JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 35, 61-86 (2019)
DOI: 10.6688/JISE.201901_35(1).0004

On Identifying Cited Texts for Citances and Classifying Their Discourse Facets by Classification Techniques*

JEN-YUAN YEH[1,+], TIEN-YU HSU[1], CHENG-JUNG TSAI[2],
PEI-CHENG CHENG[3] AND JUNG-YI LIN[4]
[1]*Department of Operation, Visitor Service, Collection and Information Management*
National Museum of Natural Science
Taichung, 404 Taiwan
E-mail: {jenyuan; dan}@mail.nmns.edu.tw
[2]*Department of Mathematics*
National Changhua University of Education
Changhua, 500 Taiwan
E-mail: cjtsai@cc.ncue.edu.tw
[3]*Department of Information Management*
Chien Hsin University of Science and Technology
Taoyuan, 320 Taiwan
E-mail: pccheng@uch.edu.tw
[4]*AI Lab, Semiconductor Business Group*
Hon-Hai Technology Group (Foxconn)
Taipei, 114 Taiwan
E-mail: jungyilin@gmail.com

Creating the faceted citation summary of a research paper involves identifying cited texts for citation sentences (*i.e.*, citances), classifying their discourse facets, and generating a structured summary from the cited texts. This paper proposes a supervised method for the first two tasks by classification techniques. The first task uses binary classification to distinguish relevant pairs of citances and reference sentences from irrelevant pairs. The second task applies multi-class classification to assign one of the predefined discourse facets to relevant pairs of the first task. The proposed method is evaluated using the CL-SciSumm 2016 datasets and found to be competitive in producing superior results compared to state-of-the-art methods.

Keywords: citation analysis, citation linkage identification, discourse facet classification, binary/multi-class classification, scientific paper summarization

1. INTRODUCTION

The explosive growth of scientific publications nowadays has created an acute need for scientific paper summarization. With the large volume of research papers, presenting a researcher with a summary greatly facilitates his/her research work. Such a summary can be [16]: an abstract (*i.e.*, the author's own summary), a faceted summary that captures all aspects of a paper, or a citation summary (*a.k.a.* the community created sum-

Received August 6, 2017; revised October 17 & December 2, 2017; accepted December 5, 2017.
Communicated by Hsin-Hsi Chen.
* This paper is an extended version of the ICSCA 2017 paper "Reference scope identification for citances by classification with text similarity measures" [55].
* This work is supported by the Ministry of Science and Technology (MOST), Taiwan (Grant number: MOST 104-2221-E-178-001).
+ Corresponding author: Tel: +886 4 23226940 ext. 728; fax: +886 4 23222621.

mary) comprising the set of citation sentences (*i.e.*, citances [43]) referring to a paper. More recently, many efforts have been devoted to automating the synthesis and updating automatic summaries of scientific papers, *e.g.*, [1, 16, 20, 26, 41, 47, 48, 54].

The citation summary is important because it reflects different perspectives on the same reference paper [24]. As noted in [25], a citation summary provides a contextual, interpretative layer to the cited texts. However, it does not consider the context of the target user, verify the claim of the citation, nor provide the context from the reference paper, in terms of the type of information cited or where it is in the reference paper. To address these drawbacks, a promising direction has emerged for creating a faceted citation summary by grouping all cited texts and citances together by facets (*e.g.*, the goal of the paper, the method, the results obtained, or the conclusions of the paper). Fig. 1 defines the task of creating the faceted citation summary for a reference paper [24, 25].

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 (*i.e.*, 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 facet of the paper it belongs to, from a predefined set of facets.
Task 2: Finally, generate a structured summary of the RP from the cited text spans of the RP.

Fig. 1. The task of creating the faceted citation summary of a reference paper [24, 25].

This paper proposes a supervised method for Task 1A and Task 1B in Fig. 1 via classification techniques. The method models instances (*i.e.*, pairs of citances and reference sentences) as vectors of features. Citation-dependent and citation-independent features are explored, including lexical, knowledge-based, corpus-based, syntactic, and surface features. Several representative classification algorithms are investigated, namely, *k*-Nearest Neighbors, Decision Tree, Logistic Regression, Support Vector Machine, Naïve Bayes, and Random Forest. Majority Voting is also exploited to combine multiple classifiers. Task 1A uses binary classification to distinguish relevant instances from irrelevant instances. In addition, a selection strategy is developed to refine the output by excluding incorrectly classified instances. Task 1B applies multi-class classification with the one-against-one reduction strategy to assign one of the predefined discourse facets to relevant instances of the first task.

The main contributions of this study are summarized as below:

(1) A classification-based method that casts Task 1A and Task 1B as binary classification and multi-class classification problems is proposed. A selection strategy is also developed, and it significantly improves performance in Task 1A.
(2) For feature extraction, this study explores a wide spectrum of citation-dependent and citation-independent features. Some are new, some directly adopt text similarity measures in the literature, and some adopt the measures with modifications. No related studies have evaluated in entirety as we have done.
(3) A systematic comparison of feasibility and performance of several representative

classification algorithms for Task 1A and Task 1B is provided.

(4) For the class imbalance problem that classification-based related studies also suffer, the use of SMOTE [12] to introduce biases towards the minority is a new approach.

(5) The proposed method is evaluated in a case study with the CL-SciSumm 2016 datasets. Comprehensive experimental studies and in-depth discussions are given from various perspectives regarding the design and the effectiveness of the method.

The rest of this paper is organized as follows. Section 2 briefly reviews some related work. Section 3 describes technical details on the proposed classification-based method. The evaluation results are presented in Section 4 and Section 5 discusses the method. Finally, Section 6 concludes this paper and points out future related work.

2. RELATED WORK

For Task 1A, most related work identifies the best-matching cited texts for citances following the intuition that a citance and the cited texts to which it refers share some similarity. In [2], three methods, namely, word classification, sequence labeling, and segment classification, are compared, and segment classification is found achieving the best performance. In [42], two approaches are investigated: tf-idf cosine similarity with multiple incremental modifications, and Support Vector Machine (SVM) [15] with subset tree kernel (a convolutional kernel on trees). In [32], different combination methods and strategies (*e.g.*, voting, Jaccard focused, and Jaccard cascade) based on various feature rules of different lexicons and similarities are used; meanwhile, SVM is also tried. In [44], each reference sentence obtains a score from a hybrid model consisting of the tf-idf cosine similarity and the similarity predicted by a single-layer neural network. Sentences are then selected via diversity-based reranking. In [11], the task is cast as a ranking problem modelled by Ranking SVM [27]. A reference paper is dismantled into n-sentence chunks (n = 1, …, 4), and the top n-sentence chunks relevant to every citance are chosen. In [29], three methods are developed. The first method applies a modified variant of TextSentenceRank [53] to incorporate the similarity of reference sentences to the citance on a textual level. The second method employs Random Forest [9] to select from the candidates extracted by the original TextSentenceRank. The third method uses Random Forest to identify the relevant sub-parts of the reference paper, and applies the original TextSentenceRank on each sub-part to extract cited texts. In [3], a rule-based method based on lexical and syntactic dependency cues is introduced. In [35], the subset of reference sentences that have the same facet as the citance are extracted. Then, the bi-directional similarity function [39] is applied to identify from the subset the most similar sentence to the citance.

Regarding Task 1B, rule-based and classification-based methods have been explored. In [32], feature rules and an SVM-based method are examined. Further, voting and fusion methods are used to combine different candidate results. In [11], Decision Tree [49] is applied with the tf-idf vector of the citance as features. In [3], a rule-based method that uses cues of section headers to identify facets is developed. In [35], the word distribution is built as a profile for each facet in citances and reference spans. The facet score for a citance (or a reference sentence) is measured by adding the profile of each constituent word, and the facet of the highest score is assigned. In [52], the task is tack-

led by SVM with a polynomial kernel using position, semantic similarity, and rhetorical features.

3. PROPOSED METHOD

For a citance, the goal is to retrieve as many relevant reference sentences from the reference paper as possible based on the intuition that when a citation relation exists between a citance and the cited text spans, they share similarity in meaning. The proposed method is designed in a supervised manner for the reason that machine learning is capable of automatically building precise models that can analyze more complex data and deliver more accurate results. Fig. 2 illustrates the proposed classification-based method with Task 1A on the left-hand side and Task 1B on the right-hand side.

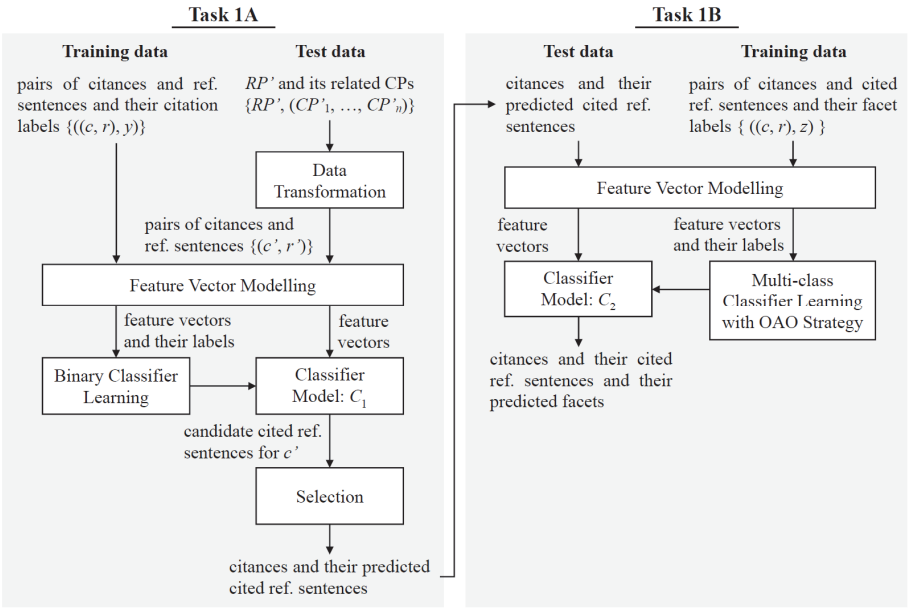

Fig. 2. Overview of the proposed classification-based method.

(A) The proposed method for Task 1A

The process comprises training and testing phases. First, the training phase is introduced. The training data is a set of pairs $\{((c, r), y)\}$, in which 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 (*i.e.*, r is the cited reference sentence of c) if y is labelled as "citation" or a non-citation instance (*i.e.*, r is not the cited reference sentence of c) if y is labelled as "non-citation". Each (c, r) is represented as a feature vector by *feature vector modelling*. The inputs to *binary classifier learning* consist of training instances and their feature vectors and citation labels. The output is a binary classifier C_1; $C_1(c, r) = y$. In the testing phase, the inputs of a new reference paper RP' and its related citing papers are converted by *data transformation* into pairs of citances and reference sentences $\{(c', r')\}$. Each (c', r') is modelled as a feature vector and C_1 predicts its citation label. For each

citance c', the reference sentences classified as its cited reference sentences form the candidate output. Finally, *selection* chooses those candidates with a degree of relatedness to c' greater than the predefined threshold as the output.

(B) The proposed method for Task 1B

The training data is a set of pairs $\{((c, r), z)\}$, where c is a citance, r is a cited reference sentence of c, and $z \in \{Aim\_Citation, Hypothesis\_Citation, Implication\_Citation, Method\_Citation, Results\_Citation\}$. The training data is modelled as feature vectors by *feature vector modelling* and inputted to *multi-class classifier learning* to build a multi-class classifier C_2. The learning process is reduced to multiple binary classification tasks via the one-against-one (OAO) reduction strategy. For the testing phase, the input data is the output of Task 1A, *i.e.*, $\{((c', r'), y) \mid y = citation\}$. For each (c', r'), C_2 takes its feature vector and assigns to it a proper discourse facet label.

3.1 Data Transformation (for Task 1A only)

Data transformation breaks down the test data into pairs of citances and reference sentences, which is the information unit that our method processes. RP' is divided into sentences $\{r' \mid r' \in RP'\}$. For a citance c', pairing each r' and c' forms the output of data transformation, *i.e.*, $\{(c', r') \mid r' \in RP'\}$. The transformation process is also performed on the *raw* training data, which consists of research papers with each coming with its related citing papers, to obtain the training data, *i.e.*, $\{((c, r), y)\}$ in Fig. 2, where each (c, r) is associated with its citation label y.

3.2 Feature Vector Modelling

Linguistic processing is first carried out using Stanford CoreNLP [37]. For each text segment (*i.e.*, citance c or reference sentence r), a tokenizer divides it into tokens (roughly words). A tagger assigns parts of speech to each word, and the base forms of words are determined by a morphological analyzer. A named entity recognizer labels sequences of words as names of organizations, people, locations, *etc.* and normalizes dates, times, and numeric quantities. Finally, a parser marks up the structure of the text in terms of phrases and syntactic dependencies.

The modelling of (c, r) depends on both citation-dependent and citation-independent features. Citation-dependent features evaluate the degree of citation relation between c and r by text similarity measures. Due to limited surface information in short texts, semantic information and syntactic information derived from deep linguistic processing are also taken into account. By contrast, citation-independent features only focus on assessing the significance of r. With the feature set F comprising an $|F|$-dimensional vector space, (c, r) is modelled as a vector of numerical features. $(c, r) = \langle x_1, x_2, \ldots, x_{|F|} \rangle$, where there is a feature extraction function ϕ_i and $\phi_i(c, r) = x_i$ w.r.t. feature f_i. Given citance c, x_i is normalized via min-max normalization over all reference sentences.

Five families of features are proposed: *lexical*, *knowledge-based*, *corpus-based*, *syntactic*, and *surface features*. The first four families are citation-dependent, while the last one is citation-independent. The following notations are used. $|c|$ is the number of words in c, $|c \cap r|$ is the size of the intersection of words in c and r, $|c \cup r|$ is the size of

the union of words in c and r, $tf_{w,c}$ is the number of times that word w appears in c, df_w is the number of documents that word w appears in, and N is the total number of documents in the collection. Note that df_w and N target documents and sentences, respectively, when idf (inverse document frequency) and isf (inverse sentence frequency) are considered.

(A) Lexical features

Features in this family measure the relatedness between c and r based on words shared by them and the occurrence statistics of words.

- **Word overlap:** The overlap of words between c and r is measured. Five word overlap measures as in Eqs. (1)-(5) are adopted: *matching coefficient*, *Dice coefficient*, *Jaccard coefficient*, *Overlap coefficient* and *cosine* (see [36]).

$$\phi(c,r) = |c \cap r| \text{ for matching coefficient} \tag{1}$$

$$\phi(c,r) = \frac{2 \times |c \cap r|}{|c| + |r|} \text{ for Dice coefficient} \tag{2}$$

$$\phi(c,r) = \frac{|c \cap r|}{|c \cup r|} \text{ for Jaccard coefficient} \tag{3}$$

$$\phi(c,r) = \frac{|c \cap r|}{\min(|c|,|r|)} \text{ for Overlap coefficient} \tag{4}$$

$$\phi(c,r) = \frac{|c \cap r|}{\sqrt{|c| \times |r|}} \text{ for cosine} \tag{5}$$

Additionally, two variations of Overlap coefficient developed in [38] are used, where one measures the proportion of words in c that also appears in r, and the other, as given in Eq. (6), further weights the overlap by the summation of idf (or isf) of words.

$$\phi(c,r) = \frac{|c \cap r|}{|c|} \left(\sum_{w \in c \cap r} \log \frac{N}{df_w} \right) \tag{6}$$

- **tf-idf measure:** The tf-idf (or tf-isf) cosine similarity [7] between c and r is computed. Eq. (7), a variant in [5] for detecting topically similar sentences, is also considered.

$$\phi(c,r) = \sum_{w \in c \cap r} \log(tf_{w,c} + 1) \log(tf_{w,r} + 1) \log\left(\frac{N+1}{df_w + 0.5}\right) \tag{7}$$

- **Identity measure:** The identity measure identifies co-derivative documents based on the intuition that similar documents have similar numbers of word occurrences [38]. It is applied to compare co-derivation between c and r, as denoted by

$$\phi(c,r) = \frac{1}{1 + \frac{\max(|c|,|r|)}{\min(|c|,|r|)}} \sum_{w \in c \cap r} \frac{\log(N / df_w)}{1 + |tf_{w,c} - tf_{w,r}|}. \tag{8}$$

- **ROUGE score:** ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [34] is employed to count as relevance the number of overlapping units, *e.g.*, *n*-grams, word sequences, and word pairs, between *c* and *r*. The selected ROUGE metrics include ROUGE-N (*N* = 1, …, 4), ROUGE-L, ROUGE-W, and ROUGE-SU4. ROUGE-N measures *n*-gram based co-occurrence statistics; ROUGE-L considers longest common subsequence (LCS) based statistics; ROUGE-W extends ROUGE-L to assign credit to consecutive in-sequence matches; and ROUGE-SU4 reflects skip-bigram with a maximum skip distance of 4 plus unigram-based co-occurrence statistics. For each metric, three scores are produced: recall, precision, and F-measure.
- **Named entity overlap:** The overlap of named entities (NEs) between *c* and *r*, measured by Jaccard and Dice coefficients. Another measure is also examined as the ratio of pairs of identical NEs in *c* and *r* to all pairs of NEs in *c* and *r*.
- **Number overlap:** The overlap of numbers between *c* and *r*, measured by Jaccard and Dice coefficients.
- **Discriminative degree of citation-related word pairs:** Word pair (w_1, w_2), in which $w_1 \in c$ and $w_2 \in r$, is viewed as a citation-related word pair if (*c*, *r*) is a citation instance. A lexicon of significant citation-related word pairs is built by collecting from the training data those pairs with high discriminative degree, implying the corresponding word pairs statistically tend to appear in citation instances. The discriminative degree of (w_1, w_2) is determined by chi-square test (χ^2), pointwise mutual information (PMI) [13], and normalized PMI (NPMI) [8].[1] Two versions of this feature are defined. One counts the number of significant citation-related word pairs in *c* and *r*, and the other adds up the discriminative degree of significant citation-related word pairs in *c* and *r*.

(B) Knowledge-based features

Features in this family measure the relatedness between *c* and *r* by considering linguistic knowledge derived from WordNet [17].

- **WordNet-based semantic similarity:** Eq. (9) shows the scoring function (adapted from [39]) for calculating the semantic similarity between *c* and *r*:

$$\phi(c,r) = 0.5 \times sim(c,r) + 0.5 \times sim(r,c) \tag{9}$$

where $sim(T_1,T_2) = \sum_{w_1 \in T_1} (\max_{w_2 \in T_2} sim(w_1,w_2) \times specificity(w_1)) \Big/ \sum_{w_1 \in T_1} specificity(w_1)$ and *specificity*(w_1) = idf_{w_1}. In this study, six WordNet-based word similarity measures are applied, including *jcn*, *lch*, *lin*, *path*, *res*, and *wup* (see [45]). Below a short description for each of these six measures is provided. Note that all measures are defined between two concepts ct_1 and ct_2, *i.e.*, word senses of two words w_1 and w_2.

The path measure is equal to the inverse of minlen(ct_1, ct_2), *i.e.*, the shortest path length between ct_1 and ct_2:

$$sim_{path}(w_1,w_2) = \frac{1}{\text{minlen}(ct_1,ct_2)}. \tag{10}$$

[1] A threshold is set for each metric to exclude insignificant word pairs. It is decided as 3.841 for χ^2 (considering $p = 0.05$ with 1 degree of freedom) and 0.0 for both PMI and NPMI. Also note that the same threshold values are set for two surface features, namely, discriminative degree of citation-related words and discriminative degree of facet-related words.

Eq. (11) defines the lch measure:

$$sim_{lch}(w_1, w_2) = -\log \frac{\text{minlen}(ct_1, ct_2)}{2 \times D} \tag{11}$$

where D is the maximum path length in the WordNet taxonomy. The wup measure as in Eq. (12) measures the depth of ct_1 and ct_2 in the taxonomy in relation to their least common subsummer (LCS).[2]

$$sim_{wup}(w_1, w_2) = \frac{2 \times \text{depth}(\text{LCS}(ct_1, ct_2))}{\text{depth}(ct_1) + \text{depth}(ct_2)} \tag{12}$$

The res measure relies on the information content (IC) of the LCS of ct_1 and ct_2. It is determined by Eq. (13):

$$sim_{res}(w_1, w_2) = \text{IC}(\text{LCS}(ct_1, ct_2)). \tag{13}$$

Here, IC is the negative log of the probability of a concept occurring in a corpus. Finally, the lin and jcn measures are two extensions of the res measure that incorporate the IC of the individual concepts:

$$sim_{lin}(w_1, w_2) = \frac{2 \times \text{IC}(\text{LCS}(ct_1, ct_2))}{\text{IC}(ct_1) + \text{IC}(ct_2)}, \tag{14}$$

$$sim_{jcn}(w_1, w_2) = \frac{1}{\text{IC}(ct_1) + \text{IC}(ct_2) - 2 \times \text{IC}(\text{LCS}(ct_1, ct_2))}. \tag{15}$$

- **ADW semantic similarity:** ADW (Align, disambiguate, and walk) [46] is a unified WordNet-based method for measuring semantic similarity of lexical items based on sense-based semantic signatures. Three signature comparison measures, namely, Cosine Similarity, Weighted Overlap, and Top-*k* Jaccard, are introduced. This study chooses Weighted Overlap, as in Eq. (16), for its observed performance in [46].

$$\phi(c, r) = \sum_{i=1}^{|S|} (rank_i^c + rank_i^r)^{-1} \Big/ \sum_{i=1}^{|S|} (2i)^{-1} \tag{16}$$

where S denotes the intersection of all senses with non-zero probability in both signatures of c and r, and $rank_i^j$ denotes the rank of sense $s_i \in S$ in signature j.

- **WordNet-based lexical overlap:** This is similar to WordNet-based semantic similarity, but $sim(T_1, T_2)$ in Eq. (9) is determined by the greedy and optimal matching methods in [50]. In the greedy matching method, $sim(T_1, T_2) = \sum_{w_1 \in T_1} (\max_{w_2 \in T_2} sim(w_1, w_2)) \Big/ \sum_{w_1 \in T_1} 1.0$, as in Eq. (9) when $specificity(w_1) = 1.0$. For the optimal matching method, by adding dummy words, T_1 and T_2 are of the same length, *i.e.*, $T_1 = \{w_{1,1}, w_{1,2}, \ldots, w_{1,n}\}$ and $T_2 = \{w_{2,1}, w_{2,2}, \ldots, w_{2,n}\}$. The method finds permutation π of $\{1, 2, \ldots, n\}$ such that $\sum_{i=1}^{n} sim(w_{1,i}, w_{2,\pi(i)})$ is maximized. Thus, $sim(T_1, T_2) = \max_{\pi} \sum_{i=1}^{n} sim(w_{1,i}, w_{2,\pi(i)}) \Big/ \sum_{w_1 \in T_1} 1.0$.

(C) Corpus-based features

Features in this family measure the relatedness between c and r by considering cor-

[2] The LCS is the most specific concept that two concepts share as an ancestor [45].

pus statistics for deriving semantic relations between words.

- **LSA-based semantic similarity:** This is similar to WordNet-based semantic similarity, but the word semantic similarity is derived by latent semantic analysis (LSA) [30]. As noted in [39], LSA yields a lower dimensional vector space that allows for a homogeneous representation of words, word sets, and texts by singular value decomposition (SVD) and dimensionality reduction on a word-by-document matrix A. Firstly, SVD decomposes A as $A = UZ_kV^T$, where U and V are column-orthogonal matrices with left and right singular vectors, respectively, in columns, and Z is a diagonal matrix of singular values ($\sigma_1, \ldots, \sigma_k$) sorted in descending order. Dimensionality reduction follows to choose k' ($k' \ll k$) and obtains the approximation $A' = UZ_{k'}V^T \simeq A$, in which $Z_{k'}$ represents the latent semantic structure (or the semantic space) extracted from A. The word similarity in the resulting vector space A', in which LSA word vector per column and LSA document vector per row, is then measured by the standard cosine similarity, and is incorporated for $sim(w_1, w_2)$ in Eq. (9).
- **LSA-based lexical overlap:** This is similar to WordNet-based lexical overlap, but the word similarity $sim(w_1, w_2)$ in Eq. (9) is specified by the cosine similarity of LSA vectors of words (see feature *LSA-based semantic similarity* for deriving the word similarity based on LSA).

(D) Syntactic features

Features in this family measure the relatedness between c and r by comparing their syntactic structures obtained by deep linguistic processing.

- **Dependency overlap:** Given sets of dependency relations R_c and R_r for c and r, a relation is a triple (l, h, t) where l is the dependency type, h is the governing lemma, and t is the dependent lemma. The matching of dependency relations between c and r is measured by Jaccard and Dice coefficients, and Simple Dependency Overlap [56], as in Eq. (17).

$$\phi(c,r) = \frac{2 \times |R_c \cap R_r| \times |R_c \| R_r|}{|R_c| + |R_r|} \tag{17}$$

- **Lexico-syntactic subsumption:** In [51], a paraphrase score is introduced as the average of $entscore(T, H)$ and $entscore(H, T)$, where $entscore(T, H)$ assesses the degree of entailment that T entails H. The score is used to quantify the degree of paraphrase between c and r. Briefly, T and H are modelled as lexico-syntactic graphs and a subsumption operation is performed to calculate $entscore(T, H)$:

$$entscore(T,H) = (\alpha \times \frac{\sum_{v_h \in V_H} \max_{v_t \in V_T} \text{match}(v_h, v_t)}{|V_H|} + \beta \times \frac{\sum_{e_h \in E_H} \max_{e_t \in E_T} \text{synt\_match}(e_h, e_t)}{|E_H|} + \gamma) \times \frac{(1 + (-1)^{\#neg\_rel})}{2} \tag{18}$$

where V_T (or V_H) is the set of vertices of the T (or H) graph; E_T (or E_H) is the set of

edges of the T (or H) graph; match(v_h, v_t) denotes word-matching based on synonyms of v_h and v_t; synt_match(e_h, e_t) checks the syntactic relation equivalence between e_h and e_t; and #neg_rel represents the number of negation relations between T and H. Parameters are set ($\alpha = \beta = 0.5$, $\gamma = 0$) according to [51].

- **Word order similarity:** The word order similarity is defined in [33] by the normalized difference of word order in sentences. It is retained to measure how similar the word order in c and r is regarding word sequence and location. See Eq. (19), where WO_c and WO_r are the word order vectors of c and r, respectively.

$$\phi(c,r) = 1 - \frac{\|WO_c - WO_r\|}{\|WO_c + WO_r\|} \tag{19}$$

(E) Surface features

Features in this family are mainly borrowed from text summarization for measuring the significance of r in the reference paper.

- **Sentence length:** The word count of r.
- **Sentence position:** The position of r from the beginning of the reference paper.
- **Similarity with title:** The tf-idf cosine similarity between r and the title of the reference paper.
- **Similarity with first-sentence:** The tf-idf cosine similarity between r and the first sentence of the reference paper.
- **Similarity with context:** The tf-idf cosine similarity between r and its previous (or next) sentence as the context.
- **Similarity with centroid:** The tf-idf cosine similarity between r and the centroid (*i.e.*, the average of all sentence vectors) of the reference paper.
- **TextRank centrality:** TextRank [40] models a document as a graph, in which each node is a sentence and an edge exists between two sentences if there is a similarity relation between them, and applies PageRank [10] to assess the centralities of sentences. The TextRank centrality, as in Eq. (20), is applied to evaluate the significance of r in the reference paper.

$$TRCentrality(r) = \frac{d}{|V|} + (1-d) \sum_{v \in adj[r]} \frac{TRCentrality(v)}{\deg(v)} \tag{20}$$

where V is the set of nodes; d is a damping factor; $adj[r]$ is the set of nodes that are adjacent to r; and deg(v) is the degree of node v.

- **Num. of named entities:** The count of named entities in r. Additionally, the ratio of constituent words of named entities to all words in r is considered.
- **Num. of numbers:** This is similar to num. of named entities, but targeting numbers.
- **Discriminative degree of citation-related words:** For word $w \in r$ and r is a cited reference sentence, w is viewed as a citation-related word. A lexicon of significant citation-related words is built by collecting from the training data those words with high discriminative degree, implying the corresponding words statistically tend to appear in cited reference sentences. The discriminative degree of w is determined by χ^2, PMI [13], and NPMI [8]. Two versions of this feature are applied, where one counts the

number of significant citation-related words in *r*, and the other adds up the discriminative degree of significant citation-related words in *r*.

- **Discriminative degree of facet-related words (for Task 1B only):** Suppose word $w \in r$, in which *r* is a cited reference sentence and the discourse facet of *r* is *d*. Then, *w* is viewed as a facet-related word. A lexicon of significant facet-related words is built by collecting from the training data those words with high discriminative degree, implying the corresponding words statistically tend to appear in cited reference sentences with the discourse facet *d*. The discriminative degree of *w* is determined by χ^2, PMI, and NPMI. Two versions of this feature are used; one is the number of significant facet-related words in *r*, and the other adds up the discriminative degree of significant facet-related words in *r*.

The aforementioned features are implemented in consideration of many factors, *e.g.*, the granularity of information 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. Besides, for some features, several measurements are adopted, and all the different measurements are included in feature vector. For example, lexical feature *word overlap* takes into account a total of seven measurements. Consequently, for an instance, a 343-dimensional feature vector and a 373-dimensional feature vector are produced in Task 1A and Task 1B, respectively.[3]

3.3 Classifier Learning

This study investigates and compares several representative classification algorithms, namely, *k*-Nearest Neighbors (*k*-NN) [4], Decision Tree [49], Logistic Regression [31], Support Vector Machine (SVM) [15],[4] Naïve Bayes [28], and Random Forest [9]. Further, Majority Voting is exploited to combine multiple classifiers. The voting strategy counts the votes of individual classifiers and selects the mode of the classes as the majority decision. It is expected to produce a classifier superior to any of the equally well performing classifiers by balancing out their weaknesses.

The aforementioned classification algorithms are directly used to create binary classifiers in Task 1A. For multi-class classification in Task 1B, this problem decomposes into a set of binary classification tasks that can be solved by these algorithms. Two decomposition methods are widely used, namely, one-against-all (OAA) and one-against-one (OAO). OAA builds *K* binary classifiers for *K* classes. Each is trained with positive samples of a given class and negatives samples of the other classes. For an unseen instance, the class is assigned by the classifier with the highest confidence score. By contrast, OAO builds $K(K-1)/2$ binary classifiers where each is trained on data from pairs of two classes. Voting is performed among the classifiers and the class with the most votes is chosen. This study adopts OAO since previous research, *e.g.*, [6, 22], has noted that OAO is in general more suitable for practical use.

---

[3] In the current implementation, feature vector modelling is time-consuming due to many features needed to be extracted and perhaps poor code efficiency. For limited time and computing resources, feature selection may help suggest subsets of features that are useful to build faster and more cost-effective predictors [18].

[4] This study tests with linear SVM for its faster training and competitive accuracy. By the kernel trick, polynomial classifiers, radial basic function networks, and three-layer sigmoid neural nets can also be learned.

The implementations of the investigated classification algorithms currently rely on Weka [19]. See Appendix A for full details on construction of classifiers.

3.4 Selection (for Task 1A only)

For a new citance c', the learned classifier C_1 is capable of classifying its cited reference sentences. To reduce the number of false positives (*i.e.*, those classified as cited reference sentences but truly non-cited) in the output, a selection strategy is introduced. For each citance c', the reference sentences classified as its cited reference sentences form the candidate output. The final output is created by selecting those candidates with a degree of relatedness to c' greater than the predefined threshold α.[5] Note that the relatedness between a citance and a reference sentence can be assessed by features (see Section 3.2). The current implementation uses tf-idf cosine similarity.[6]

4. EXPERIMENTS

4.1 Datasets

The CL-SciSumm 2016 datasets comprise a training, a development, and a test dataset. Each dataset contains ten topics and each topic consists of a research paper, its citing papers, and three summaries. In each topic, citances are identified by human annotators. Each citance is mapped to its cited texts and tagged with the information facets it represents. Table 1 presents the statistics of the datasets. Task 1A is evidently challen-

Table 1. Statistics of the CL-SciSumm 2016 datasets.

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	1.46	1.25	1.25
Avg. num. of cited reference sentences for a citance	1.84	1.50	1.37
Num. of citation instances	249	329	480
Num. of non-citation instances	27,235	54,704	87,697
Num. of Aim_Citation instances	47	53	39
Num. of Hypothesis_Citation instances	1	14	10
Num. of Method_Citation instances	150	248	350
Num. of Results_Citation instances	34	70	74
Num. of Implication_Citation instances	17	23	67
Num. of instances of more than one discourse facet	0	79	60

[5] An alternative strategy, which ranks candidates according to their relatedness to c' and selects the top-k candidates as the final output, is tried in our earlier paper [55].

[6] Values without normalization are considered. Also note that some classifiers (*e.g.*, Naïve Bayes) produce confidence scores indicating the tendency that an instance is classified. It is suggested to order candidate sentences by confidence scores when they are available.

ging. Take the test dataset as an example, for a citance, the task needs to correctly identify 1.37 cited reference sentences on average from a reference paper with an average length of 229.1 sentences. Moreover, after breaking down a dataset into instances, an imbalanced dataset is produced. For example, the training dataset has 27,235 non-citation instances, far greater than the number of citation instances (*i.e.*, 249). The class imbalance problem also occurs on multi-class classification in Task 1B. Building predictive models using imbalanced data tends to ignore the minority class of more interest and overfit to the majority class. SMOTE (Synthetic Minority Over-sampling Technique) [12] is applied to tackle the class imbalance problem (see Section 4.3).

Fig. 3 illustrates a citation annotation example. *Citation offset* represents the sentence (id: 21) in the citing article (id: C04-1194), *reference offset* indicates the sentences (id: 37) in the reference article (id: E03-1020), and *discourse facet* denotes the facet of the citation. In the test dataset, the information of reference offset, reference text, and discourse facet is not provided and needs to be identified.

Citance Number: 7 | Reference Article: E03-1020.xml | Citing Article: C04-1194.xml | Citation Marker Offset: ['21'] | Citation Marker: Dorow and Widdows, 2003 | **Citation Offset: ['21']** | Citation Text: <S sid ="21" ssid = "21">The last trend, explored by (Véronis, 2003), (Dorow and Widdows, 2003) and (Rapp, 2003), starts from the cooccurrents of a word recorded from a corpus and builds its senses by gathering its cooccurrents according to their similarity or their dissimilarity.</S> | **Reference Offset: ['37']** | Reference Text: <S sid ="37" ssid = "10">To detect the different areas of meaning in our local graphs, we use a cluster algorithm for graphs (Markov clustering, MCL) developed by van Dongen (2000).</S> | **Discourse Facet: Method_Citation** | Annotator: Ankur Khanna, NUS |

Fig. 3. A citation annotation example.

4.2 Evaluation Metrics

Precision, recall and F-measure (*i.e.*, the harmonic mean of precision and recall) are used to evaluate system performance in both Task 1A and Task 1B. For Task 1A, they measure the overlap of sentence IDs between the system output and the gold standard. In Task 1B, the proportion of correctly classified discourse facets by the system are measured, contingent on the expected response of Task 1A. For a topic, Eqs. (21)-(22) and Eqs. (23)-(24) define the evaluation metrics for Task 1A and Task 1B, respectively. The reported scores are the average of those for all topics in the test dataset.[7]

$$\text{Precision}_{1A} = \frac{\sum_c |\text{sysRefOffset}(c) \cap \text{annRefOffset}(c)|}{\sum_c |\text{sysRefOffset}(c)|} \tag{21}$$

$$\text{Recall}_{1A} = \frac{\sum_c |\text{sysRefOffset}(c) \cap \text{annRefOffset}(c)|}{\sum_c |\text{annRefOffset}(c)|} \tag{22}$$

[7] The official evaluation scripts of the CL-SciSumm 2016 are used to avoid result differences caused by different implementations of the evaluation metrics.

where, given a citance c, sysRefOffset(c) is the set of sentence IDs in the Reference Offset field identified by the evaluated system, and annRefOffset(c) is the set of sentence IDs in the Reference Offset field labelled by the human annotators.

$$\text{Precision}_{1B} = \frac{\sum_c |\text{sysFacet}(c) \bigcap \text{annFacet}(c)|}{\sum_c |\text{sysFacet}(c)|} \tag{23}$$

$$\text{Recall}_{1B} = \frac{\sum_c |\text{sysFacet}(c) \bigcap \text{annFacet}(c)|}{\sum_c |\text{annFacet}(c)|} \tag{24}$$

where, given a citance c, sysFacet(c) is the set of facets in the Discourse Facet field identified by the evaluated system, and annFacet(c) is the set facets in the Discourse Facet field labelled by the human annotators.[8]

4.3 Experimental Settings

In order to learn an effective classifier using more training data, the training and development datasets are merged. Synthetic minor class instances are created using SMOTE [12] to introduce biases towards the minority. Currently, 10 minor class nearest neighbors are used to generate synthetic instances until the number of instances per class is equal. In Task 1A, 81,361 synthetic citation instances are produced. The resulting dataset has 163,878 instances in total (half per class). In Task 1B, 1,333 synthetic instances are produced, including 298 Aim_Citation instances, 383 Hypothesis_Citation instances, 358 Implication_Citation instances, and 294 Results_Citation instances. The resulting dataset has 1,990 instances in total (358 instances per class). The two new datasets are used to build classifiers. Each is divided into five folds using stratified sampling. Various model settings in Appendix A are examined by 5-fold cross-validation. The selection threshold α is estimated via one-dimensional grid search. For each classifier, F-measure scores of different model settings are compared via statistical significance testing, and the best setting is selected. Finally, the classifiers with the best model settings are trained on the new training datasets and used to predict the test dataset following the guidelines of Task 1A and Task 1B.

Tables 2 and 3 list the best model parameters. Parameters not presented are unchanged as in Appendix A, and the top k models with the best F-measure are selected to conduct Majority Voting. Furthermore, two basic baselines, Random and OneR [21], and two advanced baselines, DecisionStump [23] and JRip (*i.e.*, RIPPER) [14], are tested for comparisons. Note that given a citance, a.Random randomly extracts a number (no more than 5) of reference sentences as its cited reference sentences, and b.Random randomly assigns a discourse facet to each cited reference sentence. The reported results are averaged from 10 random runs. Finally, in Task 1A, a simple unsupervised method, a.CosineSim, which outputs reference sentences that have values of tf-idf cosine similarity greater than α, is also compared.

[8] According to Eqs. (23) and (24), the connection between the outputs of Task 1A and Task 1B is not taken into account. That is, the performance of Task 1B is evaluated based on the output facets no matter the predicted sentences in Task 1A are correct or incorrect. It is worthy of investigation to find more proper evaluation methods and metrics to reflect the fact that the error of Task 1A would propagate to Task 1B.

Table 2. Settings of model parameters for Task 1A.

Model	Parameter Setting	
	Classifier Parameters	α value
Baseline: a.Random	N/A	N/A
Baseline: a.DecisionStump	Weka's Default	0.0582
Baseline: a.JRip	Weka's Default	0.0383
Baseline: a.OneR	Weka's Default	0.0381
Baseline: a.CosineSim	N/A	0.0806
a.IBk	k=1	0.0529
a.J48	confidenceFactor=0.25, reducedErrorPruning=false	0.0384
a.Logistic	useConjungateGradientDescent=true	0.0455
a.L1-SVM	SVMType=L2-regularized L1-loss support vector classification (dual), cost=0.5 (*i.e.*, 2^−1)	0.0455
a.L2-SVM	SVMType=L2-regularized L2-loss support vector classification (dual), cost=1.0 (*i.e.*, 2^0)	0.0455
a.LR-SVM	SVMType=L2-regularized logistic regression (dual), cost=22.627417 (*i.e.*, 2^4.5)	0.0455
a.NaïveBayes	useKernelEstimator=true	0.0782
a.RandomForest	numFeatures=19 (≈sqrt(343)), numIterations=500	0.0580
a.VotingTop3	Majority voting of a.L1-SVM, a.L2-SVM, and a.Logistic	0.0455
a.VotingTop5	Majority voting of a.L1-SVM, a.L2-SVM, a.LR-SVM, a.Logistic, and a.RandomForest	0.0455

Table 3. Settings of model parameters for Task 1B.

Model	Parameter Setting
Baseline: b.Random	N/A
Baseline: b.DecisionStump	Weka's Default
Baseline: b.JRip	Weka's Default
Baseline: b.OneR	Weka's Default
b.IBk	k=4
b.J48	reducedErrorPruning=true
b.Logistic	useConjungateGradientDescent=true
b.L1-SVM	SVMType=L2-regularized L1-loss support vector classification (dual), cost=0.353553391 (*i.e.*, 2^−1.5)
b.L2-SVM	SVMType=L2-regularized L2-loss support vector classification (dual), cost=0.25 (*i.e.*, 2^−2)
b.LR-SVM	SVMType=L2-regularized logistic regression (dual), cost=2.0 (*i.e.*, 2^1)
b.NaïveBayes	useKernelEstimator=true
b.RandomForest	numFeatures=20 (≈sqrt(373)), numIterations=750
b.VotingTop3	Majority voting of b.LR-SVM, b.Logistic, and b.NaïveBayes
b.VotingTop5	Majority voting of b.L1-SVM, b.LR-SVM, b.Logistic, b.NaïveBayes, and b.RandomForest

4.4 Results

(A) Task 1A results

Table 4 presents Task 1A results and their 95% confidence intervals in brackets.

The ranking of models according to F-measure is also provided. The following parts compare model performance in F-measure. First of all, a.Random has the worst results as expected. Among the baselines, a.DecisionStump and a.JRip produce relatively good results. Most models of the proposed method substantially surpass the baselines, except a.IBk and a.J48. The results for the unsupervised method, a.CosineSim, indicate that our supervised method benefits from classification techniques and delivers more accurate results. Secondly, both a.IBk and a.J48 perform poorly but a.J48 is inferior to a.IBk. a.J48 predicts many false positives, leading to low precision and F-measure. Next, the literature has proven the effectiveness of SVM in classification and regression problems and, as expected, a.L1-SVM, a.L2-SVM, and a.LR-SVM are competitive in producing superior results. Indeed, a.L2-SVM and a.L1-SVM are the best and the second best, respectively. a.Logistic also obtains good results. Overall, models using functional classifiers, including a.L1-SVM, a.L2-SVM, a.LR-SVM, and a.Logistic, outdo the other models.

In addition, a.NaïveBayes produces moderate results. a.NaïveBayes has the highest recall for it predicting more citation instances, raising the chance of hitting correctly classified instances. Furthermore, as an ensemble method, a.RandomForest is supposed to yield excellent results. However, a.RandomForest is ranked 7th with F-measure close to that of a.NaïveBayes. a.RandomForest has the highest precision but relatively low recall. Preliminary analysis finds that a.RandomForest predicts 384 citation instances, much lower than 859.4, the average amount that a.L1-SVM, a.L2-SVM, a.LR-SVM, a.Logistic, and a.NaïveBayes predict. Finally, Majority Voting works competitively, but none of voting-based models show improvements over a.L2-SVM. a.VotingTop3 and a.L2-SVM tie for first place and a.VotingTop5 is ranked 5th. Also, with more classifiers, the voting performance deteriorates, which merits further analysis.

Table 4. Task 1A results of models (best performance bold-faced).

Model	Rank	Precision	Recall	F-measure
Baseline: a.Random	15	0.0048 [±0.0041]	0.0107 [±0.0090]	0.0065 [±0.0056]
Baseline: a.CosineSim	12	0.0642 [±0.0132]	0.3304 [±0.0507]	0.1050 [±0.0183]
Baseline: a.DecisionStump	10	0.1005 [±0.0187]	0.1691 [±0.0561]	0.1159 [±0.0208]
Baseline: a.JRip	9	0.1316 [±0.0322]	0.1295 [±0.0621]	0.1240 [±0.0462]
Baseline: a.OneR	14	0.0874 [±0.0285]	0.0853 [±0.0401]	0.0808 [±0.0294]
a.IBk	11	0.1282 [±0.0324]	0.1125 [±0.0539]	0.1128 [±0.0385]
a.J48	13	0.0847 [±0.0203]	0.1285 [±0.0639]	0.0930 [±0.0294]
a.L1-SVM	**3**	0.1187 [±0.0303]	0.2220 [±0.0633]	0.1456 [±0.0298]
a.L2-SVM	**1**	0.1245 [±0.0349]	0.2217 [±0.0680]	**0.1499 [±0.0348]**
a.LR-SVM	6	0.1177 [±0.0262]	0.2070 [±0.0629]	0.1407 [±0.0258]
a.Logistic	4	0.1212 [±0.0318]	0.2123 [±0.0634]	0.1449 [±0.0307]
a.NaïveBayes	8	0.1002 [±0.0190]	**0.2359 [±0.0320]**	0.1348 [±0.0167]
a.RandomForest	7	**0.1723 [±0.0737]**	0.1568 [±0.0709]	0.1375 [±0.0395]
a.VotingTop3	**1**	0.1246 [±0.0321]	0.2211 [±0.0641]	**0.1499 [±0.0314]**
a.VotingTop5	5	0.1208 [±0.0319]	0.2112 [±0.0634]	0.1443 [±0.0308]

Table 5 shows the official Task 1A results of the top 5 CL-SciSumm 2016 systems, and the results of the median and the worst machine. F-measure scores indicate that eight models of the proposed method significantly outperform the best machine, Sys15$tfidf+ st+sl. They are a.L1-SVM with improvement of 8.82%,[9] a.L2-SVM with improvement

[9] The improvement is calculated as $(b–a)/a*100$ when b is compared to a.

of 12.03%, a.LR-SVM with improvement of 5.16%, a.Logistic with improvement of 8.3%, a. NaïveBayes with improvement of 0.75%, a.RandomForest with improvement of 2.77%, a.VotingTop3 with improvement of 12.03%, and a.VotingTop5 with improvement of 7.85%.

Table 5. Part of the official Task 1A results in the CL-SciSumm 2016.

SYSID	Precision	Recall	F-measure
Sys15$tfidf+st+sl (1/23)	0.0961 [±0.0269]	0.2250 [±0.0777]	0.1338 [±0.0390]
Sys8$Fusion Method (2/23)	0.0828 [±0.0228]	0.2621 [±0.0904]	0.1251 [±0.0358]
Sys8$Jaccard Focused Method (2/23)	0.0828 [±0.0228]	0.2621 [±0.0904]	0.1251 [±0.0358]
Sys8$Voting Method1 (4/23)	0.0763 [±0.0224]	0.2418 [±0.0891]	0.1152 [±0.0352]
Sys8$Voting Method2 (5/23)	0.0706 [±0.0229]	0.2236 [±0.0853]	0.1067 [±0.0354]
Median machine (Sys5$Default)	0.0423 [±0.0230]	0.0349 [±0.0192]	0.0381 [±0.0207]
Worst machine (Sys9$sect-class-tsr)	0.0117 [±0.0146]	0.0063 [±0.0083]	0.0081 [±0.0105]

Table 6. Task 1B results of models (best performance bold-faced).

Model	Rank	Precision	Recall	F-measure
Baseline: a.Random+ b.Random	15	0.0983 [±0.0994]	0.0091 [±0.0100]	0.0164 [±0.0177]
Baseline: a.CosineSim+ b.JRip	5	0.3154 [±0.0461]	0.2355 [±0.0422]	0.2639 [±0.0357]
Baseline: a.DecisionStump+b.JRip	9	0.3383 [±0.0528]	0.1479 [±0.0502]	0.1944 [±0.0456]
Baseline: a.JRip+b.JRip	12	0.3939 [±0.0653]	0.1070 [±0.0551]	0.1583 [±0.0635]
Baseline: a.OneR+b.JRip	14	0.2602 [±0.1016]	0.0648 [±0.0367]	0.0977 [±0.0474]
a.IBk+b.NaïveBayes	13	**0.5600 [±0.1329]**	0.0950 [±0.0394]	0.1543 [±0.0564]
a.J48+b.RandomForest	11	0.4554 [±0.0902]	0.1252 [±0.0610]	0.1820 [±0.0722]
a.L1-SVM+ b.VotingTop5	**2**	0.4624 [±0.0777]	0.2022 [±0.0550]	0.2732 [±0.0588]
a.L2-SVM+ b.VotingTop5	4	0.4471 [±0.0720]	0.1993 [±0.0594]	0.2658 [±0.0640]
a.LR-SVM+ b.VotingTop5	8	0.4515 [±0.0759]	0.1868 [±0.0563]	0.2561 [±0.0596]
a.Logistic+ b.VotingTop5	6	0.4562 [±0.0797]	0.1927 [±0.0581]	0.2629 [±0.0629]
a.NaïveBayes+ b.RandomForest	**1**	0.4481 [±0.0852]	**0.2053 [±0.0449]**	**0.2743 [±0.0495]**
a.RandomForest+ b.RandomForest	10	0.4554 [±0.2031]	0.1250 [±0.0740]	0.1871 [±0.0990]
a.VotingTop3+ b.VotingTop5	**3**	0.4571 [±0.0771]	0.1993 [±0.0562]	0.2691 [±0.0603]
a.VotingTop5+ b.VotingTop5	7	0.4536 [±0.0799]	0.1913 [±0.0581]	0.2612 [±0.0630]

(B) Task 1B results

Table 6 provides Task 1B results. For each Task 1A model, its outputs are inputted into models in Task 1B. Note that baselines are paired with baselines only, except for a.Random and b.Random, which are a pair separately. 113 (1+4×3+10×10=113) combinations are evaluated, and the best results w.r.t. F-measure that each Task 1A model obtains in Task 1B are reported. The following compares model performance in F-measure. First of all, a.Random+b.Random clearly performs the worst. Among the baselines, the combinations with b.JRip produce the best results. However, except for a.CosineSim+b. JRip, which obtains a high F-measure, the other baselines have moderate results. For the proposed method, three models, namely, b.NaïveBayes, b.RandomForest, and b.VotingTop5, are suggested. Six models with b.VotingTop5, including a.L1-SVM+b.Voting Top5, a.L2-SVM+b.VotingTop5, a.LR-SVM+b.VotingTop5, a.VotingTop3+b.VotingTop5, and

a.VotingTop5+b.VotingTop5, produce competitively superior results. Models with b. RandomForest have moderate results except a.NaïveBayes+b.RandomForest, which has the highest recall and F-measure. Finally, although b.NaïveBayes is suggested, the only model, a.IBk+b.NaïveBayes, does not produce satisfactory results except for its precision.

Table 7 lists the official Task 1B results of the top 5 CL-SciSumm 2016 systems. Regarding F-measure, no models of the proposed method surpass the best machine, Sys15$tfidf+st+sl. Our best model, a.NaïveBayes+b.RandomForest, is only superior to Sys8$Voting Method2, with a slight increase of 0.4%. The second and third best models are a.L1-SVM+b.VotingTop5 and a.VotingTop3+b.VotingTop5. The former has the same F-measure as Sys8$Voting Method2, and the latter loses to Sys8$Voting Method2 with a decrease of −1.5%. Table 1 shows that 60 instances belong to more than one discourse facet. However, the proposed method classifies each instance in one discourse facet. For further improvement, multi-label classification is worthy of investigation.

Table 7. Part of the official Task 1B results in the CL-SciSumm 2016.

SYSID	Precision	Recall	F-measure
Sys15$tfidf+st+sl (1/19)	0.9000 [±0.1859]	0.2735 [±0.0928]	0.4073 [±0.1188]
Sys8$Jaccard Focused Method (2/19)	0.5812 [±0.0894]	0.2308 [±0.0954]	0.3143 [±0.0949]
Sys8$Fusion Method (3/19)	0.5319 [±0.0932]	0.2268 [±0.0943]	0.2994 [±0.0887]
Sys8$Voting Method1 (4/19)	0.5717 [±0.1141]	0.2177 [±0.0958]	0.2933 [±0.0904]
Sys8$Voting Method2 (5/19)	0.5971 [±0.1282]	0.1934 [±0.0876]	0.2732 [±0.1005]
Median machine (Sys15$Tkern1-4)	0.5000 [±0.3099]	0.0406 [±0.0334]	0.0730 [±0.0590]
Worst machine (Sys12$Default)	0.1250 [±0.1866]	0.0055 [±0.0077]	0.0105 [±0.0147]

5. DISCUSSION

(A) Effect of selection strategy in Task 1A

Table 8 presents Task 1A results of a.L2-SVM with different selection strategies. Note that the Top-k strategy orders and selects the top k candidates as the output. The k values range from 1 to 5, and NoLimit denotes that all candidates are outputted. Two α values are tested: 0.0500 and 0.0455. The former is equal to 1.5 standard deviations from the mean of degree of relatedness distributed in the test dataset. The latter is decided by cross-validation (see Section 4.3). Top-1 has the highest precision and NoLimit obtains the highest recall. Typically, recall is improved as a larger k is considered since more sentences are outputted, but such cases increase the number of false positives and lower precision, and vice versa for a smaller k. For the Top-k strategy, the highest F-measure occurs for Top-2, which concurs with the statistics of the average number of cited reference sentences for a citance in Table 1. For NoLimit, the recall of 0.4373 indicates that 56.27% cited reference sentences are not identified. The low precision of 0.0236 implies many output sentences are false positives. Overall, the threshold strategy obtains better precision and F-measure. Owing to α, the strategy is capable of outputting various numbers of cited reference sentences for each citance. This makes the proposed method behave similarly to manual annotation. The results also suggest a default value for α as 1.5 standard deviations from the mean of degree of relatedness. Similar phenomena occur for the other models, but are not explicated here due to the limited space.

Table 8. Task 1A results of a.L2-SVM with different selection strategies (best performance bold-faced).[10,11]

Selection strategy		Rank	Precision	Recall	F-measure
Top-k	1	5	**0.1330 [±0.0488]**	0.1038 [±0.0410]	0.1161 [±0.0445]
	2	**2**	0.1168 [±0.0287]	0.1892 [±0.0567]	0.1437 [±0.0384]
	3	4	0.0939 [±0.0222]	0.2263 [±0.0661]	0.1320 [±0.0334]
	4	6	0.0752 [±0.0159]	0.2397 [±0.0649]	0.1139 [±0.0258]
	5	7	0.0676 [±0.0148]	0.2671 [±0.0774]	0.1073 [±0.0251]
	NoLimit	8	0.0236 [±0.0031]	**0.4373 [±0.1163]**	0.0438 [±0.0062]
Threshold	0.0500	**3**	0.1206 [±0.0253]	0.2022 [±0.0610]	0.1425 [±0.0291]
α	0.0455	**1**	0.1245 [±0.0349]	0.2217 [±0.0680]	**0.1499 [±0.0348]**

(B) Effect of reduction strategy for multi-class classification in Task 1B

Table 9 lists Task 1B results of the top 3 models in Table 6 with different reduction strategies being applied. It is evident that the OAO strategy surpasses the OAA strategy in all metrics. This concurs with previous research, *e.g.*, [6, 22], which has noted that the OAO strategy is generally more suitable for practical use.

Table 9. Task 1B results of the top 3 models in Table 6 with different reduction strategies (best performance bold-faced).

Reduction strategy	Metrics	Models		
		a.NaiveBayes+ b.RandomForest	a.L1-SVM+ b.VotingTop5	a.VotingTop3+ b.VotingTop5
OAO (1-against-1)	Precision	**0.4481 [±0.0852]**	**0.4624 [±0.0777]**	**0.4571 [±0.0771]**
	Recall	**0.2053 [±0.0449]**	**0.2022 [±0.0550]**	**0.1993 [±0.0562]**
	F-measure	**0.2743 [±0.0495]**	**0.2732 [±0.0588]**	**0.2691 [±0.0603]**
OAA (1-against-ALL)	Precision	0.4182 [±0.0861]	0.4458 [±0.1357]	0.4365 [±0.1380]
	Recall	0.2039 [±0.0440]	0.1765 [±0.0679]	0.1737 [±0.0695]
	F-measure	0.2661 [±0.0477]	0.2431 [±0.0795]	0.2386 [±0.0821]

(C) Classifier performance in Task 1A without applying the selection strategy

Table 10 provides the classification results of classifiers in Task 1A without applying the selection strategy. Positive means a citation instance and negative means a non-citation instance. The percentage in parentheses stands for the proportion of the total. For example, L2-SVM identifies 181 true positives, *i.e.*, 37.71% of 480 positive instances in the test dataset (see Table 1). Apparently, most classifiers identify small numbers of true positives, but many false positives. This implies the necessity for the selection strategy to reduce the number of false positives in the output. In addition, substantial room for improvement remains due to commonly low classification accuracy of positives.

(D) Comparison between the proposed method and related studies

This work is similar to other classification-based related studies, *e.g.*, [2, 29, 32, 42,

[10] The Top-2 results are slightly different from those in our earlier paper [55]. The previous results are obtained by our own evaluation scripts but the results here are obtained by the official scripts of the CL-SciSumm 2016.

[11] Since the reported F-measure is the average of F-measure scores over all topics and is not the harmonic mean of the average precision and recall, the phenomenon happens that the Top-2 strategy has a higher F-measure than that of the threshold strategy ($\alpha = 0.05$) even though both the precision and recall of the Top-2 strategy are lower than that of the threshold strategy ($\alpha = 0.05$).

Table 10. Classifier performance in Task 1A without applying the selection strategy.

Classifier	# of true positives	# of false positives	# of true negatives	# of false negatives
IBk	93 (19.38%)	2,982	84,715 (96.60%)	387
J48	108 (22.50%)	6,047	81,650 (93.10%)	372
L1-SVM	183 (38.13%)	8,049	79,648 (90.82%)	297
L2-SVM	181 (37.71%)	7,156	80,541 (91.84%)	299
LR-SVM	178 (37.08%)	7,229	80,468 (91.76%)	302
Logistic	183 (38.13%)	7,271	80,426 (91.71%)	297
NaïveBayes	316 (65.83%)	12,612	75,085 (85.62%)	164
RandomForest	122 (25.42%)	3,263	84,434 (96.28%)	358
VotingTop3	184 (38.33%)	7,327	80,370 (91.65%)	296
VotingTop5	180 (37.50%)	7,232	80,465 (91.75%)	300

44, 52]. However, this work explores a wide spectrum of citation-dependent and citation-independent features which, to the best of our knowledge, no related studies have evaluated in entirety as we have done. In addition, this work investigates and compares the feasibility and performance of several representative classification algorithms for Task 1A and Task 1B. In contrast, most related studies only focus on one classification algorithm. Finally, a few related studies deal with the class imbalance problem by undersampling the majority, *e.g.*, [32, 42], or re-weighting the imbalanced classes, *e.g.*, [32]. Instead, the use of SMOTE [12] in this work to oversample the minority is new.

6. CONCLUSION

This paper proposes a supervised method to identify cited texts for citances and classify their discourse facets using classification techniques. The first task uses binary classification to distinguish relevant pairs of citances and reference sentences from irrelevant pairs. In addition, a selection strategy is developed to refine the output by excluding incorrectly classified instances. The second task applies multi-class classification with the one-against-one reduction strategy to assign one of the predefined discourse facets to relevant pairs of the first task. The method is evaluated using the CL-SciSumm 2016 datasets and found to perform well with competitive results. Compared to the CL-SciSumm 2016 participants, the method is in first place in Task 1A and in fifth place in Task 1B.

There remains room for improvement. First of all, methods of combining classifiers, *e.g.*, bagging, boosting, and stacking, are worthy of investigation. Feature selection may also contribute to classifier improvement in model generalization, prediction performance, and learning efficiency. Furthermore, as mentioned in Section 4.4, multi-label classification is worth trying for Task 1B. Finally, scaling the corpus is important because a more precise classifier can be built using more observed samples.

REFERENCES

1. A. Abu-Jbara and D. Radev, "Coherent citation-based summarization of scientific papers," in *Proceedings of the 49th Annual Meeting of the Association for Computa-*

tional Linguistics: Human Language Technologies, 2011, pp. 500-509.
2. A. Abu-Jbara and D. Radev, "Reference scope identification in citing sentences," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012, pp. 80-90.
3. P. Aggarwal and R. Sharma, "Lexical and syntactic cues to identify reference scope of citance," in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries*, 2016, pp. 103-112.
4. D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, Vol. 6, 1991, pp. 37-66.
5. J. Allan, C. Wade, and A. Bolivar, "Retrieval and novelty detection at the sentence level," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003, pp. 314-321.
6. E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research*, Vol. 1, 2000, pp. 113-141.
7. R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, New York, 1999.
8. G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," in *Proceedings of International Conference of the German Society for Computational Linguistics and Language Technology*, 2009, pp. 31-40.
9. L. Breiman, "Random forests," *Machine Learning*, Vol. 45, 2001, pp. 5-32.
10. S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, Vol. 30, 1998, pp. 107-117.
11. Z. Cao, W. Li, and 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*, 2016, pp. 132-138.
12. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, Vol. 16, 2002, pp. 321-357.
13. K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, Vol. 16, 1990, pp. 22-29.
14. W. W. Cohen, "Fast effective rule induction," in *Proceedings of the 12th International Conference on Machine Learning*, 1995, pp. 115-123.
15. C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, Vol. 20, 1995, pp. 273-297.
16. A. Elkiss, S. Shen, A. Fader, G. Erkan, D. States, and 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, 2008, pp. 51-62.
17. C. Fellbaum, "WordNet(s)," in K. Brown (ed.), *Encyclopedia of Language and Linguistics*, 2nd ed., Elsevier, Oxford, 2005, pp. 665-670.
18. I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 1157-1182.
19. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, Vol. 11, 2009,

pp. 10-18.
20. C. D. V. Hoang and M.-Y. Kan, "Towards automated related work summarization," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, pp. 427-435.
21. R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, Vol. 11, 1993, pp. 63-90.
22. C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, Vol. 13, 2002, pp. 414-425.
23. W. Iba and P. Langley, "Induction of one-level decision trees," in *Proceedings of the 9th International Workshop on Machine Learning*, 1992, pp. 233-240.
24. K. Jaidka, M. K. Chandrasekaran, B. F. Elizalde, R. Jha, C. Jones, M.-Y. Ken, A. Khanna, D. Molla-Aliod, D. R. Radev, F. Ronzano, and H. Saggion, "The computational linguistics summarization pilot task," in *Proceedings of the Text Analysis Conference*, 2014.
25. K. Jaidka, M. K. Chandrasekaran, S. Rustagi, and 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*, 2016, pp. 93-102.
26. K. Jaidka, C. Khoo, and J.-C. Na, "Deconstructing human literature reviews – A framework for multi-document summarization," in *Proceedings of the 14th European Workshop on Natural Language Generation*, 2013, pp. 125-135.
27. T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 133-142.
28. G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 338-345.
29. S. Klampfl, A. Rexha, and 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*, 2016, pp. 122-131.
30. T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, Vol. 25, 1998, pp. 259-284.
31. S. le Cessie and J. C. van Houwelingen, "Ridge estimators in logistic regression," *Applied Statistics*, Vol. 41, 1992, pp. 191-201.
32. L. Li, L. Mao, Y. Zhang, J. Chi, T. Huang, X. Cong, and 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*, 2016, pp. 156-167.
33. Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, 2006, pp. 1138-1150.
34. C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proceedings of the ACL-04 Workshop on Text Summarization Branches Out*, 2004, pp. 74-81.
35. B. Malenfant and 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*, 2016, pp. 146-155.
36. C. D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA, 1999.
37. C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55-60.
38. D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel, "Similarity measures for tracking information flow," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 2005, pp. 517-524.
39. R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006, pp. 775-780.
40. R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 404-411.
41. S. Mohammad, B. Dorr, M. Egan, A. Hassan, P. Muthukrishan, V. Qazvinian, D. Radev, and D. Zajic, "Using citations to generate surveys of scientific paradigms," in *Proceedings of Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009, pp. 584-592.
42. L. Moraes, S. Baki, R. Verma, and 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*, 2016, pp. 113-121.
43. P. I. Nakov, A. S. Schwartz, and M. Hearst, "Citances: Citation sentences for semantic analysis of bioscience text," in *Proceedings of the SIGIR'14 Workshop on Search and Discovery in Bioinformatics*, 2004, pp. 81-88.
44. 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*, 2016, pp. 168-174.
45. T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity – Measuring the relatedness of concepts," in *Proceedings of the 19th National Conference on Artificial Intelligence*, 2004, pp. 1024-1025.
46. M. T. Pilehvar, D. Jurgens, and R. Navigli, "Align, disambiguate and walk: A unified approach for measuring semantic similarity," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013, pp. 1341-1351.
47. V. Qazvinian and D. R. Radev, "Scientific paper summarization using citation summary networks," in *Proceedings of the 22nd International Conference on Computational Linguistics*, 2008, pp. 689-696.
48. V. Qazvinian and 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*, 2010, pp. 555-564.

49. J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993.
50. V. Rus and M. Lintean, "A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics," in *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*, 2012, pp. 157-162.
51. V. Rus, P. M. McCarthy, M. C. Lintean, D. S. McNamara, and A. C. Graesser, "Paraphrase identification with lexico-syntactic graph subsumption," in *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference*, 2008, pp. 201-206.
52. H. Saggion, A. AbuRa'ed, and F. Ronzano, "Trainable citation-enhanced summarization of scientific articles," in *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries*, 2016, pp. 175-186.
53. C. Seifert, E. Ulbrich, R. Kern, and M. Granitzer, "Text representation for efficient document annotation," *Journal of Universal Computer Science*, Vol. 19, 2013, pp. 383-405.
54. S. Teufel and M. Moens, "Summarizing scientific articles: Experiments with relevance and rhetorical status," *Computational Linguistics*, Vol. 28, 2002, pp. 409-445.
55. J.-Y. Yeh, T.-Y. Hsu, C.-J. Tsai, and P.-C. Cheng, "Reference scope identification for citances by classification with text similarity measures," in *Proceedings of the 6th International Conference on Software and Computer Applications*, 2017, pp. 87-91.
56. T. T. Zhu and M. Lan, "ECNUCS: Measuring short text semantic equivalence using multiple similarity measurements," in *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, 2013, pp. 124-131.

Jen-Yuan Yeh (葉鎮源) is currently an Assistant Researcher of the Department of Operation, Visitor Service, Collection and Information Management at the National Museum of Natural Science. His research interests include text mining and summarization, information retrieval and extraction, digital libraries and museums, and natural language processing.

Tien-Yu Hsu (徐典裕) is currently a Research Fellow in Department of Operation, Visitor Service, Collection and Information at National Museum of Natural Science, Taiwan. He is also a Professor in Graduate Institute of Library and Information Science at National Chung Hsing University, Taiwan. His work and research interests are related to digital archive, smart museum, smart education, museum learning, and knowledge management.

Cheng-Jung Tsai (蔡政容) is currently an Associate Professor in the Department of Mathematics, Graduate Institute of Statistics and Information Science at National Changhua University of Education, Chang-Hua, Taiwan. His research interests include data mining, big data analysis, information security, e-learning and digital image processing.

Pei-Cheng Cheng (鄭培成) is currently an Assistant Professor of the Department of Information Management, Chien Hsin University of Science and Technology. His research interests include information retrieval, data mining, image retrieval, and machine learning.

Jung-Yi Lin (林忠億) Ph.D. is currently an Assistant Manager of AI Lab, SBG, Hon-Hai Technology Group (Foxconn). His research interests include evolutionary computation, machine learning, artificial intelligence, and data mining.

APPENDIX A: LIST OF EVALUATED CLASSIFIERS

For each classification algorithm, a classifier is generated for each combination of the parameters specified below.

Type	Classification algorithm	Name in Weka	Parameter combination
Lazy	*k*-Nearest Neighbors (*k*-NN)	lazy.IBk	k={1, 2, 4, 16, 32, 64}, nearestNeighbourSearchAlgorithm=LinearNNSearch (with distanceFunction=EuclideanDistance)
Bayesian	Naïve Bayes	bayes.NaïveBayes	useKernelEstimator={true, false}
Tree	Decision Tree: C4.5	trees.J48	binarySplits=false, minNumObj=2, unpruned=false, {reducedErrorPruning=true,

			{confidenceFactor={0.25, 0.5, 0.75}, reducedErrorPruning=false}}
	Random Forest	trees.RandomForest	maxDepth=0 (*i.e.*, unlimited), numFeatures=sqrt(#(features)), numIterations={10, 50, 100, 300, 500, 750, 1000, 3000, 5000}
Function	Logistic Regression	functions.Logistic	maxIts=−1(*i.e.*, unlimited), ridge=1.0E−8, useConjungateGradientDescent={true, false}
	Support Vector Machine (SVM)*	functions.LibLINEAR	(1) L1-SVM: SVMType=L2-regularized L1-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 (2) L2-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 (3) LR-SVM: SVMType=L2-regularized logistic regression (dual), bias=1.0, cost={2^−14, 2^−12, …, 2^0, …, 2^12, 2^14}, eps=1.0E−4, epsilonParameter=0.1, maximumNumberOfIterations=50000
Meta	Majority Voting	meta.Vote	combinationRule=Majority Voting (1) Base classifiers for Task 1A: L1-SVM, L2-SVM, LR-SVM, Logistic, RandomForest (2) Base classifiers for Task 1B: L1-SVM, LR-SVM, Logistic, NaïveBayes, RandomForest
	Multi-class Classification	meta.MultiClassClassifier	method=1-against-1, usePairwiseCoupling =true Base classifiers: IBk, J48, L1-SVM, L2-SVM, LR-SVM, Logistic, NaïveBayes, RandomForest

* To find the value of the cost parameter, a coarse search is followed by a finer search to find the best option. For example, if the coarse search finds 2^0 to be the best value, a finer search is conducted on its neighborhood (2^−2, 2^−1.5, 2^−1, 2^−0.5, 2^0, 2^0.5, 2^1, 2^1.5, 2^2).