FEATURE SELECTION AND CLASSIFICATION INTEGRATED METHOD FOR IDENTIFYING CITED TEXT SPANS FOR CITANCES ON IMBALANCED DATA

Jen-Yuan Yeh[1*], Cheng-Jung Tsai[2], Tien-Yu Hsu[3], Jung-Yi Lin[4], Pei-Cheng Cheng[5]

[1]Dept. of Operation, Visitor Service, Collection and Information Management,
National Museum of Natural Science, Taichung 40453, Taiwan

[2]Graduate Institute of Statistics and Information Science,
National Changhua University of Education, Changhua 50007, Taiwan

[3]Dept. of Science Education,
National Museum of Natural Science, Taichung 40453, Taiwan

[4]IP Affairs Division,
Hon Hai Precision Ind.Co.,Ltd (Foxconn), Taipei 11492, Taiwan

[5]Dept. of Information Management,
Chien Hsin University of Science and Technology, Taoyuan 32097, Taiwan

Email: jenyuan@nmns.edu.tw[1*] (corresponding author), cjtsai@cc.ncue.edu.tw[2], dan@nmns.edu.tw[3],
jungyilin@gmail.com[4], pccheng@uch.edu.tw[5]

DOI: https://doi.org/10.22452/mjcs.vol34no4.3

ABSTRACT

Recent studies in scientific paper summarization have explored a new form of structured summary for a reference paper by grouping all cited and citing sentences together by facet. This involves three main tasks: (1) identifying cited text spans for citances (i.e., citing sentences), (2) classifying their discourse facets, and (3) generating a structured summary from the cited text spans and citances. This paper focuses on the first task, and approaches the task as binary classification to distinguish relevant pairs of citances and reference sentences from irrelevant pairs. We propose a new method that integrates feature selection and classification techniques to enhance classification performance. The proposed method investigates combinations of six feature selection methods (χ^2-Statistics, Information Gain, Gain Ratio, Relief-F, Significance Attribute Evaluation, and Symmetrical Uncertainty), and five classification algorithms (k-Nearest Neighbors, Decision Tree, Support Vector Machine, Naïve Bayes, and Random Forest). Additionally, to address imbalanced data during training, we apply SMOTE (Synthetic Minority Over-sampling Technique) to introduce synthetic biases towards the minority. Experiments are conducted using the CL-SciSumm corpora to compare the effect of feature selection applied to classification. The results reveal the benefits of feature selection in significantly boosting performance of F_1 score metric, and show that our method is competitive to the state-of-the-art methods in the CL-SciSumm evaluations.

Keywords: Citation analysis, cited text spans identification, feature selection, classification, class imbalance, performance evaluation, scientific paper summarization

1.0 INTRODUCTION

Manual summarization of scientific literature requires considerable time and effort, and the rate at which new scientific papers are published makes it difficult to keep up. Thus, there has been extensive investigation into automatic summarization of scientific papers. Scientific paper summarization is one of the most challenging applications of automatic text summarization. Such summarization systems need to produce a concise, informative, and fluent summary conveying the key information from the paper(s), and must also synthesize the summary for certain argumentative purposes [21].

Numerous approaches have been developed to automate the synthesis and updating of automatic summaries of scientific papers, e.g., [1], [12], [15], [19], [21], [38], [45], [46], [51]. Recently, the interest in scientific paper summarization has focused on citation-based summarization, which uses citations[1] to a paper to form its summary [1], [38]. This type of summary is called the *citation summary* of a paper, and comprises a set of citation sentences[2] (i.e., citances [40]) where the paper is cited. This form of summarization offers a view of the cited paper with information deemed to be important by peers, and can be seen as a community-created summary of that paper [15], [47]. However, as articulated by [22], a citation summary does not consider the context of the target user, verify the claim of the citation, or provide context from the reference paper.

To strike a balance between context and community insight, a new promising direction has emerged: the creation of a citance-focused faceted summary that groups all cited and citing sentences by facet (e.g., the goal of the paper, methods, results obtained, and conclusions) [22]. As shown in Fig. 1, this involves three main tasks: (1A) identifying cited text spans for citances, (1B) classifying their discourse facets, and (2) generating a structured summary from the cited text spans and citances.

Given: A topic consisting of a reference paper (RP) and a set of citing papers (CPs) that all contain citations to the RP. In each CP, the text spans (i.e., citances) have been identified that pertain to a particular citation to the RP.
Task 1A: For each citance, identify the spans of text (cited text spans) in the RP that most accurately reflect the citance. These are of the granularity of a sentence fragment, a full sentence, or several consecutive sentences.
Task 1B: For each cited text span, identify what discourse facet of the paper it belongs to, from a predefined set of facets. Discourse facets describe the type of information in the cited text span.
Task 2: Finally, generate a structured summary of the RP from the cited text spans of the RP and all of the community discussion represented in the citances.

Fig. 1: The three main tasks for creating a citance-focused faceted summary of a paper [22]

This work deals with Task 1A. It builds on our prior effort [53], but more specifically focuses on investigating the use of feature selection techniques to enhance classification performance in the context of identifying cited text spans for citances. We propose a new method that integrates feature selection and classification (see Fig. 2). The feature selection techniques investigated are filter-based, and are independent of the classifiers used. A preferred sequence of features is built according to their individual predictive power, and a reduced feature set is computed by removing features of low predictive power. We consider six feature-goodness criteria: χ^2 (Chi-Squared)-Statistics, Information Gain, Gain Ratio, Relief-F, Significance Attribute Evaluation, and Symmetrical Uncertainty. As for classification, we use five notable algorithms: *k*-Nearest Neighbors, Decision Tree, Support Vector Machine (SVM), Naïve Bayes, and Random Forest. Additionally, we employ an over-sampling approach, SMOTE (Synthetic Minority Over-sampling Technique) [11], to tackle the class imbalance problem during training. The primary goal of this study is the evaluation and comparison of classification results after applying feature selection to classification results from the original problem.

The main contributions of this paper are summarized below:

1. By extending the classification-based method in [53], we propose a new method for Task 1A that further enhances classification performance by feature selection. The proposed method investigates multiple feature selection techniques to identify a subset of features that can accurately represent the data, and explores combinations of feature selection and classification to produce the best results.
2. By integrating data sampling techniques (e.g., SMOTE [11] in this study) into the pipeline of the proposed method, the effect caused by class imbalance during training can be mitigated.
3. The proposed method is evaluated in a case study using the CL-SciSumm corpora. Experimental results reveal the benefits of feature selection in boosting performance of F_1 score metric.
4. A comparison is conducted to understand the influences of variables (e.g., type of classifier, type of feature selection method, and number of selected features) that can affect the performance. Empirical analysis on the

[1] Different citations to the same paper often focus on distinct aspects of that paper [15]. Hence, the literature has taken advantage of citations to understand the main findings and contributions of a paper and how that paper affects other papers.
[2] We use the terms *citing sentences* and *citation sentences* interchangeably.

relations between feature selection methods is also performed, and a breakdown by category of the most prominent features is provided.

The rest of this paper is organized as follows: Section 2.0 briefly reviews related work, Section 3.0 details the proposed method, Section 4.0 presents and discusses the evaluation results, and Section 5.0 concludes and suggests future research directions.

2.0 RELATED WORK

Previous work identifies the best-matching cited text spans for citances, assuming that a citance and the cited text spans to which it refers share similarity of meaning.[3] We roughly categorize methods in the literature into information retrieval (IR)-based, classification-based, learning to rank (L2R)-based, and hybrid methods.

IR-based methods: [36] extracted a subset of reference sentences that are with the same facet as the citance. Then, a bi-directional similarity was applied combining word-to-word similarity and word specificity to identify from the subset the most similar sentence to the citance. [16] created an index that holds all the different spans of text of the reference paper and transformed each citance into a query. Each query was subsequently used to retrieve the most relevant spans of text, depending on term frequency-inverse document frequency (TF-IDF) similarity and BM25. [8] utilized word embeddings based similarity to identify relevant sentences from reference papers. They also studied several variations, including rank optimization, normalized embeddings, and average embeddings over a window.

Classification-based methods: [2] compared methods of word classification, sequence labelling, and segment classification, and found that segment classification performed best. [53] used binary classification to distinguish relevant pairs of citances and reference sentences from irrelevant pairs. They compared several classification algorithms, including *k*-nearest neighbors, decision tree, logistic regression, support vector machine (SVM), naïve Bayes, random forest, and ensembles of individual classifiers. They also explored a wide spectrum of citation-dependent and citation-independent features. [35] applied various classifiers (SVM, decision tree, logistic regression, and nearest neighbors), and combined their results by voting. They used similarity-based, rule-based, and position-based features.

Some studies try both IR-based and classification-based methods, but do not treat them in combination. [39] investigated TF-IDF similarity with multiple incremental modifications and SVMs with a tree kernel. [54] developed a search-based method that considers TF-IDF similarity at sentence- and character-level and word2vec similarity. Besides, they examined a logistic regression classifier.

L2R-based methods: [10] cast the task as a ranking problem by Ranking SVM. A reference paper was dismantled into *n*-sentence chunks, and the top *n*-sentence chunks relevant to each citance was extracted. [29] trained an L2R model with features indicating lexical overlap and semantic similarity between sentences. The top-ranked reference sentence and its adjacent sentences (if they also appeared high in the ranking) were chosen.

Hybrid methods: [26] developed three methods based on TextSentenceRank. The first applied a modified TextSentenceRank to incorporate the similarity of reference sentences to the citance on textual level. The second employed random forest to select from the candidates extracted by the original TextSentenceRank. The third used random forest to identify the relevant sub-parts of the reference paper, and applied the original TextSentenceRank to each sub-part to extract cited text spans. [41] scored each reference sentence using a hybrid model that considers TF-IDF similarity and the similarity predicted by a single-layer neural network. Sentences were selected via diversity-based re-ranking. [42] utilized an artificial neural network (ANN) as filtering to find candidate reference sentences. To determine the cited text, TF similarity between candidate sentences and the citance was measured. [43] proposed a joint scoring method that weights surface-level closeness and semantic relation. The surface-level closeness incorporated TF-IDF similarity and the longest common subsequence score, and the semantic relation was learned from a pairwise neural network ranking model. [31] explored different combination strategies (e.g., voting, Jaccard focused, and Jaccard cascade) on the basis of various feature rules of different lexicons and similarities, and also tested SVM. [32] extended [31] and employed additional features based on the deep semantic information obtained by WordNet and a convolutional neural network (CNN). Based on [32], [30] adopted word mover‘s distance (WMD) and improved latent Dirichlet allocation (LDA) model to calculate sentence similarity. [24] ranked

[3] This assumption follows the findings of [15] that co-citation implies textual similarity.

reference sentences by structural correspondence learning, positional language models, and textual entailment techniques. Further, they attempted three method combinations: (1) a linear combination of the methods, (2) the use of one method as a "filter" for another, and (3) the use of L2R algorithms which are fed the scores of individual methods. [4] introduced a voting system that leverages the best results from word embeddings distance, modified Jaccard, and BabelNet embeddings distance. Following [4], [3] designed a voting scheme based on supervised (convolutional neural network) and unsupervised techniques using word embeddings representations and features computed from linguistic and semantic analysis of the documents. [52] used random forest with multiple features. They integrated random forest with BM25 and VSM (vector space model) model, and applied a voting strategy to select the most related text spans. In addition, they integrated language models with embeddings into the voting system to improve performance. [14] extracted candidate cited text spans using handcrafted patterns, and applied *k*-nearest neighbors with lexical and syntactic features (cosine similarity, LDA score, and WMD score) to group similar sentences in one cluster. Top scored sentences were selected according to their Jaccard and TF-IDF similarities.

2.1 Comparison Between This Work and Previous Work

This work approaches the task of identifying cited text spans for citances as binary classification, and thus is essentially similar to previous classification-based studies. The main difference from earlier studies is that this work more specifically focuses on investigating the application of feature selection techniques to identify a subset of features that can accurately represent the data, reduce the complexity of the feature space, and enhance classification performance. This work also brings forth new ideas of integrating data sampling techniques into the pipeline of the proposed method to tackle the class imbalance problem during training. Finally, this study, as far as we know, is the first systematic evaluation on investigating and comparing the feasibility and performance of various combinations of feature selection strategies and classification methods.

3.0 PROPOSED METHOD

Fig. 2 illustrates the proposed method, which integrates feature selection and classification to enhance classification performance and uses data sampling to address imbalanced data during training. The process consists of two phases: *training* and *prediction*, which we detail as follows.

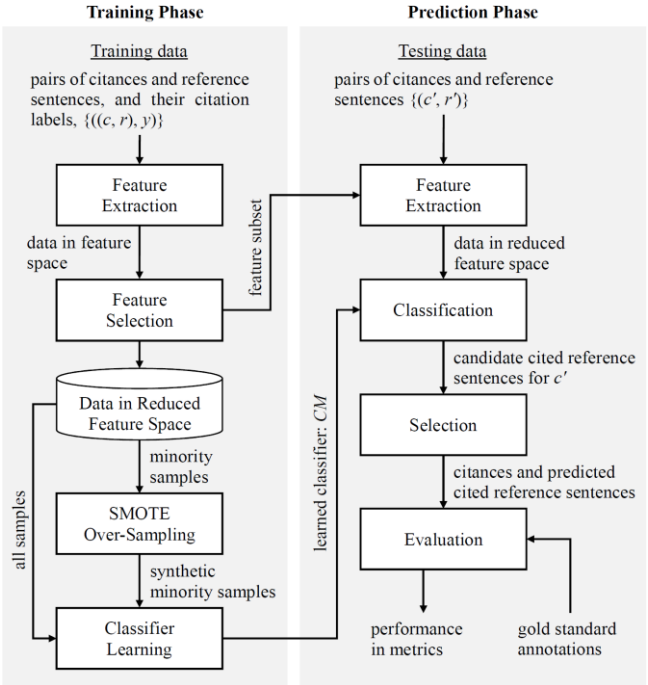

Fig. 2: Overview of the proposed method

Training phase. Recall that in Fig. 1 the given topic consists of a reference paper (RP) and a set of citing papers (CPs) that all contain citations to the RP, and in each CP, the text spans (i.e., citances) have been identified that pertain to a particular citation to the RP. We break down the given topic into pairs of citances and reference

sentences as the training data. Formally, the training data is a set of pairs $\{((c, r), y)\}$, where c is a citance pertaining to a reference paper *RP*, r is a reference sentence of *RP*, and $y \in \{citation, non\text{-}citation\}$. (c, r) is a citation instance if y is labelled as "citation" (i.e., r is the cited reference sentence of c), and is a non-citation instance if y is labelled as "non-citation" (i.e., r is not the cited reference sentence of c). Note that the training data is class imbalanced since the number of citation instances is far less than the number of non-citation instances (see Section 4.1). Each (c, r) is modelled via *feature extraction* and *feature selection*. Feature extraction transforms the data into numerical values of ad hoc features. Feature selection selects relevant (or discriminatory) features to accurately represent the data in the reduced feature space. With the reduced feature set F comprising an $|F|$-dimensional vector space, (c, r) is represented as a feature vector, i.e., $(c, r) = \langle x_1, x_2, \ldots, x_{|F|} \rangle$, where ϕ_i is a feature extraction function and $\phi_i(c, r) = x_i$ w.r.t. feature f_i.[4] To balance the training data, *SMOTE over-sampling* applies SMOTE (Synthetic Minority Over-sampling Technique) [11] to introduce biases towards the minority. It creates synthetic minority class instances (i.e., citation instances) by forming convex combinations of neighboring instances. The input to *classifier learning* uses training instances, and their feature vectors and citation labels. The output is a binary classification model, *CM*, and ideally $CM(c, r) = y$ for all training instances. The learning step trains a predictive model in order to optimize for some specific performance metrics (e.g., classification accuracy, error rate) with the observed data.

Prediction phase. Given an unseen instance (c', r') in feature vector using the same feature subset, *CM* decides its proper citation label. The reference sentences classified as cited reference sentences of c' compose the candidate output. *Selection* further chooses candidates with high relatedness to c' as the final output. For evaluation, performance metrics are calculated by comparing the match between the output and the gold standard.

3.1 Feature Extraction

Following [53], we consider five families of features: *lexical*, *knowledge-based*, *corpus-based*, *syntactic*, and *surface features*. Table 1 lists all the features. The first four families are citation-dependent, and the last is citation-independent. While citation-dependent features evaluate the citation relation between c and r using text similarity measures, citation-independent features focus only on assessing the significance of r.

Table 1: List of features used in this study (excerpted from [53])

Feature family	Feature name
Lexical	Word overlap; TF-IDF measure; Identity measure; ROUGE score; Named entity overlap; Number overlap; Discriminative degree of citation-related word pairs
Knowledge-based	WordNet-based semantic similarity; ADW semantic similarity; WordNet-based lexical overlap
Corpus-based	LSA-based semantic similarity; LSA-based lexical overlap
Syntactic	Dependency overlap; Lexico-syntactic subsumption; Word order similarity
Surface	Sentence length; Sentence position; Similarity with title; Similarity with first-sentence; Similarity with context; Similarity with centroid; TextRank centrality; Num. of named entities; Num. of numbers; Discriminative degree of citation-related words

To measure the relatedness between c and r, lexical features use words shared by them and word occurrence statistics, knowledge-based features consider linguistic knowledge derived from WordNet [37], corpus-based features apply corpus statistics to derive semantic relations between words, and syntactic features compare their syntactic structures obtained by deep linguistic analysis. Surface features are mainly borrowed from text summarization to measure the significance of r in the *RP*. Please refer to [53] for the technical details of feature extraction.

Our implementation extracts a total of 343 features in consideration of various factors, e.g., the granularity of units, the forms of words (e.g., single words, n-grams, composite words, and lemmas), the use of parts-of-speech, the removal of stopwords, the term-weighting schemes, the use of the context of r, and the parameter settings in feature extraction.

[4] Given citance c, the x_i are normalized by min-max normalization over all reference sentences.

3.2 Feature Selection

Feature selection determines a subset of features useful for building a good predictor. Techniques of feature selection can be divided into *filter*, *wrapper*, and *embedded* methods [17]. This study adopts a filter method – *feature ranking* – due to its simplicity, scalability, and good empirical success. Using a feature-goodness criterion, the method defines a preferred sequence of features, where a feature with high goodness is potentially more relevant to the problem. For classification, the criterion is usually measured by the dependence of individual features to the target class. For feature elimination, a subset of features is constructed by removing low-scored features experimentally or according to a threshold.

We consider six feature-goodness criteria: χ^2 (Chi-Squared)-Statistics [34], Information Gain [20], Gain Ratio [48], Relief-F [27], Significance Attribute Evaluation [6], and Symmetrical Uncertainty [44].

- **χ^2 (Chi-Squared)-Statistics (CHI) [34]**

CHI evaluates each feature according to its value of the chi-squared statistic in relation to the classes. For feature *f*, suppose *k* is the number of classes; A_{ij} is the number of instances in the *i*-th interval[5], *j*-th class; R_i is the number of instances in the *i*-th interval, $R_i = \sum_{j=1}^{k} A_{ij}$; C_j is the number of instances in the *j*-th class, $C_j = \sum_{i=1}^{2} A_{ij}$; N is the total number of instances, $N = \sum_{i=1}^{2} R_i$; and E_{ij} is the expected frequency of A_{ij}, $E_{ij} = \frac{R_i \times C_j}{N}$. The χ^2 value of feature *f* is calculated by

$$CHI(f) = \sum_{i=1}^{2}\sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \quad . \tag{1}$$

In practice, a feature of an χ^2 value less than 3.841 is eliminated for $p = 0.05$ with 1 degree of freedom.

- **Information Gain (IG) [20]**

IG measures the amount of information obtained for class prediction in bits that the presence or absence of a feature gives about the classes. It estimates the reduction in entropy between the prior entropy of classes {*C*} and the posterior entropy, given values {*V*} for feature *f*:

$$IG(f) = -\sum_{C} P(C)\log_2 P(C) - \left(-\sum_{V} P(V) \sum_{C} P(C|V)\log_2 P(C|V) \right), \tag{2}$$

where $P(C)$ is the prior probability of class *C*, $P(V)$ is the prior probability that an instance has value *V* for feature *f*, and $P(C|V)$ is the conditional probability that instances with value *V* for feature *f* belong to class *C*.

- **Gain Ratio (GR) [48]**

As an adaptation of IG, GR compensates for the information gain's bias in favor of features with more values, and introduces a split information value to normalize IG. The split information indicates the potential information generated by splitting the training data into partitions, corresponding to values {*V*} of feature *f*, and is calculated as

$$SI(f) = -\sum_{V} P(V)\log_2 P(V) \, , \tag{3}$$

where $P(V)$ is the prior probability that an instance has value *V* for feature *f*. The gain ratio of feature *f* is the ratio of the information to the split information:

$$GR(f) = \frac{IG(f)}{SI(f)} \, . \tag{4}$$

[5] The range of a numeric feature needs to be discretized into intervals, using, e.g., the entropy-based discretization method [34]. In most cases, there are only two intervals for a feature.

Here, $IG(f)$, as shown in Eq. (2), is the information gain for feature *f*.

- **Relief-F (RE) [27]**

Relief [25] estimates the quality of features according to how well their values distinguish between the instances of the same and different classes that are near each other. Relief is applicable only to two-class problems and cannot handle incomplete data. Its variation, Relief-F [27], can deal with multiclass problems, is more robust, and can tackle incomplete and noisy data. Fig. 3 illustrates the Relief-F algorithm, which iterates *m* times. In each cycle, for a randomly selected training instance *R* (selected without replacement), the algorithm finds the *k* nearest neighbors from the same class (i.e., nearest hits; denoted $H = \{H_1, H_2, \ldots, H_k\}$) and the *k* nearest neighbors from each of the different classes (i.e., nearest misses; denoted $M(C) = \{M_1(C), M_2(C), \ldots, M_k(C)\}$ for class *C*). It subsequently updates the quality estimation *W* for all the features based on the average of the contribution of all the hits and all the misses.[6] For feature *f*, RE(*f*) equals *W*[*f*] when the process ends.

***Algorithm* Relief-F**

Input: for each training instance, a vector of feature values and its class
Output: the vector *W* of estimations of the quality of features

1. set initial weight for each feature *f*, *W*[*f*] = 0.0
2. **for** *i* = 1 to *m* **do begin**
3. randomly select an instance *R*
4. find the *k* nearest hits *H*
5. **for** each class $C \neq class(R)$ **do**
6. from class *C* find the *k* nearest misses *M*(*C*)
7. **for** each feature *f* **do**
8. $$W[f] = W[f] - \sum_{j=1}^{k} \frac{\text{diff}(f, R, H_j)}{m \times k} + \sum_{C \neq class(R)} \left(\frac{P(C)}{1 - P(class(R))} \sum_{j=1}^{k} \frac{\text{diff}(f, R, M_j(C))}{m \times k} \right)$$
9. **end**

Fig. 3: Pseudocode for the Relief-F algorithm [27]. diff(*f*, I_1, I_2) calculates the difference between the values of feature *f* for two instances I_1 and I_2, $P(C)$ is the prior probability of class *C*, $1–P(class(R))$ is the sum of probabilities for the misses' classes, and *m* is a user-defined parameter ($m \leq$ number of training instances). Note that diff is used also to measure the distance between instances to find the nearest neighbors, and the total distance is simply the sum of differences over all features

- **Significance Attribute Evaluation (SAE) [6]**

SAE assigns a conditional probability based significance to every feature, determined by its separability and capability, to distinguish instances of distinct classes. The significance of a feature is defined as a two-way function of its association to the class decision. Suppose the *feature-to-class association* of feature *f* is a function of the mean of the discriminating powers of all possible values of *f*:

$$\sigma_1(f) = \left(\frac{1}{m} \sum_{i=1}^{m} \vartheta^i \right) - 1.0, \tag{5}$$

where *m* is the number of distinct feature values for feature *f*, and ϑ^i is the discriminating power of a feature value. Further, assume the *class-to-feature association* for feature *f* is computed as the mean of the separability of its values:

$$\sigma_2(f) = \left(\frac{1}{k} \right) \times \left(\sum_{j=1}^{k} \Lambda^j \right) - 1.0, \tag{6}$$

[6] The contribution for each class of the misses is weighted by the prior probability of that class and divided by the factor $1 – P(class(R))$ [27].

where Λ^j is the separability of the feature values of feature *f* with respect to class *j*, and *k* is the number of different classes. The significance of feature *f* is designated as $SAE(f) = \frac{\sigma_1(f) + \sigma_2(f)}{2}$, i.e., the average of its feature-to-class and class-to-feature association values.

- **Symmetrical Uncertainty (SU) [44]**

SU estimates the feature-to-class correlation. In terms of classes {*C*} and values {*V*} of feature *f*, the symmetrical uncertainty of feature *f* is given by

$$SU(f) = \frac{2 \times IG(f)}{H(C) + H(V)} \text{ where } H(X) = -\sum_X P(X) \log_2 P(X). \quad (7)$$

In the equation, $IG(f)$ is the information gain for feature *f* (see Eq. (2)), *P*(*C*) is the prior probability of class *C*, and *P*(*V*) is the prior probability that an instance has value *V* for feature *f*.

3.3 SMOTE Over-sampling

Constructing predictive models using imbalanced data tends to ignore the minority class of high interest, and will highly favor the majority class (i.e., the class typically carrying lower cost of misclassification) in order to maximize overall accuracy. As noted in [50], two techniques can be adopted to address imbalanced data: cost-sensitive learning and data sampling. For the latter, over-sampling and under-sampling are the two most common re-sampling techniques, both forcing the learner to focus more on correctly classifying instances of the minority class by altering the class distribution of the training data.

We employ an over-sampling approach, SMOTE (Synthetic Minority Over-sampling Technique) [11], to mitigate the effect caused by class imbalance during training. SMOTE creates synthetic minority class instances to introduce more coverage of the minority class, thus allowing a classification algorithm to carve broader decision regions. As described in [11], SMOTE over-samples the minority class by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the *k* minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the *k* nearest neighbors are randomly chosen. In the implementation, we generate synthetic citation instances until the number of instances of citation and non-citation classes is balanced.

3.4 Classifier Learning

The use of classification algorithms includes *k*-Nearest Neighbors [5], Decision Tree [49], Support Vector Machine [13], Naïve Bayes [23], and Random Forest [9].

***k*-Nearest Neighbors (*k*-NN)** [5] is an instance-based classifier that classifies an unseen instance by majority voting of its *k* nearest training instances.

Decision Tree [49] creates a tree-like classification model by iteratively identifying the most significant attribute (i.e., feature in this study), that splits the data into homogeneous sets. In a decision tree, a leaf denotes a class label, an internal node is a test on an attribute, and a branch is the test outcome. This study applies C4.5 [49], which adopts the normalized information gain as the splitting criterion.

Support Vector Machine (SVM) [13] constructs a hyperplane (or set of hyperplanes) of the maximum margin that separates data of distinct classes. The basic form of SVMs learns a linear classifier. By the kernel trick, the algorithm can learn polynomial classifiers, radial basic function networks, and three-layer sigmoid neural nets. This study tests linear SVM for its faster training and competitive accuracy, and adopts L2-regularized L2-loss linear SVM (referred to as L2-SVM in the literature) in implementation.

Naïve Bayes [23] builds a classifier based on Bayes' theorem with the assumption that features are conditionally independent given the class. To classify an unseen instance, a Naïve Bayes classifier uses Bayes' rule to compute the probability of each class, given the vectors of observed values for predictive features. It then predicts the most probable class using Maximum a Posteriori (MAP).

Random Forest [9] is an ensemble method that generates a forest of uncorrelated decision trees and makes a

prediction by the most votes across all the trees. Each tree is grown from a sub-sample drawn with replacement from the training data. At each split, the best split on a random subset of features is used to split a node, and each tree is grown to the largest extent possible (i.e., there is no pruning).

3.5 Selection

To decrease the number of false positives (i.e., those classified as cited reference sentences, but truly non-cited) in the output, we apply our previously developed selection strategy [53]. For citance c', the reference sentences classified as its cited reference sentences compose the candidate output. The final output is made up of the candidates of relatedness to c' greater than a predefined threshold α. The relatedness is currently scored by the TF-IDF similarity between a citance and a reference sentence.

4.0 EVALUATION

4.1 Data

For experiments, we use the CL-SciSumm 2016 and 2017 corpora. Each corpus has two datasets: one for training and one for testing. Note that the CL-SciSumm 2016 corpus has an additional development dataset. Each dataset contains 10-30 topics, and each topic consists of a research paper, its citing papers, and three types of summaries. In each topic, citances are identified by human annotators. Each citance is mapped to its cited text spans and annotated with the information facet(s) it stands for. Fig. 4 shows a citation annotation example where *Citation Offset* indicates the citing sentences (ids: 1 and 2) in the citing article (id: W13-4011), *Reference Offset* indicates the cited sentences (ids: 155 and 156) in the reference article (id: J00-3003), and *Discourse Facet* denotes the facet of the citation. In the test dataset, the fields of Reference Offset and Discourse Facet are not provided and need to be respectively identified in Task 1A and Task 1B.

Citance Number: 5 | Reference Article: J00-3003.xml | Citing Article: W13-4011.xml | Citation Marker Offset: ['1'] | Citation Marker: Stolcke et al., 2000 | **Citation Offset: ['1','2']** | Citation Text: <S sid="1" ssid="1">Conversational feedback is mostly performed through short utterances such as yeah, mh, okaynot produced by the main speaker but by one of the other participants of a conversation.</S><S sid="2" ssid="2">Such utterances are among the most frequent in conversational data (Stolcke et al., 2000).</S> | **Reference Offset: ['155', '156']** | Reference Text: <S sid="155" ssid="75">A backchannel is a short utterance that plays discourse-structuring roles, e.g., indicating that the speaker should go on talking.</S><S sid="156" ssid="76">These are usually referred to in the conversation analysis literature as "continuers" and have been studied extensively (Jefferson 1984; Schegloff 1982; Yngve 1970).</S> | **Discourse Facet:** Method_Citation | Annotator: Muthu Kumar Chandrasekaran, NUS |

Fig. 4: A citation annotation example

Each dataset is transformed into pairs of citances and reference sentences. Table 2 and Table 3 present, respectively, the statistics of the CL-SciSumm 2016 and 2017 corpora. It is observed that the datasets are imbalanced. For example, the training dataset of the CL-SciSumm 2016 corpus has 27,235 non-citation instances compared with 249 citation instances.

Table 2: Statistics of the CL-SciSumm 2016 corpus

Statistics	Training	Development	Test
Num. of topics	10	10	10
Avg. num. of sentences in a reference paper	218.3	223.2	229.1
Avg. num. of citing papers in a topic	8.4	15.3	23.9
Avg. num. of citances in a topic	13.5	21.9	35
Avg. num. of citing sentences in a citance	1.5	1.3	1.3
Avg. num. of cited reference sentences for a citance	1.8	1.5	1.4
Num. of citation instances	249	329	480
Num. of non-citation instances	27235	54704	87697

Table 3: Statistics of the CL-SciSumm 2017 corpus

Statistics	Training	Test
Num. of topics	30	10
Avg. num. of sentences in a reference paper	223.3	201.4
Avg. num. of citing papers in a topic	15.8	10.3
Avg. num. of citances in a topic	19.8	15.9
Avg. num. of citing sentences in a citance	1.4	1.4
Avg. num. of cited reference sentences for a citance	1.6	1.5
Num. of citation instances	929	231
Num. of non-citation instances	147386	32204

4.2 Performance Metrics

We use the official CL-SciSumm metrics: *precision (P)*, *recall (R)*, and F_1 *score* (F_1). For every topic, it measures the overlap of sentence IDs between the system output and the gold standard. Precision is the fraction of outputted reference sentences that are truly cited reference sentences, recall is the fraction of truly cited reference sentences that are outputted, and F_1 score is the harmonic mean of precision and recall:

$$F_1 = 2 \times \frac{P \times R}{P + R} \text{ where} \tag{8}$$

$$P = \frac{\sum_c |\text{sysRef}(c) \cap \text{annRef}(c)|}{\sum_c |\text{sysRef}(c)|} \text{ and } R = \frac{\sum_c |\text{sysRef}(c) \cap \text{annRef}(c)|}{\sum_c |\text{annRef}(c)|}.$$

Note that, given citance c, sysRef (c) is the set of sentence IDs in the Reference Offset field identified by the evaluated system, and annRef (c) is the set of sentence IDs in the Reference Offset field labelled by the human annotators.

The reported scores are the average of those for all topics in the test dataset. The higher the score value, the better the system performance.

4.3 Experimental Setup

To build a classifier with more observed data, we merge the training and the development datasets of the CL-SciSumm 2016 corpus as a new training dataset. As for the CL-SciSumm 2017 corpus, the original training dataset is used. Our implementation of classifiers and feature selection methods relies on Weka [18]. The parameters of feature selection methods are set as Weka's defaults for simplicity. The considered feature numbers are: 5, 10, 30, 50, 100, 150, 200, 250, 300.

For a feature selection method and a feature number, the training data in the reduced feature space is produced and combined with synthetic minority class instances to form new training data. As mentioned in Section 3.3, SMOTE takes each minority class sample and introduces synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Currently, we use 10 nearest neighbors. The new training data is used to construct a classifier. Numerous parameter settings for a classifier (see Table 4) are examined using stratified 5-fold cross-validation. The results are compared via statistical significance testing and the best setting is selected. Repeating the above steps, we collect all the best classifier settings for distinct feature numbers, given that a particular feature selection method is considered. The winner among the best settings is decided as the one of the highest mean F_1 over different rounds of cross-validation. We use the winning classifier setting and the corresponding feature selection method and feature number to train a classification model, and test it on the test dataset following the guidelines of Task 1A. For fair comparisons, the selection threshold α (see Section 3.5) is configured the same as 0.05[7] for classifiers with and without feature selection.

[7] The value is suggested by [53], and is the 1.5 standard deviations from the mean of degree of relatedness distributed in the test dataset.

Table 4: List of the evaluated classifiers. DT stands for Decision Tree, NB stands for Naïve Bayes, and RF stands for Random Forest

Algorithm	Parameter combination
k-NN	k={1, 2, 4, 16, 32, 64}, nearestNeighbourSearchAlgorithm=LinearNNSearch (with distanceFunction=EuclideanDistance)
DT	binarySplits=false, minNumObj=2, unpruned=false, {reducedErrorPruning=true, {confidenceFactor={0.25, 0.5, 0.75}, reducedErrorPruning=false}}
SVM	SVMType=L2-regularized L2-loss support vector classification (dual), bias=1.0, cost={2^−14, 2^−12, …,2^0, …, 2^12, 2^14}, eps=1.0E−4, maximumNumberOfIterations=50000
NB	useKernelEstimator={true, false}
RF	maxDepth=0 (*i.e.*, unlimited), numFeatures=sqrt(#(features)), numIterations={10, 50, 100, 300, 500, 750, 1000, 3000, 5000}

4.4 Results and Discussions

Tables 5-9 respectively list the scores of *k*-NN, Decision Tree (DT), SVM, Naïve Bayes (NB), Random Forest (RF) classifiers, in combination with different feature selection methods. WAF (a.k.a. with-all-features) denotes that all 343 features are used (i.e., no feature selection is applied), NF represents the considered feature numbers, and Impr. means the relative improvement[8] obtained from applying feature selection. The observations of the results suggest the following considerations:

- The classification results of the use of the reduced feature sets (i.e., the proposed method) are generally improved compared to those corresponding to the use of the whole feature set. The results on the CL-SciSumm 2016 corpus show that: (1) the improvements of *k*-NN and DT with feature selection are significant (averagely 9.6%[9] for *k*-NN and 30.8% for DT); (2) the improvements of NB and RF with feature selection are moderate (averagely 4.4% for NB and 3.3% for RF); and (3) for most SVM combinations, only slight improvements (around 1-2% and averagely 0.9%) are obtained. SVM+GR and SVM+IG are two exceptions, which have inferior results to the direct SVM. The results on the CL-SciSumm 2017 corpus show that DT with feature selection has significant improvements (averagely 19.5%), while the improvements of *k*-NN, SVM, NB, and RF with feature selection are moderate (averagely 4% for *k*-NN, 2.7% for SVM, 6.1% for NB, and 3.9% for RF). Note that DT using the whole feature set is seen predicting many false positives, and has relatively poor results on both the CL-SciSumm 2016 and 2017 corpora. We conjecture that DT may overfit the training data in the full feature space. In this viewpoint, DT with feature selection helps reduce overfitting, leading to substantial increases in performance. Additionally, SVM with feature selection has relatively slight improvements on both the CL-SciSumm 2016 and 2017 corpora. SVM is an approximate implementation of a bound on the generalization error, that depends on the margin, but is independent of the dimensionality of the feature space [7]. Thus, feature selection methods might have no increased performance guarantees, provided that the regularization parameters are properly tuned over the use of the whole feature set.
- There is no single feature selection that outperforms all the others. For example, GR outperforms CHI on the CL-SciSumm 2016 corpus when integrated with *k*-NN. On the same data, conversely, CHI is superior to GR when integrated with DT. It is noted that the effectiveness of the whole classification system is generally due to the combined effect of classifier and feature selection.
- The number of selected features is generally less than the total number of available features. It is also viewed that the number of the most useful features depends on the classification algorithm, and varies for distinct feature selection methods. The observations on both the CL-SciSumm 2016 and 2017 corpora illustrate that: (1) relatively small feature subsets (around 5-50 features) are suggested for *k*-NN; (2) regarding DT and NB, small feature subsets (around 30-100 features) are effective in most cases, although 200-300 features are

[8] The relative improvement is calculated by (b−a)/a×100 when b is compared to a.

[9] The average is computed over performance increases obtained by distinct feature selection methods. In this case, (9.3%+12.3%+9.3%+12.8%+10.9%+2.7%)/6=9.6%.

selected in some cases; (3) for SVM and RF[10], in contrast, large feature subsets (around 200-300 features) are implied.

Table 5: Results of *k*-NN with distinct feature selection methods (best performance bold-faced)

	CL-Scisumm 2016			**CL-SciSumm 2017**		
	NF	**F_1**	**Impr.**	**NF**	**F_1**	**Impr.**
WAF	All	0.1098	--	All	0.1070	--
CHI	10	0.1200	9.3%	5	0.1112	3.9%
GR	30	0.1233	12.3%	30	0.1104	3.2%
IG	10	0.1200	9.3%	10	0.1118	4.5%
RE	30	**0.1238**	12.8%	50	0.1123	5.0%
SAE	10	0.1218	10.9%	30	**0.1129**	5.5%
SU	5	0.1128	2.7%	10	0.1091	2.0%

Table 6: Results of DT with distinct feature selection methods (best performance bold-faced)

	CL-Scisumm 2016			**CL-SciSumm 2017**		
	NF	**F_1**	**Impr.**	**NF**	**F_1**	**Impr.**
WAF	All	0.0900	--	All	0.0957	--
CHI	50	**0.1343**	49.2%	30	0.1082	13.1%
GR	150	0.1096	21.8%	50	0.1144	19.5%
IG	50	0.1253	39.2%	50	**0.1209**	26.3%
RE	5	0.1073	19.2%	30	0.1104	15.4%
SAE	200	0.1141	26.8%	100	0.1153	20.5%
SU	50	0.1155	28.3%	50	0.1168	22.0%

Table 7: Results of SVM with distinct feature selection methods (best performance bold-faced)

	CL-Scisumm 2016			**CL-SciSumm 2017**		
	NF	**F_1**	**Impr.**	**NF**	**F_1**	**Impr.**
WAF	All	0.1425	--	All	0.1303	--
CHI	250	0.1442	1.2%	200	0.1332	2.2%
GR	250	0.1402	–1.6%	200	0.1324	1.6%
IG	200	0.1421	–0.3%	250	0.1319	1.2%
RE	250	0.1454	2.0%	250	0.1343	3.1%
SAE	150	0.1449	1.7%	200	0.1347	3.4%
SU	150	**0.1456**	2.2%	150	**0.1364**	4.7%

Table 8: Results of NB with distinct feature selection methods (best performance bold-faced)

	CL-Scisumm 2016			**CL-SciSumm 2017**		
	NF	**F_1**	**Impr.**	**NF**	**F_1**	**Impr.**
WAF	All	0.1288	--	All	0.1141	--
CHI	30	0.1338	3.9%	10	0.1212	6.2%
GR	300	0.1317	2.3%	100	0.1183	3.7%
IG	30	0.1358	5.4%	50	0.1163	1.9%
RE	250	0.1305	1.3%	50	**0.1274**	11.7%
SAE	30	**0.1436**	11.5%	30	0.1259	10.3%
SU	200	0.1310	1.7%	150	0.1172	2.7%

[10] RF performs well when the feature number is high since it generates a tree based on a sub-sample of the data and a random subset of features [9].

Table 9: Results of RF with distinct feature selection methods (best performance bold-faced)

	CL-Scisumm 2016			CL-SciSumm 2017		
	NF	**F_1**	**Impr.**	**NF**	**F_1**	**Impr.**
WAF	All	0.1310	--	All	0.1239	--
CHI	250	0.1346	2.7%	200	0.1264	2.0%
GR	200	**0.1365**	4.2%	200	0.1311	5.8%
IG	300	0.1355	3.4%	250	**0.1324**	6.9%
RE	300	0.1339	2.2%	200	0.1239	0.0%
SAE	200	0.1360	3.8%	300	0.1302	5.1%
SU	300	0.1357	3.6%	250	0.1283	3.6%

Table 10 and Table 11 summarize the best results of the proposed method. For comparison purposes, the results obtained by using the whole feature set and the official Task 1A results of the top 5 CL-SciSumm 2016 and 2017 systems are also presented. Table 10 shows that for the CL-SciSumm 2016 corpus, *k*-NN+RE outperforms *k*-NN by 12.7%, DT+CHI outperforms DT by 48.9%, SVM+SU outperforms SVM by 2.1%, NB+SAE outperforms NB by 11.6%, and RF+GR outperforms RF by 4.6%. Table 11 shows that for the CL-SciSumm 2017 corpus, *k*-NN+SAE outperforms *k*-NN by 5.6%, DT+IG outperforms DT by 26.0%, SVM+SU outperforms SVM by 4.6%, NB+RE outperforms NB by 11.4%, and RF+IG outperforms RF by 6.5%. Besides, Table 10 presents that three models of the proposed method have substantially superior performance to the Top-1 CL-SciSumm 2016 system (Sys15$tfidf+st+sl). They are SVM+SU with an improvement of 9.0%, NB+SAE with an improvement of 7.5%, and RF+GR with an improvement of 2.2%. DT+CHI ties with Sys15$tfidf+st+sl. Table 11 presents that three models of the proposed methods have substantially superior performance to the Top-1 CL-SciSumm 2017 system (NJUST Run 2). They are SVM+SU with an improvement of 9.7%, NB+RE with an improvement of 2.4%, and RF+IG with an improvement of 6.5%. Overall, the results reveal the benefits of feature selection in significantly boosting classification performance. Our method is also found performing competitively, compared to the Top5 CL-SciSumm 2016 and 2017 systems.

Table 10: Performance comparison of classifiers with and without feature selection and the top 5 CL-SciSumm 2016 systems (best performance bold-faced)

System	**Rank**	**F_1**
A. Classifiers without feature selection		
k-NN	13	0.110
DT	15	0.090
SVM	3	0.143
NB	8	0.129
RF	7	0.131
B. Classifiers with feature selection		
k-NN+RE (NF: 30)	11	0.124
DT+CHI (NF: 50)	5	0.134
SVM+SU (NF: 150)	1	**0.146**
NB+SAE (NF: 30)	2	0.144
RF+GR (NF: 200)	4	0.137
C. Top 5, median, and worst CL-SciSumm 2016 systems		
Sys15$tfidf+st+sl [39]	5	0.134
Sys8$Fusion [31]	9	0.126
Sys8$Jaccard Focused [31]	9	0.126
Sys8$Voting1 [31]	12	0.116
Sys8$Voting2 [31]	14	0.108
Median system	16	0.039
Worst system	17	0.008

Table 11: Performance comparison of classifiers with and without feature selection and the top 5 CL-SciSumm 2017 systems (best performance bold-faced)

System	Rank	F_1
A. Classifiers without feature selection		
k-NN	13	0.107
DT	15	0.096
SVM	3	0.130
NB	9	0.114
RF	5	0.124
B. Classifiers with feature selection		
k-NN+SAE (NF: 30)	11	0.113
DT+IG (NF: 50)	8	0.121
SVM+SU (NF: 150)	1	**0.136**
NB+RE (NF: 50)	4	0.127
RF+IG (NF: 250)	2	0.132
C. Top 5, median, and worst CL-SciSumm 2017 systems		
NJUST Run 2 [35]	5	0.124
NJUST Run 5 [35]	7	0.123
NJUST Run 4 [35]	9	0.114
TUGRAZ Run 2 [16]	12	0.110
CIST Run 1 [32]	13	0.107
Median system	16	0.074
Worst system	17	0.014

Measuring the similarity between the feature subsets generated by different feature selection methods can be useful, for example, to identify diverse feature selection methods for constructing ensembles. As proposed in [28], the similarity can be calculated by the consistency index for two subsets. More precisely, $sim(A,B) = \frac{rn-k^2}{k(n-k)}$, in which A and B are two feature subsets of the full feature set FS, $|A| = |B| = k$, $0 < k < |FS| = n$, and $r = |A \cap B|$. Analyzed from the CL-SciSumm 2016 corpus, Table 12 presents the similarity between different feature selection methods, given 30 features selected by each method. Pairs of (CHI, IG), (GR, SU), and (IG, SAE) have a large similarity value greater than 0.7, implying that similar features are selected. Pairs of (CHI, RE), (GR, SAE), (IG, RE), and (RE, SAE) have a similarity lower than 0.3, in contrast, implying that dissimilar features are selected. Note that we also observe different similarity results for varying number of selected features.

Table 12: Similarity between different feature selection methods (30 features selected by each method), analyzed from the CL-SciSumm 2016 corpus

	CHI	GR	IG	RE	SAE	SU
CHI	–	0.45	**0.89**	0.23	0.67	0.63
GR		–	0.34	0.49	0.16	**0.82**
IG			–	0.12	**0.78**	0.53
RE				–	0.05	0.49
SAE					–	0.34
SU						–

Lastly, Table 13 provides a breakdown by category of the top 50 most prominent features in the CL-SciSumm 2016 corpus. For simplicity, the ranking of features is built via Borda count [33]. The aggregation ranks every feature according to its accumulated order of preferences made by different feature selection methods. The column "# of features" means, for a feature, the number of its variations listed in the top 50. It is seen that the lexical features are the majority, which dominates 82%. The surface features account for the remaining 18%. It is somewhat surprising that complex features (e.g., knowledge-based, corpus-based, and syntactic features) do not make the top 50 feature list. We perform further analysis regarding two factors: the granularity of words and the context of *r*. For the

granularity of words, 1-gram (i.e., single word) accounts for 78%, 2-gram for 6%, longest common subsequence (LCS) for 10%, and skip-bigram with a maximum skip distance of 4 plus unigram (SU4 for short) for 6%.[11] Regarding the context of *r*, no context considered accounts for 50%, the use of *r*'s previous sentence for 26%, and the use of *r*'s next sentence for 24%. The preliminary analysis is valuable, for it can help further feature engineering to discover potentially more effective features.

Table 13: Statistics of the top 50 most prominent features in the CL-SciSumm 2016 corpus

Feature family	Feature name	# of features
Lexical (82%)	Word overlap	8 (16%)
	ROUGE score	15 (30%)
	Discriminative degree of citation-related word pairs	18 (36%)
Surface (18%)	Num. of named entities	1 (2%)
	Sentence length	1 (2%)
	Similarity with title	1 (2%)
	TextRank centrality	6 (12%)

5.0 CONCLUSIONS AND FUTURE WORK

This paper focuses on identifying cited text spans for citances, the first step towards scientific paper summarization. We approach the task as binary classification to distinguish relevant pairs of citances and reference sentences from irrelevant pairs. We propose a new method to enhance classification performance by integrating feature selection and classification techniques (see Fig. 2). Various combinations of feature selection methods and classification algorithms are explored. The feature selection techniques investigated are filter-based. They are χ^2-Statistics, Information Gain, Gain Ratio, Relief-F, Significance Attribute Evaluation, and Symmetrical Uncertainty. Once the most relevant (or discriminatory) features are selected, the classification algorithms, namely, *k*-Nearest Neighbors, Decision Tree, Support Vector Machine, Naïve Bayes, and Random Forest, are applied to build predictive classifiers. Practically, the classification task suffers the class imbalance problem since the training data has far less relevant pairs of citances and reference sentences (i.e., citation instances) than irrelevant pairs (i.e., non-citation instances). We apply SMOTE to introduce synthetic biases towards the minority, forcing the learner to focus more on correctly classifying the minority. The proposed method is evaluated using the CL-SciSumm 2016 and 2017 corpora. Experimental results reveal that the application of feature selection helps identify a subset of features that can accurately represent the data, reduce the complexity of the feature space, and produce substantial improvements in performance (see Tables 5-9). It is also found that our method is competitive to the state-of-the-art methods in the CL-SciSumm evaluations (see Table 10 and Table 11).

Future work will consider the following: (1) investigating the feasibility of additional feature selection techniques, e.g., wrapper and embedded methods, (2) studying combinations of feature selection methods for constructing ensembles, and (3) further feature engineering based on the analysis of the most prominent features to design potentially more effective features.

REFERENCES

[1] A. Abu-Jbara, & D. Radev, "Coherent Citation-Based Summarization of Scientific Papers", in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, Portland, OR, 2011, pp. 500-509.

[2] A. Abu-Jbara, & D. Radev, "Reference Scope Identification in Citing Sentences", in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2012)*, Montréal, QC, Canada, 2012, pp. 80-90.

[3] A. Abura'ed, A. Bravo, L. Chiruzzo, & H. Saggion, "LaSTUS/TALN+INCO @ CL-SciSumm 2018 – Using Regression and Convolutions for Cross-document Semantic Linking and Summarization of

[11] LCS and SU4 are considered only for the feature "ROUGE score".

Scholarly Literature", in *Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018)*, Ann Arbor, MI, 2018, pp. 150-163.

[4] A. Abura'ed, L. Chiruzzo, H. Saggion, P. Accuosto, & A. Bravo, "LaSTUS/TALN @ CLSciSumm-17: Cross-document Sentence Matching and Scientific Text Summarization Systems", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 55-66.

[5] D. W. Aha, D. Kibler, & M. K. Albert, "Instance-Based Learning Algorithms". *Machine Learning*, Vol. 6, No. 1, 1991, pp. 37-66.

[6] A. Ahmad, & L. Dey, "A Feature Selection Technique for Classificatory Analysis". *Pattern Recognition Letters*, Vol. 26, No. 1, 2005, pp. 43-56.

[7] P. Bartlett, & J. Shawe-Taylor, "Generalization Performance of Support Vector Machines and Other Pattern Classifiers", in *Advanced in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, & A. J. Smola, Eds. The MIT Press, 1998, pp. 43-54.

[8] G. Baruah, & M. Kolla, "Klick Labs at CL-SciSumm 2018", in *Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018)*, Ann Arbor, MI, 2018, pp. 134-141.

[9] L. Breiman, "Random Forests". *Machine Learning*, Vol. 45, No. 1, 2001, pp. 5-32.

[10] Z. Cao, W. Li, & D. Wu, "PolyU at CL-SciSumm 2016", in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, Newark, NJ, 2016, pp. 132-138.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, & W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique". *Journal of Artificial Intelligence Research*, Vol. 16, 2002, pp. 321-357.

[12] A. Cohan, & N. Goharian, "Scientific Article Summarization Using Citation-Context and Article's Discourse Structure", in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, Lisbon, Portugal, 2015, pp. 390-400.

[13] C. Cortes, & V. Vapnik, "Support-Vector Networks". *Machine Learning*, Vol. 20, No. 3, 1995, pp. 273-297.

[14] D. Debnath, A. Achom, & P. Pakray, "NLP-NITMZ @ CLScisumm-18", in *Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018)*, Ann Arbor, MI, 2018, pp. 164-171.

[15] A. Elkiss, S. Shen, A. Fader, G. Erkan, D. States, & D. Radev, "Blind Men and Elephants: What Do Citation Summaries Tell Us About a Research Article?". *Journal of the American Society for Information Science and Technology*, Vol. 59, No. 1, 2008, pp. 51-62.

[16] T. Felber, & R. Kern, "Graz University of Technology at CL-SciSumm 2017: Query Generation Strategies", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 67-72.

[17] I. Guyon, & A. Elisseeff, "An Introduction to Variable and Feature Selection". *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 1157-1182.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, & I. H. Witten, "The WEKA Data Mining Software: An Update". *SIGKDD Explorations*, Vol. 11, No. 1, 2009, pp. 10-18.

[19] C. D. V. Hoang, & M.-Y. Kan, "Towards Automated Related Work Summarization", in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010): Posters*, Beijing, China, 2010, pp. 427-435.

[20] E. B. Hunt, J. Marin, & P. J. Stone, *Experiments in Induction*. Academic Press, 1966.

[21] K. Jaidka, C. S. G. Khoo, & J.-C. Na, "Deconstructing Human Literature Reviews – A Framework for Multi-Document Summarization", in *Proceedings of the 14th European Workshop on Natural Language Generation (ENLG 2013)*, Sofia, Bulgaria, 2013, pp. 125-135.

[22] K. Jaidka, M. K. Chandrasekaran, S. Rustagi, & M.-Y. Kan, "Overview of the CL-SciSumm 2016 Shared Task", in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, Newark, NJ, 2016, pp. 93-102.

[23] G. H. John, & P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers", in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*, Montréal, QC, Canada, 1995, pp. 338-345.

[24] S. Karimi, L. Moraes, A. Das, & R. Verma, "University of Houston @ CL-SciSumm 2017: Positional Language Models, Structural Correspondence Learning and Textual Entailment", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 73-85.

[25] K. Kira, & L. A. Rendell, "A Practical Approach to Feature Selection", in *Proceedings of the 9th International Workshop on Machine Learning (ML92)*, Aberdeen, Scotland, UK, 1992, pp. 249-256.

[26] S. Klampfl, A. Rexha, & R. Kern, "Identifying Referenced Text in Scientific Publications by Summarisation and Classification Techniques", in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, Newark, NJ, 2016, pp. 122-131.

[27] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF", in *Proceedings of the 7th European Conference on Machine Learning (ECML-94)*, Catania, Italy, 1994, pp. 171-182.

[28] L. I. Kuncheva, "A Stability Index for Feature Selection", in *Proceedings of the 25th IASTED International Multi-Conference on Artificial Intelligence and Applications*, Innsbruck, Austria, 2007, pp. 390-395.

[29] A. Lauscher, G. Glavas, & K. Eckert, "University of Mannheim @ CLSciSumm-17: Citation-Based Summarization of Scientific Articles Using Semantic Textual Similarity", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 33-42.

[30] L. Li, J. Chi, M. Chen, Z. Huang, Y. Zhu, & X. Fu, "CIST@CLSciSumm-18: Methods for Computational Linguistics Scientific Citation Linkage, Facet Classification and Summarization", in *Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018)*, Ann Arbor, MI, 2018, pp. 84-95.

[31] L. Li, L. Mao, Y. Zhang, J. Chi, T. Huang, X. Cong, & H. Peng, "CIST System for CL-SciSumm 2016 Shared Task", in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, Newark, NJ, 2016, pp. 156-167.

[32] L. Li, Y. Zhang, L. Mao, J. Chi, M. Chen, & Z. Huang, "CIST@CLSciSumm-17: Multiple Features Based Citation Linkage, Classification and Summarization", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 43-54.

[33] D. Lippman, *Math in Society*, 2017. [Online]. Available: http://www.opentextbookstore.com/mathinsociety/.

[34] H. Liu, & R. Setiono, "Chi2: Feature Selection and Discretization of Numeric Attributes", in *Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 1995)*, Herndon, VA, 1995, pp. 338-391.

[35] S. Ma, J. Xu, J. Wang, & C. Zhang, "NJUST @ CLSciSumm-17", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 16-25.

[36] B. Malenfant, & G. Lapalme, "RALI System Description for CL-SciSumm 2016 Shared Task", in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, Newark, NJ, 2016, pp. 146-155.

[37] G. A. Miller, "WordNet: A Lexical Database for English". *Communications of the ACM*, Vol. 38, No. 11, 1995, pp. 39-41.

[38] S. Mohammad, B. Dorr, M. Egan, A. Hassan, P. Muthukrishan, V. Qazvinian, D. Radev, & D. Zajic, "Using Citations to Generate Surveys of Scientific Paradigms", in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2009)*, Boulder, CO, 2009, pp. 584-592.

[39] L. Moraes, S. Baki, R. Verma, & D. Lee, "University of Houston at CL-SciSumm 2016: SVMs with Tree Kernels and Sentence Similarity", in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, Newark, NJ, 2016, pp. 113-121.

[40] P. I. Nakov, A. S. Schwartz, & M. A. Hearst, "Citances: Citation Sentences for Semantic Analysis of Bioscience Text", in *Proceedings of the SIGIR 2004 Workshop on Search and Discovery in Bioinformatics*, Sheffield, UK, 2004, pp. 81-88.

[41] T. Nomoto, "NEAL: A Neurally Enhanced Approach to Linking Citation and Reference", in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, Newark, NJ, 2016, pp. 168-174.

[42] A. Pramanick, S. Mandi, M. Dey, & D. Das, "SciSumm 2017: Employing Word Vectors for Identifying, Classifying and Summarizing Scientific Documents", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 94-98.

[43] A. Prasad, "WING-NUS at CL-SciSumm 2017: Learning from Syntactic and Semantic Similarity for Citation Contextualization", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 26-32.

[44] W. H. Press, S. A. Teukolsky, W. T. Vetterling, & B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[45] V. Qazvinian, & D. R. Radev, "Scientific Paper Summarization Using Citation Summary Networks", in *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, Manchester, UK, 2008, pp. 689-696.

[46] V. Qazvinian, & D. R. Radev, "Identifying Non-Explicit Citing Sentences for Citation-Based Summarization", in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, Uppsala, Sweden, 2010, pp. 555-564.

[47] V. Qazvinian, D. R. Radev, S. M. Mohammad, B. Dorr, D. Zajic, M. Whidby, & T. Moon, "Generating Extractive Summaries of Scientific Paradigms". *Journal of Artificial Intelligence Research*, Vol. 46, 2013, pp. 165-201.

[48] J. R. Quinlan, "Induction of Decision Trees". *Machine Learning*, Vol. 1, No. 1, 1986, pp. 81-106.

[49] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1992.

[50] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, & A. Napolitano, "A Comparative Study of Data Sampling and Cost Sensitive Learning", in *Proceedings of the 2008 IEEE International Conference on Data Mining Workshops (ICDM Workshops 2008)*, Pisa, Italy, 2008, pp. 46-52.

[51] Teufel S., & Moens M., "Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status". *Computational Linguistics*, Vol. 28, No. 4, 2002, pp. 409-445.

[52] P. Wang, S. Li, T. Wang, H. Zhou, & J. Tang, "NUDT @ CLSciSumm-18", in *Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018)*, Ann Arbor, MI, 2018, pp. 102-113.

[53] J.-Y. Yeh, T.-Y. Hsu, C.-J. Tsai, P.-C. Cheng, & J.-Y. Lin, "On Identifying Cited Texts for Citances and Classifying Their Discourse Facets by Classification Techniques". *Journal of Information Science and Engineering*, Vol. 35, No. 1, 2019, pp. 61-86.

[54] D. Zhang, & S. Li, "PKU @ CLSciSumm-17: Citation Contextualization", in *Proceedings of the 3rd Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*, Tokyo, Japan, 2017, pp. 86-93.