each category lying several topics, each topic can be represented as a polynomial distribution over terms or words. The goal of the proposed method in the penult step is to select document instances as subsets, stating all the generated topics and their latent opinions. Meanwhile, these subsets should still be sufficient for the opinion classification.

3.3. Candidate documents selection

The training set is a collection of documents, paragraphs, or sentences. We define one of them as d_i, and its features can be $F_d = tfidf_{w,d}$, where the $tfidf_{w,d}$ means the TFIDF features of words in d_i across all documents in V. V is denoted as $V = \{(d_i, y_i)\}_i^n$, where the whole training set contains n different documents and label tuples. $d \in \mathscr{X}$ are samples from a finite document set $\mathscr{X}$. We define V^y as all the documents labeled with $y, y \in [POS, NEG, NEU]$, which means positive, negative, and neutral. The set V is considered as a combination of documents of each category, $V = V^{POS} \cup V^{NEG} \cup V^{NEU}$. From set V, for each label y, different latent topics are generated for V^y. Our objective is to create a subset for the training documents under each category. Hence, we apply LDA to each V^y to generate top $|\mathscr{T}|$ topics, and we denote each topic by $\alpha \in \mathscr{T}$, where $\mathscr{T}$ is a set of topic vectors. For each topic α, we have a polynomial distribution over words $\{p_\alpha(w_i)\}_i^k$, where k is the dimension of the word features and $\sum_i^k p_\alpha(w_i) = 1$ (see Table 1).

However, not all of the words in the training set are semantically close to its category, or its topics. The corpus in the real world can easily contain thousands of documents and millions of words. Some of these documents may be expressed as positive but falsely labeled as negative. For example, a user writes, "I like the cake in this restaurant." and gives a 1-star rate for this comment. To address this kind of outlier problem, for each topic α, we rank the semantic closed documents first and then eliminate the tail instances. We name this pre-selected set as a *candidate document set*. This method can increase the efficiency of creating document subsets and enable the subsets to have a comparatively strong correlation with its topics.

To select a candidate document set, for all documents under each category y, cosine similarity between the topic α and the document d is used

$$c(\alpha, d) = \frac{\alpha F_d}{\|\alpha\| \|F_d\|}. \tag{1}$$

The most desirable documents are selected. They are sorted from low to high by the values of $c(\alpha, d)$, forming a *candidate document set* for each topic, denoted as V_α^y. This set is then concatenated to construct the candidate document set for each category V_c^y, where $V_c^y = V_{\alpha_1}^y \cup V_{\alpha_2}^y \cup \cdots \cup V_{|\mathscr{T}|}^y$. After this process, the candidate document set becomes the input data for the next stage. The number of documents in the candidate set under each topic is

$$|V_\alpha^y| = \frac{|V_c^y|}{|\mathscr{T}|} = CandiRate * \frac{|V^y|}{|\mathscr{T}|}, \tag{2}$$

where $|V^y|$ is the cardinality of the document instances under a particular category (with the same label). The *CandiRate* is called the *candidate rate*, which expresses the portion of documents kept for each topic. $|\mathscr{T}|$ is the number of topics in a specific category. Similar to V, V_c can also be represented as $V_c = V_c^{POS} \cup V_c^{NEG} \cup V_c^{NEU}$.

Table 1
List of important symbols.

Symbol	Description
d	a document
F_d & $tfidf_{w,d}$	TFIDF features of words in d
V^y	all the documents labeled with y
V^y_α	candidate document set for each topic
V^y_c	candidate document set for each category
α	a topic vector
$\mathcal{T}$	a set of topic vectors
$c(\alpha, d)$	cosine similarity between the topic α and the document d
$\mathcal{U}$	the document subset
$\mathcal{U}_b$	the optimal opinion document subset
$p_\alpha(w)$	the probability of word w in topic α

Algorithm 1 describes the process of selecting the candidate set under each subcorpus with a specific opinion label. The topics are generated first, where the number of topics is given. Then the documents with the same label are filtered according to their similarity with each topic. After removing the lower-ranked documents, the remained are concatenated to form the candidate document set. Fig. 2 depicts the procedure of the candidate document selection.

Algorithm 1. Algorithm for Candidate Set Selection

```
Input: $CandiRate$, $V^y$
Output: $V^y_c$
Data: Training set $V^y$ where $y \in [POS, NEG, NEU]$
$V^y_c \leftarrow \emptyset$
$\mathcal{T} \leftarrow LDA(V^y)$
$C_t \leftarrow V^y$
for topic $\alpha \in \mathcal{T}$ do
    $V^y_{sort} \leftarrow argsort_{d \in V^y} c(\alpha, d)$
    $V^y_\alpha \leftarrow top(V^y_{sort}, CandiRate)$    // top(V, x) can get the top x portion of the sorted items in set V.
    $V^y_c \leftarrow V^y_c \cup V^y_\alpha$
    $C_t \leftarrow V^y \backslash V^y_\alpha$
```

3.4. Submodular objective function

For selecting the most reliable subjective document subset representing the whole training document, submodular maximization is used for creating a document subset for each topic. We propose an objective function for subset selection under each category y as

$$\mathcal{U}_b = \max_{\mathcal{U} \in \xi_{|\mathcal{T}|}} \{f(\mathcal{U}) : |\mathcal{U}| \leqslant B\}, \tag{3}$$

where $\mathcal{U}_b$ is the optimal opinion document subset, and $\mathcal{U}$ is the document subset. $|\mathcal{U}|$ is the length of the subset, which means the number of documents exists in $\mathcal{U}$. $f(\mathcal{U})$ is the score function for creating the optimal document subset of each topic α. The whole objective function explains for itself. Our objective is to maximize the $f(\mathcal{U})$ under the limitation of B, defined as

$$B = SelectRate * |V^y| = \frac{SelectRate}{CandiRate} * |V^y_c|, \tag{4}$$

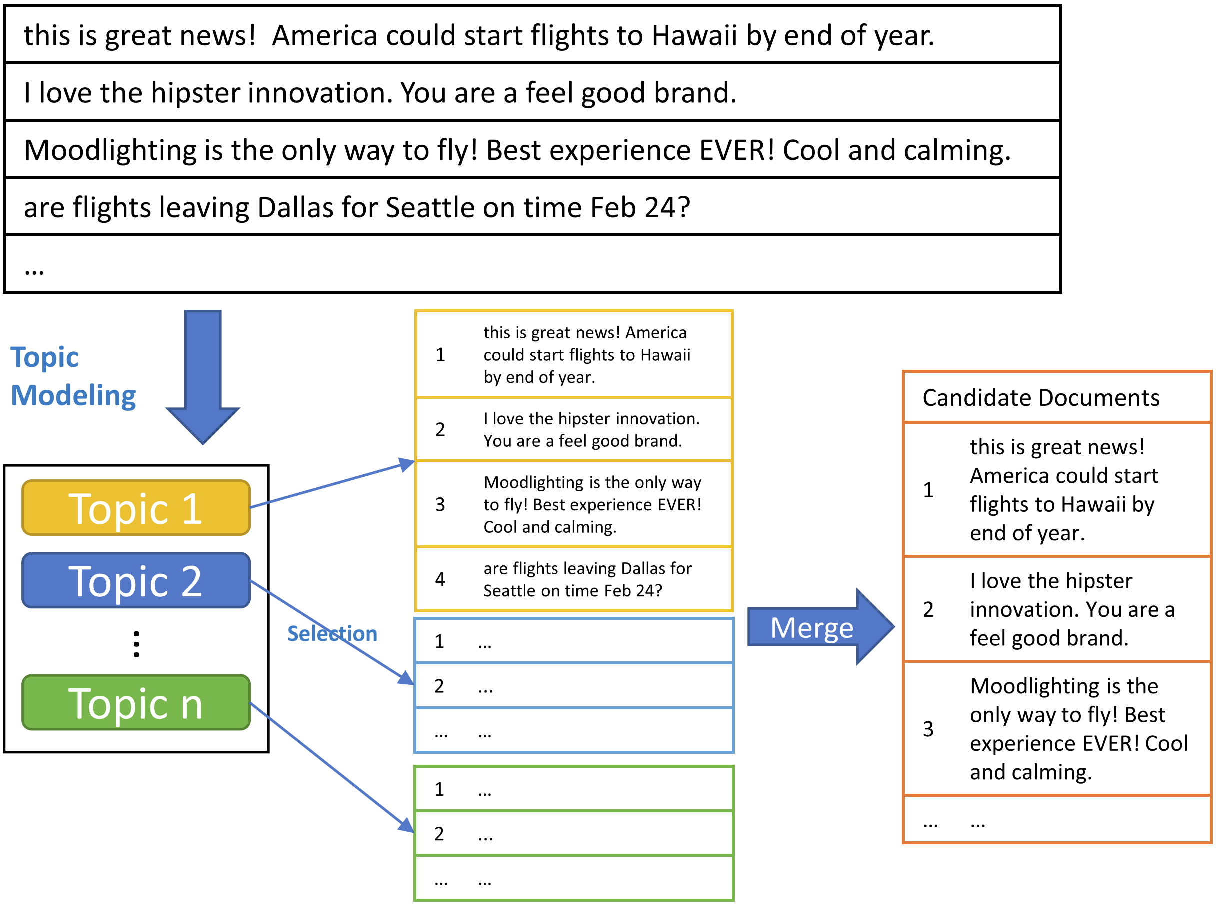

this is great news! America could start flights to Hawaii by end of year.
I love the hipster innovation. You are a feel good brand.
Moodlighting is the only way to fly! Best experience EVER! Cool and calming.
are flights leaving Dallas for Seattle on time Feb 24?
...
Topic Modeling
Topic 1
Topic 2
⋮
Topic n
Selection
1 this is great news! America could start flights to Hawaii by end of year.
2 I love the hipster innovation. You are a feel good brand.
3 Moodlighting is the only way to fly! Best experience EVER! Cool and calming.
4 are flights leaving Dallas for Seattle on time Feb 24?
1 ...
2 ...
... ...
1 ...
2 ...
... ...
Merge
Candidate Documents
1 this is great news! America could start flights to Hawaii by end of year.
2 I love the hipster innovation. You are a feel good brand.
3 Moodlighting is the only way to fly! Best experience EVER! Cool and calming.
... ...

Fig. 2. The candidate documents selection procedure.

where the *SelectRate* is a pre-defined coefficient, called the *selection rate*. It controls the percentage of document instances to keep after the greedy selection for document set under each category V^y, namely, for the whole document set V. B can also be regarded as the sum of the number of documents kept for representing each topic after filtering the redundant data.

According to the linguistic characteristics of the opinion documents, the optimal document subset should represent the topics and its corresponding category subjectively. In the context of topics, as defined in Section 3, the selected subsets should contain items that fulfill high *diversity*, *relevance*, and *opinion* for their topics generated from the original corpus. *Diversity* is a property that the subset should cover different aspects of the original semantic meaning as much as possible. Since the subset usually holds more than one documents, it should be able to describe the whole distribution of words in a topic rather than focus on only a few synonyms. *Relevance* represents that the subset should have strong semantical relevance with its category. This property is an inside nature that any subsets should have. High relevance is a crucial hallmark to evaluate the generated subset. *Opinion* makes the sentiment expressed in the subset of one topic different from sentiments in subsets of other topics under the same category. Considering that the topics under the same category may have similar sentiment expressions, the property of *opinion* can help to generate a subjective subset as unique as possible for representing the associated topic. Hence, we can get the most reliable opinion document subset by maximizing the opinion score.

For the three listed properties, we propose a submodular score function

$$f(\mathscr{U}) = \mathscr{D}(\mathscr{U}) + \mathscr{R}(\mathscr{U}) + \mathscr{O}(\mathscr{U}). \tag{5}$$

This function evaluates the overall quality of the selected document subsets, where $\mathscr{D}(\mathscr{U})$ gives the diversity score of the input subset $\mathscr{U}$, $\mathscr{R}(\mathscr{U})$ measures the relevance between the subset and the target topic, $\mathscr{O}(\mathscr{U})$ encourages the subset to be as subjective as possible for the corresponding representation. The submodular function $f(\mathscr{U})$ should fit the definition, which is called *diminishing returns*, that for any set $\mathscr{J} \subseteq \mathscr{K} \subseteq \mathscr{V}$ and every $v \in \mathscr{V} \setminus \mathscr{K}$, we have

$$f(\mathscr{K} \cup \{v\}) - f(\mathscr{K}) \leqslant f(\mathscr{J} \cup \{v\}) - f(\mathscr{J}). \tag{6}$$

It means that the incremental value of function $f(\mathscr{U})$ caused by v decreases along with the context v expending from $\mathscr{J}$ to $\mathscr{K}$. To ensure that the $f(\mathscr{U})$ is submodular, we can use the theorem brought by [28]:

Theorem 1 (*Composition property*). *For any set functions* $\mathscr{P} : 2^V \rightarrow \mathrm{R}$ *and* $f : \mathrm{R} \rightarrow \mathrm{R}, \mathscr{F} := f \circ \mathscr{P}$ *is a non-decreasing submodular for every non-decreasing submodular* $\mathscr{P}$, *if and only if* f *is monotone concave.* $\circ$ *is a composition operator.*

If we have a set of non-decreasing submodular functions$\{f_i(x)\}^n$, then $F(x) = \sum_i^n \omega_i f_i(x)$ s.t. $\omega_i \in \Omega$ is still non-decreasing submodular, where Ω is a finite set of non-negative constants. In other words, the weighted linear combination of a series of submodular functions is still submodular. In the next few sections, we describe three different submodular score functions in detail. Their weighted sum $f(\mathscr{U})$ can keep the submodularity for further optimization by using the greedy algorithm [44,24]. In Section 3.5, we propose an opinion-sensitive greedy algorithm for constructing a subset for each candidate document set with $1 - 1/e$ lower bound.

3.4.1. Diversity function

The utility of the diversity function is to encourage the score function $f(\mathscr{U})$ to contain more documents with different word features. The following function can help us achieve the goal:

$$\mathscr{D}(\mathscr{U}) = \sum_{\alpha \in \mathscr{T}} \Psi(d), \tag{7}$$

where

$$\Psi(d) = \sqrt{\sum_{d \in \mathscr{U}, w \in \alpha \cap d} p_\alpha(w) F_d}, \tag{8}$$

where $p_\alpha(w)$ is the probability of word w in topic α and $F_d = tfidf_{w,d}$.

The proposed function (7) is a monotone submodular function that the proof is straightforward. Since the root-linear function $\Psi(x) = \sqrt{\sum_i^n x_i}$, s.t $x_i \geqslant 0$ is submodular [17], we can use the composition property mentioned in Theorem 1 that a non-negative weighted linear combination of submodular function is still submodular. Hence, the proposed diversity function $\mathscr{D}(\mathscr{U})$ is a monotone submodular function.

Since $\Psi(d)$ is a concave non-decreasing function, we have $\Psi(d_1 + d_2) \leqslant \Psi(d_1) + \Psi(d_2)$ for different documents d_1 and d_2. If $\mathscr{U} = \{d_1\}$ and α is fixed, we should select d_2 into $\mathscr{U}$ according to the value of $\Psi(d_1 + d_2)$. The more substantial the word feature differences between d_1 and d_2, the higher the value of $\Psi(d_1 + d_2)$. When the word features in d_1 and d_2 are completely different, then $\Psi(d_1 + d_2) = \Psi(d_1) + \Psi(d_2)$. In other words, this property enables the subset to extract the document with more diverse word features. That is to say, $f(\mathscr{U})$ can have high utility in its diversity.

3.4.2. Relevance function

This score function encourages the relevance between subset and candidate selection set by measuring the cosine similarity between them. It is defined as following

$$\mathscr{R}(\mathscr{U}) = \sum_{\tilde{d} \in V_c^y} \min\left\{\sum_{d \in \mathscr{U}} c\left(\tilde{d}, d\right), \frac{|\mathscr{U}|}{|V_c^y|} \sum_{d \in V_c^y} c\left(\tilde{d}, d\right)\right\}, \tag{9}$$

where $c\left(\tilde{d}, d\right)$ is the cosine similarity between two document vectors $F_{\tilde{d}}$ and F_d. $\sum_{d \in \mathscr{U}} c\left(\tilde{d}, d\right)$ measures the similarity between two document feature vectors, one is in the candidate set for one category V_c^y, and another is in the selected subset $\mathscr{U}$. $\frac{|\mathscr{U}|}{|V_c^y|} \sum_{d \in V_c^y} c\left(\tilde{d}, d\right)$ is the average document similarity in one particular category from the candidate document set V_c^y that using $|\mathscr{U}|$ documents can represent.

The diversity function is also a monotone submodular function. The $f(x) = \min(x, a)$, where $a \geqslant 0$ holds the concavity, and it is monotone non-decreasing. Hence, $f(x)$ holds the submodularity. The composition property in Theorem 1 still holds in this place. Since the weighted linear combination of the submodular function is still submodular, $\mathscr{R}(\mathscr{U})$ is a monotone submodular function.

It can be found that the former part in the *min* function can be regarded as a function of $\tilde{d}$ and the latter part is a constant, denoted as a. When $f\left(\tilde{d}\right) \geqslant a$, it means that the $\tilde{d}$ from V_c^y is too similar to the document instances already existed in subset $\mathscr{U}$. Adding $\tilde{d}$ to $\mathscr{U}$ will no longer provide a noticeable improvement in the relevance score.

3.4.3. Opinion function

Opinion function is a fine-grained opinion document selector. It facilitates the function to capture subtle but subjective documents for the selected subset.

$$\mathcal{O}(\mathcal{U}) = \frac{1}{|\mathcal{T}|}\sum_{\alpha\in\mathcal{T}} O(\alpha)\max_{d\in V_\alpha^y\cap\mathcal{U}} O(d,y), \tag{10}$$

where $|\mathcal{T}|$ is the total number of topics defined in Section 3.3. V_α^y is a set of documents in the candidate set V_c^y representing topic α. $O(d,y)$ is the opinion function for calculating the score in each document d. $O(\alpha)$ is for calculating the opinion score of each topic vector.

$O(d,y)$ is defined as

$$O(d,y) = \frac{1}{|d|}\sum_{w\in d} O_d(w,y), \tag{11}$$

where $|d|$ is the total number of words in document d. $O_d(w,y)$ is the opinion function to calculate subjective polarity of each word in a document d labeling with y. A pre-trained fine-grained opinion knowledge base introduced in Section 2.3 is adopted. It provides a subjective polarity of a word in four dimensions, including attention, aptitude, sensitivity, and pleasantness. Respecting these four fine-grained opinion types, we propose $O_d(w,y)$ as following

$$O_d(w,y) = \begin{cases} pla(w) + |att(w)| + apt(w), \\ \text{if} \quad y = POS \wedge pla(w) > 0 \wedge apt(w) > 0; \\ |pla(w)| + |sen(w)| + |apt(w)|, \\ \text{if} \quad y = NEG \wedge pla(w) < 0 \wedge apt(w) < 0; \\ |pla(w) + |att(w)| - |sen(w)| + apt(w)|, \\ \text{if} \quad y = NEU, \end{cases} \tag{12}$$

where the $pla(w)$ is the function for the pleasantness score, $att(w)$ is for attention score, $sen(w)$ is for sensitivity score, and $apt(w)$ is for aptitude score. The value of each of these functions is in the range $[-1,1]$ [8].

To obtain the opinion polarity in topics, we define $O(\alpha)$ as

$$O(\alpha) = \sum_{w\in\alpha} O_t(w), \tag{13}$$

where $O_t(w)$ is the function to get the opinion for each word token in topic α, defined as

$$O_t(w) = p_\alpha(w) * |inte(w)|, \tag{14}$$

where the definition of $p_\alpha(w)$ is the same as which in Section 3.4.1. $inte(w)$ is the opinion polarity intensity of a word token, where $inte(w) \in [-1,1]$ [8].

The proposed opinion function (10) is a monotone submodular function. Since $O_d(w,y) \geqslant 0 \Rightarrow O(d,y) \geqslant 0$ and $O_t(w) \geqslant 0 \Rightarrow O(\alpha) \geqslant 0$, then $\mathcal{O}(\mathcal{U})$ is a non-negative monotone submodular function. This function can also be considered as a fine-grained opinion form of the facility location function [18,22] in the following scenario: The topic (customer) tries to select a document (facility) from a candidate set (locations) for maximizing the opinion score of all documents (facility rating) in the selected set.

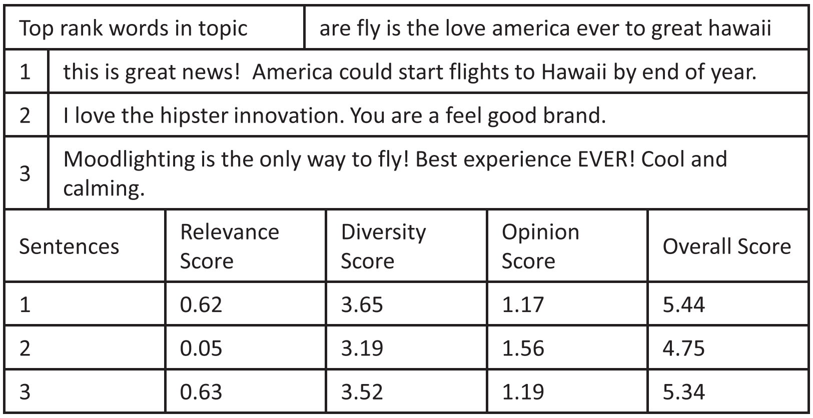
Fig. 3 depicts the score of the proposed diversity, relevance, and opinion function on document instance examples. Given the top rank words in the topic generated from these three documents, we can refer that document 1 and 3 achieve a higher score in the relevance. As for diversity, document 1 provides more information and description than document 3, but document 3 offers more opinions than document 1. However, document 2 presents a higher opinion score because of the highest-ranked topic word "love", a strong opinion word.

3.5. Algorithm for selection

In the previous section, we prove that the proposed objective function (3) is submodular. The following greedy selection Algorithm 2 can optimize this function and consider the opinions in documents simultaneously. In this algorithm, the greedy selection is performed on the input candidate set to optimize the proposed submodular objective function. At each iteration, the document that obtains the most significant objective function gain per opinion word is added to the selected subset. However, this process cannot transgress the limit B, which is directly related to the number of documents in the candidate set, the selection rate, and the candidate rate.

Compared with the general submodular optimization algorithm [35], we introduced a factor $|d_{ow}|$, where $d_{ow} = \{w \in d | inte(w) \in [-1,1]\}$ in line 7 Algorithm 2. This factor represents the number of existed opinion words in d concerning the pre-trained opinion knowledge base. Assuming two documents can contribute the same difference increment in line 7, the document with fewer opinion words can be selected. The opinion score increment contributed by each opinion word will be more significant in that document, and the diversity and relevance score. Hence the created document subset can be more sensitive to every opinion word.

Since the introduced factor $|d_{ow}|$ is not related to the limit B, the performance of the greedy subset selection procedure is still lower bounded by $1-\frac{1}{e}$ [35]. $\frac{f(\mathcal{U}^y\cup\{d\})-f(\mathcal{U}^y)}{|d_{ow}|}$ in line 7 Algorithm 2 can also be regarded as a non-negative constant $\frac{1}{|d_{ow}|}$ that



Top rank words in topic	are fly is the love america ever to great hawaii
1	this is great news! America could start flights to Hawaii by end of year.
2	I love the hipster innovation. You are a feel good brand.
3	Moodlighting is the only way to fly! Best experience EVER! Cool and calming.

Sentences	Relevance Score	Diversity Score	Opinion Score	Overall Score
1	0.62	3.65	1.17	5.44
2	0.05	3.19	1.56	4.75
3	0.63	3.52	1.19	5.34

Fig. 3. The Diversity, Relevance, and Opinion Scores of Examples in Twitter US Airline Sentiment Dataset.

multiplies the proposed submodular function $f(\mathscr{U})$ then calculates the difference. Hence the modified function is still non-negative monotone submodular. From this perspective, the modified function can be directly optimized by the general submodular optimization algorithm. We define the factor $|d_{ow}|$ in Algorithm 2 because it can be employed to optimize other submodular-based methods for subset selection shown in Section 4.1, though these methods are not targeted at opinion subset selection.

Algorithm 2. Algorithm for Opinion Subset Selection

```
Input: $SelectRate, CandiRate, V_c^y$
Output: $\mathcal{U}^y$
Data: Training set $V^y$ where $y \in [POS, NEG, NEU]$
assert $SelectRate < CandiRate$
$\mathcal{U}^y \leftarrow \emptyset$
$B \leftarrow \frac{SelectRate}{CandiRate} * |V_c^y|$
$C \leftarrow V_c^y$
$\Delta \leftarrow 0$
while $C \neq \emptyset$ do
    $d' \leftarrow \text{argmax}_{d \in C, |d_{ow}|>0} \frac{f(\mathcal{U}^y \cup \{d\}) - f(\mathcal{U}^y)}{|d_{ow}|}$
    if $|\mathcal{U}^y| < B$ and $\Delta \geq 0$ then
        $\mathcal{U}^y \leftarrow \mathcal{U}^y \cup \{d'\}$
    $\Delta \leftarrow f(\mathcal{U}^y \cup \{d'\}) - f(\mathcal{U}^y)$
    $C \leftarrow C \backslash \{d'\}$
```

The overall opinion subset selection is Algorithm 3. The original documents with the same label are firstly divided, then form the candidate document set by Algorithm 1. The greedy selection in Algorithm 2 creates the opinion subset within one category. Finally, the selected subsets from every category are combined to produce the output opinion subset. The time complexity of the candidate selection procedure is $O(n)$, and the greedy subset selection procedure is $O(kn)$ [23], where k is a cardinality constraint, and n the size of the input set. Hence the time complexity of Algorithm 3 is $O(kn)$.

Algorithm 3. Algorithm for Opinion Subset Selection

```
Input: $SelectRate$, $CandiRate$, $V_c^y$
Output: $\mathcal{U}$
Data: Training set $V$
$\mathcal{U} \leftarrow \emptyset$
for category $y \in [POS, NEG, NEU]$ do
    $V_c^y \leftarrow CandidateSetSelection(CandiRate, V^y)$      // Algorithm 1
    $\mathcal{U}^y \leftarrow GreedySubsetSelection(SelectRate, CandiRate, V_c^y)$
      // Algorithm 2
    $\mathcal{U} \leftarrow \mathcal{U} \cup \mathcal{U}^y$
```

4. Experiment

The objective of opinion subset selection is to serve the opinion analysis defined in Section 3.1. We validate the efficacy of the proposed opinion subset selection framework by investigating classifiers' and regressors' performance on the opinion subsets. These subsets are selected by different methods in a specified selection rate range. The results of the experiment for classification are shown in Tables 3–14 and Figs. 4–8, and the regression results are shown in Tables 15–19 and Figs. 9–12. The datasets, comparative methods, and the experiment setup are depicted in Section 4.1. The analysis and discussion of the results are in Section 4.2 and 4.3.

4.1. Evaluation setup

To evaluate the validity of our proposed opinion subset selection framework, we randomly choose at most 1500 instances from five different datasets, including, *Amazon Fine Food Reviews (AFFR)* [30], *Datafiniti Hotel Reviews (DHR)*,[1] *Hotel Reviews from Chennai (HRC)*,[2] *Restaurant Reviews in San Francisco (RRSF)*,[3] and *Twitter US Airline Sentiment (TAS)*.[4] The selected document instances are shuffled and then divided into training and testing sets with a 50% splitting rate. The statistical details are depicted in Table 2.

Six different baseline methods, namely Diversity + Relevance (D + R), Opinion (OP), Submodular Wan (SubWan) [47], Submodular Jayanth (SubJay) [18], Submodular Dimovski (SubDimovski) [12] and TextRank [5], are used for comparison. Five of them (D + R, OP, SubWan, SubDimovski, and SubJay) belong to the submodular methods family, and TextRank is a graph-based method. The details of each baseline are listed below:

Baseline 1 – Diversity + Relevance (D + R): This baseline method is constructed with only the first two parts of the score function, which are diversity function (7) and relevance function (9).

Baseline 2 – Opinion (OP): Within the same proposed framework, but it only employs the opinion function (10) as the objective submodular function.

Baseline 3 – Submodular Wan (SubWan): The score function of this method contains three different parts, which are relevance, coverage, and discrimination. Their function is submodular. Hence it could fit into the same framework we proposed. We remove the top 500 words limitation and replace $len(s)^\varepsilon$ with $|d_{ow}|$ in their algorithm to fit the optimization. Based on the explanation in their work, three parameters α for relevance, β for coverage, and γ for discrimination are empirically set to $0.05, 250$, and 300.

Baseline 4 – Submodular Jayanth (SubJay): This method uses the coverage function $L(s)$ in [25] and the A_4 facility location function in [18] to replace the proposed score function. According to their experiment setup, γ for $L(s)$ is 0.5, β for A_4 is $1 - \alpha$, and α for $L(s)$ is 0.3. This baseline can be adapted to the proposed subset selection framework because these two functions hold monotone submodularity.

[1] https://www.kaggle.com/datafiniti/hotel-reviews.

[2] https://www.kaggle.com/ranjitha1/hotel-reviews-city-chennai.

[3] https://www.kaggle.com/jkgatt/restaurant-data-with-100-trip-advisor-reviews-each.

[4] https://www.figure-eight.com/data-for-everyone.

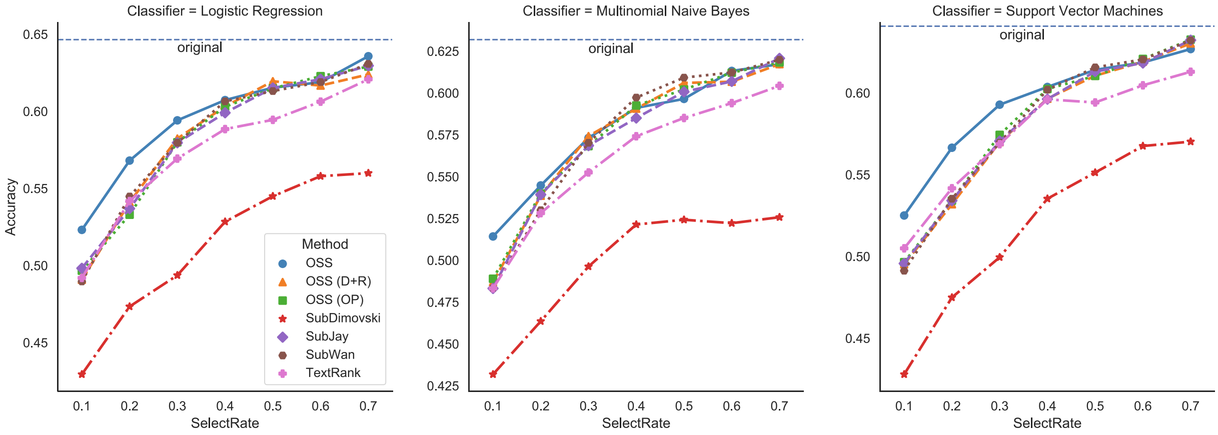


Fig. 4. The Accuracy (*mean*) versus Selection Rate of Three Classifiers on Five Benchmark Datasets.

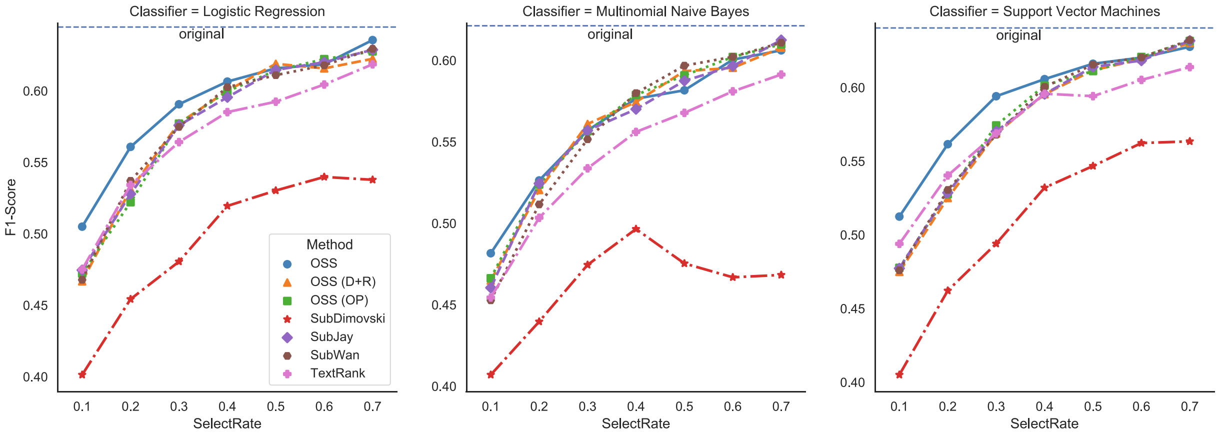


Fig. 5. The F1-Score (*mean*) versus Selection Rate of Three Classifiers on Five Benchmark Datasets.

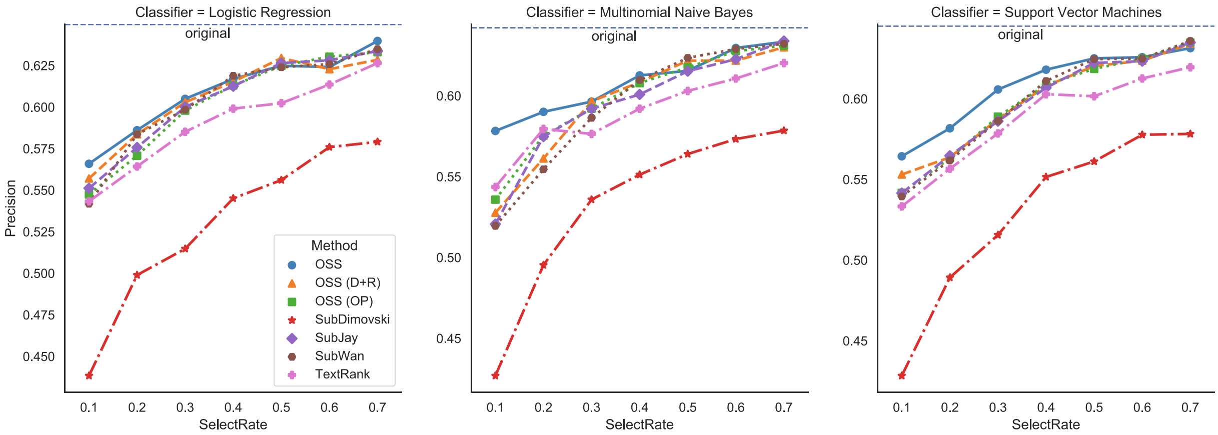


Fig. 6. The Precision (*mean*) versus Selection Rate of Three Classifiers on Five Benchmark Datasets.

Baseline 5 – Submodular Dimovski (SubDimovski): This method employs a score function $F(\cdot)$ called Ratio-Penalty Marginal Gain [12]. In the original work, they proposed a sentence similarity function using the sent2vec [37] as the feature input. We replace this function with the cosine similarity for balancing its performance. This baseline can be integrated into

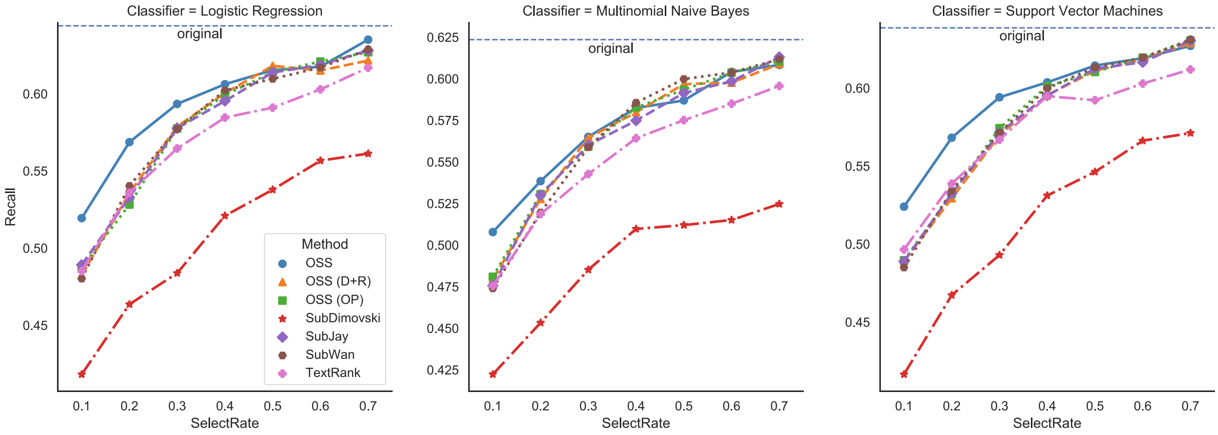


Fig. 7. The Recall (*mean*) versus Selection Rate of Three Classifiers on Five Benchmark Datasets.

the proposed subset selection framework because the $F(\cdot)$ is a submodular function. The introduced factor $|d_{ow}|$ is removed for algorithm comparison in this baseline.

Baseline 6 – TextRank: We apply an updated version of the TextRank algorithm [5] for subset selection under each category. Then the selected subsets are concatenated together to form a full subset. Rather than submodular baselines, this one is graph-based that each document instance is a node, and the similarity weight the edge between every two nodes. Compared with the original TextRank, which utilized cosine similarity as the weights, this version employed Okapi Best Match 25 (BM25) [19] as the new scheme for the edge.

In the evaluation, we heuristically set *CandiRate* for Algorithm 3 to 0.8, which can remove the outliers (mentioned in Section 3.3) and provide computation relief at the same time. For the parameters of LDA, the number of topics $|\mathscr{T}|$ is set to 5 under an empirical assumption that every 50 documents can almost support one topic. This assumption is based on the vocabulary size (VSize in Table 2) that most of the user expressions in the dataset are less than 50 words. The *SelectRate* varies from 0.0 to 0.7, with 0.1 as the stepsize. The golden labels in the datasets are star-ratings, ranging from 1 to 5, given by the online user. The regression evaluation can be directly performed on a $0.0 - 5.0$ continuous score range. For classification, we divide them into three categories, positive (4–5 stars), neutral (3 stars), and negative (1–2 stars).

4.2. Classification evaluation with subsets

We evaluate the performance of our proposed framework by using Scikit-learn [38]. The experiment contains three typical classifiers for opinion analysis [33], which are Logistic Regression (LR), Multinominal Naive Bayes (mNB), and Support Vector Machine (SVM). The standard metrics of opinion classification are employed to evaluate our proposed opinion subset selection framework. These standard metrics are accuracy, precision, recall, and F1-Score. *Accuracy* measures how many documents are correctly classified into their opinions for the whole document dataset. *Precision* measures the percentage of documents that are correctly classified out of all documents with the same assigned opinion label by the classifier. The fraction of corrected classified documents out of all documents labeled with that opinion by the user is the *Recall*. *F1-Score* is the weighted harmonic mean of recall and precision.

We use Tables 3–12 to illustrate every subset selection method's performance and robustness for classification on different datasets. These tables contain the comparative results on five corpora, namely AFFR, DHR, HRC, RRSF, and TAS. On the other hand, Figs. 4–7 are used to depict the performance of subset selection methods at different selection rates. Fig. 8 shows the average training time for the three classifiers on subsets selected at different rates. The comparative results of the metric standard deviation (STD) at *SelectRate* 0.6 are shown in Tables 13 and 14 to illustrate the stability of our proposed OSS method.

The average metric score differences achieved by three classifiers are listed in Tables 3–12, obtained by training on the subset created by each selection method at full *SelectRate* range. The average metric score differences of all classifiers are also listed as the key performance indicator in this evaluation. It is worth noting that a high value of the average metric score difference is preferable.

For a specific classifier on a corpus, metric score differences are calculated between the score obtained by training on the subset created at every selection rate and which achieved on the original data. Then the mean of these metric score differences is calculated for each classifier, i.e., $\overline{\text{AccDiff}}_{\text{SVM}}(OSS) = 1/|SR| * \sum_{SR}[Acc_{OSS}(SR) - Acc_{Origi}]$, where $\overline{\text{AccDiff}}_{\text{SVM}}(OSS)$ is the average accuracy difference of SVM training on OSS created subset, SR is short for *SelectRate*, $Acc_{OSS}(SR)$ represents the accuracy of SVM training on the OSS subset selected at SR, and Acc_{Origi} is the accuracy of SVM training on original data. The four average metric score (Accuracy, F1-Score, Precision, and Recall) differences are denoted as $\overline{\text{AccDiff}}, \overline{\text{F1Diff}}, \overline{\text{PreDiff}}$, and

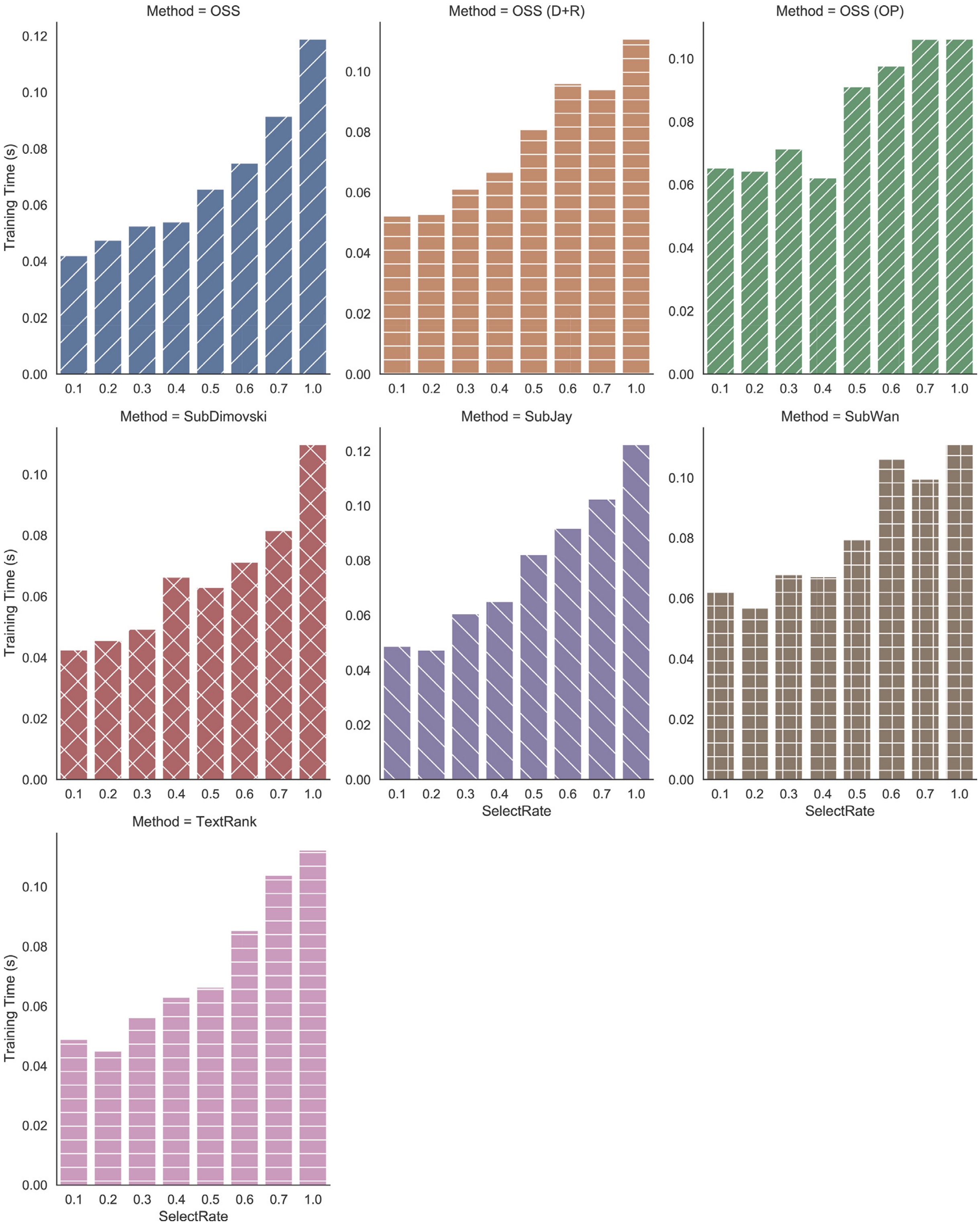


Fig. 8. The Average Training Time (*second*) Using Subsets versus Selection Rate of Three Classifiers on Five Benchmark Datasets.

$\overline{\text{RecDiff}}$. These four differences together reflect the overall performance of the subset selection method at full *SR* range and how it may vary on different classifiers. The metric mean of the average score difference of all classifiers is also defined for

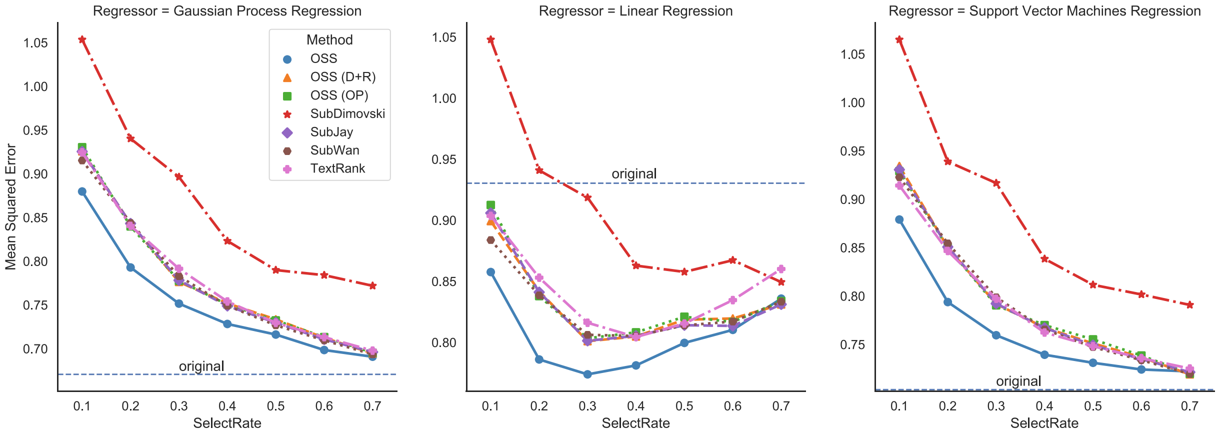


Fig. 9. The MSE (*mean*) versus Selection Rate of Three Regressors on Five Benchmark Datasets.

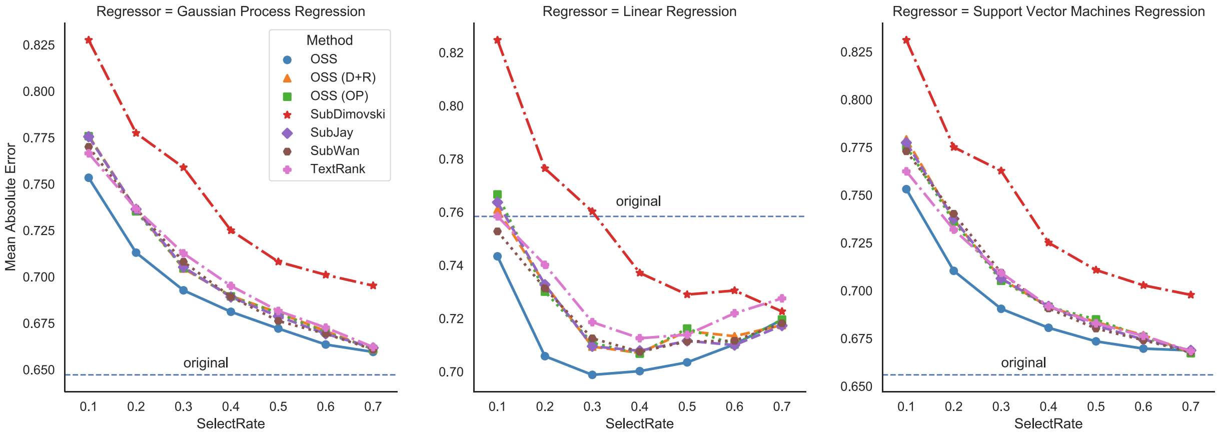


Fig. 10. The MAE (*mean*) versus Selection Rate of Three Regressors on Five Benchmark Datasets.

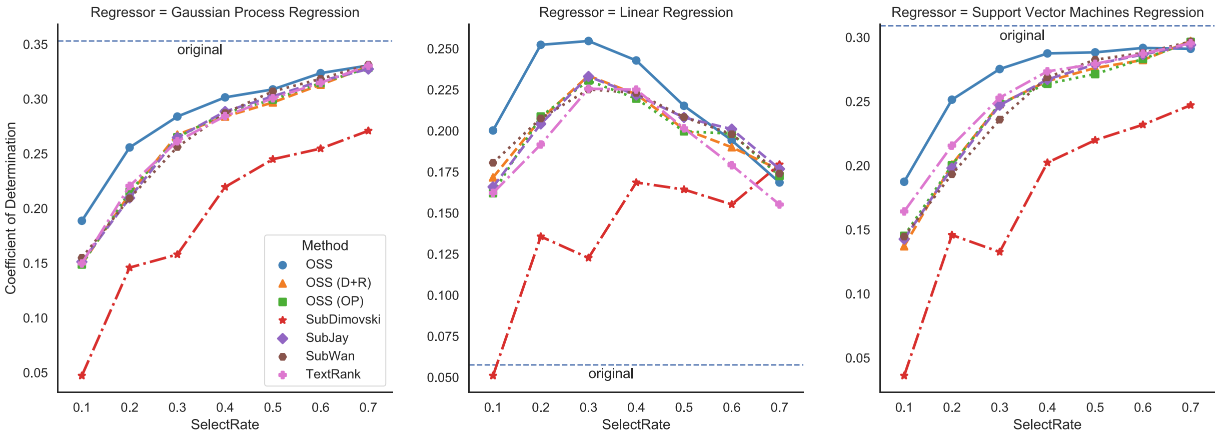


Fig. 11. The R^2 (*mean*) versus Selection Rate of Three Regressors on Five Benchmark Datasets.

evaluation, denoted as $\overline{\mathrm{AccDiff}}_c$, $\overline{\mathrm{F1Diff}}_c$, $\overline{\mathrm{PreDiff}}_c$, and $\overline{\mathrm{RecDiff}}_c$. Referring to the average score difference, the higher the metric mean of all classifiers, the better and more robust of the subset selection method.

In Tables 3, 4, 9 and 10, our proposed OSS method delivers the best mean performance for all classifiers on the AFFR and RRSF. For a corpus with long text lengths, different methods' performance can be affected by complex sentence structure and

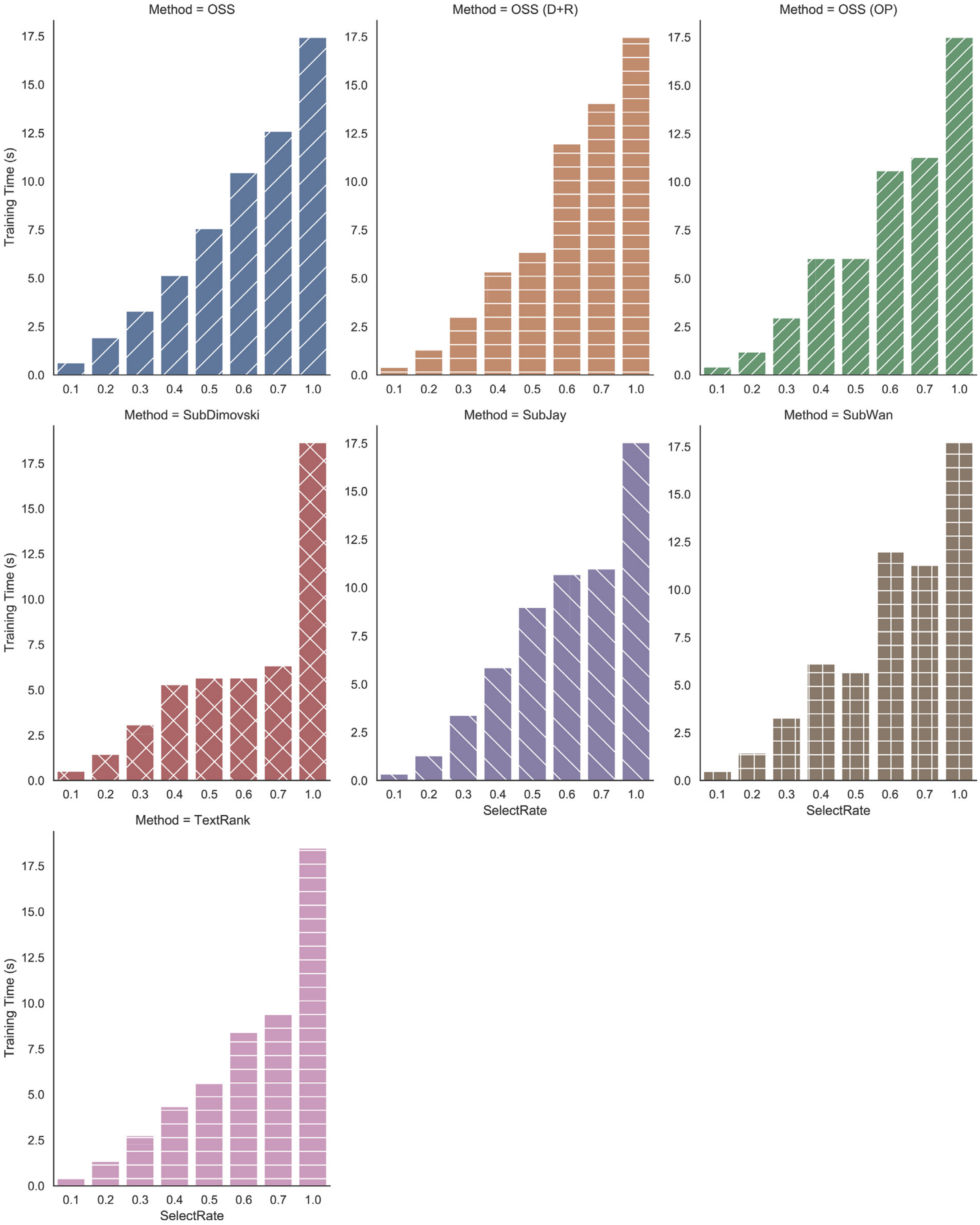


Fig. 12. The Average Training Time (*second*) Using Subsets versus Selection Rate of Three Regressors on Five Benchmark Datasets.

the repetition of opposite opinion words in the same sentence. This kind of semantic feature may lead to similar performance for different subset selection methods, such as on AFFR. In Tables 5 and 6, the OSS performs the best $\overline{\text{RecDiff}_c}$ on the DHR

Table 2
Opinion Dataset Specification.

Dataset	AvgLen	VSize	Training Set			Test Set	Total
			POS	NEG	NEU		
AFFR	47	3945	229	258	263	750	1500
DHR	40	3490	236	272	242	750	1500
HRC	24	1942	252	172	244	669	1337
RRSF	53	4171	236	256	258	750	1500
TAS	10	1473	259	240	251	750	1500

The average length (AvgLen) is the average number of words in document instances. The vocabulary size (VSize) is the total number of distinct word features containing in each dataset.

Table 3
Average Classification Accuracy and F1 Score Difference for All SelectRates on the Dataset AFFR.

Method	$\overline{\text{AccDiff}}$				$\overline{\text{F1Diff}}$			
	SVM	mNB	LR	$\overline{\text{AccDiff}_c}$	SVM	mNB	LR	$\overline{\text{F1Diff}}$
SubWan	−0.076	−0.082	−0.068	−0.075	−0.075	−0.082	−0.067	−0.075
SubJay	−0.074	−0.083	−0.073	−0.077	−0.073	−0.084	−0.072	−0.076
SubDimovski	−0.140	−0.189	−0.143	−0.157	−0.146	−0.227	−0.159	−0.177
TextRank	−0.079	−0.088	−0.074	−0.080	−0.077	−0.088	−0.072	−0.079
OSS (D + R)	−0.076	−0.078	−0.070	−0.075	−0.074	−0.079	−0.069	−0.074
OSS (OP)	**−0.071**	−0.082	**−0.068**	−0.074	**−0.070**	−0.082	**−0.067**	−0.073
OSS	*−0.074*	***−0.073***	*−0.073*	***−0.073***	*−0.072*	***−0.073***	*−0.072*	***−0.072***

Table 4
Average Classification Precision and Recall Difference for All SelectRates on the Dataset AFFR.

Method	$\overline{\text{PreDiff}}$				$\overline{\text{RecDiff}}$			
	SVM	mNB	LR	$\overline{\text{PreDiff}_c}$	SVM	mNB	LR	$\overline{\text{RecDiff}_c}$
SubWan	−0.069	−0.082	−0.058	−0.070	−0.076	−0.082	**−0.067**	−0.075
SubJay	−0.067	−0.083	−0.064	−0.071	−0.074	−0.083	−0.072	−0.076
SubDimovski	−0.129	−0.152	−0.125	−0.135	−0.138	−0.192	−0.143	−0.158
TextRank	−0.068	−0.084	−0.059	−0.070	−0.077	−0.087	−0.072	−0.079
OSS (D + R)	−0.068	−0.079	−0.061	−0.069	−0.075	−0.078	−0.069	−0.074
OSS (OP)	−0.065	−0.080	−0.061	−0.069	**−0.071**	−0.082	**−0.067**	−0.073
OSS	***−0.053***	***−0.067***	***−0.053***	***−0.058***	***−0.071***	***−0.072***	*−0.070*	***−0.071***

Table 5
Average Classification Accuracy and F1 Score Difference for All SelectRates on the Dataset DHR.

Method	$\overline{\text{AccDiff}}$				$\overline{\text{F1Diff}}$			
	SVM	mNB	LR	$\overline{\text{AccDiff}_c}$	SVM	mNB	LR	$\overline{\text{F1Diff}_c}$
SubWan	−0.045	−0.035	−0.048	−0.043	−0.047	−0.037	−0.046	−0.043
SubJay	−0.044	−0.033	−0.046	−0.041	−0.047	−0.034	−0.047	−0.043
SubDimovski	−0.129	−0.166	−0.155	−0.150	−0.131	−0.195	−0.160	−0.162
TextRank	−0.044	−0.073	−0.055	−0.057	−0.049	−0.086	−0.060	−0.065
OSS (D + R)	−0.050	−0.036	−0.051	−0.046	−0.053	−0.037	−0.050	−0.047
OSS (OP)	−0.036	**−0.029**	−0.045	**−0.037**	−0.040	**−0.029**	**−0.045**	**−0.038**
OSS	***−0.030***	*−0.051*	***−0.039***	*−0.040*	***−0.036***	*−0.058*	*−0.047*	*−0.047*

dataset. For $\overline{\text{AccDiff}_c}$, $\overline{\text{F1Diff}_c}$, and $\overline{\text{PreDiff}_c}$ on the DHR, the OSS (OP) is slightly better than the OSS. For the dataset with clear opinion expression, such as the DHR, the OSS (OP) can be sufficient enough for subset selection. However, the OSS can still create a subset with adequate opinions and semantic features on the recall focused task. In Tables 7, 8, 11 and 12, our proposed OSS method achieves the highest $\overline{\text{AccDiff}_c}$, $\overline{\text{F1Diff}_c}$, and $\overline{\text{RecDiff}_c}$ on the HRC and TAS datasets. As for $\overline{\text{PreDiff}_c}$, the OSS offers the second, which is only slightly lower than the TextRank in Table 8 and the SubWan in Table 12. From the results in these two tables, we can see that our OSS method can offer sufficient robustness and performance for short text opinion analysis compared with other subset selection methods. It exhibits that our proposed opinion, relevance, and diversity property are essential for this kind of analysis.

Table 6
Average Classification Precision and Recall Difference for All SelectRates on the Dataset DHR.

Method	$\overline{\text{PreDiff}}$				$\overline{\text{RecDiff}}$			
	SVM	mNB	LR	$\overline{\text{PreDiff}}_c$	SVM	mNB	LR	$\overline{\text{RecDiff}}_c$
SubWan	−0.034	−0.022	−0.033	−0.030	−0.048	−0.034	−0.051	−0.044
SubJay	−**0.031**	−0.024	−**0.031**	−0.029	−0.047	−0.032	−0.049	−0.043
SubDimovski	−0.115	−0.099	−0.116	−0.110	−0.127	−0.159	−0.150	−0.145
TextRank	−0.057	−0.051	−0.056	−0.055	−0.044	−0.070	−0.054	−0.056
OSS (D + R)	−0.037	−0.025	−0.034	−0.032	−0.053	−0.035	−0.054	−0.047
OSS (OP)	−**0.031**	−**0.017**	−0.033	−**0.027**	−0.040	−**0.028**	−0.048	−0.039
OSS	*−0.034*	*−0.030*	*−0.038*	*−0.034*	−***0.029***	*−0.048*	−***0.038***	−***0.038***

Table 7
Average Classification Accuracy and F1 Score Difference for All SelectRates on the Dataset HRC.

Method	$\overline{\text{AccDiff}}$				$\overline{\text{F1Diff}}$			
	SVM	mNB	LR	$\overline{\text{AccDiff}}_c$	SVM	mNB	LR	$\overline{\text{F1Diff}}_c$
SubWan	−0.067	−0.066	−0.097	−0.077	−0.064	−0.098	−0.108	−0.090
SubJay	−0.071	−0.072	−0.097	−0.080	−0.068	−0.092	−0.104	−0.088
SubDimovski	−0.081	−0.076	−0.107	−0.088	−0.102	−0.118	−0.149	−0.123
TextRank	−0.059	−**0.062**	−0.091	−0.071	−0.063	−0.094	−0.109	−0.089
OSS (D + R)	−0.070	−0.067	−0.092	−0.076	−0.066	−0.092	−0.100	−0.086
OSS (OP)	−0.082	−0.070	−0.099	−0.084	−0.079	−**0.091**	−0.108	−0.093
OSS	−***0.053***	*−0.072*	−***0.060***	−***0.062***	−***0.048***	*−0.104*	−***0.062***	−***0.071***

Table 8
Average Classification Precision and Recall Difference for All SelectRates on the Dataset HRC.

Method	$\overline{\text{PreDiff}}$				$\overline{\text{RecDiff}}$			
	SVM	mNB	LR	$\overline{\text{PreDiff}}_c$	SVM	mNB	LR	$\overline{\text{RecDiff}}_c$
SubWan	−0.047	−0.060	−0.064	−0.057	−0.062	−0.075	−0.103	−0.080
SubJay	−0.044	−0.044	−0.059	−0.049	−0.066	−0.076	−0.100	−0.081
SubDimovski	−0.085	−0.079	−0.110	−0.091	−0.098	−0.081	−0.133	−0.104
TextRank	−**0.026**	−**0.009**	−**0.040**	−**0.025**	−0.063	−**0.070**	−0.105	−0.079
OSS (D + R)	−0.041	−0.046	−0.054	−0.047	−0.064	−0.074	−0.096	−0.078
OSS (OP)	−0.053	−0.039	−0.058	−0.050	−0.077	−0.074	−0.102	−0.084
OSS	*−0.033*	*−0.019*	*−0.042*	*−0.031*	−***0.044***	*−0.076*	−***0.058***	−***0.059***

Table 9
Average Classification Accuracy and F1 Score Difference for All SelectRates on the Dataset RRSF.

Method	$\overline{\text{AccDiff}}$				$\overline{\text{F1Diff}}$			
	SVM	mNB	LR	$\overline{\text{AccDiff}}_c$	SVM	mNB	LR	$\overline{\text{F1Diff}}_c$
SubWan	−0.057	−0.074	−0.058	−0.063	−0.061	−0.066	−0.060	−0.062
SubJay	−0.060	−0.072	−0.059	−0.064	−0.065	−0.065	−0.061	−0.064
SubDimovski	−0.111	−0.133	−0.114	−0.119	−0.115	−0.156	−0.122	−0.131
TextRank	−0.079	−0.071	−0.075	−0.075	−0.078	−0.069	−0.073	−0.073
OSS (D + R)	−0.054	−0.069	−0.056	−0.060	−0.059	−0.062	−0.059	−0.060
OSS (OP)	−0.055	−0.071	−0.058	−0.061	−0.060	−0.064	−0.059	−0.061
OSS	−***0.051***	−***0.039***	−***0.054***	−***0.048***	−***0.055***	−***0.032***	−***0.056***	−***0.048***

The data in these tables confirm that the proposed OSS method can perform adequately on all five opinion corpora. It indicates that the OSS can offer a robust subset for opinion classification in different contexts and complex application environments where both the target selection rate and the classification method are unknown. On the other hand, the high impact of our proposed opinion factor in the algorithm can be explicitly confirmed when compared with the results of TextRank and SubDimovski in all tables.

The experimental results of every metric for the specified selection rate range are exhibited in Figs. 4–7. These figures consist of two different representations. One is the metric scores of a particular classifier (i.e., the accuracy of SVM) on five datasets at each selection rate. Another is the point and line, which displays the metric score changed along with the selec-

Table 10
Average Classification Precision and Recall Score Difference for All SelectRates on the Dataset RRSF.

Method	$\overline{\text{PreDiff}}$				$\overline{\text{RecDiff}}$			
	SVM	mNB	LR	$\overline{\text{PreDiff}}_c$	SVM	mNB	LR	$\overline{\text{RecDiff}}_c$
SubWan	−0.062	−0.050	−0.056	−0.056	−0.057	−0.072	−0.058	−0.062
SubJay	−0.065	−0.051	−0.058	−0.058	−0.060	−0.071	−0.059	−0.063
SubDimovski	−0.105	−0.115	−0.102	−0.107	−0.111	−0.133	−0.114	−0.119
TextRank	−0.073	−0.059	−0.069	−0.067	−0.079	−0.070	−0.075	−0.075
OSS (D + R)	−0.059	−0.048	−0.055	−0.054	−0.054	−0.068	−0.057	−0.060
OSS (OP)	−0.061	−0.050	−0.058	−0.056	−0.056	−0.070	−0.058	−0.061
OSS	***−0.044***	***−0.031***	***−0.046***	***−0.040***	***−0.051***	***−0.038***	***−0.054***	***−0.048***

Table 11
Average Classification Accuracy and F1 Score Difference for All SelectRates on the Dataset TAS.

Method	$\overline{\text{AccDiff}}$				$\overline{\text{F1Diff}}$			
	SVM	mNB	LR	$\overline{\text{AccDiff}}_c$	SVM	mNB	LR	$\overline{\text{F1Diff}}_c$
SubWan	−0.054	−0.031	−0.045	−0.043	−0.067	−0.034	−0.057	−0.053
SubJay	−0.054	−0.039	−0.044	−0.046	−0.066	−0.040	−0.056	−0.054
SubDimovski	−0.153	−0.107	−0.150	−0.137	−0.162	−0.106	−0.160	−0.143
TextRank	−0.070	−0.065	−0.070	−0.068	−0.070	−0.063	−0.070	−0.068
OSS (D + R)	−0.059	−0.038	−0.051	−0.049	−0.075	−0.038	−0.064	−0.059
OSS (OP)	−0.052	−0.035	−0.049	−0.045	−0.065	−0.036	−0.062	−0.054
OSS	***−0.033***	***−0.032***	***−0.032***	***−0.032***	***−0.035***	***−0.034***	***−0.034***	***−0.034***

Table 12
Average Classification Precision and Recall Score Difference for All SelectRates on the Dataset TAS.

Method	$\overline{\text{PreDiff}}$				$\overline{\text{RecDiff}}$			
	SVM	mNB	LR	$\overline{\text{PreDiff}}_c$	SVM	mNB	LR	$\overline{\text{RecDiff}}_c$
SubWan	**−0.024**	−0.029	−0.017	**−0.023**	−0.054	**−0.031**	−0.045	−0.043
SubJay	−0.032	−0.037	**−0.016**	−0.028	−0.054	−0.039	−0.043	−0.045
SubDimovski	−0.147	−0.105	−0.147	−0.133	−0.153	−0.108	−0.150	−0.137
TextRank	−0.069	−0.060	−0.071	−0.067	−0.070	−0.066	−0.070	−0.069
OSS (D + R)	−0.027	−0.035	−0.017	−0.026	−0.059	−0.037	−0.051	−0.049
OSS (OP)	−0.028	−0.033	−0.026	−0.029	−0.052	−0.035	−0.049	−0.045
OSS	***−0.024***	***−0.024***	*−0.026*	*−0.025*	***−0.033***	*−0.032*	***−0.032***	***−0.032***

Table 13
Accuracy and F1 Score Standard Deviation (STD) at *SelectRate* 0.6.

Method	Accuracy				F1-Score			
	SVM	mNB	LR	$\overline{\text{STD}}$	SVM	mNB	LR	$\overline{\text{STD}}$
SubWan	0.045	**0.041**	0.044	**0.043**	0.045	**0.048**	0.042	**0.045**
SubJay	0.045	0.044	0.043	0.044	0.045	0.053	**0.041**	0.046
SubDimovski	0.049	0.076	0.054	0.060	0.048	0.101	0.057	0.069
TextRank	0.043	0.052	**0.041**	0.045	0.042	0.055	0.038	**0.045**
OSS (D + R)	0.045	0.044	0.044	0.044	0.044	0.054	0.042	0.047
OSS (OP)	0.045	0.043	0.046	0.044	0.044	0.051	0.045	0.047
OSS	***0.040***	*0.051*	*0.043*	*0.045*	***0.040***	*0.059*	*0.042*	*0.047*

tion rate for each classifier. To illustrate the subset's utility, a blue dotted line at each figure shows the classifier's performance using the original data.

In Fig. 4, the performance of our proposed OSS method is better than that of others at the selection rate of 0.1–0.3 for LR and SVM. Except for the significantly lower performance of the SubDimovski, all other methods have similar performance for mNB. Specifically, at the selection rate of 0.6 shown in Table 13, all methods can support the accuracy of all classifiers quite stable (with lower STD) except the SubDimovski, which does not contain the proposed opinion factor for submodular optimization. The f1-score and recall in Figs. 5 and 7 follow a similar trend as accuracy. For mNB, the SubDimovski subset cannot offer acceptable performance at all ranges. From the results in Tables 13 and 14, all methods except the SubDimovski holds the performance stability at the full selection rate range for f1-score and recall. In Fig. 6, the OSS is better than other methods

Table 14
Precision and Recall Standard Deviation (STD) at *SelectRate* 0.6.

Method	Precision				Recall			
	SVM	mNB	LR	$\overline{\text{STD}}$	SVM	mNB	LR	$\overline{\text{STD}}$
SubWan	0.045	0.045	0.045	0.045	0.045	0.048	0.044	0.046
SubJay	0.045	0.044	0.044	0.044	0.045	0.053	0.042	0.047
SubDimovski	0.045	0.046	**0.037**	**0.043**	0.047	0.067	0.051	0.055
TextRank	0.046	0.059	0.044	0.050	0.041	0.053	**0.039**	**0.045**
OSS (D + R)	0.045	0.043	0.044	0.044	0.045	0.053	0.044	0.047
OSS (OP)	0.045	**0.041**	0.049	0.045	0.045	**0.051**	0.046	0.047
OSS	***0.039***	*0.049*	*0.040*	***0.043***	***0.040***	*0.057*	*0.042*	*0.046*

at the selection rate of 0.1–0.4 for SVM and 0.1–0.2 for mNB. The precision achieved on the SubDimovski subset is still significantly lower than other subsets. All other methods have similar performance for LR. Considering the information in Fig. 6 and Table 14, though the SubDimovski can produce a stable precision result, it can hardly provide adequate precision performance. On the other hand, the subset created by our proposed OSS method can support all classifiers to deliver stable and relatively high performance.

In Fig. 8, our proposed OSS method's training time is consistently dropped along with the selection rate. However, all other subset selection methods faced the issue that the training time bounced higher during the selection rate decrease. For instance, classifiers' average training time, using the SubWan subset, at a selection rate of 0.6, 0.3, and 0.1 is higher than 0.7, 0.4, and 0.2, respectively. This time bounce is due to the selected documents of SubWan containing more distinct words at a lower selection rate than the higher one. It can lead to a more sparse input feature matrix for the classifier, resulting in a relatively longer training time. On the other hand, benefited from the shrinkage of the corpora's size, almost all methods can offer a significant training time drop since the selection rate of 0.6 and halves the training time at the selection rate of 0.4. Considering the results in Figs. 4–7, we can conclude that the trade-off between the metric scores and the training time can still be well-balanced at the selection rate of 0.4, where all metric score drops less than 0.05 but offers only half of the training time, compared with using the original data.

This analysis validates the significance of employing semantic and sentiment information in the subset selection task for opinion classification, especially at a relatively low selection rate where the word features have a high level of peculiarity and a low level of redundancy.

The experimental results shown in Tables 3–14 and Figs. 4–8 indicate that the document subset can offer enhanced or equivalent performance for opinion classification with fewer instances. The subsets created by the OSS, which contain both semantic and opinion information, can support a classifier to provide more competitive and robust performance.

4.3. Regression evaluation with subsets

The regression experiment follows similar configurations as the classification. We evaluate the proposed framework's performance for three regressors commonly used for the opinion documents analysis [13,41,36], which are Gaussian Process Regression (GPP), Linear Regression (LrR), and Support Vector Machines Regression (SVM-R). Three typical metrics are adopted to the subset selection framework evaluation for regression [7,13,41], which are mean squared error (MSE), mean absolute error (MAE), and coefficient of determination (R^2 score). MSE measures the average squared difference between the predicted opinion and the user labeled one. On the other hand, MAE is a metric that calculates the average absolute error between these two scores. Both MSE and MAE indicates the error magnitude of the opinion predictions. The lower the score, the better is the regression model's performance. The best value of these two metrics is 0.0. Other than the previous two metrics, a higher R^2 score indicates the opinion regression model has a better capability to fit the input corpora. The best possible R^2 score is 1.0, which means the model can always predict a new document's opinion.

Tables 15–19 show different selection methods' robustness and performance for the three regressors on all datasets. These tables contain the corresponding experimental results on the same five corpora as classification. Figs. 9–11 illustrates the mean metric scores changing along with the selection rate for depicting the subset's validity for different selection methods. The comparative results of the three regressors' average training time are shown in Fig. 12 to show the efficacy by applying subset selection in opinion regression.

Three regressors' average metric score differences are listed in Tables 15–19, obtained by training on the subset generated by each selection method at all selection rates. The mean metric difference of all the regressors is listed as the key performance indicator in this assessment. It is worth noting that a low value of MSE difference ($\overline{\text{MSEDiff}}$) and MAE difference ($\overline{\text{MAEDiff}}$) and a high value of the R^2 score difference ($\overline{\text{R}^2\text{Diff}}$) is preferable.

Like the classifier, the regressor's metric difference is calculated following the same pattern, between the score obtained using the original data and each subset. Then the mean of these metric score differences is calculated for each regressor, i.e., $\overline{\text{MSEDiff}}_{\text{LrR}}(OSS) = 1/|SR| * \sum_{SR}\left[MSE_{OSS}(SR) - MSE_{Origi}\right]$, where $\overline{\text{MSEDiff}}_{\text{LrR}}(OSS)$ is the average MSE difference of the linear regressor training on OSS created subset, SR is short for *SelectRate*, $MSE_{OSS}(SR)$ represents the MSE of the linear regressor

Table 15
Average Regression Metric Score Difference for All SelectRates on the Dataset AFFR.

Method	$\overline{\text{MSEDiff}}$				$\overline{\text{MAEDiff}}$				$\overline{\text{R}^2\text{Diff}}$			
	GPR	LrR	SVM-R	$\overline{\text{MSEDiff}_c}$	GPR	LrR	SVM-R	$\overline{\text{MAEDiff}_c}$	GPR	LrR	SVM-R	$\overline{\text{R}^2\text{Diff}_c}$
SubWan	0.001	−0.231	−0.022	−0.084	−0.001	−0.094	−0.008	−0.034	−0.001	0.247	0.023	0.090
SubJay	0.002	−0.231	−0.022	−0.084	−0.001	−0.094	−0.007	−0.034	−0.002	0.248	0.023	0.090
SubDimovski	**−0.001**	**−0.264**	−0.036	**−0.100**	**−0.003**	**−0.109**	**−0.017**	**−0.043**	**0.001**	**0.282**	0.039	**0.107**
TextRank	0.013	−0.254	**−0.041**	−0.094	0.002	−0.105	−0.015	−0.039	−0.013	0.272	**0.044**	0.101
OSS (D + R)	0.002	−0.235	−0.025	−0.086	−0.001	−0.094	−0.008	−0.034	−0.002	0.251	0.027	0.092
OSS (OP)	0.001	−0.227	−0.018	−0.081	−0.001	−0.093	−0.007	−0.034	−0.001	0.243	0.019	0.087
OSS	*0.004*	*−0.237*	*−0.040*	*−0.091*	*0.000*	*−0.093*	*−0.013*	*−0.035*	*−0.004*	*0.253*	*0.043*	*0.097*

Table 16
Average Regression Metric Score Difference for All SelectRates on the Dataset HRC.

Method	$\overline{\text{MSEDiff}}$				$\overline{\text{MAEDiff}}$				$\overline{\text{R}^2\text{Diff}}$			
	GPR	LrR	SVM-R	$\overline{\text{MSEDiff}_c}$	GPR	LrR	SVM-R	$\overline{\text{MAEDiff}_c}$	GPR	LrR	SVM-R	$\overline{\text{R}^2\text{Diff}_c}$
SubWan	0.082	−0.042	0.055	0.032	0.058	−0.005	0.050	0.034	−0.135	0.069	−0.090	−0.052
SubJay	0.077	−0.046	0.050	0.027	0.056	−0.007	0.048	0.032	−0.126	0.076	−0.082	−0.044
SubDimovski	0.094	−0.044	0.053	0.034	0.063	−0.008	0.049	0.035	−0.154	0.073	−0.088	−0.056
TextRank	**0.047**	**−0.070**	**0.016**	**−0.002**	**0.039**	−0.024	**0.026**	**0.014**	**−0.078**	**0.116**	**−0.027**	**0.004**
OSS (D + R)	0.078	−0.044	0.051	0.028	0.056	−0.005	0.048	0.033	−0.128	0.072	−0.083	−0.046
OSS (OP)	0.078	−0.042	0.050	0.029	0.057	−0.005	0.048	0.033	−0.128	0.070	−0.083	−0.047
OSS	*0.060*	*−0.069*	*0.034*	*0.008*	*0.042*	***−0.027***	*0.035*	*0.017*	*−0.099*	*0.114*	*−0.056*	*−0.014*

Table 17
Average Regression Metric Score Difference for All SelectRates on the Dataset DHR.

Method	$\overline{\text{MSEDiff}}$				$\overline{\text{MAEDiff}}$				$\overline{\text{R}^2\text{Diff}}$			
	GPR	LrR	SVM-R	$\overline{\text{MSEDiff}_c}$	GPR	LrR	SVM-R	$\overline{\text{MAEDiff}_c}$	GPR	LrR	SVM-R	$\overline{\text{R}^2\text{Diff}_c}$
SubWan	0.161	−0.062	0.170	0.090	0.080	−0.020	0.081	0.047	−0.094	0.036	−0.099	−0.052
SubJay	0.169	−0.052	0.175	0.097	0.083	−0.016	0.084	0.050	−0.098	0.031	−0.102	−0.056
SubDimovski	0.394	0.137	0.397	0.309	0.171	0.054	0.168	0.131	−0.230	−0.080	−0.231	−0.180
TextRank	0.177	−0.024	0.196	0.116	0.090	−0.002	0.094	0.061	−0.103	0.014	−0.114	−0.068
OSS (D + R)	0.169	−0.051	0.176	0.098	0.084	−0.015	0.084	0.051	−0.099	0.030	−0.103	−0.057
OSS (OP)	0.172	−0.049	0.177	0.100	0.084	−0.016	0.084	0.051	−0.100	0.028	−0.103	−0.058
OSS	***0.102***	***−0.127***	***0.120***	***0.032***	***0.058***	***−0.043***	***0.062***	***0.026***	***−0.059***	***0.074***	***−0.070***	***−0.018***

Table 18
Average Regression Metric Score Difference for All SelectRates on the Dataset RRSF.

Method	$\overline{\text{MSEDiff}}$				$\overline{\text{MAEDiff}}$				$\overline{\text{R}^2\text{Diff}}$			
	GPR	LrR	SVM-R	$\overline{\text{MSEDiff}_c}$	GPR	LrR	SVM-R	$\overline{\text{MAEDiff}_c}$	GPR	LrR	SVM-R	$\overline{\text{R}^2\text{Diff}_c}$
SubWan	0.234	0.116	0.225	0.192	0.105	0.054	0.101	0.087	−0.142	−0.070	−0.136	−0.116
SubJay	0.236	0.118	0.227	0.194	0.106	0.055	0.102	0.088	−0.143	−0.071	−0.138	−0.117
SubDimovski	0.347	0.232	0.350	0.310	0.153	0.102	0.153	0.136	−0.210	−0.141	−0.212	−0.188
TextRank	0.242	0.130	0.232	0.201	0.106	0.057	0.102	0.088	−0.146	−0.079	−0.141	−0.122
OSS (D + R)	0.236	0.116	0.227	0.193	0.105	0.054	0.102	0.087	−0.143	−0.070	−0.138	−0.117
OSS (OP)	0.240	0.122	0.227	0.196	0.107	0.056	0.101	0.088	−0.145	−0.074	−0.138	−0.119
OSS	***0.198***	***0.085***	***0.180***	***0.154***	***0.086***	***0.038***	***0.077***	***0.067***	***−0.120***	***−0.052***	***−0.109***	***−0.094***

training on the OSS subset selected at *SR*, and MSE_{Origi} is the MSE of the linear regressor training on original data. Together, these three differences reflect the subset selection method's overall and specific performance for different regressors. The metric mean of the average score difference of regressors is defined for the comprehensive assessment, denoted as $\overline{\text{MSEDiff}_c}$, $\overline{\text{MAEDiff}_c}$, and $\overline{\text{R}^2\text{Diff}_c}$. By referring to the average score difference, for MSE and MAE, the lower the metric mean of all regressors, the better and more robust the subset selection method. An opposite score trend should be expected for the R^2 score comparing with the previous two metrics.

In Table 15, the SubDimvoski achieves the best performance on AFFR. However, other subset selection methods can offer similar performance with only a small score compromise, e.g., TextRank and OSS. On the other hand, the SubDimvoski sub-

Table 19
Average Regression Metric Score Difference for All SelectRates on the Dataset TAS.

Method	$\overline{\text{MSEDiff}}$				$\overline{\text{MAEDiff}}$				$\overline{\text{R}^2\text{Diff}}$			
	GPR	LrR	SVM-R	$\overline{\text{MSEDiff}}_c$	GPR	LrR	SVM-R	$\overline{\text{MAEDiff}}_c$	GPR	LrR	SVM-R	$\overline{\text{R}^2\text{Diff}}_c$
SubWan	0.041	−**0.291**	0.015	−**0.078**	**0.030**	−**0.123**	0.021	−**0.024**	−0.062	**0.442**	−0.022	**0.119**
SubJay	0.044	−0.288	0.015	−0.076	0.032	−0.121	0.021	−0.023	−0.067	0.437	−0.022	0.116
SubDimovski	0.141	−0.181	0.122	0.027	0.090	−0.058	0.084	0.039	−0.214	0.276	−0.185	−0.041
TextRank	0.062	−0.228	0.029	−0.046	0.046	−0.080	0.030	−0.001	−0.094	0.347	−0.044	0.070
OSS (D + R)	0.046	−0.283	0.019	−0.073	0.033	−0.120	0.024	−0.021	−0.070	0.430	−0.028	0.111
OSS (OP)	0.041	−0.289	0.013	−**0.078**	**0.030**	−0.121	**0.020**	−**0.024**	−0.062	0.439	−0.020	**0.119**
OSS	***0.039***	*−0.273*	***0.010***	*−0.075*	*0.031*	*−0.109*	***0.020***	*−0.019*	*−**0.060***	*0.414*	*−**0.015***	*0.113*

set's capability can hardly be maintained on an equivalent level with others given the metric comparison in Figs. 9–11. Considering all the mentioned outcomes, Table 15 reflects that the AFFR cannot give a distinguishable regression experimental result. On the other hand, the metric scores in Tables 17 and 18 on DHR and RRSF, which have similar average length features as AFFR, indicates that the OSS has a significantly better performance on the lengthy opinion text analysis. This result shows the potential of combining the semantic and opinion features for creating the subset for opinion regression. The HRC dataset in Table 16 shows that the TextRank's subset can support the classifier with the best metric result. Our OSS method offers slightly lower performance but at the same level. The results indicate that the graph-based subset selection can show their capability to discover the latent semantic linkage between different documents on medium length corpora like HRC. However, when considering the results demonstrated in Figs. 9–11, TextRank cannot maintain its stability on all datasets. Hence, our OSS method can offer a more robust and effective subset compared with TextRank. When on a concise corpus like TAS in Table 19, the OSS (OP) and SubWan can achieve a comparatively better score. However, the LrR contributes the most to this improvement if checked precisely. When limited to the GPR and SVM-R only, the proposed OSS still holds the best performance. In Fig. 12, SubWan additionally faces the issue that the training time bounced back along with the selection rate decrease, e.g., at the selection rate of 0.4 and 0.6. It is also worth noting that the opinion features in OSS (OP) can offer a similar average performance level without the semantic features in SubWan, which indicates the value of considering the opinion features for subset selection in opinion analysis.

These tables confirm that the proposed OSS method can perform appropriately on most opinion corpora. It indicates that the OSS can offer robust subsets for opinion regression on both short and long context opinion corpora, even the target selection rate and the regressor are hidden. Moreover, the experiment confirmed the impact of our proposed diversity, relevance, and opinion feature for opinion regression in the algorithm.

The experimental results of regression metrics for the selection rate range of 0.1–0.7 are exhibited in Figs. 9–11. Similar to the classification figures, the regression ones depict each regressor's metric scores varied with the selection rate. A blue dotted line displays the average score obtained by the regressor on the original data.

In Fig. 9, the mean MSE of our proposed OSS is significantly better than others at almost all three regressors' selection rates. Except for the SubDimovski, all other methods have similar scores for these regressors. The mean MAE score in Fig. 10 displays a similar trend as MSE in Fig. 9. The OSS is significantly better. All other methods, except SubDimovski, can obtain scores close to each other. For MSE and MAE, our proposed method can achieve similar performance as other baselines at the selection rate of 0.6 and 0.7 for linear regressor. For the R^2 score in Fig. 11, OSS can achieve better result in selecting rates of $0.1 - 0.4$. On the other hand, all methods except SubDimovski can offer a comparatively similar outcome between the selection rate range of $0.5 - 0.7$. The OSS displays consistent stability in our proposed framework for all regressors.

Figs. 9–11 illustrate a clear LR performance improvement utilizing the subsets. Almost all subsets can provide a better metric score than using the original data for LR, even at a selection rate of 0.1 where the size of the input dataset reduces 90%. The metric score of GPR and SVM-R increases with the selection rate, but the LR achieves the best at 0.4.

Fig. 12 shows that regressors' average training time can significantly drop at the selection rate of 0.5. Specifically, when at the selection rate of 0.4 where LR can achieve the best performance, the training time is roughly one-third of the original. Considering the performance trend in the figures, OSS can significantly improve the regressor's training efficiency and efficacy since the selection rate of 0.4.

This figure analysis validates the importance of involving relevance, diversity, and opinion features in the subset selection for opinion regression. The subset can halve the training time with only a small metric compromise. For regressor like LR, the subset can provide performance enhancement on accuracy and efficiency at the same time.

The experimental results in Tables 15–19 and Figs. 9–12 reveal the validity and efficacy of using document subset in opinion regression. Overall, the OSS subset can support different regressors to deliver a robust and well-balanced performance improvement between training time and metrics.

5. Conclusion

In this paper, the concept of opinion subset selection, together with a framework and an opinion-sensitive algorithm, is proposed. The opinions classification and regression tasks are performed, and their results are evaluated and analyzed. Based on the result's analysis, our proposed framework can offer an opinion-sensitive solution for addressing the low-performance issue existing in conventional subset selection for opinion analysis.

In our framework, the candidate document selection procedure can first eliminate the opinion mislabeled documents and relieve the computational burden for the following process. After that, the proposed score function can capture both the opinion and semantic features for subset selection. The score function consists of relevance, diversity, and fine-grained subjective opinion as fundamental properties in a submodular manner. Additionally, we introduced an opinion-sensitive factor in the algorithm for optimizing this submodular function. Experimental results from classifiers and regressors on five benchmark datasets reveal that the proposed framework can select a reliable opinion document subset. This subset can offer adequate efficiency and robust performance for different classifiers and regressors in the opinion analysis task.

In the future, the framework validity could be empirically evaluated in a dialogue system. The other tasks in natural language processing could also benefit from our work. On the other hand, this work could assist a mobile or embedded system with limited hardware conditions in performing offline opinion analysis.

CRediT authorship contribution statement

Yang Zhao: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Writing - review & editing, Visualization, Investigation, Validation, Formal analysis. **Tommy W.S. Chow:** Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the City University of Hong Kong under Project 9042996.

References

[1] S. Angelidis, M. Lapata, Multiple instance learning networks for fine-grained sentiment analysis, Transactions of the Association for Computational Linguistics 6 (2018) 17–31.

[2] M.E. Aragón, A.P. López-Monroy, L.C. González-Gurrola, M. Montes-y Gómez, Detecting depression in social media using fine-grained emotions, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1481–1486, https://doi.org/10.18653/v1/N19-1151, URL: https://www.aclweb.org/anthology/N19-1151.

[3] O. Bachem, M. Lucic, A. Krause, Scalable k -means clustering via lightweight coresets, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining, KDD '18, ACM, New York, NY, USA, 2018, pp. 1119–1127, https://doi.org/10.1145/3219819.3219973.

[4] G. Balikas, S. Moura, M.-R. Amini, Multitask learning for fine-grained twitter sentiment analysis, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1005–1008, https://doi.org/10.1145/3077136.3080702.

[5] F. Barrios, F. López, L. Argerich, R. Wachenchauzer, Variations of the Similarity Function of TextRank for Automated Summarization, arXiv e-prints (2016) arXiv:1602.03606..

[6] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[7] Y.E. Cakra, B. Distiawan Trisedya, Stock price prediction using linear regression based on sentiment analysis, in: 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2015, pp. 147–154, https://doi.org/10.1109/ICACSIS.2015.7415179.

[8] E. Cambria, S. Poria, D. Hazarika, K. Kwok, Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018..

[9] T. Campbell, T. Broderick, Automated scalable bayesian inference via hilbert coresets, Journal of Machine Learning Research 20 (2019) 1–38.

[10] D. Cavaliere, S. Senatore, V. Loia, Context-aware profiling of concepts from a semantic topological space, Knowledge-Based Systems 130 (2017) 102–115.

[11] T.W.S. Chow, D. Huang, Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information, IEEE Transactions on Neural Networks 16 (2005) 213–224.

[12] M. Dimovski, C. Musat, V. Ilievski, A. Hossman, M. Baeriswyl, Submodularity-inspired data selection for goal-oriented chatbot training based on sentence embeddings, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 4019–4025. https://doi.org/10.24963/ijcai.2018/559. 10.24963/ijcai.2018/559..

[13] A. Drake, E. Ringger, D. Ventura, Sentiment regression: Using real-valued scores to summarize overall document sentiment, in: 2008 IEEE International Conference on Semantic Computing, 2008, pp. 152–157.

[14] A. Dubey, M. Chatterjee, N. Ahuja, Coreset-based neural network compression, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 469–486.

[15] D. Feldman, M. Volkov, D. Rus, Dimensionality reduction of massive sparse datasets using coresets, in: D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29, Curran Associates Inc, 2016, pp. 2766–2774, URL: http://papers.nips.cc/paper/6596-dimensionality-reduction-of-massive-sparse-datasets-using-coresets.pdf.

[16] Y. Fu, X. Zhu, A.K. Elmagarmid, Active learning with optimal instance subset selection, IEEE Transactions on Cybernetics 43 (2013) 464–475.

[17] M.X. Goemans, N.J.A. Harvey, S. Iwata, V. Mirrokni, Approximating submodular functions everywhere, in: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '09, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009, pp. 535–544. http://dl.acm.org/citation.cfm?id=1496770.1496829..

[18] J. Jayanth, J. Sundararaj, P. Bhattacharyya, Monotone submodularity in opinion summaries, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, pp. 169–178. http://aclweb.org/anthology/D15-1017. 10.18653/v1/D15-1017..

[19] K.S. Jones, S. Walker, S.E. Robertson, A probabilistic model of information retrieval: development and comparative experiments, Information Processing and Management 36 (2000) 779–808.

[20] K. Kirchhoff, J. Bilmes, Submodularity for data selection in machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 131–141.

[21] J. Kittler, Feature selection and extraction., Handbook of Pattern Recognition and Image Processing (1986) 59–83..

[22] A. Krause, D. Golovin, Submodular Function Maximization, Cambridge University Press, 2014, pp. 71–104. 10.1017/CBO9781139177801.004..

[23] A. Kuhnle, Interlaced greedy algorithm for maximization of submodular functions in nearly linear time, in: H. Wallach, H. Larochelle, A. Beygelzimer, F.D. Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates Inc, 2019, pp. 2374–2384.

[24] H. Lin, J. Bilmes, Multi-document summarization via budgeted maximization of submodular functions, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Los Angeles, California, 2010, pp. 912–920, URL: https://www.aclweb.org/anthology/N10-1134.

[25] H. Lin, J. Bilmes, A Class of Submodular Functions for Document Summarization, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 510–520. http://www.aclweb.org/anthology/P11-1052..

[26] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2015. 10.1017/CBO9781139084789..

[27] P. Liu, S. Joty, H. Meng, Fine-grained opinion mining with recurrent neural networks and word embeddings, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 1433–1443, https://doi.org/10.18653/v1/D15-1168, URL: https://www.aclweb.org/anthology/D15-1168.

[28] L. Lovász, Submodular Functions and Convexity, Springer Berlin Heidelberg, Berlin, Heidelberg, 1983, pp. 235–257. https://doi.org/10.1007/978-3-642-68874-4_10..

[29] S. Mahabadi, P. Indyk, S.O. Gharan, A. Rezaei, Composable core-sets for determinant maximization: a simple near-optimal algorithm, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, PMLR, Long Beach, California, USA, 2019, pp. 4254–4263.

[30] J.J. McAuley, J. Leskovec, From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews, in: Proceedings of the 22Nd International Conference on World Wide Web, WWW '13, New York, NY, USA, 2013, pp. 897–908, https://doi.org/10.1145/2488388.2488466.

[31] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, Fast constrained submodular, maximization: personalized data summarization, in: M.F. Balcan, K.Q. Weinberger (Eds.), Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, PMLR, New York, New York, USA, 2016, pp. 1358–1367, URL: http://proceedings.mlr.press/v48/mirzasoleiman16.html.

[32] H. Morita, R. Sasano, H. Takamura, M. Okumura, Subtree extractive summarization via submodular maximization, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2013, pp. 1023–1032. http://aclweb.org/anthology/P13-1101..

[33] M.E. Moussa, E.H. Mohamed, M.H. Haggag, A survey on opinion summarization techniques for social media, Future Computing and Informatics Journal 3 (2018) 82–109.

[34] T. Nasukawa, J. Yi, Sentiment analysis: capturing favorability using natural language processing, in: Proceedings of the 2Nd International Conference on Knowledge Capture, ACM, New York, NY, USA, 2003, pp. 70–77, https://doi.org/10.1145/945645.945658.

[35] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions—I, Mathematical Programming 14 (1978) 265–294.

[36] I. Onal, A.M. Ertugrul, R. Cakici, Effect of using regression on class confidence scores in sentiment analysis of Twitter data, in: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 136–141, https://doi.org/10.3115/v1/W14-2622, URL: https://www.aclweb.org/anthology/W14-2622.

[37] M. Pagliardini, P. Gupta, M. Jaggi, Unsupervised learning of sentence embeddings using compositional n-gram features, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 528–540, https://doi.org/10.18653/v1/N18-1049, URL: https://www.aclweb.org/anthology/N18-1049.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[39] S. Poria, E. Cambria, G. Winterstein, G.-B. Huang, Sentic patterns: Dependency-based rules for concept-level sentiment analysis, Knowledge-Based Systems 69 (2014) 45–63.

[40] O. Rouane, H. Belhadef, M. Bouakkaz, Combine clustering and frequent itemsets mining to enhance biomedical text summarization, Expert Systems with Applications 135 (2019) 362–373.

[41] A.D. S, S.M. Rajendram, T.T. Mirnalinee, SSN_MLRG1 at SemEval-2017 task 5: Fine-grained sentiment analysis using multiple kernel Gaussian process regression model, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 823–826. https://www.aclweb.org/anthology/S17-2139. 10.18653/v1/S17-2139..

[42] Y. Shinohara, A submodular optimization approach to sentence set selection, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 4112–4115, https://doi.org/10.1109/ICASSP.2014.6854375.

[43] Q. Song, J. Ni, G. Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, IEEE Transactions on Knowledge and Data Engineering 25 (2013) 1–14.

[44] M. Sviridenko, A note on maximizing a submodular set function subject to a knapsack constraint, Operations Research Letters 32 (2004) 41–43.

[45] F. Tang, L. Fu, B. Yao, W. Xu, Aspect based fine-grained sentiment analysis for online reviews, Information Sciences 488 (2019) 190–204.

[46] H. Van Lierde, T.W.S. Chow, Learning with fuzzy hypergraphs: A topical approach to query-oriented text summarization, Information Sciences 496 (2019) 212–224.

[47] X. Wan, T. Wang, Automatic Labeling of Topic Models Using Text Summaries, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 2297–2305. http://www.aclweb.org/anthology/P16-1217..

[48] K. Wei, R. Iyer, J. Bilmes, Submodularity in data subset selection and active learning, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research PMLR, Lille, France, 2015, pp. 1954–1963.

[49] J. Wiebe, Learning subjective adjectives from corpora, in: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, AAAI Press, 2000, pp. 735–740, URL: http://dl.acm.org/citation.cfm?id=647288.721121.

[50] Y. Zheng, J.M. Phillips, Coresets for kernel regression, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, ACM, New York, NY, USA, 2017, pp. 645–654, https://doi.org/10.1145/3097983.3098000.