IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 21, 2020, accepted January 4, 2021, date of publication January 11, 2021, date of current version January 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3050547

Citation Intent Classification Using Word Embedding

MUHAMMAD ROMAN[1], ABDUL SHAHID[1], SHAFIULLAH KHAN[1], ANIS KOUBAA[2,3], AND LISU YU[4,5], (Member, IEEE)
[1]Institute of Computing, Kohat University of Science and Technology, Kohat 26000, Pakistan
[2]Robotics and Internet of Things Laboratory, Prince Sultan University, Riyadh 12435, Saudi Arabia
[3]CISTER/INESC-TEC, Polytechnic Institute of Porto, 4200 Porto, Portugal
[4]School of Information Engineering, Nanchang University, Nanchang 330031, China
[5]State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Lisu Yu (lisuyu@ncu.edu.cn)

This work was supported by the State Key Laboratory of Computer Architecture (ICT, CAS) Open Project under Grant CARCHB202019.

ABSTRACT Citation analysis is an active area of research for various reasons. So far, statistical approaches are mainly used for citation analysis, which does not look into the internal context of the citations. Deep analysis of citation may reveal interesting findings by utilizing deep neural network algorithms. The existing scholarly datasets are best suited for statistical approaches but lack citation context, intent, and section information. Furthermore, the datasets are too small to be used with deep learning approaches. For citation intent analysis, the datasets must have a citation context labeled with different citation intent classes. Most of the datasets either do not have labeled context sentences, or the sample is too small to be generalized. In this study, we critically investigated the available datasets for citation intent and proposed an automated citation intent technique to label the citation context with citation intent. Furthermore, we annotated ten million citation contexts with citation intent from Citation Context Dataset (C2D) dataset with the help of our proposed method. We applied Global Vectors (GloVe), Infersent, and Bidirectional Encoder Representations from Transformers (BERT) word embedding methods and compared their Precision, Recall, and F1 measures. It was found that BERT embedding performs significantly better, having an 89% Precision score. The labeled dataset, which is freely available for research purposes, will enhance the study of citation context analysis. Finally, It can be used as a benchmark dataset for finding the citation motivation and function from in-text citations.

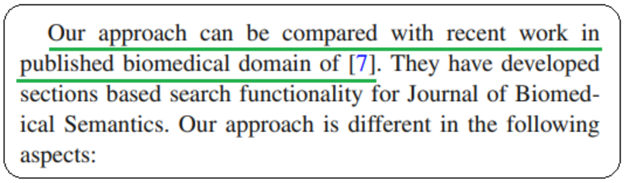
INDEX TERMS Citation intent, citation analysis, citation context, citation motivation, citation function classification, word embedding, scholarly dataset.

I. INTRODUCTION

Citing research articles has always been an integral part of a research paper. Scientific contents need to cite other works for various reasons [1], called citation intent. Finding citation intent of a citation is crucial for analyzing scientific literature and the relationship among scientific articles. A number of tasks, including citation intent classification, context analysis, research article recommendation, finding relevant papers, and creating citation networks, all require state-of-the-art scholarly datasets. For each of these mentioned problems, the approaches proposed to require different types of information. Citation intent can play a role in measuring the worth of a journal, area, and publication. Purely quantitative citation analysis has long been criticized by researchers, pointing out that many citations are made out of "politeness, policy or piety" [2]. The authors further argue that citations, merely for the sake of knowledge, should not be counted as much central bidding in a paper than the one used as a starting point of research. Moravcsik and Murugesan [3] divides the reason of citation into four classes and found out that 40 percent of the citations are classified into perfunctory class (i.e., cited paper is providing an understanding of the work). Their work further doubts the quantitative approach of citation analysis. Teufel *et al.* [1] named the intention of citation as citation function and classified them into three classes; Weak, Positive, and Neutral.

Different approaches have been used to identify the citation intent. Due to its complex nature, most of the earlier work is performed manually [3]–[5], where citations

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.



Our approach can be compared with recent work in published biomedical domain of [7]. They have developed sections based search functionality for Journal of Biomedical Semantics. Our approach is different in the following aspects:

FIGURE 1. Example of citation context.

were manually annotated with citation intent class. Newer approaches utilised automated methods to classify the citation intent [1], [6]–[9]. They normally use citation context, which includes a range of text, defined by a context window of a certain type, in a citing paper that describes a referenced article. The citation context, primarily, signals the citation reason of a referenced paper [1], [10], [11]. For instance, Figure 1 [12] shows an example of a citation context underlined in which a paper has been cited. The citation intent of the cited paper can be clearly identified from the citation context, which is *Comparison* in this case.

The earlier models used feature-based approaches to model the citation context on the bases of cue phrases [1], or linguistic patterns [13]. Due to the advancements in natural language processing and deep learning, pre-existing language modeling has also helped us understand the context of the in-text citation [7]–[9]. These models provided an automatic mechanism of finding the citation context's intent; still, they require a pre-annotated dataset to train the model to predict the class of intent. Unfortunately, most of the baseline datasets are very small, and in most cases, they are labeled through crowd-sourcing platform [7]. The human annotators manually annotated the citation context while not necessarily having the knowledge of that domain. Therefore, they fail to provide a level of confidence for using those datasets as benchmarks for citation analysis problems. This article has proposed a new technique of annotating the citation context with the citation intent. We created clusters of the sentences based on their contextual meanings. We then analyzed each of the clusters and tagged them for citation intent. As a result, we have annotated all the sentences automatically without human involvement. We evaluated our model using Precision, Recall, and F1 score to understand how well our proposed method is performing. We also compared our method by annotating and then comparing the Sci-Cite [7], a pre-annotated dataset. The main contribution of this article is as follows:

- This study critically evaluated the available datasets in this domain from the perspective of discovering citation intent analysis
- This study proposed a fully automatic approach for citation intent annotation as most of the state-of-the-art are manually or semi-automatic
- This study produced an extended dataset having the illustration of citation intent between the research articles

The rest of this article is organized as follows: in Section II, we introduce existing datasets in citation analysis and citation intent classification. In Section III, we propose our methodology and various steps required to perform. In Section IV, we discuss the experimental settings. Section V explains the results and evaluates the model. In Section VI, we present a new annotated dataset C2D-I, and finally, we conclude our work in Section VII.

II. RELATED WORK

A number of citation analysis tasks need state-of-the-art datasets. Different methods are used to create datasets. The earlier approaches were based on statistical information typically found in metadata and bibliographic information of the research papers. Therefore, the datasets were mostly based on bibliographic and metadata information. Metadata includes information like title, keywords, authors, journal, issue, volume, date of publication, pages, and citations. For instance, the Digital Bibliography & Library Project (DBLP)[1] is indexing over 4.4 million publications by more than 2.2 million authors from approximately 79,000 journals, conferences, and workshops in the major of computer science. It provides manually curated metadata; however, it lacks some valuable information like abstract. It also provides search APIs for three search services; publications, authors, and venues.

CiteSeer^X dataset [14], produced from *CiteSeer^X* indexing service,[2] provides the citation context of the cited papers. *CiteSeer^X* crawls and indexes files openly available on the web. Currently, it has nearly 6 million documents, having a total of 20 million citations. The raw *CiteSeer^X* contains noise [15], therefore various cleaned versions of *CiteSeer^X* datasets are prepared. Callaham *et al.* [16] released a cleaner version of *CiteSeer^X* by linking it to DBLP, indexing over 2 million publications. Valenzuela *et al.* [6] used a supervised learning approach to match entities among scholarly datasets and provided a cleaner version of *CiteSeer^X*.

CrossRef [17] metadata contains over 106 million publications in 13 different content types, including journals, conferences, books, grants, and others. It includes abstracts and links to full-text along with the basic publication metadata. It is a considerable contribution to research in scientometrics, including finding trends in research and measuring the growth and impact of science. The services provided by CrossRef can be accessed through API's; however, they also provide some downloadable datasets in exceptional cases like 65GB JSON dataset containing records that might be related to COVID-19 pandemic.[3]

PubMed[4] provides access to the life sciences and biomedical literature, health, behavioral science, chemical science, and bioengineering, containing more than 30 million citations, as of July 2020. It contains the abstracts of the publication but not the full-text. However, the full-text may be accessed for which the links are provided, where

[1] https://dblp.org/

[2] https://citeseerx.ist.psu.edu/

[3] https://www.crossref.org/blog/free-public-data-file-of-112-million-crossref-records/

[4] https://pubmed.ncbi.nlm.nih.gov/

available. PubMed has several components, including MedLine, PubMed Central (PMC), and Bookshelf. Medline is the largest component consisting of 5,200 journals having approximately 12 million articles, updated every seven days. MedLine does not provide the full-text of the publications; however, the full-text may be retrieved from the link to the published articles. PMC, being the second-largest component of PubMed, is a free full-text archive having 6.2 million articles from over 2,000 journals. The Bookshelf is the final component of PubMed, including citations from books, reports, and other documents from health and life sciences.

AMiner [18] is a citation network dataset containing over 3 million publications (nodes) with 25.2 million citations (edges), making a network of papers. It has two different datasets; AMiner DBLP and AMiner ACM. AMiner DBLP is based on the DBLP database and is considered very clean. AMiner ACM is constructed on ACM publications from 2.4 million articles and 9.7 million citations.

Microsoft Academic Graph (MAG) [19] provides a knowledge graph of publications and their associative entities like authors, venues, institutes, and major. Data is available offline only through MAG subscriptions, while real-time applications may use it through Microsoft Academic Knowledge Exploration Service (MAKES) API. The knowledge graph consists of over 239 million publications by 242 million authors from over 53,000 journals.

Open Academic Graph (OAG) [19] is an extensive knowledge graph based on MAG and AMiner. It contains over 166 million publications from MAG and 154 million publications from AMiner. It has metadata about the publications and does not provide the full-text of the articles. The dataset is downloadable and can be used directly, unlike MAG.

Springer SciGraph[5] exposes a Linked Open Data platform that aggregates datasets from Springer and key partners from scholarly domains. The high-quality dataset from trusted and reliable sources provides a semantic description of how data is related by visualizing the scholarly innovations. SciGraph dataset is incrementing continuously and is downloadable in the form of metadata for persons, books, technical articles, subjects, journal datasets.

The CoRA[6], [7] dataset consists of 2,708 publications and 5,429 links, divided in 7 classes. The dataset is available for download in the form of MySQL script.

The other category of datasets is having access to the full-text of the publications, therefore, providing in-depth knowledge of those scholarly articles. For instance, ACL Anthology Network (ANN) Corpus [20] is a manually curated networked dataset of citations, collaborations, and summaries in the field of Computational Linguistics. The authors manually annotated publications in the Association of Computational Linguistics (ACM) collection, including more than 20k research articles by 17k authors. Along with other metadata about the publication, the ANN also included citation context, which they named the citing area. Although this dataset provides citation sentences, those sentences are not annotated for the citation intent.

ACL Anthology Reference Corpus (ACL-ACR) [11] performed their first most extensive behavioral study of citations and developed a state-of-the-art classifier. They labeled the citation sentences with the intent of citation. This dataset is based on ANN and consists of nearly 2,000 citations only from 186 publications. These citations were divided into sets of training, validation, and testing with 85 percent training data. The data is annotated with six intent categories; Background, Motivation, Uses, Extension, Comparison, and Future Work.

CORE [22] provides both metadata only and full-text datasets. As of July 2020, it includes over 123 million metadata items, 85.5 million records with abstract whereas 9.8 million items with full-text. The dataset can be used through API's and is also downloadable.

Cohan *et al.* [7] proposed structural scaffolding for classification of citation intent and, as a result, introduced the Sci-Cite dataset. The authors compared the new dataset with ACL-ARC and achieved a 13 percent increase in FI score. Unlike ACL-ARC, instead of six classes, Sci-Cite categorized intents into three classes; Background, Method, and ResultComparison. The citation sentences are extracted from papers in Semantic Scholar Corpus [7], consisting of articles from computer science and medical domains. Eight hundred fifty workers then annotate these sentences on a crowd-sourcing platform. A total of 9,159 instances were annotated after filtering them by experts. The Sci-Cite dataset is free to download and use.

Jeong *et al.* [27] proposed a recommender model and created a new dataset. This dataset is based on the ANN dataset, and its size is less than it. They have removed the PDFs that did not use Latex or were very noisy to be used by arXiv Vanity. They used context sentences for recommending citations on the fly, white writing a research paper. The dataset description is not available, as the ANN dataset policy does not disclose modification without receiving a grant from them.

arXiv CS [24] dataset is based on arXiv.org data of computer science domain. References are linked to DBLP entries, where possible. Similarly, unarXiv [20] dataset is based on arXiv CS dataset consisting of over 1 million full-text publications and 2.7 million annotated in-text citation contexts. The in-text citations are annotated through global identifiers. All entries are linked to Microsoft Academic Graph. The dataset is freely available for use by researchers.

Khadka and Knoth [25] investigated the effects of incorporating the textual information in closed proximity of the citation on the performance of the recommender system. The authors introduced a new Context Citation Dataset (C2D), which contains the citation information along with the citation context. The citation window includes the sentence in which in-text citation is used, the sentence before and after it.

[5] https://www.springernature.com/gp/researchers/scigraph

[6] https://graphsandnetworks.com/the-cora-dataset/

[7] https://linqs-data.soe.ucsc.edu/public/lbc/cora.tgz

TABLE 1. Overview of the existing scholarly datasets.

dataset	Size	Information	Usage	Released	Updates	Context	Intent
S2ORC [9]	81.1M	8.1M Full-text, meta-data	Downloadable: JSON	July, 2020	Manual	Yes	No
Sci-Ci [7]	8,243	Meta-data and citation sentences	Downloadable: CSV	Sept, 2019	Manual	Yes	Yes
Sci-Graph [23]	2B triples	Meta-data, Abstract, Citation graph	Downloadable [8]	April, 2019	Regular bases	Yes	No
Sci-Bert Model [8]	1.14M	Trained BERT Model	Installable directly within Huggingface's framework under the allenai org.	Feb, 2019	Manual	No	No
CORE [24]	123M	9.8M Full-text, 85M meta-data	API, downloadable	2019	Automatic	No	No
unarXiv [25]	1M Documents and 29.2M Citation Contexts	Full-text	Downloadable [9]	2019	Manual	Yes	Yes
arXiv [26]	1.7M	Meta-data	Downloadable: JSON, API	April, 2018	Manual	No	No
ACL-ARC [11]	2,000 citations	Citation context	Downloadable	2018	Manual	Yes	Yes
C2D [27]	53M	In-text citation sentence, before and after in-text sentence	Downloadable: CSV	2018	Automatic	Yes	No
AMiner [18]	3M publications with 25.2M citations	Full-text	Downloadable [10]	April, 2016	Automatic by crawling	Yes	No
OAG [19]	320M	Meta-data and Abstract	Downloadable	2015	Regular updates	No	No
MAG [19]	239M	Meta-data	Azure storage and MAKES API	2015	Regularly weekly bases	No	No
ANN [20]	20,000	Meta-data, citation context	Downloadable [11]	2013	Manual	Yes	No
CoRA [12]	2,708	Citation Iformation	Downloadable: txt [13]	July, 2012	Manual	No	No
PubMed [14]	30M	Only abstract	Downloadable XML [15]	April, 2009	Updates released yearly with daily incremental files	No	No
CiteSeerX [14]	120M	Full-text, meta-data	URL access	1998, 2008	Regular by Crawling	No	No
CrossRef [17]	106M	Meta-data	API's in JSON [16] and XML [17]	2000	Regular monthly bases	No	No
DBLP [28] [18]	4.86M	Meta-data	Downloadable XML [19]	1993	Automatic Monthly	No	No

C2D includes 53 million unique citation records, created from 2 million research articles provided by CORE. The citation sentences are, however, not annotated for citation intent. Therefore, they do not provide the reason for the citation

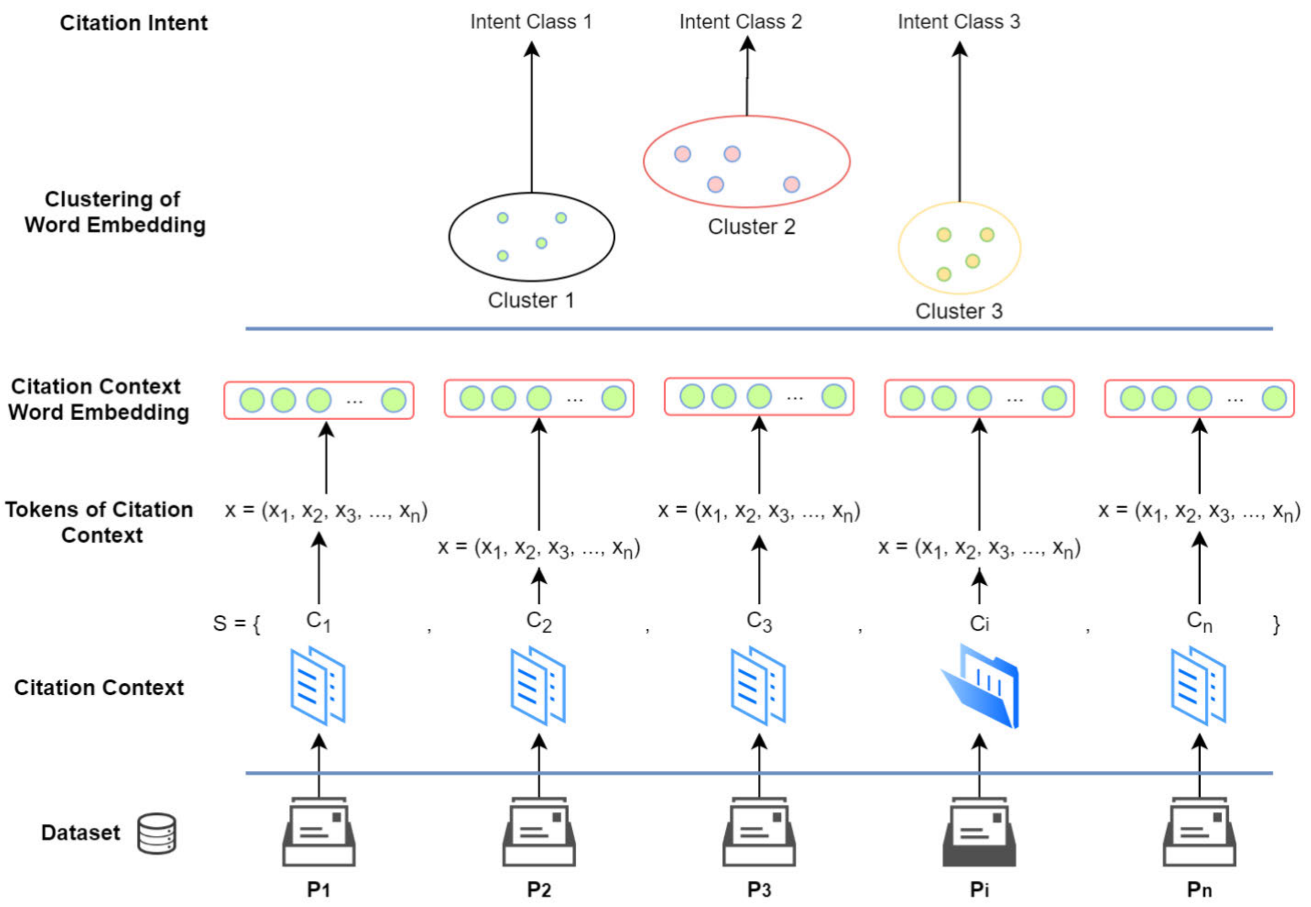

Citation Intent
Intent Class 1
Intent Class 2
Intent Class 3
Clustering of Word Embedding
Cluster 1
Cluster 2
Cluster 3
Citation Context Word Embedding
Tokens of Citation Context
x = (x1, x2, x3, ..., xn)
S = { C1 , C2 , C3 , Ci , Cn }
Citation Context
Dataset
P1
P2
P3
Pi
Pn

FIGURE 2. Clustering based citation intent classification architecture.

of a paper. However, Anika *et al.*[20] has also developed a very small citation network-based papers from RecSys conference. The dataset is divided into four files having author, publications, selected author, and selected publication. This dataset also includes the intent of citations.

Semantic Scholar Open Research Corpus (S2ORC) [9] contains around 200 million unique papers from trusted sources. 156.5 million of the publications have citation contexts. In comparison, the rest have abstract, section information, citation details, text in tables, and mathematical equations. However, the citation sentences are not annotated for their intent. The papers are taken from a more diverse set of domains than any other resources. The dataset can be download freely using Amazon Web Services in JSON format in individual files.

A detailed discussion about the existing dataset is provided in Table 1. The table provides information for each dataset, including the *Size* in terms of available records and publications. *Information* column specifies which information is included in the dataset, e.g., full-text, meta-data, abstract. The *Usage* field describes the accessibility mechanism of the dataset, which includes accessing it through an API or direct download in XML/JSON format. The *Released* and *Updates* attributes mention the initially released date and the dataset's update mechanism. Some datasets are periodically updated while others have no updates available after their initial release, labeled as *Manual* in the table. The *Context* and *Intent* show if the dataset has context sentences and the intent of each citation. The datasets are arranged in descending order of their initial release date, although they may have been recently updated, which is the case in most of the automatic updates.

Although studying citation intent has long been studied, there are only a few datasets that truly provide citation function classification. Most of the datasets are manually annotated and too small to be used as a benchmark, especially when using deep learning approaches. The datasets usually are not adequately evaluated, or the sample set is too small to develop a level of confidence. Therefore, we have proposed an automated method of classifying citations' intent, discussed in the next section.

III. PROPOSED METHODOLOGY

In this section, we provide the basic model of our proposed technique for labeling the citation intent. We also describe the word embedding and citation clustering techniques for finding the citation intent. The architecture of our proposed citation intent labeling technique is provided in Figure 2. We collected the research articles from a dataset. We extracted the citation context, containing the in-text

[20] https://ordo.open.ac.uk/articles/Citation_Knowledge_based_Dataset/10673132

citation, from each of the articles. The citation sentences were then pre-processed, and the citation context vectors were created using word embedding techniques. In the final stage, we created clusters of the word embeddings, and after studying the samples from created clusters, we assign a citation intent to each of these clusters.

Let S be the complete set of citation context, given by $S = C_1, C_2, \ldots, C_n$ taken from all the papers in the dataset D. Then C_i denotes the set of n number of citation contexts in a paper i, given by $C_i = c_1, c_2, \ldots, c_n$. V_i is a word vector of dimension d for paper i. The word vectors represent sentences in a dimension so that semantically similar citation contexts come closer and have similar representations. The word vectors are created through one of the embedding techniques given in Table 2. Once the citation context of a paper is converted to vector representations, the proposed technique uses a clustering algorithm to group similar sentences in the same cluster. We have used different clustering algorithms, given in Table 3, combined with various word embedding techniques. In order to create vectors using one of the embedding techniques, we pre-processed the citation context for better text classification.

A. CITATION CONTEXT PRE-PROCESSING

Pre-processing is one of the critical steps in text classification tasks. Uysal and Gunal [28] discussed four common text classification steps; stop word removal, tokenization, case conversion, and stemming/lemmatization. In our case, the text includes the citation context extracted from the papers in our dataset. We have removed stop words from the citation context. Stop words are the words that frequently occur in text irrelevant to the topic. Stop words include includes prepositions, conjunctions, articles. Stop word removal is a language-specific task requiring knowledge of a particular language, which is English in our case. We have used Natural Language Toolkit (NLTK) [29] for removing stop words in Python. It has a massive list of stop words already defined in 16 different languages. We have also extended a list of stop words provided by NLTK by adding some numbers and special characters that do not change the meaning of sentences.

In the next step of pre-processing, we changed all the upper case words to lower, as they are assumed to have the same meaning in our case. We have converted the case regardless of their position and form (full form or abbreviation).

In the final stage of pre-processing, we stemmed from the words to their root form. Stemming has remarkable effects on word embedding, as studied by Kantrowitz *et al.* [30] on TF-IDF [31]. Stemming converts the words to their root form from their derived ones. Stemming is also a language-specific task, and different algorithms have been proposed for this task [32]. We have used WordNetLemmatizer from the NLTK package in Python for lemmatization. We have observed an improvement using lemmatization instead of stemming in our experiments. Both of these techniques find the root words, but the difference is that stem word might not be a dictionary word; instead, lemmatization finds an actual word in a language.

After the pre-processing stage, we are ready to apply the word embedding technique to create vectors of cleaned citation context.

B. WORD EMBEDDING

In order to capture the distance between individual words within a citation context and among the citation sentences, we convert the text representation to a numerical form. By doing so, we can apply machine learning algorithms to text effectively. Word embedding is a numeric representation of words in a dense format where similar words have similar learned representations. Word embedding is a crucial breakthrough in applying deep learning to natural language processing tasks. The choice of selecting the best word embedding is essential as a pre-processing step in natural language processing tasks like text classification. There are different types of word embedding techniques; mainly categorized in frequency-based and prediction-based approaches, shown in Table 2. The table also lists the strengths and limitations of each of these approaches. Count Vector, Term Frequency–Inverse Document Frequency (TF-IDF), and Co-occurrence matrix base Global Vectors (GloVe) are discussed from the frequency-based embedding techniques category, also called count based embedding. Frequency-based word embedding techniques typically do not provide any contextual information. Baroni *et al.* [33] has provided a comparison of both type of approaches and has verified the claim that prediction-based approaches are superior to frequency-based vectors in various scenarios. Therefore, we have only selected the prediction-based word embedding techniques. Prediction-based algorithms include Word2Vec, Infersent, Embeddings from Language Models (ELMO), and Bidirectional Encoder Representations from Transformers (BERT). Word2Vec is a combination of two techniques are known as Continuous Bag of Words (CBOW) and Skip-gram models. Both of these models are based on shallow neural networks and learn weights that act like vector representations. They have their strengths and weaknesses listed in Table 2. Infersent, ELMO, and BERT are also prediction-based, but these are context sensitive embedding techniques. These three have almost similar strengths and limitations, therefore, we have combined them in Table 2. We have selected one technique from the frequency-based, and two from the prediction-based for checking their effectiveness in our case.

For text clustering, we have selected a variety of embedding techniques for our experiments to measure their feasibility. We have selected one technique from frequency-based and two techniques from prediction-based for checking their effectiveness in our case. In order to find the best embedding in our case, we have applied each of these embedding techniques on our dataset and selected the one providing the best results for our dataset generation, discussed in the results section.

TABLE 2. Word embedding techniques.

#	Algorithm	Type of Embedding	Strengths	Limitations
1	Count Vector [36]	Frequency-based Embedding	• Very simple and accurate • Unlike a one-hot-vector counts the total number of occurrence of a word in a single document	• Each vector has dimensions equal to the size of our total vocabulary. Thus the vector size is huge • Sparse vectors, having lots of zero in vector • Consumes a large amount of memory • Takes into account the frequency of a word within a document only, without considering its occurrence in the entire corpus • Cannot maintain a linear relationship in vector space, e.g.; 'king - man + woman = queen.' • Provides no semantic or relational information
2	TF-IDF [34]	Frequency-based Embedding	• Can easily compute the similarity between documents • Counts not only the occurrence of a word in a single document but in the entire corpus • Weight is directly proportional to word frequency within a document while inversely proportional its frequency within documents • Common words like 'while', 'but', 'the', 'is' does not have importance as compared to words that are rare in documents	• Vector size is large as document similarity is measured directly in word-count space • Does not consider co-occurrence of words and text position in a document • Semantic similarity between documents is not counted • Suffers from the inherent over-sparsity problem • Fail to under word-level synonymy and polysemy, e.g., In 'Apple is my favourite fruit' and 'I like Apple laptop' *Apple* has a different meaning, but TF-IDF fail to capture the difference in such cases
3	Co-Occurrence Matrix [37], **GloVe** [38]	Frequency-based Embedding	• Learns how frequently a set of words appear together in large text corpora • A hybrid method using machine learning based on the statistic matrix • It can preserve the semantic similarity between King and Queen • Dimensions are reduced using dimensionality reduction techniques, thus producing more accurate vectors	• Co-Occurrence matrix are costly in terms of memory, as it needs to store the co-occurrences
2	Word2Vec [39]	Prediction-based Embedding	• Provides probabilities to the words • State of the art for word analogies and word similarities • Solved magical 'King - man + woman = Queen.' • 'king:man as queen:woman' can be inferred by word vectors • Learned weights acts as word vector representations – maps words to target words • Probabilistic methods generally perform superior to deterministic methods [36] • Does not need huge memory, as compared to the co-occurrence • CBOW predicts the probability of a word given a single or group of words • Skip-gram predicts the context given a word	• Since the size of the vocabulary is too large; the model can get very difficult to train • In case of CBOW, it takes an average of the context for polysemy words. For example, in places word Apple in between a cluster of fruit and company. But in case of Skip-gram it will have two vector representations of Apple; one for the company and other for the fruit
4	**Infersent** [40], ELMO [41], **BERT** [29]	Prediction-based Word Embedding	• Can generate different word embedding for a word depending on the position of a word in a sentence - polysemy words problem is resolved. For example 'He went to the prison cell with his cell phone to extract blood cell samples from inmates'. These techniques will generate different vectors for the three 'cell' contexts.	• Compute-intensive due to generation of contextualized word embeddings

C. CITATION CONTEXT CLUSTERING

Once the sentences are converted to numerical representation, we apply clustering algorithms for cluster formation of the citation context. The clustering algorithms put the citation context of similar word embedding to the same cluster. Thus, each of the clusters contains a group of citation sentences

TABLE 3. Clustering methods.

#	Clustering Type	Algorithms	Strenghts	Limitations
1.	Flat, Partition-based Clustering	**K-Means** [43], K-Medoids [44]	• Divides the data into k number of such that $k <= n$ • The clusters are exclusively separated, each group of members belong to only one cluster • Most of the methods are distance-based (Euclidean, cosine) • Creates an initial partitioning and then uses an iterative relocation technique to approach local optimum (adopting greedy heuristic-based)	• Global optimum is often computationally prohibitive as it requires exhaustive enumeration • Data points sharing the same cluster that is far away are not captured within the cluster • Normally works well in case of spherically shaped clusters but suffers as the geometric shapes of clusters deviate from spherical shapes • Dependant on suitable values of k and *seed*
2.	Hierarchical Clustering	**HDBSCAN** [45], BIRCH [46], Chameleon [47]	• Creates a hierarchical decomposition of giving data points • The number of clusters to be created are not pre-defined, rather decided by clustering algorithms on the bases of data points • Normally, easy to implement • Provides a good understanding of data relationships using a dendrogram • Can be classified being either agglomerative or divisive, based on how the hierarchical decomposition is formed.	• Clustering cannot be undone once a data point has been added to a group • May require long computational time in comparison to k-Means, for example • In case of very large dataset may become hard to find right number of clusters

having similar nature of intent. We assign all the sentences in a cluster to a citation class after studying the sentences in that cluster. There are a number of algorithms for text clustering, as shown in Table 3. The table lists two essential types of clustering, flat and hierarchical, and discusses each of these groups' strengths and limitations. Hierarchical methods can be classified as Agglomerative or Divisive, based on how the hierarchical decomposition is formed. The agglomerative approach is a bottom-up, while the divisive is a top-down approach. Hierarchical methods can be distance-based or density-based. We have selected K-Means from flat clustering while Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) from hierarchical clustering category for our experiments. HDBSCAN is an extension of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [39] and is a density-based hierarchical approach combining the benefits of both hierarchical and density-based methods. After selecting word embedding and clustering techniques, we defined several settings by combining the word embedding and clustering algorithms, in which we perform our experiments, given in Table 4.

IV. EXPERIMENTAL SETUP

In this section, we provide the details of the experimental setups in which we performed our experiments. We discuss the dataset that we used for experiments. We also explain various settings that combine the embedding and clustering techniques. We used online Kaggle platform,[21] with GPU accelerator, for performing the experiments and evaluating the results.

TABLE 4. Experimental settings combining embedding and clustering techniques.

Setup	Embedding	Clustering Algorithm	Setting Name
1.	Infersent	KMeans	Infersent-KMeans
2.	GloVe	KMeans	GloVe-KMeans
3.	BERT	KMeans	BERT-KMeans
4.	Infersent	HDBSCAN	Infersent-HDBSCAN
5.	GloVe	HDBSCAN	GloVe-HDBSCAN
6.	BERT	HDBSCAN	BERT-HDBSCAN

A. DATASET

Sci-Cite [7] dataset is used to apply the various types of embedding and compare their results. This dataset has 8,243 crowd-sourced instances of citation contexts that are annotated for citation intent. The dataset is divided into training and testing sets. The testing set has 1,861 records, which are also annotated. The Sci-Cite dataset is unbalanced, and the number of instances in each category is not the same. The training set has 4,840 from Background, 2,294 from Method, and only 1,109 instances from Results intent class. This may be due to the fact that most of the citations are perfunctory, meaning that, cited paper is providing an understanding of the work.

Sci-cite includes citation context, label/intent, the starting and ending location of citation context, citing and cited paperId, the section in which citation is made. The dataset has three citation intent classes; Background, Method,

[21] https://www.kaggle.com/

and Result. The citation sentences annotated for the same citation intent have similar internal representation. We made clusters of the sentences with similar representation using GloVe, Infersent, and BERT word vectors. In the case of clustering techniques, which requires a pre-defined number of clusters like K-Means, we have set the parameter $k = 3$. This means the citation sentences will be grouped into three different classes: Background, Method, and Result. While other clustering techniques, which do not have a pre-defined number of clusters like HDBSCAN, we selected the clusters having a significant number of candidates listed in those clusters. The clusters having very few candidates in them were merged with larger ones.

Clustering is an unsupervised method in which the first cluster is not necessarily for the first class of citation intent and similarly for others. Therefore, to label the clusters, we assigned the intent of majority samples within a cluster. This means that a class of intent will be assigned to a cluster having the highest number of occurrences of a particular intent class.

B. EXPERIMENTAL SETTINGS

We performed the experiments in different settings of word embedding and clustering algorithms, shown in Table 4. In each setting, we changed the word embedding and clustering algorithms. We selected three different word embedding techniques based on the recommendations of [45]. Similarly, we selected the most suited clustering algorithms for the text clustering task. We compared the results of all the combinations of our chosen algorithms to check which setup performs the best in our environment.

We used Transformers libraries [46] for BERT vectorization, which provides over 32 pre-trained models in over a hundred languages. For Infersent [37] sentence embeddings, we loaded pre-trained model from the model path[22] to encode our sentences. We used 'Common Crawl' pre-trained GloVe vectors, for GloVe vectorization, from,[23] which includes 1.9M word vocabulary. We used HDBSCAN [42] library with scikit-learn [47] and imported KMeansClusterer from nltk.cluster library, for HDBSCAN and K-Means clustering, respectively. We set the cluster size $k = 3$ for K-Means and $min_cluster_size = 30$ while setting parameters for clustering algorithms. For initialization of K-Means we used K-Means++ [48], $init =$ '$k - means{+}{+}$'. K-Means algorithm keeps on updating the centroids while assigning clusters to data points. Therefore, we set a stopping criterion by setting the number of repetitions $repeat = 100$, as recommended by Fränti and Sieranoja *et al.* [49]. We used $cosine_distance$ as a distance metric between data points.

V. RESULTS

We performed experiments of our proposed model in different settings, as shown in Table 4. We compared the results of these settings on Sci-Cite dataset using Precision, Recall, and F1 measures. These measures provide detail of whether the cluster formation in a category is correct. Precision calculates the fraction of pairs correctly put in a cluster, Recall is the fraction of actual pairs that were identified, and F1-measure is a balance between Precision and Recall and is the harmonic mean of these values. By evaluating the results, we wanted to see how well our proposed model is performing? Whether the model is useful, and how will it perform on other datasets? Figure 3 presents the confusion matrix for Sci-Cite actual and predicted values by using our proposed method. The confusion matrix is generated using various setups of vectorization and clustering using sklearn [47], NumPy [50], and seaborn [51] machine learning libraries for making the confusion matrix. It is easy to inspect the prediction errors by using the confusion matrix. All the correct categorization are placed in the diagonal of the matrix. A confusion matrix having comparatively strong points on the diagonals is considered to be having better results. For example, in our case, BERT-KMeans and BERT-HDBSCAN having intense colors in the diagonal, signifying the best models in our case.

For obtaining Precision, we divide the true-positive instances by true-positive and false-positive. The true-positive is the citation instances categorized in their correct categories, whereas the false-positive is the instance correctly not categorized in a particular category.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Table 5 is a multi-class confusion matrix for Infersent-KMeans settings, which we use to calculate Precision, Recall, and F1 score. The table also describes how we labeled the predicted clusters based on the actual intent class values. For example, Cluster-1 has been labeled with Background class as Background class in Actual columns has the highest values. Similarly, Cluster-2 and Cluster-3 are labeled with Method and Result class, respectively.

To calculate the Precision of a model, we calculate the individual Precision of each category in the confusion matrix and then find the average Precision of that model, as given below:

$$\begin{aligned} Precision_{background} &= \frac{3300}{3300 + (760 + 170)} \\ &= 0.78 \\ Precision_{method} &= 0.55 \\ Precision_{result} &= 0.42 \\ Precision_{Average} &= (0.78 + 0.55 + 0.42)/3 \\ &= 58\% \end{aligned}$$

Thus the average Precision of Infersent-KMeans is 58%. Similarly, we calculate the Precision of other confusion matrices as well. Table 6 provides a complete list of the Precision measures for each setting.

Precision answers the question, "What proportion of positive identifications was actually correct?" but it cannot measure "What proportion of actual positives was identified correctly?". We calculate Recall in order to understand the

[22] encoder/infersent%s.pkl

[23] https://nlp.stanford.edu/projects/glove/

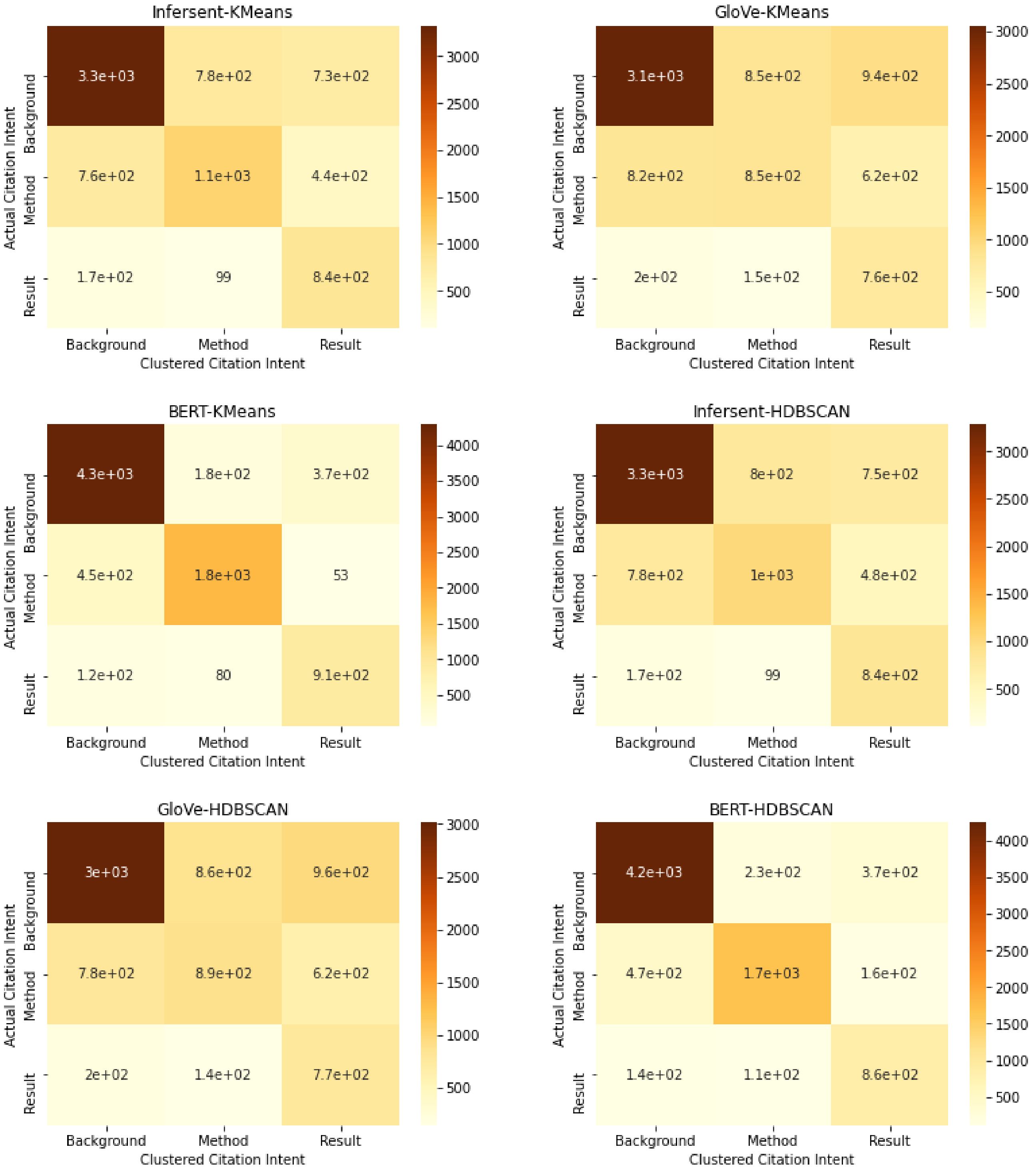


FIGURE 3. Confusion matrix of various experimental settings.

proportion of actual positives correctly identified. We used the following formula to calculate the Recall value:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

The Recall is calculated by finding the individual Recall values of each category in a matrix and then calculating the average Recall of that particular model, as given below:

$$\begin{aligned} Recall_{background} &= \frac{3300}{3300 + (780 + 730)} \\ &= 0.69 \\ Recall_{method} &= 0.48 \\ Recall_{result} &= 0.75 \\ Recall_{Average} &= (0.69 + 0.48 + 0.42)/3 \\ &= 64\% \end{aligned}$$

TABLE 5. Infersent-KMeans multi-class confusion matrix.

Predicted Clusters	Label Assigned	Actual Intent Classes		
		Background	Method	Result
Cluster 1	Background	**3300**	760	170
Cluster 2	Method	780	**1100**	99
Cluster 3	Result	730	440	**840**

TABLE 6. Comparison of Precision, Recall and F-Measures of various setups.

Setup	Setting Name	Precision	Recall	F1
1.	Infersent-KMeans	58%	64%	60%
2.	GloVe-KMeans	51%	56%	51%
3.	BERT-KMeans	81%	82%	81%
4.	Infersent-HDBSCAN	57%	63%	58%
5.	GloVe-HDBSCAN	52%	57%	52%
6.	BERT-HDBSCAN	77%	79%	78%

The average Recall of Infersent-KMeans is 64%. We calculated the Precision of other matrices in the same manner. Precision and Recall values are always in tension, meaning that increasing one value will result in decreasing the other value and vice versa. We must examine both Precision and Recall to evaluate a system entirely. Therefore, we use the weighted average of both Precision and Recall, called the F1 score. F1 score takes both the false-positive and false-negative into account for evaluating a model. We have used the formula given below to calculate the F1 score:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In order to calculate the average F1 score, we calculated the individual F1 score as follows:

$$\begin{aligned} F1_{background} &= 2 * \frac{Precision_{background} * Recall_{background}}{Precision_{background} + Recall_{background}} \\ &= 73\% \\ F1_{method} &= 51\% \\ F1_{result} &= 54\% \\ F1_{Average} &= (0.73 + 0.51 + 0.54)/3 \\ &= 60\% \end{aligned}$$

Table 6, provides a comparison of the measures for each setup. The experimental results show that BERT vectors provide the best results with the KMeans clustering technique. The choice of clustering technique here does not have much impact as it can be seen in measures of setup 3 and 6, where KMeans and HDBSCAN clustering algorithms have been used with BERT vectors, respectively. Replacing KMeans with HDBSCAN has an insignificant impact, which may also be due to the Sci-Cite dataset contents and may provide the same results in other cases. In conclusion, a significant role is played by the choice of embedding technique, which in our case, BERT has provided comparatively better results. BERT uses the contextual relations between words and sub-words in text data. It is pre-trained on massive datasets, which have made it able to understand and encode the contextual relations among words, which is not possible in the case of non-contextual vectors like GloVe.

Sci-Cite has labeled the citation sentences with three different citation intents; Background, Method, and Result having support 4840, 2290, and 1109, respectively. The burst chart in Figure 4 provides a comparison of the measures against each citation intention. The inner layers of the figure mention various settings. Each of the branches provides individual metrics values, including Precision, Recall, and F1, for a particular setting. The support of each of these groups is almost double to its preceding group, which is due to the fact that most of the citations are made for definition, topic introduction, and background study [3]. The values of Precision, Recall, and F1 are very consistent in the case of BERT with KMeans compared to the other settings. The measures are not consistent for different categories of citations, especially in the case of GloVe.

In some cases, like the Background category of GloVe-HDBSCAN, the Precision is high while Recall is low, which means that the model is useful in grouping background citations in the Background category but not that much good in grouping all the background citations. This may be because the Background records ratio in the Sci-Cite dataset is high compared to the Method and Result categories. Similarly, the values of Precision for Result category in GloVe-HDBSCAN are too low, but the Recall value is high.

VI. C2D-I DataSet

After comparing the results based on different setups of our proposed technique, we selected the one having the best results and implemented that model on an un-annotated dataset of C2D. The C2D dataset contains 53 million unique citation information records containing source Id, chapter and paragraph details, author, and publication information, but it does not have the citation intent specified. It was not easy for vectorization and clustering to handle such massive data due to the hardware and time requirements. Sampling methods can effectively reduce the amount of data and help speed up the data processing. Liu and Zhang [52] provide a comparison of different techniques for sampling a vast dataset. In the case of our dataset, we are interested in sampling items belonging to different citation intents. As we do not have the citation reason details provided in the C2D dataset, we use a stratified sampling method to divide the population into groups based on the section in which the citation has been made. We use section information for grouping because citation reason directly relates to the section in which the citation has been made [53]. We divided the C2D dataset into four groups based on the IMRaD structure of research articles for sampling. We then chose an equal number of citation instances from each group to make a sample of 10 million records.

Once the sample dataset was created, we pre-processed the citation context of the C2D dataset, as explained in Section III-A. After pre-processing, we performed word

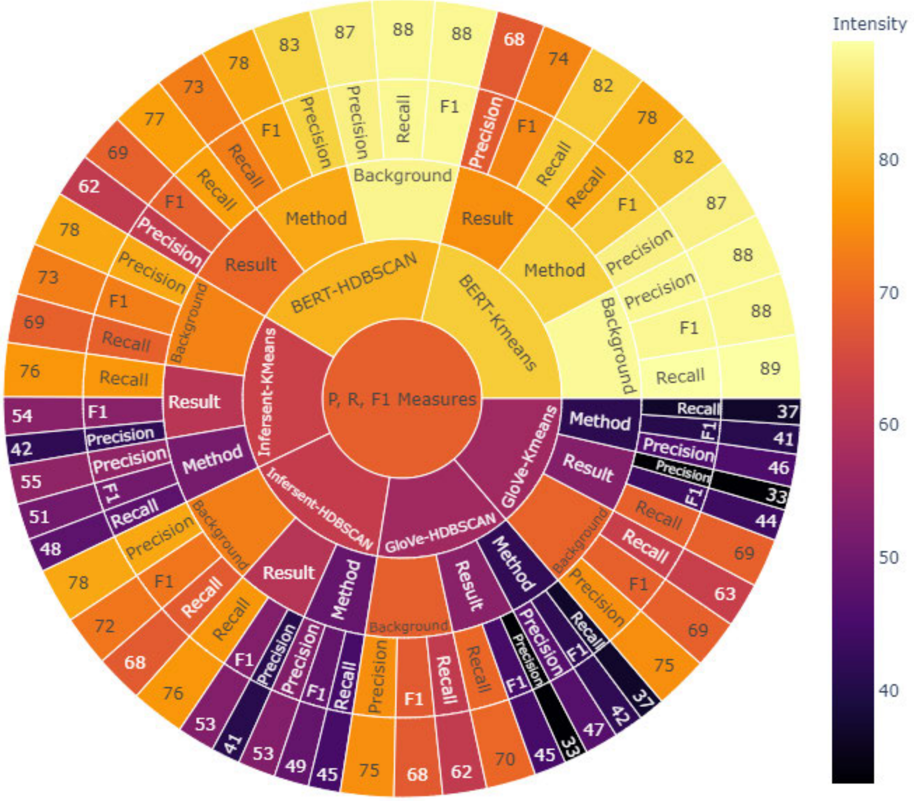


FIGURE 4. Citation reason category wise Precision, Recall and F1 measures in various setups.

embedding of citation context using the BERT word embedding technique. We applied the KMeans algorithm to it in order to form clusters of the citation contexts of C2D. We then manually evaluated the clusters by randomly taking any sentences from each cluster. This way, we mapped the clusters, created by clustering algorithms, to citation intent classes. As a result, we annotated a 10 million citation context using our proposed technique. The new annotated dataset is known as C2D with Intent (C2D-I) and can be provided upon request. C2D-I is a comparatively large dataset, and it includes the citation context as well as citation intent; therefore, it is best suited for deep analysis.

VII. CONCLUSION

Finding citation intent is vital for citation analysis, for which we need a substantial labeled dataset. This study provided a critical analysis of the existing datasets and discussed their limitations while using them for citation intent extraction. We proposed a text clustering based mechanism to annotate un-annotated dataset by using the citation context. For that, we first proposed our method and evaluated it on the pre-annotated dataset, Sci-Cite. We achieved outstanding results in terms of Precision, Recall, and F1 measures. We observed that contextual embedding could play an essential role in grouping the citation sentences, as the contextual sentences provided better results than non-contextual embedding. After evaluating our proposed mechanism, we annotated the C2D un-annotated dataset and created a new dataset on top of it, called C2D with intent (C2D-I), which can be provided on request. In the future, this annotated dataset can be used for evaluating newly proposed frameworks for citation intent classification. We have used only limited citation intent classes, including Background, Method, and Result provided in the Sci-Cite dataset. There are several other citation reason classes reported in the literature [1], [3]. In further studies, we shall evaluate our proposed technique on different citation intent classes from a variety of dataset.

REFERENCES

[1] S. Teufel, A. Siddharthan, and D. Tidhar, "Automatic classification of citation function," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2006, pp. 103–110.

[2] J. M. Ziman, *Public Knowledge: An Essay Concerning the Social Dimension of Science*, vol. 519. Sydney, NSW, Australia: CUP Archive, 1968. [Online]. Available: https://www.aclweb.org/anthology/volumes/W06-16/

[3] M. J. Moravcsik and P. Murugesan, "Some results on the function and quality of citations," *Social Stud. Sci.*, vol. 5, no. 1, pp. 86–92, Feb. 1975.

[4] J. Swales, *Genre Analysis: English in Academic and Research Settings*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[5] N. Harwood, "An interview-based study of the functions of citations in academic writing across two disciplines," *J. Pragmatics*, vol. 41, no. 3, pp. 497–518, Mar. 2009.

[6] M. Valenzuela, V. Ha, and O. Etzioni, "Identifying meaningful citations," in *Proc. 29th AAAI Conf. Artif. Intell. Workshops*, 2015, p. 13.

[7] A. Cohan, W. Ammar, M. van Zuylen, and F. Cady, "Structural scaffolds for citation intent classification in scientific publications," Sep. 2019, *arXiv:1904.01608*. [Online]. Available: http://arxiv.org/abs/1904.01608

[8] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," 2019, *arXiv:1903.10676*. [Online]. Available: http://arxiv.org/abs/1903.10676

[9] K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. Weld, "S2ORC: The semantic scholar open research corpus," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4969–4983.

[10] A. Abu-Jbara, J. Ezra, and D. Radev, "Purpose and polarity of citation: Towards NLP-based bibliometrics," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 596–606.

[11] D. Jurgens, S. Kumar, R. Hoover, D. McFarland, and D. Jurafsky, "Measuring the evolution of a scientific field through citation frames," *Trans. Assoc. Comput. Linguistics*, vol. 6, pp. 391–406, Dec. 2018.

[12] A. Shahid and M. T. Afzal, "Section-wise indexing and retrieval of research articles," *Cluster Comput.*, vol. 21, no. 1, pp. 481–492, Mar. 2018, doi: 10.1007/s10586-017-0914-4.

[13] H. Small, "Interpreting maps of science using citation context sentiments: A preliminary investigation," *Scientometrics*, vol. 87, no. 2, pp. 373–388, May 2011.

[14] C. L. Giles, K. D. Bollacker, and S. Lawrence, "CiteSeer: An automatic citation indexing system," in *Proc. 3rd ACM Conf. Digit. Libraries (DL)*, 1998, pp. 89–98.

[15] A. Prasad, M. Kaur, and M.-Y. Kan, "Neural ParsCit: A deep learning-based reference string parser," *Int. J. Digit. Libraries*, vol. 19, no. 4, pp. 323–337, Nov. 2018.

[16] M. Callaham, R. L. Wears, and E. Weber, "Journal prestige, publication bias, and other characteristics associated with citation of published studies in peer-reviewed journals," *JAMA*, vol. 287, no. 21, pp. 2847–2850, 2002.

[17] G. Hendricks, D. Tkaczyk, J. Lin, and P. Feeney, "Crossref: The sustainable source of community-owned scholarly metadata," *Quant. Sci. Stud.*, vol. 1, no. 1, pp. 414–427, Feb. 2020.

[18] J. Tang, "AMiner: Toward understanding big scholar data," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, Feb. 2016, p. 467.

[19] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. Hsu, and K. Wang, "An overview of microsoft academic service (MAS) and applications," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 243–246.

[20] D. R. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara, "The ACL anthology network corpus," *Lang. Resour. Eval.*, vol. 47, no. 4, pp. 919–944, Dec. 2013.

[21] B. Yaman, M. Pasin, and M. Freudenberg, "Interlinking scigraph and dbpedia datasets using link discovery and named entity recognition techniques," in *Proc. 2nd Conf. Lang., Data Knowl. (LDK)*. Wadern, Germany: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019, pp. 1–8.

[22] D. Upton, P. Upton, T. Jones, K. Jutlla, D. Brooker, and H. Grove. (2019). *Core–Aggregating the World's Open Access Research Papers 2011, Evaluation of the Impact of Touch Screen Technology on People With Dementia and Their Carers Within Care Home Settings*. Accessed: Jun. 16, 2019. [Online]. Available: https://core.ac.uk/download/pdf/51151130.pdf

[23] T. Saier and M. Färber, "Bibliometric-enhanced arxiv: A data set for paper-based and citation-based tasks," in *Proc. BIR@ ECIR*, 2019, pp. 14–26.

[24] M. Färber, A. Thiemann, and A. Jatowt, "A high-quality gold standard for citation-based tasks," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018, pp. 1–5.

[25] A. Khadka and P. Knoth, "Using citation-context to reduce topic drifting on pure citation-based recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 362–366.

[26] M. Ley, "DBLP: Some lessons learned," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1493–1500, Aug. 2009.

[27] C. Jeong, S. Jang, E. Park, and S. Choi, "A context-aware citation recommendation model with bert and graph convolutional networks," *Scientometrics*, vol. 124, pp. 1–16, Jul. 2020.

[28] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manage.*, vol. 50, no. 1, pp. 104–112, Jan. 2014.

[29] E. Loper and S. Bird, "NLTK: The natural language toolkit," in *Proc. ACL Workshop Effective Tools Methodologies Teaching Natural Lang. Process. Comput. Linguistics*. Philadelphia, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70.

[30] M. Kantrowitz, B. Mohit, and V. Mittal, "Stemming and its effects on TF-IDF ranking (poster session)," in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*. New York, NY, USA: Association for Computing Machinery, 2000, pp. 357–359, doi: 10.1145/345508.345650.

[31] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 60, no. 5, pp. 493–502, Oct. 2004.

[32] P. Willett, "The porter stemming algorithm: Then and now," *Program*, vol. 40, no. 3, pp. 219–223, Jul. 2006.

[33] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 238–247.

[34] G.-H. Liu and J.-Y. Yang, "Image retrieval based on the texton co-occurrence matrix," *Pattern Recognit.*, vol. 41, no. 12, pp. 3521–3527, Dec. 2008.

[35] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[37] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 670–680. [Online]. Available: https://www.aclweb.org/anthology/D17-1070

[38] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. NAACL*, 2018, pp. 2227–2237.

[39] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.

[40] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[41] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, Mar. 2009.

[42] L. McInnes, J. Healy, and S. Astels, "Hdbscan: Hierarchical density based clustering," *J. Open Source Softw.*, vol. 2, no. 11, p. 205, Mar. 2017.

[43] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: An efficient data clustering method for very large databases," *SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, Jun. 1996, doi: 10.1145/235968.233324.

[44] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[45] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C.-J. Kuo, "Evaluating word embedding models: Methods and experimental results," *APSIPA Trans. Signal Inf. Process.*, vol. 8, no. 1, pp. 1–14, Jul. 2019, Art. no. E19. [Online]. Available: https://www.cambridge.org/core/services/aop-cambridge-core/content/view/EDF43F837150B94E71DBB36B28B85E79/S204877031900012Xa.pdf/div-class-title-evaluating-word-embedding-models-methods-and-experimental-results-div.pdf

[46] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*. Stroudsburg, PA, USA: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[48] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 2006-13, Jun. 2006. [Online]. Available: http://ilpubs.stanford.edu:8090/778/

[49] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognit.*, vol. 93, pp. 95–112, Sep. 2019.

[50] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[51] M. Waskom. (Sep. 2020). *Mwaskom/Seaborn*. [Online]. Available: https://seaborn.pydata.org/citing.html, doi: 10.5281/zenodo.592845.

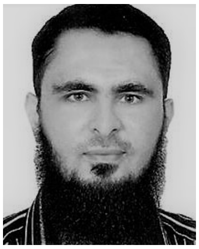
[52] Z. Liu and A. Zhang, "A survey on sampling and profiling over big data (technical report)," 2020, *arXiv:2005.05079*. [Online]. Available: http://arxiv.org/abs/2005.05079

[53] A. Y. Khan, A. K. Shahid, and M. T. Afzal, "Extending co-citation using sections of research articles," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 26, no. 6, pp. 3345–3355, 2018.

MUHAMMAD ROMAN received the M.S. degree in computer science from the Kohat University of Science and Technology, Kohat, Pakistan, in 2015, where he is currently pursuing the Ph.D. degree in computer science with the Institute of Computing. He is currently a senior software developer for the last 12 years. His research interest includes investigating maps of science using contextual proximity of citations, recommending relevant documents, information systems, deep learning, and natural language processing.

ABDUL SHAHID received the Ph.D. degree in computer science from the Capital University of Science and Technology, Islamabad, Pakistan. He is currently a Faculty Member with the Institute of Computing, Kohat University of Science and Technology, Pakistan. He is a professional software engineer and a consultant with software companies for the last 13 years. His research focuses on information system, digital libraries, recommending relevant documents with the help of in-text citation frequencies and patterns. In this field, he has authored or coauthored a number of good quality articles in different international conferences and journals.

SHAFIULLAH KHAN received the Ph.D. degree in wireless networks security from Middlesex University, U.K. He is currently an Associate Professor with the Institute of Information Technology, Kohat University of Science and Technology, Pakistan. His research mainly focuses on wireless broadband network architecture, security and privacy, security threats, and mitigating techniques. He is serving as an editor in many well-reputed international journals.



ANIS KOUBAA is currently a Professor in computer science and the Leader of the Robotics and Internet of Things Research Laboratory, Prince Sultan University. He is also a Research and Development Consultant with Gaitech Robotics, China, and a Senior Researcher with CISTER/INESC TEC and ISEP-IPP, Porto, Portugal. He is also a Senior Fellow of the Higher Education Academy in U.K. He is also an ACM Distinguished Speaker. He received several distinctions and awards, including the Rector Research Award in 2010 from Al-Imam Mohamed bin Saud University and the Rector Teaching Award in 2016 from Prince Sultan University. He has been the Chair of the ACM Chapter in Saudi Arabia since 2014.

LISU YU (Member, IEEE) received the B.E. degree from the Mao Yisheng Honors College, School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, in 2014, and the Ph.D. degree from the Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, in 2019. He was a Visiting Scholar with the University of Arkansas, Fayetteville, AR, USA, and the University of Houston, Houston, TX, USA, from 2017 to 2019. He is currently a Distinguished Associate Professor with the School of Information Engineering, Nanchang University, China. His main research interests include advanced wireless communications, coded modulation, non-orthogonal multiple access, fiber wireless communication, ultra-dense networks, unmanned aerial vehicle, and visible light communication. He serves as a member of the IEEE Communications Society Technical Committee on Green Communications and Computing and the Signal Processing and Computing for Communications. He has served as the Student Activities Chair for the IEEE Communication Society Chengdu Chapter and several international conferences technical program committee member, section chair, and special track chair. He is currently serving as an Academic Editor for the PLOS ONE.