Ensembling Approaches to Citation Function Classification and Important Citation Screening

Xiaorui Jiang

Centre for Computational Sciences and Mathematical Modeling, Coventry University, United Kingdom,

xiaorui.jiang@coventry.ac.uk

Abstract

Compared to feature engineering, deep learning approaches for citation context analysis have yet fully leveraged the myriad of design options for modeling in-text citation, citation sentence, and citation context. In fact, no single modeling option universally excels on all citation function classes or annotation schemes, which implies the untapped potential for synergizing diverse modeling approaches to further elevate the performance of citation context analysis. Motivated by this insight, the current paper undertook a systematic exploration of ensemble methods for citation context analysis. To achieve a better diverse set of base classifiers, we delved into three sources of classifier diversity, incorporated five diversity measures, and introduced two novel diversity re-ranking methods. Then, we conducted a comprehensive examination of both voting and stacking approaches for constructing classifier ensembles. We also proposed a novel weighting method that considers each individual classifier's performance, resulting in superior voting outcomes. While being simple, voting approaches faced significant challenges in determining the optimal number of base classifiers for combination. Several strategies have been proposed to address this limitation, including meta-classification on base classifiers and utilising deeper ensemble architectures. The latter involved hierarchical voting on a filtered set of meta-classifiers and stacked meta-classification. All proposed methods demonstrate state-of-the-art results on, with the best performances achieving more than 5% and 4% improvements on the 11-class and 6-class schemes of citation function classification and by 3% on important citation screening. The promising empirical results validated the potential of our ensembling approaches for citation context analysis.

Keywords

Citation function classification; Important citation screening; ensemble; majority voting; classifier stacking

1. Introduction

Citation context analysis (Zhang et al., 2013) is an important task in scientific text understanding. A citation context tells the reason for the citing authors to make a citation, i.e., citation function, and how important or relevant the cited paper is, i.e., citation importance, to the citing study. A plethora of studies have been made on machine learning algorithms for citation function classification (Teufel et al., 2006a; Aggarwal et al., 2010; Dong & Schäfer, 2011; Jochim & Schütze, 2012; Abu-Jbara et al., 2013; Iorio et al., 2013; Li et al., 2013; Jha et al., 2016; Hernández-Alvarez et al., 2017; Meng et al., 2017; Jurgens et al, 2018; Ihsan et al., 2023) and important citation screening (Wan & Liu, 2014; Zhu et al., 2014; Valenzuela et al., 2015; Hassan et al., 2017; Pride & Knoth, 2017; Qayyum & Afzal, 2019; Nazir et al., 2020; ; Aljohani et al, 2021; Qayyum et al., 2021). Deep learning methods further pushed the states of the art (SOTA) significantly (Cohan et al., 2019; Beltagy et al., 2019; Zhang et al., 2022; Jiang & Chen, 2023; Qi et al., 2023).

Despite the significant progress, several shortcomings remain unresolved in existing studies. Citations should be encoded in context. Citation context is a window of surrounding sentences. Example 1 in Figure 1 shows such an extreme example. To avoid misclassifying the citation "[Miller et al.]" in sentence S-124, it is necessary to look backward to the meta-statement of comparison in S-119. Several recent studies have explored citation context modelling (Lauscher et al., 2022; Jiang & Chen, 2023; Zhang et al., 2023; Qi et al, 2023). Being less discussed, most deep learning approaches generated a feature vector for the whole citation context or sentence (Munkhdalai et al., 2016; Lauscher et al., 2017; Bakhti et al., 2018; Su et al., 2019), even some reporting SOTA performances (Cohan et al., 2019; Beltagy et al., 2019; Zhang et al., 2023; Qi et al., 2023), rather than individual in-text citations. This is problematic when applied to citation sentences with multiple in-text citations of different functions, illustrated by the examples from the dataset of this study, i.e., Example 2-3 in Figure 2. In-text citations should be modelled separately, apart from the context they occur.

Example 1: Meta-statement of comparison and contrast. (Teufel, 2010, pp. 434).

We will outline here the main parallels and differences between our method and previous work. In cooccurrence smoothing [Brown et al. 1993] (CoCoGM), as in our method, a baseline model is combined with a similarity-based model that refines some of its probability estimates. In Brown et al's work, given a baseline probability model P, which is taken to be the MLE, the confusion probability EQN between conditioning words EQN and EQN is defined as EQN and the probability that EQN is followed by the same context words as EQN. Then the bigram estimate derived by cooccurrence smoothing is given by EQN. In addition, the cooccurrence smoothing method sums over all words in the lexicon. [Miller et al] (CoCoGM) suggest a similar method... They do...

Figure 1: Citation context is necessary for correct citation function classification.

Example 2: "Weak(ness)" and "Neut(ral)" citations appear in the same citation sentence.

From: https://aclanthology.org/W00-1804.

S-1. While Optimality Theory (OT) (Prince et al. 1993) [Weak] has been successful in explaining certain phonological phenomena such as conspiracies (Kisseberth 1970) [Neut], it has been less successful for computation. (...more weaknesses...)

Example 3: "PSim" (similarity) and "Neut" citations appear in the same citation sentence. Context sentence S-2 is needed to infer the functions of the first two citations in the citation sentence S-1 (forming a citation segment and having the same function).

From: https://aclanthology.org/J00-1004.

S-1. Formalisms for finite-state and context-free transduction have a long history (e.g., Lewis and Stearns 1968; Aho and Ullman 1972) [PSim], and such formalisms have been applied to the machine translation problem, both in the finite-state case (e.g., Vilar et al. 1996) [Neut] and the context-free case (e.g., Wu 1997) [Neut]. S-2. In this paper we have added to this line of research by providing a method for automatically constructing fully lexicalized statistical dependency transduction models from training examples.

Figure 2: Multiple in-text citations may have different functions.

Indeed, Jiang and Chen (2023) have explored a large design space of in-text citation encoding, citation sentence encoding, and citation context encoding towards contextualised citation modelling. They observed that various strong models had their own advantages and disadvantages in recognising different citation functions. The abundant combinations of citation modeling options allow high promise to fuse the strong baselines into a more competent ensemble model for citation context analysis. In machine learning literature, classifier ensemble (Zhou, 2014), or multiple-classifier system (Kuncheva, 2014), has proven to be an effective at improving predictive performance in many subject areas (Jahrer et al., 2010; Xiao et al., 2018; Cao et al., 2020), including a diverse range of natural language text classification tasks (Szidarovszky et al., 2010; Rajani et al., 2015; Rajani & Mooney, 2018; Malmasi &

Dras, 2018; Barrault et al., 2019; Wang et al., 2020; Lin et al., 2022). The success of ensemble learning lies in the diversity among base classifiers (Brown et al., 2005; Ruta & Gabrys, 2005; Sesmero et al., 2021), which is fortunately guaranteed by the wide spectrum of contextualised citation modelling approaches. Therefore, the focus of the current paper is to present a comprehensive study of ensembling approaches to citation context analysis.

The main contributions of the current paper are three-fold. To the best of our knowledge, it is the first comprehensive study and application of ensemble methods to the important task of citation context analysis. To build a large pool of base models for citation context analysis, 175 models were trained based on 35 different citation modelling architectures as in Jiang and Chen (2023), 5 models per architecture initialized with different randomization. Then, a plethora of approaches to combining base classifiers (abbreviated to classifiers hereafter when the context is clear) were systematically evaluated. Thanks to the abundant diversity among classifiers, majority voting significantly improved citation context analysis performances on all the three annotation schemes that were adopted, and produced new states of the art. The success of ensembling is determined by classifier diversity. Our second contribution is the proposal of two heuristic methods to obtain a good diverse set of classifiers. The first method was to re-rank the pair-wise diversity analysis results, which proved to be both effective and efficient in classifier selection and ensembling. The second method was to analyse and employ five famous pair-wise diversity measures to virtually expand the exploration of the space of subset of classifiers, which further improved ensembling performance. Finally, a novel reliability-enhanced confidence-based voting method was proposed to more intelligently break ties in majority voting, which used classifiers' posterior probability (i.e., confidence) and performance (i.e., reliability).

The remaining of the paper is organised as follows. Sect. 2 reviews the related work about machine learning (Sect. 2.1) and deep learning (Sect. 2.2) approaches to citation context analysis, including important citation screening (Sect. 2.3), and the application of ensemble methods in the natural language processing field (Sect. 2.4). Sect. 3 briefly explains the methodological framework of ensembling that the current paper applied, including the ensembling framework (Sect. 3.1), the architecture of base classifiers (Sect. 3.2), sources of classifier diversity (Sect. 3.3), voting approaches to combine classifiers by simple rules (Sect. 3.4), stacking approaches to train meta-classifiers that learns to fuse classifiers (Sect. 3.5), the lattermost including building deep ensembles on top of shallow ensembles. After introducing datasets in Sect. 4, we will detail the experiments of each ensemble method in Sect. 5, more precisely, base classifiers in Sect. 5.1, voting in Sect. 5.2, stacking in Sect. 5.3, and deep stacking in Sect. 5.4. Sect. 6 concludes the paper with discussions of the pros and cons and potential future directions.

2. Related Work

2.1. Machine Learning for Citation Function Analysis

The first machine learning approach might belong to the seminar work by Teufel et al. (2006a). They developed a comprehensive set of features to capture the common cue phrases for expressing scientific concepts and to extract the syntactic information around these cue phrases or the main verbs of citation sentences. An Instance-Based k-nearest-neighbor classifier (IBk) was employed to classify citation functions. To facilitate developing machine learning algorithms, for the first time, a comprehensive and operationalisable 12-class annotation scheme was proposed along with a carefully annotated dataset (Teufel et al., 2006b). Most subsequent studies, especially in the computer science and engineering domain including the current one, inherit from Teufel with certain simplifications, so these annotations schemes are to some extent mappable to each other (Dong & Schäfer, 2011; Abu-Jbara et al., 2013; Jha et al., 2017; Hernández-Alvarez et al., 2017; Jurgens et al., 2018; Su et al., 2019). The exception is Jochim and Schütze (2012), which categorised citations into quadchotomic dimensions of Moravcsik and Murugesan (1975): conceptual vs. operational, organic vs. perfunctory, evolutionary vs. juxtapositional, and confirmative vs. negational.

Teufel et al.'s foundational work spurred much research to refine and enrich the feature set for citation context analysis (Agarwal et al., 2010; Dong & Schäfer, 2011; Li et al., 2013; Abu-Jbara et al., 2013; Jha et al., 2017; Hernández-Alvarez et al., 2017; Meng et al., 2017; Ihsan et al., 2023). In summary, features are syntactic and lexical patterns around manually identified informative cue-phrases for different classes. Amongst them, Jochim and Schütze (2012) also highlighted the importance of named entity features, such as names of dataset, software, algorithm and method, which might be indicators of a usage citation. The most state-of-the-art feature engineering approach came from Jurgens et al. (2018), who used a simplified annotation scheme of six classes (see Table 1), which was later used by the Citation Context Classification (3C) shared tasks (Kunnath et al., 2020). To improve classification performance, novel features were introduced, like citation context topics, linguistic patterns bootstrapped around citations, and PageRank rankings (Jurgens et al., 2018).

2.2. Deep Learning for Citation Function Analysis

More recently, deep learning techniques have been applied to citation function classification. Initial works employed Convolutional Neural Networks (CNNs; Lauscher et al., 2017; Bakhti et al., 2018; Aljohani et al., 2023), Bidirectional Long-Short Term Memory (BiLSTM; Munkhdalai et al., 2016), or CNNs stacked over BiLSTM (Yousif et al., 2019)

to summarize citation sentence or citation context into a feature vector. To enhance contextual understanding, either pretrained word embeddings (Cohan et al., 2019; Roman et al., 2021) or contextualized language models (Beltagy et al., 2019; Maheshwari et al., 2021) were utilized. Witnessing the obvious class imbalance of citation function categories, Aljohani et al. (2023) applied focal loss and class weights to improve classification performance, while Jiang and Chen (2023) tried to merge and re-annotate six datasets in the computational linguistics Teufel et al.'s annotation scheme (Teufel et al., 2006b) to increase the sizes of the minority classes, such as "PSup" and "PBas". There have been a few studies with a particular focus on signifying the importance of properly encoding citation context (Lauscher et al., 2022; Zhang et al., 2022; Jiang & Chen, 2023). For example, Lauscher et al. (2022) created a new dataset with manually annotated minimal set of context sentences that are necessary for citation function classification. This was similar to Jiang and Chen (2023), but the particular merit of the former is that context sentences are not limited to citations neighbourhood, instead can appear anywhere in the paper. While both datasets leave much space for research in the identification of useful context, or citation block according to Kaplan et al. (2016), Lauscher et al. (2022) used gold-standard citation context for citation function classifiers to demonstrate the necessity of it while Jiang and Chen (2023) empirically encoded 2 and 3 context sentences before and after the citation sentence without performing useful citation context sentence identification. As we pointed out in Sect. 1, most of these studies encoded the whole citation context or citation sentence, rather than individual in-text citations.

In parallel, there was also an obvious trend of multi-task learning to enhance citation function classification by jointly training and optimising both the primary task and complementary tasks that are semantically related. Su et al. (2019) used a CNN to encode citation context and used the same encodings for both citation function classification and citation provenance recognition, with the assumption that the two tasks are semantically close. Yousif et al. (2019) used BiLSTM to encode citation sentence and stacked another CNN layer to summarise the meaning of citation sentence. The encoded feature vector was used for both citation function and citation sentiment classification. Cohan et al. (2019) used a self-attention mechanism to summarise the BiLSTM encodings of citation context for citation function classification. The same encodings were also used for two auxiliary tasks, citation worthiness and section role predictions, which had much larger data sources for enhanced representation learning. The same auxiliary tasks were also used in subsequent studies (Oesterling et al., 2021; Qi et al., 2023). Oesterling et al. (2021) extended Cohan et al.'s work by incorporating hand-crafted features like cue list and TF-IDF vectors. Qi et al. (2023) expanded the SciBERT embeddings of each work with manual features such as part-of-speech tag, syntactic pattern, sentiment score,

and TF-IDF values. Different from previous works which used a shared-parameter structure (Zhou & Yang, 2018), Qi et al. decoupled the SciBERT encoders for the three tasks, with the main task further enhanced by a multi-head self-attention mechanism. In addition, all of them relied on one way of encoding in the wide spectrum of modelling options, e.g., self-attention over contextualised word embeddings such as SciBERT, which made them incapable of utilizing the pros of different modelling methods.

2.3. Approaches to Important Citation Screening

A closely related but not central task is important citation screening — recognising meaningful citations that play a significant role to the citing paper, which was embarked by several studies (Wan & Liu, 2014; Zhu et al., 2014; Valenzuela et al., 2015) and flourished in subsequent research (Hassan et al., 2017; Pride & Knoth, 2017; Qayyum & Afzal, 2019; Wang et al., 2020; Qayyum et al., 2021; Aljohani et al., 2021). This classification can be viewed as a simplified version of citation function classification, as citation importance is fundamentally linked to citation function. The distinction lies in the fact that citation function applies to each in-text citation, while citation importance has been evaluated per pair of citing and cited papers by previous studies. Consequently, these studies mainly used paper-level metadata (Wan & Liu, 2014; Valenzuela et al., 2015) and basic full-text features such as cue phrases and textual similarities (Zhu et al., 2014; Hassan et al., 2018; Qayyum & Afzal, 2019; Ghosh et al., 2022). Deep learning approaches to this task encountered the same challenges as in citation function classification that were discussed in the Introduction section (Yousif et al., 2019; Aljohani et al., 2021b; Maheshwari et al., 2021). Recently, Aljohani et al. (2023) reported much better performance on the task by use of focal loss to alleviate the issue of high degree of class imbalance. All existing paper handled the task of screening important citations at the paper level for each pair of citing and cited papers. The current paper, on the contrary, handles the problem at the in-text citation level. Ensembles of deep learning methods were proposed to identify important in-text citations, which could be easily amalgamated into important citation screening in the traditional sense.

2.4. Ensemble Approaches to Natural Language Processing

Ensemble approaches have been successfully applied to a wide range of natural language processing problems, for example, word alignment for machine translation (Wu & Wang, 2005), hedge identification (Szidarovsky et al., 2010), item recommendation (Jahrer et al., 2010), semantic lexicon induction (Qadir & Riloff, 2012), information extraction (Rajani et al., 2015), natural language identification (Malmasi & Dras, 2018), text generation for abstractive

summarization (Kobayashi, 2018), named entity normalization (Deng et al., 2019), neural machine translation (Wang et al., 2020), medication mentioning identification in tweets (Dang et al., 2021), harmful news identification (Lin et al., 2022), etc. Notably, a lot of participants of the GermEval-2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments used classifier ensembles, e.g., Akomeah et al. (2021), Tran & Kruschwitz (2021), etc. In fact, one of the most important findings of the 2019 International Workshop on Machine Translation was that most state-of-the-art systems were based ensemble methods (Barrault et al., 2019).

Most applications of ensemble methods in natural language processing were naïve, simply by combing a limited number of classifiers. Some used homogeneous classifiers or model architectures. Deng et al. (2019) combined several CNN-based architectures while Dang et al. (2021) and Lin et al. (2022) combined several BERT-based models. Others combined heterogeneous classifiers, like Jahrer et al. (2010), Rajani et al. (2015), Malmasi and Das (2018). There are several ways of generating homogeneous base classifiers, for example by using different input features (method used by the current paper), by using different model hyperparameters such as Random Forest (a combination of small decisions trees of different sizes), by training models on bootstrapped datasets, i.e., boosting (Zhou et al., 2014) such as Wu & Wang (2005), and by adding randomness to the training process (widely used for training and aggregating various deep learning model using different random seeds, also used by the current paper). The current paper explored the vast design space of citation modelling options for citation context analysis. For each citation modelling option, five seeds were used for training. Therefore, both the first and last methods were adopted to generate a pool of homogeneous base classifiers in the current paper, while boosting was not used due to prohibitive high cost of training a large number of deep learning models.

There are in general two ways of ensembling base classifiers, by combining base classifiers' predictions using certain rules, often majority voting, or by developing a learnable combiner, called meta-classifier, to segregate base classifiers' predictions. While most ensembling papers in the natural language processing domain used very simple combination rules, such as majority voting (Wu & Wang, 2005; Qadir & Riloff, 2012; Rajani et al., 2015; Kobayashi, 2018; Deng et al., 2019; Dang et al., 2021) or as simple as an OR connective (Szidarovsky et al., 2010), some studies trained a meta-classifier to combine base models' predictions (Jahrer et al., 2010; Wang et al., 2020; Lin et al., 2022). Malsami and Dras (2018) was the most comprehensive study amongst the ensemble-based natural language processing studies we were aware of. They systematically studied a wide range of combination rules, different types of meta-classifiers, and stacked meta-classifiers (Sesmero et al., 2015), i.e., level-2 meta-classifiers trained on the outputs of

level-1 ensembles. The current paper also made a comprehensive exploration of both majority voting and meta-classifier approaches for ensembling. In addition to stacked meta-classifier, we also studied stacked voter (Sect. 3.5). Besides, we also proposed a novel majority voting method, detailed in Sect. 3.4.

Note that, none of the reviewed papers studied the selection of proper base classifiers to ensemble because their base classifier pool sizes were small. In our work, more than 180 base classifiers were trained. A brute-force combination of all the base classifiers would fail to make meaningful improvements. Diversity analysis is an approach that was recognised as one of the key factors for building a successful ensemble (Nam et al., 2021). Kuncheva & Whitaker (2003) and Brown et al. (2005) were good resources for classifier diversity, covering most famous diversity measures, except ratio of errors (Aksela, 2003). Interested readers can refer to Kuncheva (2014) and Zhou (2014) for a more comprehensive coverage of diverse topics of building an classifier ensemble, while Sesmero et al. (2021) had a particular focus on learning a stacked ensemble.

3. Ensembling Methodology

3.1. Framework

Figure 3 illustrates of the framework of building citation context analysis ensemble. The ensembling pipeline starts with a set of T base classifiers, either for citation function classification or important citation screening. Sect. 3.2 explains technical basis of building them, while Sect. 3.3.1 explains in more details the different modelling options towards building the base classifiers. Due to the large number of base classifiers, the next step is to select R "best" candidates to combine in the follow-up stage. A naïve way is to select the top-R candidates according to their classification performance, but this is often suboptimal. It was widely believed more useful to select a diverse subset of classifiers which make different errors so that the large number of peers have a chance to rectify each other's errors (Nam et al., 2021; Sesmero et al., 2021). This was done by the Diversity Analysis module based on five diversity measures widely used in the literature (Sect. 3.3.2). Mere diversity ranking may still lead to suboptimal results. On the one hand, it was important to include the few best-performing classifiers by observing a sharp performance drop of most classifiers from the top end. On the other hand, diversity ranking sometimes gave lower ranks to these top-performing classifiers and often tended to include many suboptimal classifiers (merely because their predictions were different even though maybe incorrect). Therefore, the Diversity Re-ranking component was intended to rectify this

suboptimal behaviour (Sect. 3.3.3). After re-ranking, the Classifier Selection stage retained *R* classifiers to fuse. Here the predictions made by the base classifiers were called *level-1 predictions*.

After selecting the top-*R* classifiers that achieved a better trade-off between diversity and accuracy, the Classifier Combination stage used the level-1 predictions to build ensembles, either using majority voting methods (Sect. 3.4) or through training a meta-classifier, i.e., classifier stacking (Sect. 3.5). Note that the classifiers in this paper were homogeneous classifiers because they were trained following the same deep learning architecture (Sect. 3.2), but with different feature extraction (i.e., citation modeling) methods (Sect. 3.3.1). Both majority voting and meta-classifier could directly generate the final class label. In this case, we say a *level-1 ensemble* classifier was built. Predictions of level-1 ensembles could also be used for classifier combination. For example, in Figure 3, results of majority voting could be used to vote again or to train a *level-2 meta-classifier* (the downward arrow). Similarly, results of meta-classifiers could also be used to build a *level-2 voter* (the upward arrow) or to train a level-2 meta-classifier. Results of all these options will be discussed in Sect. 5.4.

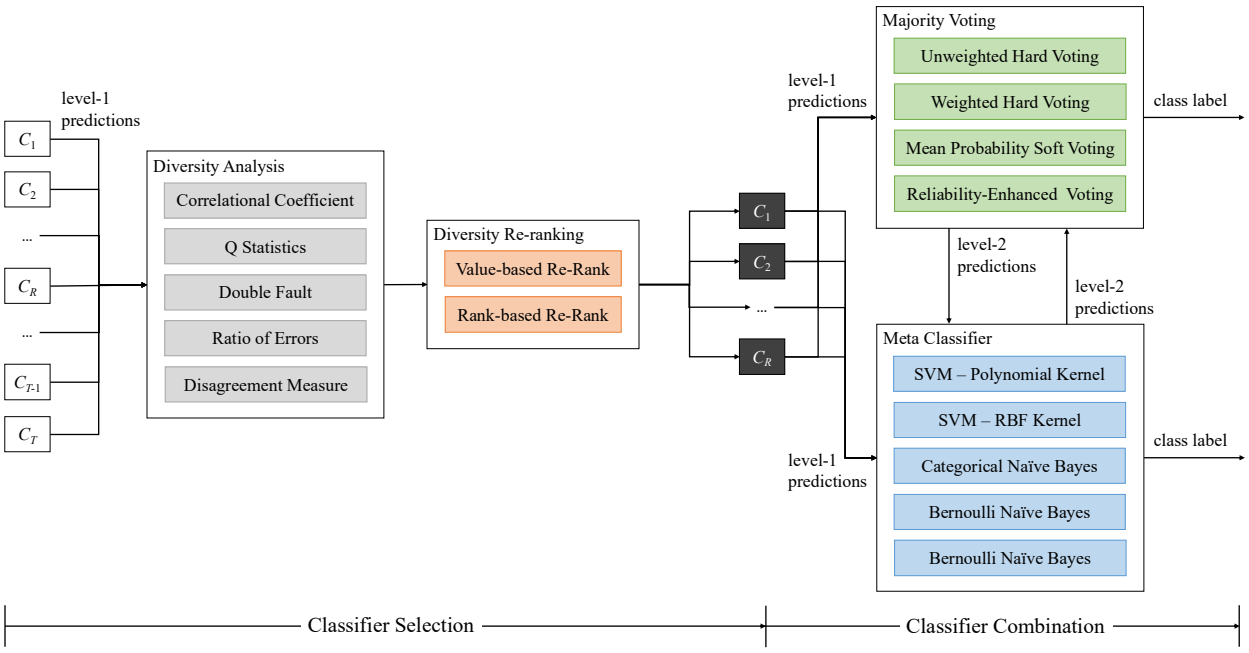

Figure 3: The Framework of Ensembling for Citation Context Analysis.

3.2. Architecture of Base Classifiers

We used all modelling options presented in Jiang and Chen (2023) to train base classifiers for citation context analysis and the cross-disciplinary pretrained language model SciBERT (Beltagy et al., 2019) for encoding citation contexts

and fine-tuning. As illustrated in Figure 4, three factors were considered: the target citation string (converted to a pseudoword "CITSEG")[1], the enclosing citation sentence, and the surrounding citation context. Following the BERT tradition (Devlin et al., 2019), the token sequences of each sentence were separated by the sequence separator "[SEP]". However, we also tested a different setup without inserting the sequence separator (detailed in Sect. 3.3.1). In addition, the whole token sequence of the citation context was prepended by the sequence classification symbol "[CLS]". SciBERT was used to encode the citation context.

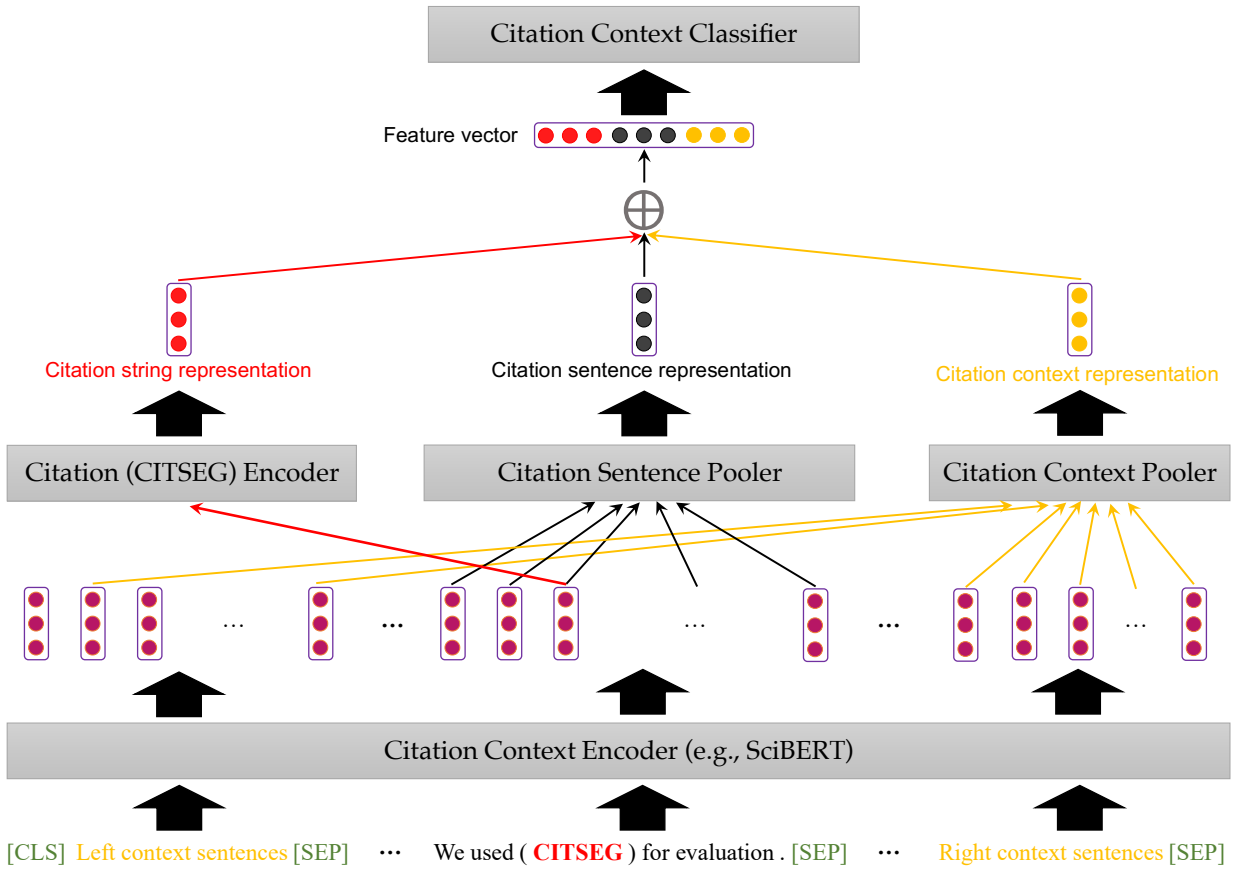

Figure 4: Architecture for Citation Context Analysis.

[1] Following Jiang and Chen (2023), consecutive in-text citation strings were merged into a citation segment, represented by a pseudoword "CITSEG". This is because all these in-text citations must have the same rhetorical role.

The feature vector for citation context analysis consisted of three parts to be discussed below, while the tested citation modelling options are introduced in Sect. 3.3.1. (i) The in-text Citation Encoder (i.e., CITSEG Encoder) used the encodings of the pseudoword "CITSEG" as the *citation string representation*, denoted by **h,** which is necessary for distinguishing between different citations in the same citation sentence. To this end, the pseudoword "CITSEG" was added to the vocabulary of SciBERT and its embeddings were learned during the fine-tuning process. (ii) Lauscher et al. (2022) estimated that for more than 90% citation instances the citation sentence alone is enough for correctly determining the citation function. Inspired by this finding, we used a Citation Sentence Pooler to produce the *citation sentence representation*, denoted by **s,** by pooling over all tokens of the citation sentence. (iii) To handle cases requiring multi-sentence contexts, the Citation Context Pooler was used to generate the *citation context representation*, denoted as **c**, from the whole citation context. In this study, the context window size was fixed to [-2, +3], i.e., two left and three right context sentences, including the central citation sentence. Indeed, Lauscher et al.'s annotations demonstrated that it is very rare to go beyond a citation context of six sentences to find the useful context sentences for determining citation functions. The final feature vector **f** was the concatenation of these three optional parts. In our experiments the Citation Context Classifier was a Multiple-Layer Perceptron (MLP) with one hidden layer.

3.3. Sources of Diversity

3.3.1. Citation Modelling

The first source of classifier diversity comes from the citation modelling options for each component in Figure 4. A large part of this subsection is inherited from Jiang and Chen (2003) (see the "Citation function classification algorithms" section) and Jiang et al. (2022), the shorter conference version of the former paper (see Sect 4. CITATION FUNCTION CLASSIFICATION MODELS). For clarity purposes, we re-structured and expanded the descriptions of the citation modelling options in the current paper.

Citation modelling in context? To distinguish between multiple citation (segments) in the same citation sentence, we assume that the citation string representation **h** is always used. There are several options for whether and how to incorporate the context where the citation string is encoded. The most extreme case is citation is encoded in context, but no context information is utilised for classification, i.e., $\mathbf{f} = \mathbf{h}$. If either (both) citation sentence representation or (and) citation context representation is (are) considered, then we have the following modelling options: $\mathbf{f} = [\mathbf{h}; \mathbf{s}]$ (both citation string and citation sentence are encoded in context and the latter is deemed to be

helpful for determining citation function or importance, suitable for citation instances whose roles can be determined using the citation sentence alone), $\mathbf{f} = [\mathbf{h}; \mathbf{c}]$ (the meaning of the citation context is used to enhance citation string representation, to account for the cases which require looking over the citation sentence to a larger surrounding context), or $\mathbf{f} = [\mathbf{h}; \mathbf{s}; \mathbf{c}]$ (with the hope of enjoying the benefits of both the previous two methods).

Sequential or hierarchical context? We define two types of citation context: a *sequential context* concatenates all tokens of all sentences in the context without inserting the special sequence separator symbol (e.g., "[SEP]" in SciBERT and all models in the BERT family), while a *hierarchical context* inserts the sequence separator symbol after each sentence. The distinction impacts the way of pooling sentence representations (e.g., for the citation sentence representation **s**) and context representation.

Pooling sentence representations. In case of a sequential context, sentence representations can be pooled by applying a sentence mask to the tokens of each sentence. Either max pooling or self-attention could be used. For hierarchical context, there is one more option, i.e., the encodings of the sequence separator. These sentence poolers apply to both citation sentence and context sentence.

Pooling context representations. Again, either max pooling or self-attention can be used. In case of a sequential context, the citation context representation is pooled from its tokens. The latter is similar to the approach used in Munkhdalai et al. (2016). For hierarchical context, the Citation Context Pooler applies the pooling operation, either max pooling or self-attention, to over all the sentence representations in the context. As described in "**Pooling sentence representations**", three options exist for pooling sentence representations: max pooling, self-attention, and the sequence separator.

Summary. Table 2 summarises the citation context analysis models that were used in the current paper. Below detail the parameters that control the modellings options discussed above.

- Context type (`ctx_type`): Sequential context (sequential) v.s. hierarchical context (hierarchical).
- Sentence pooler (`sent_pooler`): max pooling (max_pool) v.s. self-attention (self_attend) v.s. [SEP] (resp. N/A) when a hierarchical context is used (resp. in case of a sequential context).
- In-text Citation Encoder (`citseg`): By default, it is always used (O) because it was found key to strong performance (Jiang and Chen, 2023).

- Citation Sentence Pooler (`cita_pooler`): max pooling (max_pool) v.s. self-attention (self_attend) v.s. none (X); pooling was performed on word/token embeddings when a sequential context was used, or on sentence representations in case of a hierarchical context.
- Citation Context Pooler (`ctx_pooler`): max pooling (max_pool) v.s. self-attention (self_attend) v.s. none (X); pooling was performed on word/token embeddings when sequence context was used, or on sentence representations in case of hierarchical context.

3.3.2. Diversity Measure

The second source of classifier diversity comes from the combination of subsets of classifiers that are used to build ensembles. In ensemble learning, it is intuitively more plausible to choose the most "diverse" set of classifiers which make different prediction mistakes so that there is a higher chance to rectify single classifier's prediction mistake by peers (Kuncheva & Whitaker, 2003). There are basically two categories of diversity measures: pairwise and non-pairwise. Non pairwise measures calculate the overall diversity averaged across a subset of classifiers. In this paper, we trained 180 citation context analysis classifiers (36 citation modelling options × 5 seeds per option). Because the total number of possible subsets of classifiers is exponentially large, i.e., 2^{180}, we refrained to choose pairwise diversity measures for the sake of computational feasibility.

Following the notations used in Kuncheva and Whitaker (2003), let C_i and C_k (out of in total T classifiers) be a pair of classifiers working on a dataset of N samples. We defined four values based on the correctness of classifications to quantify pairwise diversity: (1) N^{11} – the number of samples that are correctly classified by C_i and C_k; (2) N^{10} – the number of samples that correctly classified by C_i but misclassified by C_k; (3) N^{01} – the number of samples that misclassified by C_i but correctly classified by C_k; and (4) N^{00} – the number of samples that are misclassified by both C_i and C_k. We have $N = N^{11} + N^{10} + N^{01} + N^{00}$. The pairwise diversity measures experimented in this paper included *correlation coefficient* (Div_{CC}), *Q statistic* (Div_{Q}), *double fault* (Div_{DF}), *disagreement measure* (Div_{DM}), and *ratio of errors* (Div_{RO}) (Aksela, 2003), which are defined in Eqs. (1–5). A note is deserved for *ratio of errors*, where $N^{00}_{different}$ is the number of samples that are misclassified by both classifiers but misclassified into different classes and N^{00}_{same} is the number of samples that are misclassified by both classifiers in the same way. Ratio of errors reflects the most extreme and worst setting for ensembling because it means "several classifiers agree on an incorrect result" (Aksela, 2003). We also note that correlation coefficient, Q statistics and double fault are inversely proportional to

diversity, so we deliberately add a negative sign in Eq. (1–3). Although our definitions of Div_{CC}, Div_{Q}, Div_{DF} slightly differ from their original definitions, they allow for sorting classifier diversity in a consistent way.

Div_{CC}:
$$\rho_{i,k} = -\frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11}+N^{10})(N^{01}+N^{00})(N^{11}+N^{01})(N^{10}+N^{00})}} \quad (1)$$

Div_{Q}:
$$Q_{i,k} = -\frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \quad (2)$$

Div_{DF}:
$$DF_{i,k} = -\frac{N^{00}}{N^{11}+N^{10}+N^{01}+N^{00}} \quad (3)$$

Div_{DM}:
$$Dis_{i,k} = \frac{N^{10}+N^{01}}{N^{11}+N^{10}+N^{01}+N^{00}} \quad (4)$$

Dic_{RO}:
$$RE_{i,k} = \frac{N^{00}_{different}}{N^{00}_{same}} \quad (5)$$

3.3.3. Diversity Re-Ranking

The base classifiers used in this paper were all deep learning methods and the number of classifiers was big, therefore we decided to select the top R "most diverse" subset of classifiers (from T candidate classifiers). Using diversity measures discussed in Sect. 3.3.2, we could greedily select R most diverse classifiers, while the diversity of one classifier was defined as the sum of all pairwise diversities between it and all other classifiers in the candidate set. However, this method was flawed because candidate classifiers' performances varied a lot. When looking only at classifier diversity but totally ignoring classifier performance, the selected subset often included many weak classifiers and, what was more severe, often missed the strongest ones. This was caused by the symmetry of pairwise diversity measures and the fact that diversity measures were defined by classifier errors (Brown et al., 2005). More specifically, the weakest classifiers that make the most mistakes might have made many unique classification errors, potentially resulting in higher diversity. This could be seen from the empirically results of majority voting on a subset of weak classifiers that the ensemble could rival but hardly beat the strongest classifier, which was ,missed by diversity ranking (Table 3-5).

Therefore, this paper proposed two simple but effective diversity re-ranking methods to avoid this inferior situation. We relied on two things: classifier performance (e.g., macro F1), and classifier diversity (e.g., either one of the five diversity measures). The first method was *value-based re-ranking*, which was simply sorting classifiers in descending order of the sum of normalised classifier diversity and normalised classifier performance. Here normalised

diversity is calculated based on the sign of its value: If positive, the normalised diversity of a classifier in the candidate set is the diversity of the classifier divided by the maximal diversity; otherwise, the normalised diversity is the maximal diversity divided by the diversity of the classifier. The second method was *rank-based re-ranking*, which first sort classifiers in descending orders of classifier performance and classifier diversity, and then re-sort the classifiers in ascending order of the sum of classifier performance rank and classifier diversity rank.

Figure 5 shows a real example on our dataset using double fault (DF), where $T = 20$, $R = 10$, i.e., selecting 10 most diverse classifiers from a pool of 20 candidates. Rank_DF (resp. Rank_F1) is the rank of classifier based on DF (resp. Macro F1) in descending order. Norm_DF and Norm_F1 are the normalised DF and normalised F1 respectively. To performance value-based and rank-based re-ranking, two weights are calculated: Weight_V = Norm_DF + Norm_F1, and Weight_R = Rank_DF + Rank_F1. Finally, ReRank_V and ReRank_R are the value-based and rank-based re-ranking results in descending order of Weight_V and Weight_R respectively. Ties are broken using classifier performance, e.g., F1. A notable case in Figure 5 is shown in bold underlined. C_{16} has the highest classification performance, beating other candidates by a large margin. However, its diversity rank is very low. Fortunately, both re-ranking methods bring it to the top-10 list, which is preferred! Another notable case is in bold italic. C_2 has very low rank in term of F1; its performance is poor. As we assumed earlier, such weak classifiers might be undesirably "diverse" only because they make too many errors, some of which may be unique. Fortunately, the rank-based reranking method is able to rule it out of the top-10 list, which may improve the performance of ensemble that is built on top of 10 selected classifiers.

Classifier	DF	F1	Rank_DF	Rank_F1
C_1	-3.9910	65.02	1	3
C_2	-4.0013	***64.12***	2	***19***
C_3	-4.0400	64.38	3	13
C_4	-4.0516	64.95	4	5
C_5	-4.0955	65.12	5	2
C_6	-4.1252	64.65	6	7
C_7	-4.1574	64.46	7	10
C_8	-4.1690	64.99	8	4
C_9	-4.1703	64.28	9	16
C_{10}	-4.1742	64.56	10	8
C_{11}	-4.1755	64.93	11	6
C_{12}	-4.1768	64.41	12	12
C_{13}	-4.2065	64.26	13	17
C_{14}	-4.2155	64.12	14	20
C_{15}	-4.2271	64.14	15	18
C_{16}	-4.2542	**66.16**	**17**	1
C_{17}	-4.2542	64.48	16	9
C_{18}	-4.2581	64.42	18	11
C_{19}	-4.2710	64.37	19	14
C_{20}	-4.2735	64.36	20	15

Norm_DF	Norm_F1	Weight_V	Weight_R
1.0000	0.9828	1.9828	4
0.9974	0.9692	1.9666	21
0.9879	0.9731	1.9610	16
0.9850	0.9817	1.9667	9
0.9745	0.9842	1.9587	7
0.9675	0.9772	1.9447	13
0.9600	0.9742	1.9342	17
0.9573	0.9823	1.9396	12
0.9570	0.9716	1.9286	25
0.9561	0.9759	1.9320	18
0.9558	0.9814	1.9372	17
0.9555	0.9736	1.9291	24
0.9488	0.9712	1.9200	30
0.9467	0.9691	1.9158	34
0.9441	0.9694	1.9135	33
0.9381	1.0001	1.9382	18
0.9381	0.9746	1.9127	25
0.9373	0.9737	1.9110	29
0.9344	0.9730	1.9074	33
0.9339	0.9729	1.9068	35

Rerank_V	Rerank_R
1	1
3	***11***
4	6
2	3
5	2
6	5
10	7
7	4
13	13
11	9
9	8
12	12
14	16
15	19
16	17
8	**10**
17	14
18	15
19	18
20	20

Figure 5: An example of re-ranking 20 candidate classifiers which are originally sorted in double fault (DF).

3.4. Majority Voting

The first ensembling approach was majority voting. Formally speaking, out of T candidate classifiers, a subset of R most diverse classifiers were selected based on diversity measure (Sect. 3.3.2) and diversity re-ranking (Sect. 3.3.3). Both hard majority voting and soft majority voting (Zhou, 2014) were evaluated. For Hard majority voting, the most basic one was *unweighted hard voting* (HARD VOTE – UNWEIGHTED in future sections and tables), which simply counts the number of votes each class label received from base classifiers and chose the class label that won the most votes, and randomly selects a label when a tie happens. Due to this randomness, we decided to report the average performance over 10 random runs in the Results and Discussions section (Sect. 5). Intuitively, we felt it reasonable to have more trust in the stronger classifiers, so the *weighted hard voting* approach (HARD VOTE – WEIGHTED) used classifier performance to weight each vote, and the score for each label is the sum of the weighted votes. In HARD VOTE – WEIGHTED, ties are avoided most of the time, so there was little need for averaging over 10 random runs.

When it comes to soft majority voting, classifier confidence on each instance, i.e., the posterior probability of a classifier, was used for fusing decisions. A lot of choices existed in past literature (Malmasi & Dras, 2018), for example, Mean Probability Rule, Median Probability Rule, Product Rule, Highest Confidence, Corda Count, etc. Malmasi and Dras reported strong performances of the mean probability and median probability rules compared to hard majority vote. In our experiments, we saw similar performances of both methods, so we opted for Mean Probability Rule (SOFT

VOTE – MEAN) as it was the best performing voting method in Malmasi and Dras (2018). See Figure 6 for the illustration of this fusing method. Meanwhile, we proposed a new soft weighting method called *Reliability-Enhanced Soft Voting* (SOFT VOTE – RELABILITY). Each classifier provided three types of information for decision fusion: *vote* (label predicted by classifier), *confidence* (posterior probability of predicted label), and *reliability* (performance of classifier, e.g., Macro F1 in this paper). Then, a *soft vote* is calculated by *confidence* × *reliability*. Then fusion decision was made by total number of votes and total number of soft votes, using the latter to break ties. This approach was proved to be an extremely effective and consistently robust voting method in our experiments.

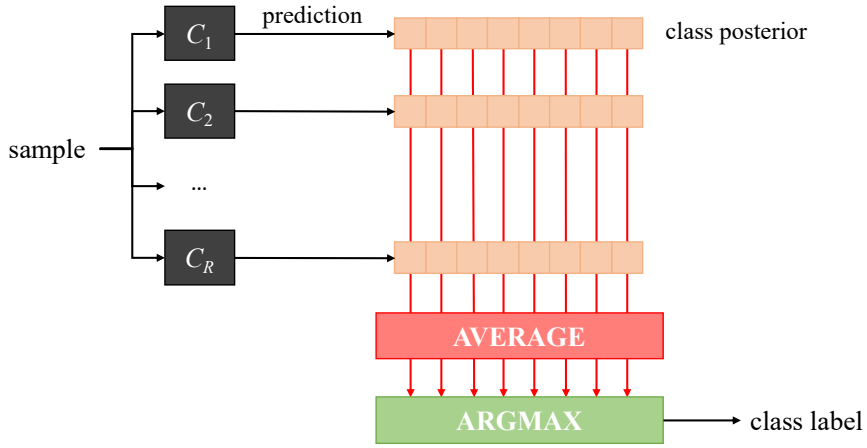

Figure 6. Soft Voting by Mean Probability Rule, Adapted from Malmasi and Dras (2018).

3.5. Classifier Stacking

Different meta-classifiers were selected in the literature for classifier stacking, such as Gradient Boosted Decision Tree (GDBT) and Neural Networks (NN) (Jahrer et al., 2010), Deep Neural Networks (Xiao et al., 2018), Logistic Regression (Shahri et al., 2020), Support Vector Machine (SVM) (Akomeah et al., 2021). Malmasi and Dras (2018) presented the most comprehensive comparison among nine meta-classifiers, including Logistic Regression (LogReg), Ridge Regression (Ridge), Linear SVM, RBF-Kernel SVM, LogReg, *k*-Nearest Neighbour (*k*-NN), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Perceptron and Decision Tree (DT). Our experimental results corroborated with Malmasi and Dras in that DT, Perceptron and QDA were not competitive. Contrastively, LogReg (with L1-regularisation or L2-regularization, the latter of which is similar to Ridge) and Linear SVM did not rival SVMs with kernels in our experiments. In addition, the experimental results of Random Forest (RF) and different variants of it were also not as convincing as Jahrer et el. showed, despite of extremely time-consuming

hyperparameter tuning. Therefore, we decided to choose and report on *k*-NN, SVM (with polynomial and RBF kernels, abbreviated as SVM-Poly and SVM-RBF respectively), LDA, Categorical Naive Bayes (CatNB), and Bernoulli Naïve Bayes (BerNB). Different from most of the literature, we used both the predicated labels and posterior probabilities of base classifiers as inputs to meta-classifiers. Both SVM-Poly and SVM-RBF accepted posterior probabilities as input, while *k*-NN, CatNB and BerNB accepted class label as input. We did not choose Gaussian Naïve Bayes because we believed the posterior probability distribution of classifier predictions is not Gaussian. Out initial experimental results also confirmed this assumption through its inconspicuous performance, which were omitted to save space.

Note that meta-classifier needs data for training another classifier, which will then be applied to the test data. Different from voting methods which directly worked on the test data, we used two ways to evaluate meta-classifier performance. The first way was to only use test data, but this meant no particular held-out data for meta-classifier training. In this case, we adopted 5-fold cross validation, a machine learning approach to training more robust classifier and reporting more robust performance on datasets of limited size. The second way was to use the original validation data which were used to fine-tune the deep learning base classifiers. Ideally, there should be a held-out dataset just for meta-classifier training (Zhou, 2014), a portion of which should be reserved for meta-classifier hyperparameter tuning, as in Jahrer et al. (2010). However, this was impossible in our case. Therefore, we decided to enlarge the validation data with misclassified samples in the training set. We found that (1) the training instances were classified by all base classifiers with an extremely high accuracy, and (2) these classifiers proved to be able to generalize to the validation and test data as there was no catastrophic performance drop from validation to test. So, we expanded validation data with the training instances that were mis-classified by at least two base classifiers, and then used 5-fold cross-validation to tune meta-classifier performance. Details will be given in the Dataset section (Sect. 4.2) and Results and Discussion section (Sect. 5).

4. Dataset

4.1. Citation Context Dataset

We used the citation context dataset proposed in Jiang and Chen (2023). This dataset was created by re-annotating citation instances from six datasets in the computational linguistics (CL) domain. The six datasets were proposed by previous studies about citation function classification (Teufel et al., 2006a; Dong & Schäfer, 2011; Abu-Jbara et al., 2013; Hernández-Alvarez et al., 2017; Jurgens et al., 2018; Su et al., 2019). The dataset contains 3356 citation contexts,

4784 in-text citations and 3854 citation targets with annotations. Note that, in this dataset, consecutive citation strings in each citation sentence were merged into a citation segment, represented by a pseudoword "CITSEG". Each citation segment is a citation target and annotations were made to each citation segment. For example, in the exemplar citation sentence "SHRDLU (Winogard, 1973) was intended to address this problem." (Figure 3), the in-text citation target "Winogard, 1973" was replaced by the pseudoword "CITSEG". So, the citation sentence was tokenized into ["SHRDLU", "(", "CITSEG", ")", "was", "intended", "to", "address", "this", "problem", "."]. For experiments, the dataset was randomly split into a training split (60%), a validation split (15%) and a test split (25%), making sure that each split had the same class distribution (Jiang & Chen, 2023). This paper used exactly the same data splits.

The dataset was originally annotated using a classical 12-class annotation scheme (Teufel et al., 2006a) plus a common function "Future (work)". The annotation scheme was then mapped to a more coarse-grained and widely used 6-class scheme (Jurgens et al., 2016). "CoCoXY" means comparison and contrast between two cited papers. "Weak" means weakness of the cited paper. "CoCoGM" (resp. "CoCoR0") means objective comparison and contrast about research goal and method (resp. empirical results), while "CoCo-" means the cited paper is inferior to the citing paper, i.e., a negative comparison. "PSim" means similarity between citing and cited papers. "PSup" means the citing and cited papers support each other theoretically, either technically or empirically. "PMot" means the citing paper is motivated by the cited paper. "PUse" means the citing paper uses some intellectual assets proposed by the cited paper. "PModi" means technical modification of the cited paper while "PBas" means ideational basis on the cited paper. Finally, "Neut" means anything else unable to be classified into other categories, or "neutral" citations, or often "background" citations. The authors of the dataset mapped the original annotations to a slightly simplified 11-class scheme, in which the "CoCo-" class was spread into "CoCoGM" (goal and method comparison) and "CoCoRes" (result comparison) because the former mixes comparisons about both methods and results), and the "Basis" class merged "PBas" and "PModi" because these classes were still too small. Citation functions could also be mapped to citation importance, for which mapping from citation function to citation importance by Valenzuela et al. (2015) was used. Citation importance is binary, either important or unimportant.

12-class + "Future" (Teufel et al., 2006a)				11-class			6-class (Jurgens et al., 2016)			2-grade (Valenzuela et al., 2015)		
label	**size**		**ratio**	**label**	**size**	**ratio**	**label**	**size**	**ratio**	**grade**	**size**	**ratio**
	citstr	**citseg**	**citseg**									
Future	97	85	2.21%	Future	85	2.21%	Future	85	2.21%	Important	2937	76.21%
CoCoXY	200	152	3.94%	CoCoXY	152	3.94%	Background	1615	41.90%			
Neut	1924	1463	37.96%	Neutral	1463	37.96%						
Weak	223	158	4.10%	Weakness	158	4.10%	ComOrCon	944	24.49%			
CoCoGM	390	299	7.76%	CoCoGM	328	8.51%						
CoCo-	108	80	2.08%									
CoCoR0	107	100	2.59%	CoCoRes	151	3.92%						
PSup	123	100	2.59%	Support	100	2.59%						
PSim	247	207	5.37%	Similar	207	5.37%						
PMot	365	288	7.47%	Motivation	288	7.47%	Motivation	288	7.47%			
PUse	794	755	19.59%	Usage	755	19.59%	Uses	755	19.59%	Unimportant	917	27.39%
PModi	72	65	1.69%	Basis	167	4.33%	Extends	167	4.33%			
PBas	134	102	2.65%									
Total	4784	3854										

Table 1. Citation Context Database and Annotation Schemes (adapted from Jiang and Chen (2023)).

4.2. Meta-classifier Data

The data splitting was done on citation segments. There were in total 2497 training instances, 582 validation instances and 775 test instances. The number of validation instances were comparatively small. So, we decided to expand the validation set for with training samples that were misclassified by at least TWO base classifiers. Considering "Support", "Weakness", "Basis", "Similar" were the more difficult classes for most classifiers, more instances of these classes were added to enrich the validation set. They were treated as more confusing cases, and we hoped that improvement on these samples would boost meta-classifier performance. In total, there were 2112 training samples combined with the validation set for training the meta-classifier.

5. Results and Discussions

5.1. Base Classifiers

The performances of the base classifiers on citation function classification were obtained from Jiang and Chen (2023). In addition, citation importance classifiers were trained using the same settings as in Jiang and Chen's paper. Five

random rums were done using the same seeds and the best macro F1, average macro F1 and the standard deviation were reported. All experiments were run on one GeForce RTX 3080 GPU whose CUDA version was 11.6. Table 2, which is adapted from Table 5 in Jiang and Chen (2023), shows the performances of all 36 model architectures on citation function classification (with the 11-class and 6-class citation function schemes) and important citation screening (with the 2-grade citation importance scheme). The best classifiers achieved 66.16% best F1 (across five runs) and 63.5% average F1 (across five runs) on the 11-class scheme. The 66.16% best F1 was considered strong due to the cognitive complexity of this citation function scheme. The top-3 models (indeed model architectures) in term of best (macro) F1 were shown in **bold underlined**, **bold** and underlined fonts respectively in the table. Note that, with the 11-class scheme, there was a significant performance drop from 66.16% (top-1) to 65.12% (top-2). Less extreme but still significant performance drops also happened in the top-performing models on the 6-class scheme, from 74.03% (top-1) to 73.25% (top-3), and further to 72.81% (hie-21), then suddenly to 72.11% (hie-09). After that the model performance curve, if sorted in descending order, started to be flatter. This signifies the necessity of including the best performing model(s) into the ensemble. In addition, that the performance differences between the weakest classifiers were often minor, implying a higher chance of low classifier diversity among them, so it might be wiser to avoid building ensembles mainly based on weak classifiers.

[Table 2: See Appendix]

5.2. Majority Voting

5.2.1. Experimental Setup

Due to the large number of base classifiers ($T' = 150$), most of which significantly underperformed the few top ones, we decided to first select a set of T classifiers in descending order of classifier performance as the pool of candidates. To ensure performance, the pool should be large enough, say $T = 50$. We also tested a series of different sizes: $T \in \{50, 40, 30, 20, 10\}$. Finally, a subset of R diverse classifiers were chosen from the pool to fuse. The T candidates were ranked in descending order of classifier diversity based on pair-wise diversity measures, as explained in Sect. 3.3.2. In this way, it was still difficult to determine the best subset, i.e., the best R value, to fuse, so we tested different values of R ($R = 2,3, \ldots, T$) and reported the best performance together with the corresponding ensemble size R. As introduced in Sect. 3.4, four voting methods were experimented, unweighted hard majority weighting (HARD – UNWEIGHTED), weighted hard majority voting (HARD – WEIGHTED), mean-probability soft majority voting

(SOFT – MEAN), and reliability-enhanced soft voting (SOFT – RELIABILITY). HARD – UNWEIGHTED was done 10 times and averaged[2]. With other methods, whenever there was a tie, though being very rare, macro F1 was used to break the tie. For each fusion method, the five diversity measures introduced in Sect. 3.3.2 were tested and compared. For each diversity measure applied in combination with each fusion method, we reported both the original ensemble performance (the ¬RR column in Table 3–5) in term of macro F1 and the performances of diversity re-ranking defined in Sect. 3.3.3 (the RR_Rnk and RR_Val columns).

5.2.2. Results

Table 3–5 show the results under all experimental setups with the 11-class scheme, 6-class scheme and 2-grade scheme respectively. For majority voting, it was difficult to determine the best R (classifiers to fuse) when using pair-wise diversity measures. Therefore, we reported the performance that was obtained using the best R ranging between 2 and T. This is a significant drawback of the majority voting method. It was hardly possible to reliably determine the "best" R using a held-out set, because a small difference between the distributions of the validation and test samples would be amplified, causing the R "optimised" on the validation set to become suboptimal or even poor on the test set. This drawback can be alleviated by the classifier stacking approach, which trains a meta-classifier to optimise the weights of the contribution of each classifier, making fusion results more stable. The best majority voters were HARD – WEIGHTED on the 11-class scheme with a pool of $T = 40$ candidate classifiers diversified by Q statistics and value-based re-ranking (the shaded cell in Table 3), and SOFT – RELIABILITY on the 6-class scheme with a pool of $T = 40$ candidates diversified by correlation coefficient and rank-based re-ranking (the shaded cell in Table 4). **Compared to the best single models the performance gains were significant:** for 11-class, a 4.6% absolute improvement from 66.16% (seq-08 in Table 2) to 70.78% (Table 3), and for 6-class, a 3% absolute improvement from 74.03% (seq-01 in Table 2) to 77.05% (Table 4). On the 2-grade scheme, there were quite a few ensemble settings performing equally well, topping at 89.63%. This might be caused by the fact that the important citation screening task was comparatively not as complex as the citation function classification task, which was proved by the good single model performance topping at 86.65% (Table 2). The decision space was also much simpler. The diversities among classifiers were likely

[2] The following randomly picked seeds were used: 11, 107, 211, 509, 521, 929, 971, 1061, 1753, and 1979.

to be not as obvious as in citation function classification, resulting in many ensemble classifiers with similar behaviours.

[Table 3–5: See Appendix]

Several conclusive observations could be made. Firstly**, relatively weak classifiers did contribute to a stronger ensemble performance**. When only a small number of top-performing classifiers were selected, e.g., $R = 10, 20$ for 11-class (Table 3), $R = 10, 20$ for 6-class (Table 4), and $R = 10$ for 2-grade (Table 5), the ensemble performance were not optimal. The best performances appeared when $R = 40$ for 11-class, $R = 50$ for 6-class and $R = 30$ for 2-grade schemes. Secondly, from the results on all three annotation schemes, it was safe to claim that **when the pool of candidate classifiers is large and diverse enough, diversity re-ranking methods consistently improves fusion performance** (RR > ¬RR)**.** Generally, **rank-based re-ranking was overall better than value-based re-ranking** (RR_Rnk > RR_Val). Both claims could be seen from the "AVG" rows in all three tables. The extreme opposite case was that, when $T = 10$, doing diversity re-ranking was worse than no re-ranking on all three annotation schemes. The reasons might be that the candidate pool was too small, thus missed a lot of candidates that provided commentary views of the classification task. This explanation corroborates with our first claim that many weak classifiers are indeed helpful for building a better ensemble. The situation with the 2-grade scheme was an even more extreme opposite case, where the best voter appeared at $T = 30$ without doing re-ranking (though a few value-based re-ranking results rivaled). We also noted that the best ensemble size for all these rivaling voters was all $R = 15$, which first corroborate with the first claim above and also implied that there might be many important citation screeners that performed equally well, and the fusion of a subset of them reached the performance ceiling of majority voting because they reinforced each others' decisions, reducing the ensemble's capability of integrating more classifiers' points of view. Thirdly, **cognitively more challenging tasks might require a larger candidate pool to allow more diversity.** This actually implied the effectiveness of performing diversity analysis for combining classifiers like citation function classification (Nam et al., 2021).

5.3. Classifier Stacking

5.3.1. Experimental Setup

Four types of meta-classifier were used, *k*-Nearest Neighbour (*k*-NN), Support Vector Machine (SVM), Naïve Bayes (NB) and Linear Discriminant Analysis (LDA). For *k*-NN, the following values were selected: $k = 5,7,11,13,15$.

Base classifier's predicted labels were used as inputs. For SVM, both polynomial kernels and RBF (Radial Basic Function) kernels were used. They were denoted as SVM-Poly and SVM-RBF. Base classifiers' posterior probabilities (of predicted labels) were used as inputs for both SVM and LDA. For NB, Categorical Naive Bayes (CatNB) was used for citation function classification and Bernoulli Naïve Bayes (BerNB) was used for important citation screening, both with base classifiers' predicted labels as inputs.

Table 6 summarises the hyperparameters tuned for each meta-classifier. For k-NN, the only option needs to be tuned was the weighting of instances (nearest neighbours used for voting), either "uniform" (i.e., equally weighted) or "distance" (inversely weighted based on distance to the test sample). For CatNB and BerNB, the only hyperparameter tuned was α, the additive value used for smoothing the Naïve Bayes estimate of the counts of feature values with respect to each category[3]. For SVM-RBF, the hyperparameter was γ in the RBF kernel function while SVM-Poly had one more parameter — the degree of polynomial d[4]. For both SVM-Poly and SVM-RBF, the regularisation coefficient C was a common hyperparameter[5]. Due to the large number of hyperparameter settings of SVM, we first performed grid search using a large but coarse range of C and γ values, found the poor value ranges for both parameters, and then narrowed down to a smaller but finer range of hyperparameters values as in Table 6. For all the meta-classifiers, the five diversity measures (Sect. 3.2.2) were also part of the hyperparameters to be tuned. Finally, note that only rank-based re-ranking was used in the meta-classifier experiments as this was proved an overall better re-ranker when there was abundance in candidate classifiers.

[Table 6: See Appendix]

Two groups of experiments were done for classifier stacking. The first group was done purely on the test split. For a more robust evaluation, 5-fold cross validation was done and the best performance across all hyperparameter setups was reported. The cross-validation results on the test split were regarded as the upper limit of meta-classifier. The more common practice is to optimise the meta-classifiers on a held-out set, here using the validation set enriched with training samples that caused errors to at least two base classifiers, and apply the "optimal" parameter setting to the test split. Again, 5-fold cross validation and grid search were used for hyperparameter tuning on the held-out set.

[3] https://scikit-learn.org/stable/modules/naive_bayes.html#categorical-naive-bayes
[4] https://scikit-learn.org/stable/modules/svm.html#kernel-functions
[5] https://scikit-learn.org/stable/modules/svm.html#svm-classification

Then, meta-classifiers were trained using the "optimised" hyperparameters on the whole held-out samples and then were evaluated against the test set.

5.3.2. Results

Table 7-9 show the 5-fold cross-validation results of all the meta-classifiers on the test split. Generally, *k*-NN often was not a strong meta-classifier, and SVM (either SVM-RBF or SVM-Poly) was among the most powerful meta-classifier. On the 11-class scheme, the best performance was 70.81% by SVM-RBF (Table 7), which beat reliability-enhanced soft voting, which was 70.78% (Table 3). However, a fundamental difficulty for voter was the choice of the right size (R) of the selected candidate set, for which there was no systematic way to decide. Classifier stacking removed this complexity by properly weighting the pool of candidates (of size T), which in essence softly excluded "bad" base classifiers by learning to set a small enough weight for them. Classifier stacking thus is a more convenient method to use, especially when the candidate pool is too large to manoeuvre manually. However, the on the 6-class and 2-grade schemes, the best performances of classifier stacking were 76.85% (Table 8) and 89.53% (Table 9) respectively, underperforming the voting counterparts, which reported 77.05% in Table 4 and 89.63% in Table 5. However, the performances were still significantly better than the best single classifier, by (70.78 – 66.16 =) 4.62% on the 11-class scheme, by (76.85 – 74.03 =) 2.82% on the 6-class scheme, by (89.53 – 86.65 =) 2.88% on the 2-grade scheme respectively. Note that, these performances were regarded as oracle values (imprecisely speaking upper-bounds), as they were directly obtained from the test set through cross-validation.

[Table 7–9: See Appendix]

Table 10-12, on the contrary, show the performances of the meta-classifiers that were tuned on the validation split through 5-fold cross validation, together the optimal hyperparameters for each meta-classifier. Table 13 shows the performances of these optimal meta-classifiers on the test split. Now the best performances were around 69.66% (by LDA) on the 11-class scheme (still a 3.50% increase), 77.33% (by SVM-Poly) on the 6class scheme (a significant 3.30% increase), and 88.68% (by *k*-NN when $k = 7$) on the 2-grade scheme (only a 2.03% increase). We note that different meta-classifiers exhibited vastly different performances from each other, they, called level-1 meta-classifiers, also shew abundant variety and possible could be combined further. Indeed, we did some preliminary correlation analysis of the level-1 voters and level-1 meta-classifiers, and found that level-1 voters shew significantly limited diversity among each other (and indeed either further stacked voting or stacked meta-classifier on level-1 voters could

not bring performance improvement), while classifier diversity among level-1 meta-classifiers had much higher potential for further stacking to obtain better performance. So, we will focus on deep stacking of meta-classifiers in the following subsection.

[Table 10–12: See Appendix]

[Table 13: See Appendix]

5.4. Deep Stacking

5.4.1. Experimental Setup

In the experiments, we only tested stacking on level-1 meta-classifiers, because they showed rich diversity. Reliability-enhanced soft voting was used for building the stacked voter (results in Table 14). According to the results in Table 7-9, SVM-RBF and SVM-Poly were chosen to build the stacked meta-classifier, for which 5-fold cross-validation was done on the test set for performance reporting (results in Table 15). Instead of finding the "most diverse" set of level-1 meta-classifiers, we opted to perform an ablation-style study. We ran a series of experiments by first removing each category of level-1 meta-classifiers (i.e., *k*-NN with different *k*'s, NB either CatNB or BerNB, LDA, SVM (either SVM-RBF, SVM-Poly or both) and then removing more level-1 meta-classifiers of two or more categories. We decided to test a large number of such combinations to optimise the final ensemble's performance. Both level-2 voter and level-2 meta-classifier (SVM-RBF and SVM-Poly). In Table 14 and 15, the "¬" symbol means a (number of) meta-classifiers of this type were excluded from the experiment. For *k*-NN's, we also included the *k*'s of the excluded meta-classifiers. What is more, $k = 5$ or 15 did not perform well on the 11-class and 2-grade schemes, so they were pre-excluded from the ablation study. Similarly, $k = 9$–15 were pre-excluded for any experiments on the 6-class scheme. Finally, the "*" symbol means the best configuration among all ablation experiments about *k*-NN. This best configuration was used in further ablation with other classifiers, say the "¬ NB, *k*-NN *" and "¬ LDA, *k*-NN *" rows. The top-3 performances were highlighted by **<u>bold underscored</u>**, **bold**, and <u>underscored</u> fonts respectively.

[Table 14: See Appendix]

[Table 15: See Appendix]

5.4.2. Results

Table 14 shows the performances of reliability-enhanced soft voting on different combinations of level-1 meta-classifiers. The promising aspect was that **voting on meta-classifiers significantly improved the ensemble performances over each individual level-1 meta-classifier** (refer to Table 12): 71.48% v.s. 69.66% on the 11-class scheme, 78.16% v.s. 77.33% on the 6-class scheme, and 89.67% v.s. 88.86% on the 2-grade scheme. The level-2 reliability-enhanced soft voting performances were also better than the cross-validated meta-classifier performances (refer to Table 7): 71.48% v.s. 70.81% on the 11-class scheme, 78.16% v.s. 76.85 % on the 6-class scheme, and 89.67% v.s. 89.53% on the 2-grade scheme. The performances also outperformed the best level-1 majority voters (refer to Table 3-5): 71.48% v.s. 70.78% on the 11-class scheme, 78.16% v.s. 77.05 % on the 6-class scheme, and 89.67% v.s. 89.63% on the 2-grade scheme. This is very encouraging. Meanwhile, it was very clear that using all level-1 meta-classifiers was not able to produce voting performance. On the 11-class and 6-class annotations, the "All" rows significantly underperformed other ablated meta-classifier combinations about *k*-NN. The most extreme case was the 2-grade scheme, where the best level-2 voter performance was obtained without any *k*-NN. Again, it highlights that, for majority voters, **it is a very challenging problem how to select the best subset to combine**.

Table 15 shows the performances of level-2 metaclassifier (SVM-RBF) on all three annotation schemes. First of all, the best level-2 meta-classifiers' performances rivaled the best performances of level-2 voter, and significantly outperformed any single level-1 meta-classifier (refer to Table 7) or reliability-enhanced voter (refer to Table 3-5): 71.75% v.s. 70.81% or 70.71% on the 11-class scheme, 78.03% v.s. 76.85% or 77.05 on the 6-class scheme, and 89.63% v.s. 89.53% or 89.63% on the 2-grade scheme. What is more encouraging was that level-2 meta-classifier was easier to use than level-2 voters. This was demonstrated by the good performances of the level-2 meta-classifiers trained to combine all level-1 meta-classifier predictions (the "All" row in Table 15). Indeed, on the 11-class and 6-class schemes, the best performances were obtained from learning on all level-1 meta-classifiers, while on the 2-grade scheme, this resulted in the third highest performance. This confirmed our previous hypothesis that **level-2 meta-classifier might have the ability to softly exclude unsuitable level-1 meta-classifiers** by setting the correct weights to exclude them from ensembling. Level-2 meta-classifier also stablised the performances of level-1 meta-classifiers, making the final ensemble more robust.

6. Concluding Remarks

Recently, Jiang and Chen (2023) presented a comprehensive study of the wide range of options of citation modelling and their impact on the performance of citation function classification. Their study laid the foundation for building ensemble classifier for citation context analysis, i.e., (one of the) sources of classifier diversity, including the modelling options of in-text citation, citation sentence and citation context. Motivated by their important finding that there is no single best classifier for all citation function categories, the current paper focused on experimenting and evaluating various ways of building ensemble classifiers to improve the performance of citation context analysis, extended from citation function classification to important citation screening.

Our main contribution is the exploitation of three sources of classifier diversity to facilitate ensemble building, namely citation modelling, diversity ranking and diversity re-ranking. The large space of citation modelling options allowed us to design 36 deep learning architectures and trained 180 deep learning models to perform citation context analysis, 5 models per architectures using different random seeds, out of which a diverse set of classifiers were selected as candidate for combination by using five pair-wise diversity measures. One major contribution of the current paper was the proposal of two diversity re-ranking methods to make a better trade-off of classifier performance against classifier diversity. We found that the most diverse base classifiers often tended to be weak, and the strongest ones were often excluded. Diversity ranking alone tended to result in suboptimal ensembling performances. Both our proposed diversity re-ranking methods, namely value-based re-ranking and rank-based re-reranking, had significant impact on the success of ensembles, and rank-based re-ranking method was generally more stable than value-based re-ranking (averaged across five diversity measures).

Three types of ensembling methods were used and evaluated, including majority voting, meta-classifier (equiv. classifier stacking) and deep classifier stacking. Apart from using unweighted hard voting, weighted hard voting, and mean probability soft voting, we also proposed a fourth voting method, called reliability-enhanced soft voting, which defined soft vote as the product of base classifier's performance (reliance) and posterior probability of prediction (confidence). Reliability-enhanced soft voting was proved to be an effective fusing method, evidenced by the results that reliability-enhanced soft voting and weighted hard voting were the two best methods on the 11-class and 6-class annotation schemes for citation function classification. Rank-based re-ranking proved to perform better in combination with both voting methods. Meanwhile, it was demonstrated necessary to build a large enough pool of base classifiers

for diversity analysis and classifier selection. This also implied the value of weak classifiers. The strongest classifiers and a diverse subset of relatively weak classifiers both contributed to performance improvement of ensembles.

However, it was extremely a challenging task to choose the optimal number of base classifiers to fuse, severely harming the usability of majority voting in practice. To circumvent this obstacle, meta-classifiers were trained directly on a large pool of base classifiers after diversity analysis so that the need for further classifier selection was eliminated, with the hope that useless classifiers will be softly ruled out by receiving low weights. On both the citation function classification and important citation screening tasks, kernel support vector machine proved to be the most successful. Significant performance improvement was observed, especially on the 6-class scheme with a 3.50% absolute improvement to 77.33% macro F1 compared to the state of the art reported in Jiang and Chen (2023). More experiments showed that a level-2 ensemble could exploit the diversity among level-1 meta-classifiers to further improve ensemble performance. Reliability-enhanced soft voting and kernel support vector machine (on level-1 meta-classifiers) significantly improved the performance, achieving 5.50% and 5.59% absolute increases respectively on the 11-class citation function scheme, 4.14% and 3.99% on the 6-class scheme, and 4.02% and 3.99% on the task of important citation screening. Again, meta-classifier was proved easy to use because the tedious selection of candidate classifiers was avoided. More specifically, training a level-2 meta-classifier on all level-1 meta-classifiers produced the best (or at least rivaling) ensembling performances, while reliability-enhanced soft voting on all level-1 meta-classifiers was severely suboptimal. Overall, the current study emphasized the necessity of proper diversity analysis and the superiority of deep classifier stacking in building a powerful citation context analysis ensemble.

7. Acknowledgements

The author is partially supported by National Planning Office of Philosophy and Social Science of China (18ZDA238), the International Exchange Scheme of the Royal Society of the United Kingdom (IESR1231175), and the Research Excellent Development Framework award of Coventry University (Nov 2023—July 2024).

8. Statements and Declarations

The author does not have any competing interests to declare.

9. References

Abu-Jbara, A., & Radev, D. (2012). Reference Scope Identification in Citing Sentences. Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies *(NAACL-HLT'12)*, Stroudsburg, PA, USA, 80–90. https://aclanthology.org/N12-1009.

Abu-Jbara, A., Erza, J., & Radev, D. (2013). Purpose and Polarity of Citation: Towards NLP-based Bibliometrics. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'13)*, 596–606. https://aclanthology.org/N13-1067

Agarwal, S., Choubey, L., & Yu, H. (2010). Automatically Classifying the Role of Citations in Biomedical Articles. In *Proceedings of the 2010 Annual Symposium of the American Medical Informatics Association (AMIA'10)*, 11–15. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041379

Aljohani, N.R., Fayoumi, A., Hassan, S.-U., (2021). An in-text citation classification predictive model for a scholarly search system. *Scientometrics*, 126, 5509–5529. https://doi.org/10.1007/s11192-021-03986-z

Aljohani, N.R., Fayoumi, A., & Hassan, S.-U., (2023). A novel focal-loss and class-weight-aware convolutional neural network for the classification of in-text citations. *Journal of Information Science*, *23*(1), 79–92. https://doi.org/10.1177/0165551521991022

Aksela, M. (2003). Comparison of Classifier Selection Methods for Improving Committee Performance. In: Windeatt, T., Roli, F. (eds) Multiple Classifier Systems. MCS 2003. Lecture Notes in Computer Science, vol 2709. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44938-8_9

Akomeah, K.O., Kruschwitz, U., & Ludwig, B. (2021). UR@NLP_A_Team @ GermEval 2021: Ensemble-based Classification of Toxic, Engaging and Fact-Claiming Comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments (GerEval'21)*, 95–99. https://aclanthology.org/2021.germeval-1.14

Bakhti, K., Niu, Z., Yousif, A., & Nyamawe, A.S. (2018). Citation Function Classification Based on Ontologies and Convolutional Neural Networks. In: L. Uden, D. Liberona, J. Ristvej (Eds.) Communications in Computer and Information Science: Vol 870. Learning Technology for Education Challenges. LTEC 2018 (pp. 105–115). Springer, Cham. https://doi.org/10.1007/978-3-319-95522-3_10

Barrault, L., Bojar, Costa-jussà, M.R., Federmann, C., Fishel, M., Graham, Y., et al. (2019). Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation: Shared Task Papers (WMT'19)*, pages 1–61.

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP'19),* 3615–3620. https://aclanthology.org/D19-1371

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, *6*(1), 5–20. https://doi.org/10.1016/j.inffus.2004.04.004

Budi, I., & Yaniasih, Y. (2023). Understanding the meanings of citations using sentiment, role, and citation function classifications. *Scientometrics*. *128*, 735–759. https://doi.org/10.1007/s11192-022-04567-4

Cao, Y., Geddes, T.A., Yang, J.Y.H., & Yang, P. (2020). Ensemble deep learning in bioinformatics. *Nature Machine Intelligence*, *2*, 500–508. https://doi.org/10.1038/s42256-020-0217-y

Cohan, A., Ammar, W., van Zuylen, M., & Cady, F. (2019). Structural Scaffolds for Citation Intent Classification in Scientific Publications. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'19)*, 3856–3896. https://aclanthology.org/N19-1361

Dang, H.N., Lee, K., Henry, S., & Uzuner, Ö. (2020). Ensemble BERT for Classifying Medication-mentioning Tweets. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task (#SMM4H)*, 37–41. https://aclanthology.org/2020.smm4h-1.5

Deng, P., Chen, H., Huang, M., Ruan, X., & Xu, L. (2019). An ensemble CNN method for biomedical entity normalization. In *Proceedings of the 5th Workshop on BioNLP Open Shared Tasks*, 143–149. http://dx.doi.org/10.18653/v1/D19-5721

Dong, C., & Schäfer, U. (2011). Ensemble-style Self-training on Citation Classification. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP'11)*, 623–631. https://aclanthology.org/I11-1070

Garzone, M., & Mercer, R.E. (2000). Towards an Automated Citation Classifier. In *Proceedings of the 2000 Conference of the Canadian Society for Computational Studies of Intelligence (Canadian AI20)*, 337-346. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45486-1_28

Hassan, S.-U., Akram, A., & Haddawy, P. (2017). Identifying Important Citations Using Contextual Information from Full Text. In *Proceedings of the 2017 IEEE/ACM Joint Conference on Digital Libraries (JCDL'17)*, 41–48. https://doi.org/10.1109/JCDL.2017.7991558

Hernández-Alvarez, M., & Gómez, J.M. (2016). Survey about citation context analysis: Tasks, techniques, and resources. *Natural Language Engineering*, *22*(3), 327–349. https://doi.org/10.1017/S1351324915000388

Hernández-Alvarez, M., Gómez, J.M., & Martínez-Barco, P. (2017). Citation function, polarity and influence classification. *Natural Language Engineering*, *23*(4), 561–588. https://doi.org/10.1017/S1351324916000346

Ihsan, I., Rahman, H., Shaikh, A., Sulaiman, A., Rajab, K., & Rajab, A. (2023). Improving in-text citation reason extraction and classification using supervised machine learning techniques. *Computer Speech & Language*, *82*, 101526. https://doi.org/10.1016/j.csl.2023.101526

Iorio, A.D., Nuzzolese, A.G., & Peroni, S. (2013). Towards the automatic identification of the nature of citations. In *Proceedings of the 3rd Workshop on Semantic Publishing (SePublica'13) at the 10th Extended Semantic Web Conference (ESWC'13)*, 63–74. http://ceur-ws.org/Vol-994/paper-06.pdf

Iqbal, S., Hassan, S.-U., Aljohani, N.R., Alelyani, S., Nawaz, R., & Bornmann, L. (2021). A decade of in-text citation analysis based on natural language processing and machine learning techniques: an overview of empirical studies. *Scientometrics*, *126*, 6551–6599. https://doi.org/10.1007/s11192-021-04055-1

Jahrer, M., Töscher, A., & Legenstein, R. (2010). Combining Predictions for Accurate Recommender Systems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*, 693–702. https://doi.org/10.1145/1835804.1835893

Jha, R., Abu-Jbara, A., Qazvinian, V., & Radev, D.R., (2017). NLP-driven citation analysis for scientometrics. *Natural Language Engineering*, *23*(1), 93–130. https://doi.org/10.1017/S1351324915000443

Jiang, X., Cai, C., Fan, W., Liu, T., & Chen, J. (2022). Contextualised Modelling for Effective Citation Function Classification. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval (NLPIR '22)*, 93–103. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3582768.3582769

Jiang, X., & Chen, J. (2023). Contextualised Segment-Wise Citation Function Classification. *Scientometrics*, *128*, 5117–5158. https://doi.org/10.1007/s11192-023-04778-3

Jochim, C., & Schütze, H. (2012). Towards a Generic and Flexible Citation Classifier Based on a Faceted Classification Scheme. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING'12)*, 1343–1358. https://aclanthology.org/C12-1082

Jurgens, D., Kumar, S., Hoover, R., McFarland, D., & Jurafsky, D. (2018). Measuring the Evolution of a Scientific Field through Citation Frames. *Transactions of the Association for Computational Linguistic*, *6*, 391–406. https://doi.org/10.1162/tacl_a_00028

Kaplan, D., Tokunaga, T., & Teufel, S. (2016). Citation Block Determination Using Textual Coherence. *Journal of Information Processing*, *24*(3), 540–553. https://doi.org/10.2197/ipsjjip.24.540

Kuncheva, L.I., & Whitaker, C.J. (2003). Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning*, *51*, 181–207. https://doi.org/10.1023/A:1022859003006

Kuncheva, L.I (2014). *Combining Pattern Classifiers: Methods and Algorithms (2nd Edition)*. Wiley.

Kunnath, S.N., Pride, D., Gyawali, B., & Knoth, P. (2020). Overview of the 2020 WOSP 3C citation context classification task. In *Proceedings of the 8th International Workshop on Mining Scientific Publications (WOSP'2020)*, 75–83. https://aclanthology.org/2020.wosp-1.12

Kunnath, S.N., Herrmannova, D., Pride, D., & Knoth, P. (2022). A meta-analysis of semantic classification of citations. *Quantitative Science Studies*, *2*(4), 1170–1215. https://doi.org/10.1162/qss_a_00159

Lauscher, A., Glavaš, G., Ponzetto, S.P., & Eckert, K. (2017). Investigating convolutional networks and domain-specific embeddings for semantic classification of citations. In *Proceedings of the 6th International Workshop on Mining Scientific Publications (WOSP'17)*, 24–28. https://doi.org/10.1145/3127526.3127531

Lauscher, A., Brandon, K., Kuehl, B., Johnson, S., Jurgens, D., Cohan, A., & Lo, K. (2022). MULTICITE: Modelling realistic citations requires moving beyond the single-sentence single-label setting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (*NAACL'22*), 1875–1889. http://dx.doi.org/10.18653/v1/2022.naacl-main.137

Li, X., He, Y., Meyers, A., & Grishman, R. (2013). Towards Fine-grained Citation Function Classification. In *Proceedings of the 2013 Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP'13)*, 402–407. https://aclanthology.org/R13-1052

Lin, S.-Y., Kung, Y.-C., & Leu, F.-Y. (2022). Predictive intelligence in harmful news identification by BERT-based ensemble learning model with text sentiment analysis. *Information Processing & Management*, *59*(2), page 102872. https://doi.org/10.1016/j.ipm.2022.102872

Lu, W., Meng, R., & Liu, X. (2014). A Deep Scientific Literature Mining-Oriented Framework for Citation Content Annotation. *Journal of Library Science in China*, *40*(214), 93–104. (in Chinese) https://doi.org/10.13530/j.cnki.jlis.140029

Maheshwari, H., Singh, B., & Varma, V. (2021). SciBERT Sentence Representation for Citation Context Classification. In *Proceedings of the Second Workshop on Scholarly Document Processing (SDP'21)*, 130–133. https://aclanthology.org/2021.sdp-1.17

Malmasi, S., & Dras, M. (2018). Native Language Identification With Classifier Stacking and Ensembles. *Computational Linguistics*, *44*(3), 403–446. https://doi.org/10.1162/coli_a_00323

Meng, R., Lu, W., Chi, Y., & Han, S. (2017). Automatic Classification of Citation Function by New Linguistic Features. In *Proceedings of iConference 2017*, 826–830. https://doi.org/10.9776/17349

Moravcsik, M. J. and Murugesan, P. (1975). Some results on the function and quality of citations. *Social Studies of Science*, *5*, 86–92. https://doi.org/10.1177/030631277500500106

Munkhdalai, T., Lalor, J., & Yu, H. (2016). Citation Analysis with Neural Attention Models. In *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis (LOUHI'16)*, 69–77. https://aclanthology.org/W16-6109

Nam, G., Yoon, J., Lee, Y., & Lee, J. (2021). Diversity Matters When Learning From Ensembles. In *Proceedings of 35th Conference on Neural Information Processing Systems (NeurIPS'21)*. https://proceedings.neurips.cc/paper/2021/hash/466473650870501e3600d9a1b4ee5d44-Abstract.html

Nanba, H., Kando, N., & Okumura, M. (2000). Classification of research papers using citation links and citation types: Towards automatic review article generation. In *Proceedings of the 11th ASIS SIG/CR Classification Research Workshop*, 117-134. http://dx.doi.org/10.7152/acro.v11i1.12774

Nazir, S., Asif, M., Ahmad, S., Bukhari, F., Afzal, M. T., & Aljuaid, H. (2020). Important citation identifcation by exploiting content and section-wise in-text citation count. *PLoS ONE*, *15*(3), e0228885. https://doi.org/10.1371/journal.pone.0228885

Nicholson, J.M., Mordaunt, M., Lopez, P., Uppala, A., Rosati, D., Rodrigues, N.P., Grabitz, P., & Rife, S.C. (2021). scite: A smart citation index that displays the context of citations and classifies their intent using deep learning. *Quantitiative Science Studies*, *2*(3), 882–898.

Oesterling, A., Ghosal, A., Yu, H., Xin, R., Baig, Y., Semenova, L., et al. (2021). Multitask Learning for Citation Purpose Classification. In *Proceedings of the Second Workshop on Scholarly Document Processing (SDP'21)*, 134–139. https://aclanthology.org/2021.sdp-1.18

Pride, D., & Knoth, P. (2017). Incidental or influential? - challenges in automatically detecting citation importance using publication full texts. In: J. Kamps, G. Tsakonas, Y. Manolopoulos, L. Iliadis, & I. Karydis (Eds.) Lecture Notes in Computer Science: Vol 10450. Research and Advanced Technology for Digital Libraries. TPDL 2017 (pp. 572–578). https://doi.org/10.1007/978-3-319-67008-9_48

Qadir, Q., & Rillof, E (2012). Ensemble-based Semantic Lexicon Induction for Semantic Tagging. In *Proceedings of *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 199–208. https://aclanthology.org/S12-1028

Qayyum, F., & Afzal, M. T. (2019). Identifcation of important citations by exploiting research articles' metadata and cue-terms from content. *Scientometrics*, 118(1), 21–43. https://doi.org/10.1007/s11192-021-03986-z

Qayyum, F., Jamil, H., Jamil, F., & Kim, D-H. (2021). Towards Potential Content-Based Features Evaluation to Tackle Meaningful Citations. *Symmetry*, 13(10), 1973. https://doi.org/10.3390/sym13101973

Rajani, N.F., Viswanathan, V., Bentor, Y., & Mooney, R.J. (2015). Stacked Ensembles of Information Extractors for Knowledge-Base Population. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP15)*, 177–187. http://dx.doi.org/10.3115/v1/P15-1018

Rajani, N.F., & Mooney, R. (2018). Stacking With Auxiliary Features for Visual Question Answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (NAACL'18)*, 2217–2226. http://dx.doi.org/10.18653/v1/N18-1201

Ruta, D., & Gabrys, B. (2005). Classifier selection for majority voting. *Information Fusion*, *6*(1), 63–81. https://doi.org/10.1016/j.inffus.2004.04.008

Sesmero, M.P., Iglesias, J.A., Magán, E., Ledezma, A.I., & Sanchis, A. (2021). Impact of the learners diversity and combination method on the generation of heterogeneous classifier ensembles. *Applied Soft Computing*, *111*, page 1076689. https://doi.org/10.1016/j.asoc.2021.107689

Sesmero, M.P., Ledezma, A.I., & Sanchis, A. (2015). Generating ensembles of heterogeneous classifiers using Stacked Generalization. *WIREs Data Mining Knowledge Discovery*, *5*, 21–34. https://doi.org/10.1002/widm.1143

Szidarovszky, F., Solt, I., & Tikk, D.. (2010). A Simple Ensemble Method for Hedge Identification. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*, 144–147. https://aclanthology.org/W10-3021

Su, X., Prasad, A., Kan, M.-Y., & Sugiyama, K. (2019). Neural Multi-task Learning for Citation Function and Provenance. In *Proceedings of the 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL19)*, 394–395. https://doi.org/10.1109/JCDL.2019.00122

Teufel, S., Siddharthan, A., & Tidhar, D. (2006a). Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, 103–110. https://aclanthology.org/W06-1613

Teufel, S., Siddharthan, A., & Tidhar, D. (2006b). An annotation scheme for citation function. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue (SIGdial'06)*, 80–87. https://aclanthology.org/W06-1312

Teufel, S. (2010). The Structure of Scientific Articles: Applications to Citation Indexing and Summarization. Centre for the Study of Language & Information.

Tran, H.N., & Kruschwitz, U. (2021). ur-iw-hnt at GermEval 2021: An Ensembling Strategy with Multiple BERT Models. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments (GerEval'21)*, 83–87. https://aclanthology.org/2021.germeval-1.12

Valenzuela, M., Ha, V., & Etzioni, O. (2015). Identifying Meaningful Citations. *In Proceedings of the Workshops of Scholarly Big Data: AI Perspectives, Challenges, and Ideas at the 29th AAAI Conference on Artificial Intelligence (BigScholar'15)*. https://allenai.org/data/meaningful-citations

Varanasi, K.K., Ghosal, T., Tiwary, P., & Singh, M. (2021). IITP-CUNI@3C: Supervised Approaches for Citation Classification (Task A) and Citation Significance Detection (Task B). In *Proceedings of the Second Workshop on Scholarly Document Processing (SDP'21)*, 140–145. https://aclanthology.org/2021.sdp-1.19

Wan, X., & Liu, F. (2014). Are all literature citations equally Important? Automatic citation strength estimation and its applications. *Journal of the Association for Information Science and Technology*, *65*(9), 1929–1938. https://doi.org/10.1002/asi.23083

Wang, Y., Wu, L., Xia, Y., Qin, T., Zhai, C., & Liu, T.-Y. (2020). Transductive Ensemble Learning for Neural Machine Translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(4), 6291–6298. https://doi.org/10.1609/aaai.v34i04.6097

Wang, M., Zhang, J., Jiao, S., Zhang, X., Zhu, N., & Chen, G. (2020). Important citation identification by exploiting the syntactic and contextual information of citations. *Scientometrics*, *125*, 2109–2129. https://doi.org/10.1007/s11192-020-03677-1

Wu, H., Wang, H. (2005). Improving Statistical Word Alignment with Ensemble Methods. In *Proceedings of the Second International Joint Conference on Natural Language Processing* (*IJCNLP'05*). http://dx.doi.org/10.1007/11562214_41

Xiao, Y., Wu, J., Lin, Z., & Zhao, D. (2018). A deep learning-based multi-model ensemble method for cancer prediction. *Computer Methods and Programs in Biomedicine*, *153*, 1–9. https://doi.org/10.1016/j.cmpb.2017.09.005

Yousif, A., Niu, Z., Chambua, J., & YounasKhana, Z. (2019). Multi-task learning model based on recurrent convolutional neural networks for citation sentiment and purpose classification. *Neurocomputing*, *335*, 195–205. https://doi.org/10.1016/j.neucom.2019.01.021

Zhang, G., Ding, Y., & Milojević, S. (2013). Citation content analysis (CCA): A framework for syntactic and semantic analysis of citation content. *Journal of the American Society for Information Science and Technology, 64*(7), 1490–1503. https://doi.org/10.1002/asi.22850

Zhang, Y., Wang, Y., Sheng, Q.Z., Mahmood, A., Zhang, W.E., & Zhao, R. (2021). TDM-CFC: Towards Document-Level Multi-label Citation Function Classification. In: W. Zhang, L. Zou, Z. Maamar, & L. Chen (Eds.) Lecture Notes in Computer Science: Vol 13081. Web Information Systems Engineering – WISE 2021 (pp. 363–376). Springer, Cham. https://doi.org/10.1007/978-3-030-91560-5_26

Zhang, Y., Zhao, R., Wang, Y., Chen, H., Mahmood, A., Zaib, M., Zhang W. E., & Sheng, Q. Z. (2022). Towards employing native information in citation function classification. *Scientometrics*, *127*, 6557–6577. https://doi.org/10.1007/s11192-021-04242-0

Zhao, H., Luo, Z., Feng, C., Zheng, A., & Liu, X. (2019). A Context-based Framework for Modelling the Role and Function of On-line Resource Citations in Scientific Literature. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*, 5206–5215. https://aclanthology.org/D19-1524

Zheng, A., Zhao, H., Luo, Z., Feng, C., Liu, X., & Ye, Y. (2021). Improving On-line Scientific Resource Profiling by Exploiting Resource Citation Information in the Literature. *Information Processing & Management*, *58*(5), 102638. https://doi.org/10.1016/j.ipm.2021.102638

Zhou, Z.-H. (2014). *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall.

10. Appendix (Tables)

Model	citseg	ctx_type	Encoding methods			11-class			6-class			2-grade		
			cita_pooler	ctx_pooler	sent_pooler	best	avg	std	best	avg	std	best	avg	std
seq-01	O	sequential	max_pool	CLS	N/A	63.93	62.72	1.11	**74.03**	70.88	1.87	84.27	83.37	1.29
seq-02	O	sequential	max_pool	max_pool	N/A	63.21	62.61	0.45	70.23	68.25	1.60	85.49	84.25	0.70
seq-03	O	sequential	max_pool	self_attnd	N/A	64.26	62.82	1.04	70.99	68.86	1.71	86.16	85.37	0.86
seq-04	O	sequential	self_attnd	CLS	N/A	63.12	62.07	1.00	69.96	68.22	1.58	84.74	84.13	0.53
seq-05	O	sequential	self_attnd	max_pool	N/A	64.12	62.82	1.20	71.56	69.05	1.85	85.13	83.46	1.15
seq-06	O	sequential	self_attnd	self_attnd	N/A	**65.12**	63.05	1.60	72.19	69.81	1.37	86.04	84.67	0.80
seq-07	O	sequential	X	CLS	N/A	64.65	61.01	2.21	71.48	69.75	1.07	84.80	83.99	0.48
seq-08	O	sequential	X	max_pool	N/A	**66.16**	63.53	1.55	70.98	69.90	1.21	85.88	84.21	1.04
seq-09	O	sequential	X	self_attnd	N/A	63.92	62.80	0.89	71.91	69.66	1.47	86.20	84.77	0.79
seq-10	O	sequential	max_pool	X	N/A	63.93	62.72	1.11	71.89	70.18	1.77	85.82	84.57	0.81
seq-11	O	sequential	self_attnd	X	N/A	64.42	63.01	0.89	71.32	69.69	1.01	86.00	85.11	0.57
seq-12	O	sequential	X	X	N/A	64.93	63.50	1.04	**73.56**	70.22	2.44	86.00	84.74	0.68
hie-01	O	hierarchical	SEP	max_pool	SEP	62.78	61.76	0.89	69.39	68.42	1.25	84.00	83.81	0.15
hie-02	O	hierarchical	SEP	self_attnd	SEP	61.42	61.42	0.96	71.08	69.87	1.51	84.90	83.57	0.76
hie-03	O	hierarchical	max_pool	max_pool	SEP	63.30	63.30	1.12	71.71	69.60	1.36	84.00	83.81	0.15
hie-04	O	hierarchical	max_pool	self_attnd	SEP	63.79	63.79	1.71	72.10	70.25	1.69	84.90	83.57	0.76
hie-05	O	hierarchical	self_attnd	max_pool	SEP	63.69	63.69	2.21	70.09	67.83	1.74	84.42	83.41	1.17
hie-06	O	hierarchical	self_attnd	self_attnd	SEP	63.79	63.79	1.71	72.10	70.25	1.69	84.90	83.57	0.76
hie-07	O	hierarchical	max_pool	max_pool	max_pool	62.63	62.16	0.51	70.22	67.94	1.38	85.60	84.18	1.07
hie-08	O	hierarchical	max_pool	self_attnd	max_pool	65.02	62.10	2.24	69.77	68.24	1.33	84.41	83.53	0.99
hie-09	O	hierarchical	max_pool	max_pool	self_attnd	63.38	62.45	0.59	72.11	70.07	1.8	85.74	84.06	1.10
hie-10	O	hierarchical	max_pool	self_attnd	self_attnd	63.31	62.44	0.89	71.40	70.02	1.03	85.49	84.17	1.18
hie-11	O	hierarchical	self_attnd	max_pool	max_pool	64.46	62.17	1.99	72.38	69.33	3.07	85.82	84.45	1.31
hie-12	O	hierarchical	self_attnd	self_attnd	max_pool	63.43	62.26	0.83	70.78	69.56	1.57	85.41	84.55	0.59
hie-13	O	hierarchical	self_attnd	max_pool	self_attnd	64.99	63.56	1.15	71.49	69.52	1.66	85.93	84.80	0.70
hie-14	O	hierarchical	self_attnd	self_attnd	self_attnd	63.16	62.09	1.02	71.32	68.35	2.22	86.45	85.88	0.55
hie-15	O	hierarchical	X	max_pool	SEP	61.17	59.98	1.14	73.24	70.19	2.41	84.49	83.69	0.53
hie-16	O	hierarchical	X	self_attnd	SEP	63.22	62.25	0.89	71.56	70.40	1.18	85.24	84.14	1.00
hie-17	O	hierarchical	X	max_pool	max_pool	64.56	64.16	0.39	70.90	70.04	0.94	**86.65**	84.41	1.37
hie-18	O	hierarchical	X	self_attnd	max_pool	64.95	62.82	1.64	72.09	69.35	2.11	85.05	83.64	1.16
hie-19	O	hierarchical	X	max_pool	self_attnd	62.62	61.61	1.18	71.89	70.48	1.04	85.11	83.98	0.96
hie-20	O	hierarchical	X	self_attnd	self_attnd	63.15	62.39	0.60	70.72	69.75	1.1	**86.46**	84.15	1.66
hie-21	O	hierarchical	SEP	X	N/A	63.48	61.27	1.39	72.81	70.96	1.32	85.37	84.10	1.13
hie-22	O	hierarchical	max_pool	X	N/A	63.48	61.27	1.39	72.81	70.96	1.32	85.37	84.10	1.13
hie-23	O	hierarchical	self_attnd	X	N/A	62.55	61.09	1.05	70.38	69.28	1.19	86.12	84.52	0.89
hie-24	O	hierarchical	X	X	N/A	64.37	62.80	1.51	72.07	71.21	0.70	85.88	84.94	0.69

Table 2. Base Classifiers of Citation Function Classification and Their Performances.

T		50			40			30			20			10		
Re-rank		¬RR	RR_Rnk	RR_Val	¬RR	RR_Rnk	RR_Val	¬RR	RR_Rnk	RR_Val	¬RR	RR_Rnk	RR_Val	¬RR	RR_Rnk	RR_Val
HARD – UNWEIGHT	Div_{CC}	70.24//R=44	70.42//R=40	70.23//R=29	70.17//R=29	70.57//R=36	70.20//R=25	69.72//R=30	**70.37//R=27**	69.72//R=30	69.49//R=20	69.80//R=19	69.49//R=20	69.35//R=8	69.01//R=5	69.35//R=8
	Div_{DF}	69.75//R=50	70.05//R=31	69.96//R=25	69.83//R=17	70.09//R=37	69.90//R=20	69.83//R=29	69.92//R=24	70.08//R=10	70.13//R=12	**70.64//R=14**	70.44//R=13	***69.83//R=5***	69.46//R=6	**69.83//R=5**
	Div_{Q}	69.94//R=45	70.35//R=24	70.23//R=29	70.43//R=38	70.50//R=31	70.31//R=30	70.27//R=25	**70.37//R=27**	70.02//R=27	69.49//R=20	69.80//R=19	69.53//R=12	69.35//R=8	69.01//R=5	69.30//R=9
	Div_{RE}	69.96//R=47	70.06//R=27	69.92//R=48	70.14//R=39	70.45//R=35	70.23//R=18	69.72//R=30	70.53//R=26	69.72//R=30	69.57//R=14	69.80//R=19	70.10//R=16	69.35//R=8	69.01//R=5	69.35//R=8
	Div_{DM}	69.98//R=38	70.39//R=22	**70.64//R=38**	70.43//R=19	70.37//R=17	**70.70//R=29**	69.72//R=30	**70.37//R=28**	69.77//R=27	69.49//R=20	69.80//R=19	69.69//R=9	69.35//R=8	69.01//R=5	69.35//R=8
	AVG	69.97	**70.25**	70.20	70.20	**70.40**	70.27	69.85	**70.31**	69.86	69.63	**69.97**	69.85	***69.45***	69.10	**69.44**
HARD – WEIGHTED	Div_{CC}	70.21//R=44	70.49//R=23	70.33//R=29	70.42//R=28	70.69//R=31	70.45//R=17	69.99//R=30	70.48//R=27	69.82//R=30	69.52//R=20	69.97//K-20	69.97//R=20	69.53//R=9	69.28//R=8	69.91//R=8
	Div_{DF}	69.79//R=17	70.22//R=31	70.26//R=24	69.92//R=11	70.39//R=11	70.01//R=20	70.13//R=9	70.37//R=20	70.19//R=12	70.33//R=16	**70.72//R=14**	70.67//R=10	*71.28//R=5*	70.37//R=5	69.56//R=5
	Div_{Q}	70.05//R=45	70.49//R=23	70.40//R=42	70.74//R=33	70.69//R=31	**70.78//R=22**	70.18//R=25	70.65//R=25	70.28//R=24	69.40//R=20	69.97//K-20	69.97//R=20	69.59//R=9	69.28//R=8	69.83//R=9
	Div_{RE}	69.89//R=32	70.09//R=40	70.05//R=36	70.19//R=18	70.52//R=36	70.29//R=38	69.95//R=30	70.40//R=25	69.82//R=30	69.98//R=16	70.19//R=18	69.97//R=20	69.53//R=9	69.28//R=8	69.91//R=8
	Div_{DM}	70.11//R=38	70.49//R=23	**70.58//R=39**	70.38//R=19	70.39//R=20	70.58//R=29	69.66//R=30	**70.69//R=31**	69.87//R=23	69.65//R=18	69.97//K-20	70.01//R=10	69.64//R=9	69.28//R=8	69.91//R=8
	AVG	70.01	**70.36**	70.32	70.33	**70.54**	70.42	69.98	**70.52**	70.00	69.78	**70.16**	70.12	***69.91***	69.50	69.82
SOFT – MEAN	Div_{CC}	69.73//R=42	**70.66//R=23**	70.07//R=34	69.92//R=15	**70.55//R=17**	70.04//R=24	69.90//R=29	69.76//R=28	69.90//R=29	70.00//R=12	70.15//R=14	69.91//R=13	***69.75//R=5***	69.67//R=10	**69.75//R=5**
	Div_{DF}	69.67//R=15	69.99//R=22	69.90//R=24	69.50//R=37	69.74//R=17	69.73//R=14	69.76//R=26	70.02//R=24	70.28//R=8	70.65//R=12	70.13//R=14	70.76//R=10	69.70//R=5	69.67//R=10	69.70//R=5
	Div_{Q}	69.63//R=24	**70.66//R=23**	70.38//R=28	69.92//R=15	70.27//R=16	70.10//R=16	69.90//R=29	70.03//R=21	69.92//R=21	69.67//R=14	69.69//R=12	69.91//R=17	***69.75//R=5***	69.67//R=10	69.67//R=10
	Div_{RE}	69.50//R=6	70.11//R=27	70.06//R=21	69.98//R=19	70.36//R=17	69.98//R=19	69.90//R=29	69.66//R=25	69.90//R=29	70.00//R=12	70.15//R=14	69.91//R=13	69.72//R=8	69.67//R=10	69.72//R=8
	Div_{DM}	69.63//R=24	**70.66//R=23**	69.95//R=20	69.92//R=15	70.27//R=16	69.98//R=24	69.90//R=29	**70.48//R=27**	69.90//R=29	69.51//R=15	69.69//R=14	69.78//R=13	***69.75//R=5***	69.67//R=10	**69.75//R=5**
	AVG	69.63	**70.42**	70.07	69.85	**70.24**	69.97	69.87	**69.99**	**69.98**	69.97	69.96	**70.05**	***69.73***	69.67	**69.72**
SOFT – RELIABILITY	Div_{CC}	70.17//R=44	**70.54//R=22**	70.33//R=29	70.42//R=28	70.19//R=17	70.11//R=24	69.48//R=28	70.26//R=28	69.59//R=26	69.80//R=9	69.83//R=12	69.58//R=11	69.46//R=10	69.46//R=10	69.46//R=10
	Div_{DF}	69.95//R=17	69.87//R=29	69.89//R=10	69.86//R=17	70.12//R=32	69.89//R=15	69.83//R=29	70.06//R=15	70.06//R=15	70.41//R=12	**70.55//R=14**	70.43//R=14	69.77//R=5	69.46//R=10	69.77//R=5
	Div_{Q}	69.82//R=45	70.42//R=24	70.33//R=29	70.42//R=28	70.41//R=31	70.25//R=15	70.19//R=25	**70.50//R=25**	70.03//R=27	69.48//R=20	69.83//R=12	69.83//R=12	69.46//R=10	69.46//R=10	69.46//R=10
	Div_{RE}	69.85//R=47	70.11//R=34	69.84//R=25	70.12//R=28	70.43//R=37	70.12//R=28	69.31//R=30	70.46//R=26	69.68//R=28	69.80//R=9	69.91//R=13	69.81//R=16	69.46//R=10	69.46//R=10	69.46//R=10
	Div_{DM}	69.93//R=43	**70.54//R=32**	70.30//R=39	70.34//R=38	70.26//R=30	**70.46//R=24**	69.31//R=30	70.38//R=25	69.79//R=27	69.48//R=20	69.70//R=19	70.04//R=9	69.46//R=10	69.46//R=10	69.46//R=10
	AVG	69.94	**70.30**	70.14	70.23	**70.28**	70.17	69.62	**70.33**	69.83	69.79	**69.96**	69.94	***69.52***	69.46	**69.52**

Table 3. Performances of majority voting-based ensembles for 11-class citation function classification.

T		50			40			30			20			10		
Re-rank		¬RR	RR Rnk	RR Val	¬RR	RR Rnk	RR Val	¬RR	RR Rnk	RR Val	¬RR	RR Rnk	RR Val	¬RR	RR Rnk	RR Val
HARD – UNWEIGHT	Div_{CC}	75.94 //R=11	**76.93//R =18**	76.52//R =10	75.85//R =9	75.98 //R=23	75.65//R =10	75.86//R =20	76.04//R =9	75.79//R =23	***76.33//R =5***	76.24//R =17	76.16//R =8	***75.71//R =9***	75.65//R =10	75.65//R =10
	Div_{DF}	76.41 //R=17	76.45//R =30	76.54//R =31	75.53//R =11	**76.66 //R=19**	76.15//R =16	75.62//R =28	75.62//R =28	75.67//R =25	***76.31//R =13***	76.01//R =18	75.07//R =14	***75.71//R =9***	75.65//R =10	75.65//R =10
	Div_{Q}	75.94 //R=11	76.62//R =17	76.25//R =17	75.81//R =7	75.98 //R=23	76.14//R =23	76.07//R =19	75.78//R =15	75.74//R =24	***76.33//R =5***	76.24//R =17	**76.37//R =7**	***75.71//R =9***	75.65//R =10	75.65//R =10
	Div_{RE}	76.12 //R=22	76.44//R =16	76.30//R =26	75.81//R =7	75.98 //R=23	75.81//R =7	75.86//R =20	76.10//R =28	75.67//R =24	***76.33//R =5***	76.24//R =17	76.09//R =8	***75.71//R =9***	75.65//R =10	75.65//R =10
	Div_{DM}	76.14 //R=22	76.81//R =16	76.11//R =18	75.75//R =9	75.90 //R=23	75.88//R =22	***76.19//R =20***	76.04//R =9	75.74//R =26	***76.33//R =5***	76.16//R =8	75.94//R =5	***75.71//R =9***	75.65//R =10	75.65//R =10
	AVG	76.11	**76.65**	76.34	75.75	**76.10**	75.93	***75.92***	**75.92**	75.72	***76.33***	76.18	75.93	***75.71***	75.65	75.65
HARD – WEIGHTED	Div_{CC}	76.24 //R=23	76.89//R =18	76.83//R =10	75.88//R =9	75.91//R =24	75.67//R =18	76.02//R =21	76.07//R =9	75.76//R =24	76.05//R =5	76.22//R =6	76.26//R =7	***75.92//R =10***	75.92//R =10	75.92//R =10
	Div_{DF}	76.57 //R=11	76.41//R =30	76.43//R =31	75.79//R =17	**76.67//R =19**	75.72//R =16	76.13//R =22	75.54//R =10	75.77//R =25	***76.60//R =5***	76.23//R =7	75.72//R =13	***75.92//R =10***	75.92//R =10	75.92//R =10
	Div_{Q}	76.09 //R=11	**76.94//R =16**	76.21//R =17	75.80//R =13	75.81//R =23	76.05//R =23	75.96//R =25	76.00//R =26	75.85//R =26	76.05//R =5	76.22//R =6	76.22//R =6	***75.92//R =10***	75.92//R =10	75.92//R =10
	Div_{RE}	76.55 //R=14	76.59//R =16	76.53//R =15	75.73//R =10	75.88//R =23	75.79//R =20	75.93//R =21	76.38//R =28	75.76//R =24	76.12//R =14	76.22//R =6	76.04//R =8	***75.92//R =10***	75.92//R =10	75.92//R =10
	Div_{DM}	76.34 //R=22	76.88//R =17	76.47//R =18	75.97//R =9	75.91//R =24	75.97//R =22	76.47//R =17	**76.50//R =12**	75.89//R =24	76.26//R =13	75.99//R =8	76.26//R =7	***75.92//R =10***	75.92//R =10	75.92//R =10
	AVG	76.36	**76.74**	76.49	75.83	**76.04**	75.84	***76.10***	**76.10**	75.81	***76.22***	76.18	76.10	***75.92***	**75.92**	**75.92**
SOFT – MEAN	Div_{CC}	75.61//R =33	75.82//R =18	76.15//R =10	76.09//R =23	**76.66//R =15**	75.47//R =17	76.07//R =20	76.11//R =12	75.48//R =21	***75.99//R =5***	75.48//R =6	75.72//R =8	75.72//R =5	74.93//R =10	74.93//R =10
	Div_{DF}	76.43//R =25	75.66//R =16	**76.54//R =16**	75.88//R =17	76.43//R =18	**76.66//R =16**	75.94//R =12	**76.48//R =17**	76.26//R =15	75.63//R =13	75.74//R =9	75.49//R =5	***75.79//R =5***	74.93//R =10	74.93//R =10
	Div_{Q}	75.66//R =24	75.83//R =18	75.82//R =7	75.65//R =22	76.10//R =17	75.88//R =23	76.01//R =19	75.82//R =15	76.20//R =14	***75.99//R =5***	75.73//R =13	75.72//R =8	75.72//R =5	74.93//R =10	75.07//R =7
	Div_{RE}	75.70//R =16	75.62//R =15	75.79//R =10	76.09//R =23	76.55//R =16	75.87//R =24	76.07//R =20	76.11//R =12	75.85//R =28	***75.99//R =5***	75.48//R =6	75.34//R =19	75.72//R =5	74.93//R =10	74.93//R =10
	Div_{DM}	75.62//R =7	76.03//R =14	75.41//R =20	75.54//R =11	76.56//R =16	75.82//R =21	76.01//R =19	75.82//R =15	75.61//R =13	***75.99//R =5***	75.73//R =13	75.34//R =19	75.72//R =5	75.07//R =7	75.07//R =7
	AVG	75.80	75.79	**75.94**	75.85	**76.46**	75.94	76.02	**76.24**	75.88	***75.92***	75.63	75.52	***75.73***	74.96	74.99
SOFT – RELIABILITY	Div_{CC}	76.06//R =22	**77.05//R =18**	76.13//R =18	75.89//R =7	75.96//R =22	75.95//R =17	75.80//R =25	76.13//R =9	75.73//R =29	75.84//R =19	76.35//R =17	75.88//R =8	***76.35//R =9***	75.70//R =7	75.70//R =7
	Div_{DF}	76.49//R =25	76.41//R =30	76.61//R =31	75.46//R =17	**76.59//R =19**	**76.60//R =16**	75.78//R =22	75.75//R =10	75.71//R =25	75.74//R =13	75.91//R =17	75.65//R =13	***76.35//R =9***	75.70//R =7	75.70//R =7
	Div_{Q}	76.09//R =11	76.88//R =16	76.30//R =18	75.89//R =7	75.60//R =23	76.19//R =23	76.20//R =19	75.84//R =15	76.26//R =14	76.02//R =14	76.35//R =17	76.20//R =6	***76.35//R =9***	75.70//R =7	75.70//R =7
	Div_{RE}	76.06//R =22	76.43//R =16	76.48//R =20	75.89//R =7	75.96//R =22	75.89//R =7	75.72//R =20	75.88//R =12	75.64//R =28	***76.44//R =8***	76.35//R =17	**76.44//R =8**	***76.35//R =9***	75.70//R =7	75.70//R =7
	Div_{DM}	76.27//R =22	76.82//R =16	76.13//R =18	75.82//R =13	76.15//R =13	75.91//R =22	***76.20//R =19***	76.13//R =9	75.73//R =29	75.84//R =17	76.20//R =6	75.84//R =17	***76.35//R =9***	75.70//R =7	75.70//R =7
	AVG	76.19	**76.72**	76.33	75.79	76.05	**76.11**	***75.94***	**75.95**	75.81	75.98	**76.23**	76.00	***76.35***	75.70	75.70

Table 4. Performances of majority voting-based ensembles for 6-class citation function classification.

T		50			40			30			20			10		
Re-rank		¬RR	RR_Rnk	RR_Val	¬RR	RR_Rnk	RR_Val	¬RR	RR_Rnk	RR_Val	¬RR	RR_Rnk	RR_Val	¬RR	RR_Rnk	RR_Val
HARD – UNWEIGHT	Div_{CC}	89.06//R=19	89.38//R=11	89.06//R=19	89.18//R=17	**89.22//R=17**	89.18//R=17	***89.63//R=15***	89.43//R=11	**89.63//R=15**	89.18//R=17	88.99//R=20	**89.34//R=17**	***88.93//R=9***	88.90//R=9	**88.93//R=9**
	Div_{DF}	88.65//R=23	89.22//R=13	88.74//R=19	88.79//R=22	88.97//R=9	88.86//R=21	89.02//R=19	89.22//R=11	89.03//R=18	89.18//R=17	88.99//R=20	89.18//R=17	88.78//R=10	88.90//R=7	88.78//R=10
	Div_{Q}	89.26//R=23	89.38//R=11	**89.63//R=11**	89.18//R=17	89.18//R=9	89.22//R=13	***89.63//R=15***	89.06//R=27	88.90//R=19	89.18//R=17	88.99//R=20	88.99//R=20	***88.93//R=9***	88.90//R=9	**88.93//R=9**
	Div_{RE}	88.79//R=22	89.38//R=9	88.88//R=20	88.65//R=23	89.22//R=17	89.18//R=17	***89.63//R=15***	89.22//R=11	**89.63//R=15**	89.18//R=17	88.99//R=20	89.18//R=17	***88.93//R=9***	88.90//R=9	**88.93//R=9**
	Div_{DM}	89.26//R=21	89.55//R=9	89.15//R=22	88.77//R=19	89.18//R=9	88.86//R=17	***89.63//R=15***	89.02//R=15	89.26//R=17	***89.34//R=17***	88.99//R=20	**89.34//R=17**	***88.93//R=9***	88.90//R=9	**88.93//R=9**
	AVG	89.00	**89.38**	89.09	88.91	**89.15**	89.06	***89.51***	89.19	89.29	***89.21***	88.99	**89.21**	***88.90***	**88.90**	**88.90**
HARD – WEIGHTED	Div_{CC}	89.06//R=18	89.43//R=10	89.06//R=19	89.18//R=17	89.22//R=17	89.38//R=18	***89.63//R=15***	89.43//R=11	**89.63//R=15**	89.18//R=17	89.14//R=20	89.34//R=17	89.06//R=10	89.02//R=8	89.02//R=8
	Div_{DF}	88.86//R=20	89.43//R=28	89.02//R=28	89.10//R=14	89.26//R=24	89.10//R=20	89.10//R=12	89.22//R=11	89.22//R=18	89.18//R=17	89.14//R=20	89.34//R=18	89.14//R=10	89.02//R=6	88.61//R=10
	Div_{Q}	89.26//R=23	89.38//R=11	**89.63//R=11**	89.18//R=17	89.18//R=9	89.22//R=13	***89.63//R=15***	89.26//R=26	89.06//R=22	89.18//R=14	89.14//R=20	89.14//R=20	89.31//R=8	89.02//R=8	89.02//R=8
	Div_{RE}	88.86//R=27	89.38//R=9	88.90//R=20	88.90//R=20	89.38//R=18	89.18//R=17	***89.63//R=15***	89.26//R=26	**89.63//R=15**	***89.38//R=14***	89.14//R=20	89.34//R=18	***89.38//R=10***	89.02//R=8	89.02//R=8
	Div_{DM}	89.26//R=21	89.55//R=9	89.38//R=22	88.90//R=20	89.18//R=9	**89.71//R=12**	***89.63//R=15***	89.22//R=26	89.26//R=16	89.34//R=17	89.14//R=20	89.34//R=17	89.06//R=10	88.90//R=9	89.02//R=8
	AVG	89.06	**89.43**	89.20	89.05	89.24	**89.32**	***89.52***	89.28	89.36	89.25	89.14	**89.30**	***89.19***	89.00	88.94
SOFT – MEAN	Div_{CC}	88.58//R=16	**89.38//R=11**	88.74//R=17	88.77//R=17	**89.18//R=9**	88.77//R=17	***89.63//R=15***	89.26//R=11	**89.63//R=15**	89.02//R=18	88.65//R=18	89.34//R=17	88.61//R=7	88.70//R=9	88.61//R=7
	Div_{DF}	88.49//R=23	89.31//R=10	88.74//R=22	88.62//R=16	88.86//R=8	88.94//R=14	89.06//R=19	89.06//R=11	88.90//R=18	89.02//R=18	88.65//R=18	89.02//R=18	88.72//R=7	**88.86//R=7**	88.61//R=7
	Div_{Q}	89.31//R=20	**89.38//R=11**	89.34//R=9	88.77//R=17	**89.18//R=9**	88.81//R=6	***89.63//R=15***	89.06//R=27	88.78//R=9	89.02//R=18	88.77//R=13	88.65//R=18	88.61//R=7	88.70//R=9	88.56//R9
	Div_{RE}	88.58//R=16	89.38//R=11	88.74//R=17	88.65//R=18	**89.18//R=9**	88.77//R=17	***89.63//R=15***	89.06//R=27	**89.63//R=15**	89.02//R=18	88.81//R=12	89.02//R=19	88.61//R=7	88.70//R=9	88.61//R=7
	Div_{DM}	89.31//R=20	89.34//R=9	89.31//R=20	88.44//R=19	**89.18//R=9**	88.65//R=17	***89.63//R=15***	89.10//R=24	89.26//R=17	***89.34//R=17***	88.81//R=12	89.34//R=17	88.56//R=9	88.70//R=6	88.56//R9
	AVG	88.85	**89.36**	88.97	88.65	**89.12**	88.79	***89.52***	89.11	89.24	***89.08***	88.74	89.07	88.62	**88.73**	88.59
SOFT – RELIABILITY	Div_{CC}	89.06//R=19	89.38//R=11	89.06//R=19	89.18//R=17	**89.22//R=17**	89.18//R=17	***89.63//R=15***	89.22//R=11	89.26//R=17	89.18//R=17	88.77//R=19	89.34//R=17	***88.93//R=9***	88.90//R=9	***88.93//R=9***
	Div_{DF}	88.65//R=23	**89.59//R=9**	88.74//R=19	88.65//R=23	88.97//R=9	88.94//R=14	89.06//R=19	89.02//R=15	89.02//R=17	89.18//R=17	88.77//R=19	89.18//R=17	88.72//R=7	88.90//R=7	88.56//R=7
	Div_{Q}	89.26//R=23	89.38//R=11	89.63//R=11	89.18//R=17	89.18//R=9	**89.22//R=13**	***89.63//R=15***	89.22//R=11	88.90//R=19	89.18//R=17	88.93//R=12	88.77//R=19	***88.93//R=9***	88.90//R=9	***88.93//R=9***
	Div_{RE}	88.74//R=23	89.38//R=11	88.86//R=27	88.65//R=17	**89.22//R=17**	89.18//R=17	***89.63//R=15***	89.22//R=11	89.63//R=15	89.18//R=17	88.81//R=12	89.18//R=17	***88.93//R=9***	88.90//R=9	***88.93//R=9***
	Div_{DM}	89.26//R=21	89.55//R=9	89.10//R=20	88.77//R=19	89.18//R=9	88.86//R=17	***89.63//R=15***	89.02//R=15	89.26//R=17	***89.34//R=17***	88.81//R=12	89.34//R=17	***88.93//R=9***	88.90//R=9	***88.93//R=9***
	AVG	88.99	**89.46**	89.08	88.89	**89.15**	89.08	***89.52***	89.14	89.21	***89.21***	88.82	89.16	***88.89***	**88.90**	88.86

Table 5. Performances of majority voting-based ensembles for 2-grade important citation screening.

Meta-classifier	Hyperparameters	Explanations	Range
k-NN	weight	Method of weighting nearest neighbours according to their distances to the central instance.	["uniform", "distance"]
CatNB	α	The additive value used for smoothing the Naïve Bayes estimates with respect to each category.	[0.0001, 0.0002, …, 0.001, 0.002, …, 0.01, 0.02, …, 0.1, 0.2, 1.0, 1.1, …, 6.0]
BerNB	α	Same as above.	Same as above
LDA	λ	The regularisation factor for the shrinkage estimator of covariance matrices in situations where the number of training samples is small compared to the number of features[6]: $\Sigma = (1-\lambda)\Sigma + \lambda \mathrm{I}$	[0.00, 0.05, …, 0.90, 0.95, 1.00]
SVM-RBF	C	Regularisation coefficient which controls the trade-off between errors on training data and margin maximization.	[0.5, 0.6, …, 1.0, 1.1, 1.2, …, 2, 3, …, 10]
	γ	The kernel distance coefficient in $\kappa(x, x') = \exp(-\gamma \|x - x'\|^2)$.	[0.002, 0.004, …, 0.01, 0.02, 0.04, …, 0.10, 0.12, 0.14, …, 0.20]
SVM-Poly	C	Regularisation coefficient which controls the trade-off between errors on training data and margin maximization.	Same as above
	γ	The kernel distance coefficient in $\kappa(x, x') = (\gamma\langle x, x'\rangle + r)^d$, where r was defaulted to 0.	Same as above
	d	The degree of polynomial.	[2, 3, 4]

Table 6. Summary of hyperparameters of meta-classifiers.

[6] https://en.wikipedia.org/wiki/Linear_discriminant_analysis#Practical_use

RR_Rnk; *R* =	50	40	30	20	10	BEST,
CatNB	**70.33**	69.47	69.55	69.04	68.34	**70.33//R=50**
Dis, α	N/A, 0.0008	DF, 0.0004	DF, 0.0007	DM, 0.8	DF, 0.02	0.0008
k-NN (*k* = 7)	67.1	68.3	69.12	**69.82**	68.32	**69.82//R=20**
Dis, *weighting*	N/A, uniform	DF, uniform	DF, uniform	DF, uniform	RE, uniform	DF, uniform
k-NN (*k* = 9)	67.9	69.14	69.47	**69.59**	67.92	**69.59//R=20**
Dis, *weighting*	N/A, uniform	QS/DM, distance	RE, distance	DF, uniform	CC/QS/DM, uniform	DF, uniform
k-NN (*k* = 11)	67.9	68.84	68.81	**69.33**	67.45	**69.33//R=20**
Dis, *weighting*	N/A, uniform	DF, distance	DF, distance	DM, uniform	CC/QS/DM, uniform	DM, uniform
k-NN (*k* = 13)	68.19	68.61	68.46	**69.43**	67.56	**69.43//R=20**
Dis, *weighting*	N/A, distance	CC, distance	DF, distance	CC, uniform	CC/QS/DM, uniform	CC, uniform
LDA	69.76	70.29	**70.4**	69.46	69.45	**70.40//R=30**
Dis, λ	N/A, 0.75	QS/DM, 0.8	DF, 0.45	DM, 0.3	RE, 0.2	DF, 0.45
SVM-RBF	69.53	**70.81**	70.41	70.13	68.85	**70.81//R=40**
Dis, *C*, *d*, γ	N/A, 1.2, 2, 0.02	RE, 1.4, 2, 0.01	QS/RE/DM, 0.9, 2, 0.04	DF, 0.7, 2, 0.1	DF, 1, 2, 0.08	RE, 1.4, 2, 0.01
SVM-Poly	70.18	70.03	**70.37**	70.14	68.09	**70.37//R=30**
Dis, *C*, *d*, γ	N/A, 5, 2, 0.01	QS/DM, 9, 2, 0.01	QS/RE/DM, 0.7, 2, 0.04	DF, 0.3, 2, 0.1	DF, 0.5, 2, 0.16	QS/RE/DM, 0.7, 2, 0.04

Table 7. Meta-classifier performance of 5-fold cross validation on 11-class scheme on test split.

RR_Rnk; *R* =	50	40	30	20	10	BEST
CatNB	75.07	75.27	76.06	75.90	**76.70**	**76.70//R=10**
Dis, α	N/A, 0.02	DF, 0.04	DF, 4.4	DF 1.8	CC/RE, 3.9	CC/RE, 3.9
k-NN (*k* = 5)	*73.96*	75.04	74.72	74.70	**76.66**	**76.66//R=10**
Dis, *weighting*	N/A, uniform	DF, uniform	DF, uniform	DF distance	CC/RE, uniform	CC/RE, uniform
k-NN (*k* = 7)	**76.15**	74.49	74.78	74.83	**76.15**	**76.15//R=10**
Dis, *weighting*	N/A, CC/RE, uniform	CC/RE, distance	DF, distance	DF, distance	CC/RE, uniform	CC/RE, uniform
LDA	75.39	**76.85**	75.40	75.91	76.35	**76.85//R=40**
Dis, λ	N/A, 0.25	DF, 0.2	DF, 3	RE, 0.25	QS/DM, 0.1	DF, 0.2
SVM-RBF	76.67	76.01	76.37	**76.71**	75.76	**76.71//R=20**
Dis, *C*, *d*, γ	N/A, 5, 2, 0.006	QS/DM, 8, 2, 0.004	DF, 3, 2, 0.01	DF, 0.5, 2, 0.12	QS/DM, 1.8, 2, 0.1	DF, 0.5, 2, 0.12
SVM-Poly	75.99	75.59	76.38	**76.75**	75.24	**76.75//R=20**
Dis, *C*, *d*, γ	N/A, 1.3, 2, 0.02	DF, 1.8, 3, 0.02	DF, 0.1, 2, 0.1	DF, 1.3, 3, 0.04	CC/RE, 0.2, 2, 0.18	DF, 1.3, 3, 0.04

Table 8. Meta-classifier performance of 5-fold cross validation on 6-class scheme on test split.

RR_Rnk; R =	50	40	30	20	10	BEST
BerNB	N/A, 88.24	88.43	88.16	**88.62**	87.44	**88.62//R=20**
Dis, α	N/A, 6	DM, 3.6	CC/DF, 0.0001	CC/RE, 1.0	QS/RE/DM, 0.0001	CC/RE, 1.0
k-NN (k = 7)	86.69	87.36	87.91	88.28	**88.81**	**88.81//R=10**
Dis, *weighting*	N/A, uniform	QS, uniform	CC/DF/RE, distance	DF, distance	CC, uniform	CC, uniform
k-NN (k = 9)	86.6	87.32	**88.67**	88.61	88.48	**88.67//R=30**
Dis, *weighting*	N/A, uniform	QS, uniform	RE, distance	DF, distance	CC, uniform	RE, distance
k-NN (k = 11)	86.97	87.52	88.73	88.41	**89.02**	**89.02//R=10**
Dis, *weighting*	N/A, uniform	QS/DM, uniform	CC/DF, uniform	DF, distance	CC, uniform	CC, uniform
k-NN (k = 13)	87.13	87.52	88.51	88.62	**89.15**	**89.15//R=10**
Dis, *weighting*	N/A, uniform	QS, uniform	RE, distance	DF, distance	DF, uniform	DF, uniform
LDA	88.33	**88.7**	**88.7**	88.54	88.53	**88.70//R=30**
Dis, λ	N/A, 0.9	CC/QS/RE, 1.0/0.9/1.0	QS/DM, 1.0	QS/DM, 0.7	DF, 0.9	QS/DM, 1.0
SVM-RBF	88.23	88.81	89	**89.3**	**89.3**	**89.30//R=10**
Dis, *C*, *d*, γ	N/A, 0.3, 2, 0.002	CC/RE, 0.1, 2, 0.004	QS/DM, 01, 2, 0.004	QS/DM, 0.1, 2, 0.004	CC, 0.1, 2, 0.002	CC, 0.1, 2, 0.002
SVM-Poly	88.41	88.81	89	88.81	**89.53**	**89.53//R=10**
Dis, *C*, *d*, γ	N/A, 0.1, 2, 0.006	CC/DF/QS/RE, 0.1, 2, 0.008	QS/RE/DM, 0.1, 2, 0.01/0.006/0.01	QS/DM, 0.1, 2, 0.01	CC, 0.1, 2, 0.02	CC, 0.1, 2, 0.02

Table 9. Meta-classifier performance of 5-fold cross validation on 2-grade scheme on test split.

RR_Rnk; *R* =	50	40	30	20	10	BEST
CatNB	72.62	73.45	**73.58**	72.54	72.17	**73.58//R=30**
Dis, α	N/A, 2.1	DF, 2.3	DM, 0.04	DM, 0.001	DF, 3.7	DM, 0.04
k-NN (*k* = 7)	72.13	**72.43**	72.04	72.3655	72.07	**72.43//R=40**
Dis, *weighting*	N/A, distance	RE, distance	DM, distance	DF, uniform	DF, uniform	RE, distance
k-NN (*k* = 9)	72.19	72.27	71.65	**72.28**	72.13	**72.28//R=20**
Dis, *weighting*	N/A, distance	QS/DM, distance	RE, uniform	DF, uniform	DF, distance	DF, uniform
k-NN (*k* = 11)	72.2	72.15	71.7	72.28	72.07	**72.28//R=20**
Dis, *weighting*	N/A, distance	RE, distance	DM, uniform	DF, uniform	DF, distance	DF, uniform
k-NN (*k* = 13)	**72.2**	72.02	71.64	71.63	71.99	**72.20//R=50**
Dis, *weighting*	N/A, distance	RE, distance	DM, distance	DF, distance	DF, distance	distance
LDA	72.52	**73.8**	72.82	72.51	72.08	**73.80//R=40**
Dis, λ	N/A, 0.9	QS/DM, 0.9	DF, 0.95	DM, 0.7	DM, 0.85	QS/DM, 0.9
SVM-RBF	72.8	72.83	**73.06**	72.89	72.81	**73.06//R=30**
Dis, *C*, *d*, γ	N/A, 1.9, 2, 0.04	RE, 1.9, 2, 0.04	DM, 1.2, 2, 0.06	DF, 1.7, 2, 0.04	DF, 1.2, 2, 0.14	DM, 1.2, 2, 0.06
SVM-Poly	72.92	**72.93**	72.43	72.2	71.92	**72.93//R=40**
Dis, *C*, *d*, γ	N/A, 0.1, 2, 0.08	QS/DM, 0.1, 2, 0.16	DM, 1, 2, 0.04	DF, 0.6, 2, 0.12	DF, 0.4, 2, 0.2	QS/DM, 0.1, 2, 0.16

Table 10. Meta-classifier optimisation by 5-fold cross validation on 11-class scheme (on enriched validation set).

RR_Rnk; *R* =	50	40	30	20	10	BEST
CatNB	74.73	75.71	**76.57**	75.1	75.78	**76.57//R=30**
Dis, α	N/A, 4.6	DF, 5.9	RE, 5.3	QS, 3.8	DF, 1.0	RE, 5.3
k-NN (*k* = 7)	*75.33*	76.93	**77**	76.86	75.23	**77.00//R=30**
Dis, *weighting*	N/A, uniform	CC/QS/RE/DM, distance	RE, distance	DF, distance	QS, distance	RE, distance
k-NN (*k* = 9)	75.62	76.17	**76.89**	76.83	75.61	**76.89//R=30**
Dis, *weighting*	N/A, CC/RE, uniform	CC/QS/RE/DM, distance	QS, distance	QS, distance	QS, distance	QS, distance
k-NN (*k* = 11)	75.13	76.22	76.88	**76.92**	76.08	**76.92//R=20**
Dis, *weighting*	N/A, CC/RE, uniform	CC/QS/RE/DM, distance	RE, distance	Df, distance	QS, distance	DF, distance
k-NN (*k* = 13)	75.01	76.22	**76.97**	76.76	75.94	**76.97//R=30**
Dis, *weighting*	N/A, CC/RE, uniform	CC/QS/RE/DM, distance	RE, distance	DF, distance	QS, distance	RE, distance
LDA	74.8	**75.12**	75.1	74.36	74.7	**75.12//R=40**
Dis, λ	N/A, 1	CC/QS/RE/DM, 1.0	QS, 0.5	QS/DM, 0.45/1.0	QS, 0.3	CC/QS/RE/DM, 1.0
SVM-RBF	76.31	76.85	**77.5**	77.29	75.47	**77.50//R=30**
Dis, *C*, *d*, γ	N/A, 6, 2, 0.02	CC/QS/RE/DM, 1.7, 2, 0.04	RE, 0.9, 2, 0.06	DM, 0.7, 2, 0.1	CC/RE/DM, 0.1, 2, 0.02	RE, 0.9, 2, 0.06
SVM-Poly	76.19	**76.85**	76.68	76.36	74.42	**76.85//R=40**
Dis, *C*, *d*, γ	N/A, 0.6, 2, 0.06	CC/QS/RE/DM, 0.2, 3, 0.06	CC, 0.6, 3, 0.04	DM, 0.5, 3, 0.06	QS, 1.9, 3, 0.1	CC/QS/RE/DM, 0.2, 3, 0.06

Table 11. Meta-classifier optimisation by 5-fold cross validation on 6-class scheme (on enriched validation set).

RR_Rnk; R =	50	40	30	20	10	BEST
BerNB	89.58	89.79	90.01	**90.06**	89.8	**90.06//R=20**
Dis, α	N/A, 4.9	CC/QS, 0.0001	CC/QS, 0.0001	DM, 0.0001	QS, 0.0001	DM, 0.0001
k-NN (k = 7)	90.37	90.83	90.16	**90.84**	90.68	**90.84//R=20**
Dis, *weighting*	N/A, uniform	CC, uniform	QS, uniform	DM, uniform	DF, uniform	DM, uniform
k-NN (k = 9)	90.27	**90.79**	90.15	89.86	90.79	**90.79//R=40**
Dis, *weighting*	N/A, uniform	CC/QS, uniform	DM, uniform	DM, uniform	DF, distance	CC/QS, uniform
k-NN (k = 11)	90.27	**90.59**	89.89	90.08	90.33	**90.59//R=40**
Dis, *weighting*	N/A, uniform	RE, uniform	QS, uniform	DM, uniform	DM, uniform	RE, uniform
k-NN (k = 13)	90.53	**90.58**	90.13	90.08	90.14	**90.58//R=40**
Dis, *weighting*	N/A, uniform	DF, uniform	DF, uniform	DM, uniform	DF, distance	DF, uniform
LDA	90.41	**90.68**	90.45	89.99	90.23	**90.68//R=40**
Dis, λ	N/A, 0.65	DM, 0.5	QS/DM, 0.65	DF, 0.6	DM, 0.75	DM, 0.5
SVM-RBF	90.16	90.23	90.41	90.36	**90.85**	**90.85//R=10**
Dis, *C*, *d*, γ	N/A, 0.1, 2, 0.002	DM, 4, 2, 0.18	DM, 0.1, 2, 0.04	DM, 1.3, 2, 0.006	CC/RE, 2, 2, 0.008	CC/RE, 2, 2, 0.008
SVM-Poly	90.16	90.23	90.41	90.36	**90.85**	**90.85//R=10**
Dis, *C*, *d*, γ	N/A, 0.1, 2, 0.002	DM, 4, 2, 0.18	DM, 0.1, 2, 0.04	DM, 1.3, 2, 0.006	CC/RE, 2, 2, 0.008	CC/RE, 2, 2, 0.008

Table 12. Meta-classifier optimisation by 5-fold cross validation on 2-grade scheme (on enriched validation set).

	11-class		6-class		2-grade	
	valid	test	valid	test	valid	test
NB	73.58//R=30	67.79	76.57//R=30	76.43	90.06//R=20	88.24
Dis, α	CatNB: DM, 0.04		CatNB: RE, 5.3		BerNB: DM, 0.0001	
k-NN (k = 7)	72.43//R=40	68.26	77.00//R=30	75.35	90.84//R=20	88.86
Dis, *weighting*	RE, distance		RE, distance		DM, uniform	
k-NN (k = 9)	72.28//R=20	69.26	76.89//R=30	75.52	90.79//R=40	87.45
Dis, *weighting*	DF, uniform		QS, distance		CC/QS, uniform	
k-NN (k = 11)	72.28//R=20	68.78	76.92//R=20	75.95	90.59//R=40	87.71
Dis, *weighting*	DF, uniform		DF, distance		RE, uniform	
k-NN (k = 13)	72.20//R=50	68.98	76.97//R=30	75.44	90.58//R=40	87.87
Dis, *weighting*	distance		RE, distance		DF, uniform	
LDA	73.80//R=40	**69.66**	75.12//R=40	75.75	90.68//R=40	88.18
Dis, λ	QS/DM, 0.9		CC/QS/RE/DM, 1.0		DM, 0.5	
SVM-RBF	73.06//R=30	69.60	77.50//R=30	75.40	90.85//R=10	*86.45*
Dis, *C*, *d*, γ	DM, 1.2, 2, 0.06		RE, 0.9, 2, 0.06		CC/RE, 2, 2, 0.008	
SVM-Poly	72.93//R=40	68.43	76.85//R=40	**77.33**	90.85//R=10	*86.45*
Dis, *C*, *d*, γ	QS/DM, 0.1, 2, 0.16		CC/QS/RE/DM, 0.2, 3, 0.06		CC/RE, 2, 2, 0.008	

Table 13. Meta-classifier performances after being tuned on enriched validation set.

Excluded from voting	11-class	6-class	2-grade
All	70.30 $k \neq 5,15$	77.07 $k \neq 9–15$	89.34 $k \neq 5,15$
¬ *k*-NN (1)	71.35 $k \neq 11–13,5,15$	77.62 $k \neq 7, 9–15$	89.38 $k \neq 7–9,5,15$
¬ *k*-NN (2)	71.10 $k \neq 9,11–13,5,15$	77.81* $k \neq 5, 9–15$	**89.55** $k \neq 11,7–9,5,15$
¬ *k*-NN (3)	71.14 $k \neq 7,11–13,5,15$		89.38 $k \neq 13,7–9,5,15$
¬ *k*-NN (4)	**71.66*** $k \neq 7–9,5,15$	--- ---	89.10 $k \neq 11–13,5,15$
¬ *k*-NN (5)	**71.48** $k \neq 7–9,11–13,5,15$	76.74 $k \neq 5–7, 9–15$	**89.67*** $k \neq 11–13,7–9,5,15$
¬ NB	**71.48**	**78.12**	89.34
¬ LDA	70.43	**78.16**	89.34
¬ SVM-RBF	70.39	77.31	89.34
¬ SVM-Poly	70.46	77.37	89.34
¬ SVM	70.39	77.49	88.74
¬ NB, LDA	70.17	77.56	89.14
¬ NB, SVM-RBF	70.23	78.04	89.34
¬ NB, SVM-Poly	70.23	78.10	89.34
¬ LDA, SVM-RBF	69.90	77.59	89.34
¬ LDA, SVM-Poly	70.01	78.07	89.18
¬ NB, *k*-NN *	70.76	77.04	89.50
¬ NB, *k*-NN (5)	71.29	77.40	Same as above
¬ LDA, *k*-NN *	70.68	**78.12**	89.50
¬ LDA, *k*-NN (5)	71.03	77.15	Same as above

Table 14. Performances of level-2 reliability-enhanced soft voting on level-1 meta-classifiers.

Excluded from voting	11-class	6-class	2-grade
All	**71.75***	**78.03***	89.57
	$k \neq 5,15$	$k \neq 9–15$	$k \neq 5,15$
¬ *k*-NN (1)	71.49	77.92	89.58
	$k \neq 11–13,5,15$	$k \neq 7,9–15$	$k \neq 7–9,5,15$
¬ *k*-NN (2)	71.41	77.84	89.58
	$k \neq 9,11–13,5,15$	$k \neq 5,9–15$	$k \neq 11,7–9,5,15$
¬ *k*-NN (3)	71.45	---	89.43
	$k \neq 7,11–13,5,15$	---	$k \neq 13,7–9,5,15$
¬ *k*-NN (4)	71.28	---	**89.63***
	$k \neq 7–9,5,15$	---	$k \neq 11–13,5,15$
¬ *k*-NN (5)	71.51	77.89	89.53
	$k \neq 7–9,11–13,5,15$	$k \neq 5–7,9–15$	$k \neq 11–13,7–9,5,15$
¬ NB	71.37	77.63	89.58
¬ LDA	71.07	77.84	89.41
¬ SVM-RBF	71.27	77.55	89.37
¬ SVM-Poly	**71.58**	77.76	89.37
¬ SVM	71.17	77.52	89.37
¬ NB, LDA	70.88	77.96	89.37
¬ NB, SVM-RBF	71.41	77.92	89.37
¬ NB, SVM-Poly	**71.71**	77.99	89.37
¬ LDA, SVM-RBF	70.27	77.98	89.37
¬ LDA, SVM-Poly	71.12	77.92	89.20
¬ NB, *k*-NN *	Same as ¬ NB	Same as ¬ NB	89.58
¬ NB, *k*-NN (5)	71.44	77.13	89.53
¬ LDA, *k*-NN *	Same as ¬ LDA	Same as ¬ LDA	89.58
¬ LDA, *k*-NN (5)	71.44	76.96	89.53

Table 15. Performances of level-2 meta-classifier on level-1 meta-classifiers.