CovTiNet: Covid Text Identification Network using Attention based Positional Embedding Feature Fusion

Md. Rajib Hossain[1], Mohammed Moshiul Hoque[1*], Nazmul Siddique[2] and Iqbal H. Sarker[1,3]

[1]Department of Computer Science & Engineering, Chittagong University of Engineering & Technology, Chittagong, 4349, Chittagong, Bangladesh.
[2]School of Computing, Engineering and Intelligent Systems, Ulster University, UK.
[3]Security Research Institute, Edith Cowan University, WA 6027, Australia.

*Corresponding author(s). E-mail(s): moshiul_240@cuet.ac.bd; Contributing authors: rajcsecuet@gmail.com; nh.siddique@ulster.ac.uk; m.sarker@ecu.edu.au;

Abstract

Covid text identification (CTI) is a crucial research concern in Natural Language Processing (NLP). Social and electronic media are simultaneously adding a large volume of Covid-affiliated text on the World Wide Web due to the effortless access to the internet, electronic gadgets and the Covid outbreak. Most of these texts are uninformative and contain misinformation, disinformation, and malinformation that create an infodemic. Thus, Covid text identification is essential for controlling societal distrust and panic. Though very little Covid-related research (such as Covid disinformation, misinformation and fake news) has been reported in high-resource languages (e.g., English), CTI in low-resource languages (like Bengali) is in the preliminary stage to date. However, automatic CTI in Bengali text is challenging due to the deficit of benchmark corpora, complex linguistic constructs, immense verb inflexions and scarcity of NLP tools. On the other hand, the manual processing of Bengali Covid texts is arduous and costly due to their messy or unstructured forms. This research proposes a deep learning-based network (CovTiNet) to identify Covid text in Bengali. The CovTiNet incorporates an attention-based position embedding feature fusion for text-to-feature representation and attention-based CNN for Covid text identification. Experimental results show that the proposed CovTiNet achieved the highest accuracy of 96.61±.001% on the developed dataset (***BCovC***) compared to the other methods and baselines (i.e., BERT-M, IndicBERT, ELECTRA-Bengali, DistilBERT-M, BiLSTM, DCNN, CNN, LSTM, VDCNN, and ACNN).

Keywords: Natural language processing, Covid text identification, Positional encoding, Self-attention, Embedding feature fusion, Deep-leaning, Transformers, Low-resource languages.

1 Introduction

Covid was declared a Public Health Emergency of International Concern (PHEIC) by the World Health Organization (WHO). It was first reported in Wuhan, China, in December 2019 and is spreading gradually all over the World [1]. As of 20 January 2022, the total of infected cases is 339 million, with total deaths of 5.58 million and recovered of 273.20 million in the World[1]. It is a new disease for the general people, and a so-called issue for research communities, securities agencies, health organizations, financial institutes, and country policymakers [2]. Covid Text Identification (CTI) is an emerging research issue in the realm of Natural Language Processing (NLP), where an *intelligent system* can automatically identify a piece of text has Covid-related information or not. A covid text may contain misinformation, disinformation, fake news, and other details on covid.

Most countries impose lockdowns, shutdowns, social distancing and other social activities to control the spreading of Covid. As a result, the emergency announcement, vaccination information, and other essential policymakers' information are shared using social media and electronic press for familiar people [3]. People's emotions, opinions, needs, support seeking and surrounding emergency conditions are also disseminated in the text through electronic and social media. Due to these activities, a massive volume of text is generated and included on social media and the Web. However, most of the texts are unlabelled and unstructured. As a result, it is impracticable and challenging to manually extract covid related information from the messy volumes of text. On the other hand, manual mining consumes tremendous time and incurs costs. Thus, an *intelligent CTI system* can overcome the limitations of the manual identification system with fast and effective covid text detection. It also assists policymakers and ordinary people to share covid related information through social and electronic media at a rapid pace, reducing physical movement, panic, and infodemic. CTI has also reduced the time and

[1] https://www.worldometers.info/coronavirus/

search complexity for different NLP downstream tasks such as covid fake news detection, covid misinformation and disinformation classification [4].

However, developing an *intelligent* and efficient CTI system regarding under-resourced languages like Bengali is challenging due to the unavailability of benchmark corpora, lack of features extraction techniques, and colossal word inflexion rate. Moreover, a huge variation of morphological structures (i.e., Sadhu-bhasha and Cholito-bhasha), well-off dialects, and person-tense-aspect agreement make the task more complicated [5]. For these attributes, a *single embedding (SE)* method is unable to capture holistic semantic and syntactic linguistics features of text [6]. The different embedding methods (e.g., GloVe, FastText, Word2Vec) represent different feature distributions, and the performance of the downstream model varies from one embedding to another [7]. On the other hand, GloVe and Word2Vec are not able to manage the *Out-of-Vocabularies (OOV)* issues, whereas FastText can manage the OOV issues using sub-tokenization techniques. Although several low-resource (e.g., Bengali and Urdu) text classification researches have been conducted based on statistical [8] and deep learning-based approaches [9–11]. None of these works addressed the OOV, positional encoding, and single embedding issues in Bengali. Moreover, no past studies in Bengali performed Covid text identification tasks using intrinsic and extrinsic evaluations to the best of our knowledge. To summarize the research insights, this work sought the answers to the following research questions (RQs):

- **RQ1:** How to develop a Covid text corpus in Bengali for intelligent CTI.
- **RQ2:** How to choose the best embedding model to perform the CTI task with intrinsic evaluation?
- **RQ3:** How to develop a deep-learning-based framework for CTI tasks in Bengali incorporating attention-based positional embedding feature fusion?
- **RQ4:** How does the attention-based positional embedding feature fusion improve the performance of non-contextual single embedding in Bengali CTI?

To address the research questions **(RQ1-RQ4)**, this work proposes a covid text identification network called **CovTiNet** to identify the textual information related to covid in Bengali with the development of a Bengali Covid text identification corpus (*BCovC*). The proposed network reduces the OOV problems and overcomes the limitations of non-contextual single embedding feature extraction with the positional encoding technique. The CovTiNet also evaluates the embedding and classification models using *intrinsic* and *extrinsic* methods. The notable contributions of this research and possible answers to the research questions (ARQ) are summarized as follows:

- **ARQ1:** Present a detailed development process of the Bengali Covid text corpus (*BCovC*), including data collection, preprocessing, annotation, and annotation quality measures. To the best of our knowledge, this corpus is the first developed dataset in Bengali, which may alleviate the corpus unavailability issues in developing CIT in Bengali (Sec. 4). This research also developed a Covid embedding corpus (i.e., *CovEC*) and an intrinsic evaluation dataset (i.e., IEDs) for evaluating embedding models.

- **ARQ2:** Exploration of the intrinsic evaluation methods based on Spearman and Pearson correlations which helps to select the best embedding model for the downstream task (e.g., CTI) with a reduced training time and memory storage (Sec. 5.1.1 and Sec. 7.1).
- **ARQ3:** Propose a model (CovTiNet) for CTI by integrating the attention-based positional embedding feature fusion and Attention-based Convolution Neural Network (ACNN). This model adds the word position information and fuses the semantic/syntactic features of attention-based embedding models that improve the classification performance (Sec. 5.1.1).
- **ARQ4:** Present a comparative performance analysis between the proposed system (CovTiNet) and baseline methods (e.g., LibSVM, CNN, LSTM, BiLSTM, DCNN, VDCNN and transformer-based fine-tuning) with a detailed summary of the model's weakness and strengths (Sec. 7.2).

Additionally, The presented work provides comprehensive future research directions on NLP downstream tasks for morphologically rich languages like Bengali and highlights forthcoming research scopes for the research communities of the *Bengali Covid text mining or information retrieval* domain.

The rest of the paper is arranged as follows: Section 2 presents the related work and the problem statement is described in Section 3. Section 4 illustrates the development of the Bengali Covid text identification corpus, whereas Section 5 describes the proposed CTI framework. Section 6 explores the experiments and the analysis of results are summarised in Section 7. A detailed error analysis of the model and a failure case study is explained in Sections 7.6 and 8. Section 9 concludes the work with future recommendations for improvements.

2 Related Work

Covid text identification is a new and evolving research concern in recent times. Although many essential Covid related texts are being spontaneously included on the Web at a rapid pace, unwanted or undesired textual contents are also added owing to the rapid usage of the internet, and social media [12]. A few studies recently explored Covid text mining concerning high-resource languages [13], but Covid text analysis is in a primitive stage regarding under-resourced languages like Tamil and Bengali. Therefore, CTI is a significant research challenge in low-resource languages. Kolluri et al. [14] developed a machine learning (ML) based English Covid news verification system, but their system is limited to an API request in a day involving cost per request. Ng et al. [15] built a large-scale English newspaper Covid-related text corpus containing 10 Billion words of 7,000 news. They explored the ML-based topics mining method to detect the five most frequent Covid topics (e.g., Coronavirus, Covid, Covid, nCoV, and SARS-CoV-2). A deep learning (DL) based approach (e.g., LSTM with GloVe) was deployed for social media tracking during the pandemic at New York [16]. However, the LSTM+GloVe-based DL method only experimented with English social media text. Koh et al. [17] investigated loneliness during the pandemic from Twitter data using topic-based mining. The topic-based ML mining methods explored only English Twitter texts.

Covid fake news, disinformation and misinformation identification have been trending research topics in the NLP domain. Paka et al. [18] constructed a Covid fake news text dataset (e.g., CTF) and developed an attention-based Covid fake news framework that achieved an F1-score of 95.00%. A traditional ML-based (LibSVM, DT, KNN, and NN) voting ensemble method has been developed for Covid misleading information detection system [19]. This method can not work on short-text samples. Song et al. [20] explored a Covid disinformation framework and evaluated it on the largest Covid disinformation dataset[2] of 70 countries and 43 languages. However, their system is not considered the Bengali Covid text. Ghasiya et al. [21] analyzed the public sentiment from newspaper headlines of four countries (UK, India, Japan & South Korea). More than 100,000 Covid texts were collected from newspaper headlines and achieved a maximum accuracy of 90.00%. Their unsupervised topic model method is not capable of capturing context-based information.

Covid text analysis in resource-constrained languages is an underdeveloped research field due to the shortage of annotated corpora and lack of well-tuned embedding and classification models [22]. Patwa et al. [23] built a Hindi hostile post dataset and developed an identification system for online Hindi hostile posts. They used m-BERT embedding with the LibSVM classification method for detecting hostile and non-hostile posts and achieved a maximum of 84.11% accuracy for Coarse-grained classification. Hussein et al. [24] developed an Arabic Covid infodemic detection system using tweets text. This work can classify seven predefined queries (on 2,556 Arabic tweets) and obtain maximum accuracy of 67.7% using the AraBERT framework. Mattern et al. [25] developed the German Covid fake news corpus, which contains 28,056 actual and 13,186 fake news. Their BERT + Social context system gained the maximum accuracy of 82.40% on the developed dataset. A LibSVM-based classification method was explored for the Persian fake news detection system and obtained maximum accuracy of 87.00% [26]. Harakawa et al. [27] developed a tweeter keyword extraction method for Japanese text, which only carried out the word-level feature and did not consider the sentence-level linguistics semantics.

Most previous studies of CTI were conducted in English, including fake news classification, misinformation, and disinformation detection using statistical ML and transformer-based learning [28]. In contrast, some research on CTI has been conducted in Arabic, German, Indian and Persian languages [22]. However, non of the past studies have addressed CTI in Bengali. Moreover, other resource-constrained languages only considered the single embedding and transformer-based models. However, single embedding techniques cannot represent the holistic features and can not overcome the OOV issues [29]. Therefore, to address the shortcomings of past studies, this research introduced the fusion-based embedding feature representation method for Bengali

[2]https://www.poynter.org/ifcn-Covid-misinformation/

CTI and experimented with the developed Bengali Covid text corpus with different hyperparameters settings. As far as we are concerned, this work is the first attempt to develop a CTI network in Bengali by integrating the attention-based positional embedding feature fusions and CNN. The proposed network can handle Bengali morphological variation issues and minimize the OOV problems.

3 Problem Statement

The central concern of this study is to develop a text classification framework that can identify Bengali Covid-related text. In particular, this work aims to develop a framework that can classify a Bengali text into *Covid* or *not Covid*. The framework comprises three components: (i) Covid corpus development, (ii) Leveraging Deep Models for CovTiNet Selection, and (iii) CovTiNet.

Covid corpus development: develop a Python scrapper which inputs a valid Bengali Web URL from a set of URLs taken from Social media and Newspapers. The scrapper outputs a list of unlabelled Bengali texts. The scrapper is defined by Eq. 1.

$$t_i = \Upsilon(L_j), i = 1, ..., N, j = 1, ..., Z \tag{1}$$

The scrapper function $\Upsilon(.)$ takes input URL from the list and checks the *robot.txt* policy and scrapped the Bengali Web text (t_i). The list of texts t_i can have a maximum N number of crawled texts from a set of URLs. The crawled, unlabelled and noisy texts are pre-processed and annotated. The quality of annotation is measured using Eq. 2.

$$BCovC(\{tn_k, tc_l\}) = \Gamma(t_i), k = 1, ..., n, l = 1, ..., m, i = 1, ..., N \tag{2}$$

Here, tn denotes non-Covid texts, and tc represents Covid texts. The function $\Gamma(.)$ sequentially prepossess t_i, annotates manually (e.g., by the annotators), verifies (e.g., by the domain expert), and finally measures the Kappa score of the $BCovC$ corpus.

Leveraging Deep Models for CovTiNet Selection: Initially, generate the embedding model using Eq. 3.

$$\begin{split} S_{ab} = \Omega(CovEC, a, b), a = \{GloVe, FastText, Word2Vec\}, \\ b = \{(ED_1, CW_1), ..., (ED_{E^n}, CW_{E^n})\} \end{split} \tag{3}$$

The $CovEC$ is the Covid embedding corpus, a is the set of methods, and b denotes the set of hyperparameter combinations. The E^n indicates the total number of hyperparameters combinations (i.e., embedding dimension and cortex windows). The $\Omega(.)$ produces 18 embedding models. This research applies the intrinsic evaluation to select the best-performing three embedding models to reduce the time complexity of the downstream task (e.g., text identification).

Eq. 4 selects the best three embedding models.

$$\begin{aligned} B_a = \Delta 3(S_{ab}), a = \{GloVe, FastText, Word2Vec\}, \\ b = \{(ED_1, CW_1), ..., (ED_{E^n}, CW_{E^n})\} \end{aligned} \tag{4}$$

Here, $\Delta 3(.)$ represents the intrinsic evaluator, which returns the best-performed three embedding models based on Spearman and Pearson correlation scores. Three single embedding models are used for attention-based positional embedding feature fusion purposes. Now, the $BCovC$ $=\{T^n \cup T^e\}$ is randomly split into training (T^n) and testing (T^e) sets, e.g., $T^n = \{(tn^{lb}_{k^n}, yn^{lb}_{k^n}),$ $(tc^{lb}_{l^n}, yc^{lb}_{l^n})\}$, where $k^n = 1, ..., p^{lb}, l^n = 1, ..., q^{lb}$. Here, k^{nth} non-Covid text and corresponding labelled are represented by $tn^{lb}_{k^n}$ and $yn^{lb}_{k^n}$ whereas Covid text and corresponding labelled are represented by $tc^{lb}_{k^n}$ and $yc^{lb}_{k^n}$) respectively. The p^{lb} and q^{lb} indicate the total number of training non-Covid and Covid samples in the T^n. Similarly the testing set is represented by $T^e = \{(tn^{ul}_{i^e}, yn^{ul}_{i^e}), (tc^{ul}_{j^e}, yc^{ul}_{j^e})\}$, where $i^e = 1, ..., p^{ul}, j^e = 1, ..., q^{ul}$. Here, p^{ul} and q^{ul} denote the total number of unlabelled non-Covid and Covid samples in T^e. The features of training and testing sets are extracted using Eq. 5.

$$FM_{qa}/FM_{q'a} = M(B_a, T^n_q/T^e_{q'}), q = 1, ..., (p^{lb}+q^{lb}), q' = 1, ..., (p^{ul}+q^{ul}) \tag{5}$$

Here, $M(.)$ generates the feature matrix (FM_{qa} for training & $FM_{q'a}$ for testing) of training or testing sample for B_a. The non-contextual embedding methods do not carry contextual or word position information. This study introduces the position encoding (PE_{qa}) technique to overcome this issue. The q^{th} training sample positional embedding is a feature matrix (FM_{qa}). Thus, FM_{qa} is modified by adding PE_{qa} expressed by $FM_{qa} = FM_{qa} + PE_{qa}$. The a^{th} best-performed feature matrix is calculated by employing the self-attention and producing the attention-based feature matrix (Eq. 6).

$$FM^{'}_{qa}/FM^{'}_{q'a} = Attention(W^{aQ}, W^{aK}, W^{FMqa}/W^{FMq^{'}a}, FM_{qa}/FM_{q'a}) \tag{6}$$

Here, $q/q^{'}$ denotes the embedding samples, and FM denotes the feature matrix. The trainable weight matrices are denoted by W^{aQ}, W^{aK} and $W^{FMqa}/W^{FMq^{'}a}$ respectively. The attention-based positional embedding feature matrices are denoted by the $FM^{'}_{qa}$ and $FM^{'}_{q'a}$. The value of q, $q^{'}$ and a are defined in Eqs. 4-5. The training/testing samples ($q/q^{'}$) and a^{th} best performing positional embedding feature matrix ($FM_{qa}/FM_{q'a}$) are just addition to the attention-based feature matrix ($FM^{'}_{qa}/FM^{'}_{q'a}$) and normalized using $ALN(.)$ function, i.e., $\lambda_{qa}/\lambda_{q'a} = ALN(FM^{'}_{qa}/FM^{'}_{q'a} + FM_{qa}/FM_{q'a})$. Finally, normalized feature matrices fuse the feature values using Eq. 7.

$$FM_q/FM_{q'} = \Psi_{i'}(\lambda_{qa}/\lambda_{q'a}), i^{'} = \{ConCat, Average, ConCat - PCA\} \tag{7}$$

$\Psi_{i'}(.)$ denotes the fusion function, which sequentially fuses the possible combination of normalized feature matrices using the best performing embedding model $a = \{GloVe, FastText, Word2Vec\}$. The covid-related text identification model is generated by Eq. 8.

$$\Theta_{k'} = \Phi^{tr}(FM_q), k^{'} = 1, ..., F^n, q = 1, ..., (p^{lb} + q^{lb}) \tag{8}$$

Here, Φ^{tr} indicates the Covid related text identification training method, F^n denotes the total number of Covid identification models and $\Theta_{k'}$ represents the $k^{'th}$ identification model.

In the fourth module, Covid text identification models are evaluated using the testing set T^e by Eq. 7.

$$O_{k'} = \Phi^{ts}(FM_{q'}, \Theta_{k'}), k^{'} = 1, ..., F^n, q' = 1, ..., (p^{ul} + q^{ul}) \tag{9}$$

where $O_{k'}$ denotes the $k^{'th}$ output of Covid text identification model using the testing method $\Phi^{ts}(.)$.

$$CovTiNet = max[\Theta_{k'}(O_{k'})] \tag{10}$$

CovTiNet is the best performing among F^n models with maximum $O_{k'}$, i.e, maximum accuracy.

CovTiNet: integrate attention-based position embedding averaging of GloVe and FastText (**APeAGF**) for text-to-feature representation and attention-based convolutional neural networks (**ACNN**) for Covid text identification.

4 Corpora Development

Textual data collection, preprocessing, and standardization are challenging tasks for low-resource languages due to open access to text archives and lack of research [30]. The Covid pandemic has created an opportunity for developing Covid text-related corpora. As a result, few corpora are available in the high-resource language (like English). However, no Covid identification corpus is available in Bengali to our knowledge. However, the availability of benchmark corpora is a prerequisite to developing any intelligent text processing system. Thus, this work aims to develop a few corpora to perform CTI tasks in Bengali. Fig. 1 depicts the Covid corpus development details. The following subsections illustrate the development process of the three corpora: Bengali Covid text corpus ($BCovC$), Covid embedding corpus ($CovEC$), and Intrinsic evaluation dataset ($IEDs$).

4.1 Bengali Covid Text Corpus ($BCovC$)

This work proposed two Algorithms to develop Covid text corpora. Algorithm 1 uses for scrapping Web text, whereas Algorithm 2 utilizes for preprocessing,

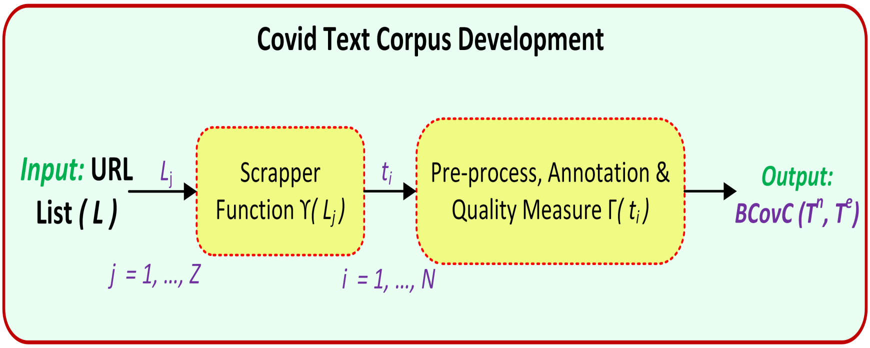


Fig. 1: Schematic Representation of Covid Corpus Development

annotation and annotation quality measures. In Algorithm 1, the function $\Upsilon(.)$ takes the list of Web URLs. The $scraper(.)$ function is a dynamic function that changes the parsing function based on a specific Web URL. The $parser(.)$ function parses the Web content to readable text and converts it to UTF-8. Finally, a total of 159,822 texts file are returned from this function (e.g., $\Upsilon()$) as list t. The texts are collected from 3 June 2020 to 15 August 2021 from popular social media sites, online news portals, and blogs.

Algorithm 1 Web Text Scrapping

$t = []$ ▷ Initial empty scrapped texts list
procedure $\Upsilon(UrlList\ L)$ ▷ Web URLs list
  for i in L **do**
    $st = scraper(i)$ ▷ Scrapping for i^{th} URL
    $pt = parser(st)$ ▷ Parse the i^{th} URL content
    $t.append(pt)$ ▷ Append the i^{th} URL UTF-8 format texts
  end for
  $return\ (t)$
end procedure

In Algorithm 2, the function $\Gamma(.)$ takes the input as noisy text list t and returns the developed corpus $BCovC$. In the first step, each text is cleaned using the text preprocessing function $Bclean(.)$. The $Bclean(.)$ function first removes all non-Bengali characters, digits and regular expressions. Then removes the THML tags, hashtags and special characters which cannot convert UTF-8. Finally, replaces the extra space, duplicate text and newline. In this step, 157,771 texts are taken, and 2,051 texts are removed due to several preprocessing operations.

Two undergraduate students manually annotated each preprocess text (pt) in the second step. The $annotator1$ manually labelled α_a and text list α_{ta}. Whereas $annotator2$ manually labelled α_b and text list is α_{tb}. If the first and

Algorithm 2 Web Text Pre-processing, Annotation & Quality Measurements

procedure $\Gamma(t)$ ▷ Noisy and unlabelled texts list
    $BCovC = \{\}$ ▷ Bengali Covid related text corpus
    $pt = []$ ▷ Reprocessed empty list
    //**First step: Text Preprocessing**
    for $i\ in\ t$ **do**
        $it = Bclean(i)$ ▷ Bengali text preprocessing
        $pt.append(it)$
    end for
    //**Second step: Text Manual Annotation**
    $\alpha_a, \alpha_{ta} = annotator1(pt)$ ▷ First manual annotation
    $\alpha_b, \alpha_{tb} = annotator2(pt)$ ▷ Second manual annotation
    $eT = [], idx = 1$
    for $i\ in\ pt$ **do**
        if $(i\ in\ \alpha_{ta})\ or\ (i\ in\ \alpha_{tb})$ **then**
            if $\alpha_{ta}[i] == \alpha_{tb}[i]$ **then** ▷ Both annotators are agreed
                $BCovC[idx = idx + 1] = i$
            end if
            if $\alpha_{ta}[i]! = \alpha_{tb}[i]$ **then** ▷ Annotators with different agreement
                $eT.append(i)$
            end if
        end if
    end for
    //**Third step: Expert Level Verification**
    $\alpha_e, \alpha_{te} = expert(eT)$
    for $i\ in\ range(1, len(eT))$ **do**
        if $\alpha_e[i] == 1$ **then** ▷ Expert is agreed
            $BCovC[idx = idx + 1] = eT[i]$
        end if
    end for
    //**Fourth step: Quality Measurements of BCovC**
    $kapp = \kappa(BCovC)$
    $return\ BCovC$
end procedure

second annotators agreed on the Covid text, i.e., the i^{th} text of pt, then it is added to the $BCovC$ corpus. When one of the annotators agreed to the Covid text, it was moved to the expert opinion. In the second step, a total of 157,771 texts are taken. Among these, 12,420 texts agreed by both annotators for Covid text, and 140,745 texts disagreed by the two annotators. Only the first annotator annotated 2,175 texts as Covid, whereas the second annotator only annotated 2,431. Thus, 4,606 texts are moved to the expert for label verification.

In the third step, a linguistics expert manually verified the texts for disagreement of annotators. A total of 1,920 texts are selected for addition to the *BCovC* corpus, and 2,686 texts are discarded from this step. In the manual annotation and expert-level verification step, 14,340 texts are included in *BCovC* as the Covid category, and randomly 14,773 texts are included in *BCovC* as the non-Covid category. Finally, both categories have 29,113 texts in the *BCovC* corpus. In the fourth step, the kappa value (κ) of *BCovC* is calculated based on the annotator's agreements and disagreement [31]. The overall kappa value of *BCovC* is 82.75%, which is an acceptable score for the corpus [32].

Table 1 shows the Covid text identification (*BCovC*) corpus statistics. The maximum of 20 words per sentence is in the Covid category, whereas the maximum of 23 words per sentence is in the non-Covid category. The minimum number of words per sentence is 4 in both categories. Though the total number of non-Covid samples is 140,745, we only randomly selected 14,773 texts (e.g., 10.5%) because of overcoming the issues of category-wise text sample imbalance [33].

Category	Attribute	Value
non-Covid	No. of words	2,866,371
	No. of unique words	122,241
	No. of samples	14,773
	No. of training/testing samples	10,331 / 4,442
	No. of sentences	318,485
Covid	No. of words	3,145,097
	No. of unique words	91,191
	No. of samples	14,340
	No. of training/testing samples	9,941 / 4,399
	No. of sentences	262,091

Table 1: *BCovC* Corpus Statistics

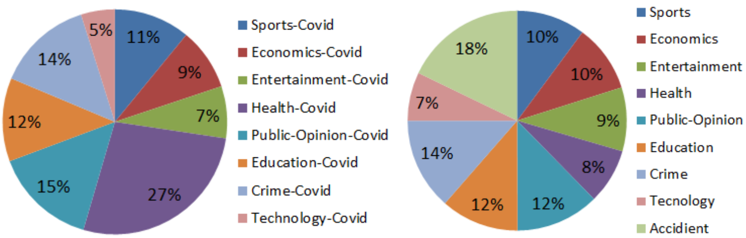
Figure 2 shows the word-cloud visualization of the most frequent 500 words of Covid and non-Covid categories. The Word cloud visualization clearly illustrates that the Covid category contains more Covid-related words, whereas the non-Covid word cloud is not. Thus, the frequent word of Covid categorizes also improved the Covid text identification performance.

Figure 3 shows the Covid and non-Covid class-wise distribution. The Covid text samples are collected from eight different Bengali categories (see Fig. 3a). Maximum 27% texts samples collected subjected to the health-Covid category and a minimum of 5% subjected to the technology-Covid category. The public-opinion-Covid indicates the social media, blogs, newspaper opinion, and public domain text comments subject to Covid.

Figure 3b depicts the non-Covid category-wise text samples. The Non-Covid text samples are annotated from nine different domains (see Fig. 3b). The crime category contained the maximum amount of text samples (14.00%) and a minimum of 7.00% included for technology. The *BCovC* was used for the text identification method evaluation and summarized to compare transformer-based fine-tuning and deep learning-based methods.

(a) Covid Word-cloud (b) Non-Covid Word-cloud

Fig. 2: Word-cloud visualization of most frequent 500 words in Bengali Covid and non-Covid training samples



(a) Covid Distribution (b) Non-Covid Distribution

Fig. 3: Covid and non-Covid domain-wise distribution of *BCovC* corpus

4.2 Covid Embedding Corpus (*CovEC*)

The *CovEC* is an unlabelled corpus developed for single embedding model generation, evaluation and Covid text identification purposes. The *CovEC* is developed based on the previous Bengali embedding corpus (*EC*) [5], and this work developed a training set of *BCovC* (e.g., T^n). Due to the enhancement of performance and training time reduction of embedding models, this research released the Bengali higher-frequency words (e.g., stop words) and the words with one frequency. After removing these words, 1,963,483 words with frequency two are included in the *CovEC* from *EC*. The *EC* data crawling duration is between January 2010 and December 2019. As a result, the Covid-related words have not existed in the past embedding corpus (i.e., EC). For this reason, we added the T^n of *BCovC* to the *CovEC*. The T^n contains 180,824 unique words. All T^n words are included in the *CovEC* corpus. Finally, 2,144,307 unique words are incorporated in the *CovEC*, used to train embedding models. Table 2 shows the key statistics of *CovEC*. This corpus contains approximately 204 million words with more than 10 million unique words.

Corpus	#Words	#Unique Words	Max. Frequency
EC [5]	200,081,093	10,067,699	11,737,198
EC-BCovC (training set)	4,199,410	180,824	47,950
Total (in *CovEC*)	204,280,503	10,248,523	11,785,148

Table 2: Statistics of *CovEC*)

Table 2 indicates that the *Bengali is an inflected language and more frequent words come from conjunction* Bengali is a heavily inflected language with a vast amount of verb and noun inflexions [34]. Thus, more frequent words come from conjunction structures. For example, single conjunction (e.g., Bengali to English translation word: oh) occurred at 5.76% of the total embedding corpus *CovEC*.

4.3 Intrinsic Evaluation Datasets (IEDs)

The intrinsic evaluation datasets (IEDs) refer to the word-level similarity measure datasets (i.e., semantic (s_m) similarity, syntactic (s_y) similarity, relatedness (s_r) similarity), and word analogy task (a_t) dataset. These datasets use to measure the embedding model performance. Recently, a dataset has been developed for intrinsic evaluation [35] of the text processing tasks. However, this dataset was not considered the Covid-related word pairs. In this research, we took 100s semantic, syntactic, relatedness and analogy tasks word pairs from the previous dataset [36]. Additionally, this work collected 50 semantic, syntactic, relatedness, and word analogy pairs according to the philosophy of 'contextual correlates of synonymy' research [37]. The first annotator collected 50 words for semantic, syntactic, and related categories, whereas the second annotator collected 50 words for each category based on the first annotator's word selection. The average of the two annotators' scores is assigned as the final score of each word pair. The annotation quality is calculated using the Spearman and Pearson correlation scores from the individual pair-wise annotators' score [38]. For the analogy task, the first annotator selects 50-word pairs based on semantic, syntactic and relatedness categories, and the second annotator also selects 50-word pairs based on the first annotator's selections. All the newly collected data are merged with the previous datasets. The Spearman and Pearson correlation scores are measured based on the combined dataset. Table 3 shows the overall summary of the developed IEDs Spearman (ρ) and Pearson (δ)

Dataset	Spearman correlation (ρ)	Pearson correlation (δ)
s_m	0.68	0.65
s_y	0.71	0.70
s_r	0.65	0.66
a_t	0.63	0.65

Table 3: Summary of IEDs concerning 150 word-pairs)

The analogy dataset is built from semantic, syntactic and relatedness categories where the absolute score difference is more than 1.8 for most word pairs. This difference occurred due to the annotator's perceptions. The maximum correlation is achieved from the syntactic category, whereas the minimum correlation is obtained from the word analogy task dataset. As a result, the a_t dataset obtained a lower correlation value than others.

5 Methodology

The central goal of this research is to develop an intelligent Covid text identification (CTI) network that can classify a piece of Bengali text into two classes: Covid or non-Covid. The methodology comprises two modules: (i) Leveraging Deep Models for CovTiNet Selection (ii) CovTiNet. Each of the modules is described in the following subsections.

5.1 Leveraging Deep Models for CovTiNet Selection

Figure 4 depicts the schematic framework for the selection procedure of CovTiNet. This study experimented with different frameworks to identify the

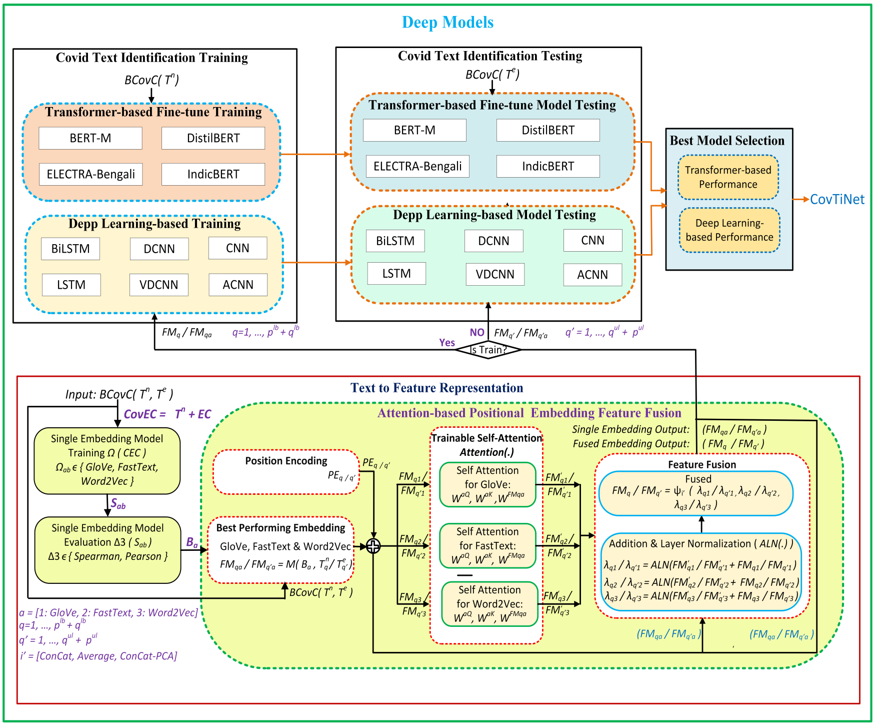


Fig. 4: Leveraging Deep Models for CovTiNet Selection

best-performing feature extraction and Covid text identification framework. The CovTiNet selection framework comprises four main modules: (i) Text to feature representation (e.g., single embedding model training, evaluation and feature fusion), (ii) Covid text identification training, (iii) Covid text identification testing, and (iv) Best model selection (i.e., CovTiNet). The following subsections describe each module in detail.

5.1.1 Text to Feature Representation Module

The function of this module is to take a Covid embedding corpus ($CovEC$) as an input and generates outputs as the attention-based positional embedding feature fusion matrix ($FM_q/FM_{q'}$), where total training and testing samples are $q \in (p^{lb} + q^{lb})$ and $q^{'} \in (p^{ul} + q^{ul})$ respectively. Initially, three embedding methods (e.g., GloVe [39], FastText [40], and Word2Vec [41]) are applied to generate 18 models (i.e., 6 for GloVe, 6 for FastText and 6 for Word2Vec). The best-performed three models are selected for the feature fusion task based on the intrinsic evaluations. Finally, the attention-based positional embedding feature fusion and representation method generate the fused feature matrix used for training and testing CovTiNet. The following paragraphs describe the overall tasks of the fusion-based feature representation module.

Single Embedding Model Training:

In this phase, the single embedding model training function $\Omega(.)$ takes the input of $CovEC$ and outputs a set of embedding models S_{ab}, where $a = \{GloVe, FastText, Word2Vec\}$ and $b = 1, ..., E^n$. Table 4 shows the overall optimized hyperparameters of three single embedding methods. The mini-

Methods	Optimized Hyperparameters
Word2Vec (SG), FastText (SG) and GloVe	ED: {200, 250, 300}, Min.Frequency: 2, Window Size: {12, 13}, Max.Frequency: 75, Epoch: 25, Mgs: 2, learning_rate: 0.01

Table 4: Optimized hyperparameters on GTX 1070 GPU & 32GB physical memory

mum grams (Mgs) are applied to the FastText training phase, and each word is split according to this value. The fastText and Word2Vec have produced the embedding model based on *centre word to context word prediction schemes.* Whereas the GloVe method prepared the embedding model using *word-word co-occurrence and frequency schemes.* In this study, three embedding dimensions (e.g., 200, 250, 300), two context window sizes (e.g., 12 & 13) and three embedding methods (e.g., GloVe, FastText & Word2Vec) accomplished a total of 18 embedding models generated for intrinsic evaluation. Statistical word frequency-based method (e.g., GloVe) and Neural embedding-based methods (e.g., Word2Vec & FastText) are trained with the tuned hyperparameters

(Shown in Table 4). The FastText SG version can carry the sub-word information at the embedding model training phase. As a result, the morphologically rich languages minimize the OOV problems [42]. A total of 18 single embedding models (6 for Word2Vec, 6 for FastText and 6 for GloVe) are generated using the combination of 3 embedding dimensions ED and 2 Window Size). All generated models are used for the intrinsic evaluation.

Single Embedding Model Evaluation:

In this step, the inputs are single embedding models and provide the best-performed embedding models as the output. Three embedding models are generated where only one best-performed model is considered from each method (GloVe, FastText and Word2Vec). The best-performed embedding models are selected based on the intrinsic evaluation in each case. The intrinsic evaluators measure the quality of an embedding model for specific NLP tasks, reduce the downstream task training time, and minimize the OOV issues [36]. Algorithm 3 illustrates the process of intrinsic evaluation.

In Algorithm 3, the function $HumanJudgementScore(.)$ returns pair-wise annotator scores of semantic (H_m), syntatic (H_y), relatedness (H_r), and analogy tasks H_{at} datasets respectively. The a^{th} embedding model (e.g., em) is evaluated based on the four datasets (e.g., s_m, s_y, s_r & a_t). Each of the datasets calculates the cosine similarity score for each word pair. The Spearman correlation ($SprCor(.)$) and Pearson correlation ($PerCor(.)$) functions sequentially take the annotator's judgement scores and cosine similarity scores for each of the datasets, which return the Spearman correlation (ρ) and Pearson correlation (δ). The Spearman correlation score of the semantic, syntactic, relatedness and analogy task is denoted by the ρ_m, ρ_s, ρ_r, and ρ_{at} respectively. Similarly, the Pearson correlation scores are represented by δ_m, δ_s, δ_r, and δ_{at} respectively. The $Pavg(.)$ function takes these six scores and returns the average score value for the combination of the b^{th} embedding model hyperparameters. In these ways, the intrinsic evaluators evaluate all the embedding models and select the best-performing embedding models using the $best(.)$ function. Finally, the $\Delta 3(.)$ function returns the best-performed three embedding models (B_a).

Attention-based Positional Embedding Feature Fusion:

The split corpus $BCovC(T^n, T^e)$ and the best-performed embedding models $B_a \in \{GloVe, FastText, Word2Vec\}$ are used as the inputs and generates the fused feature matrix ($FM_q/FM_{q'}$), where $q \in (p^{lb} + q^{lb})$, $q^{'} \in (p^{ul} + q^{ul})$. Fig. 3 shows the abstract view of the attention-based positional embedding feature fusion method. Initially, the training sample (T^n) sequentially extracts the feature using the mapping function $M(.)$ and a^{th} embedding model. The a^{th} embedding model feature matrix for q^{th} training sample is represented by $FM_{qa} \in \mathbb{R}^{sl \times ED}$, where $sl \in 256$ denotes the maximum sequence length and $ED \in 300$ indicates the optimal embedding dimension.

Algorithm 3 Intrinsic Evaluation for Word Embedding Models

procedure $\Delta 3(S)$
  $s_m := X_P$ *Semantic words pairs*
  $s_y := X_P$ *Syntactic words pairs*
  $s_r := X_P$ *Relatedness words pairs*
  $a_t := X_a$ *Analogy tasks*
  $H_m, H_y, H_r, H_{at} := HumanJudgementScore(s_m, s_y, s_r, a_t)$
  $I_r = []$
  for ab in $range(1, len(S))$ **do**
    $C_m := [],\ C_y := [],\ C_r := [],\ C_{at} = []$
    $em = S[ab]$
    for i *in* $range(1, X_P - 1)$ **do**
      $C_m.append(CosineSimilarity(s_m[i], s_m[i+1], em))$
      $C_y.append(CosineSimilarity(s_y[i], s_y[i+1], em))$
      $C_r.append(CosineSimilarity(s_r[i], s_r[i+1], em))$
    end for
    for i in $range(1, X_a)$ **do**
      $temp = a_t[i]$
      $C_{at}.append(CosineSimilarity(temp[3], temp[4], em))$
    end for
    $\{\rho_m, \rho_s, \rho_r, \rho_{at}\} := SprCor(H_m, C_m, H_y, C_y, H_r, C_r, H_{at}, C_{at})$
    $\{\delta_m, \delta_s, \delta_r, \delta_{at}\} := PerCor(H_m, C_m, H_y, C_y, H_r, C_r, H_{at}, C_{at})$
    $l_b := Pavg(\rho_m, \rho_s, \rho_r, \rho_{at}, \delta_m, \delta_s, \delta_r, \delta_{at})$
    $I_r.append(l_b)$
  end for
  $B_a := best(I_r), a = [GloVe, FastText, Word2Vec]$
  return B_a
end procedure

The q^{th} training sample sequentially produces three feature matrices, e.g., FM_{q1} for $GloVe$, FM_{q2} for $FastText$ and FM_{q3} for $Word2Vec$. Whereas $q^{'th}$ testing sample produces three feature matrices denoted by $FM_{q'1}$ for $GloVe$, $FM_{q'2}$ for $FastText$ and $FM_{q'3}$ for $Word2Vec$. The word position is crucial information for the context-aware word-level semantic, and syntactic feature representation [43]. The position-based information is added before applying the self-attention operation. The $q/q^{'}$ sample sinusoidal position encoding operation is conducted using Eq.11.

$$PE_{q/q'}[1:sl] = \begin{cases} sin(\frac{pos}{10000^{(2\times pos/ED)}}),\ if\ (pos\%2)==0\ pos=1,\ldots,sl \\ cos\frac{pos}{10000^{(2\times pos+1/ED)}},\ otherwise\ pos=1,\ldots,sl \end{cases} \tag{11}$$

Here, ED denotes the embedding dimension and the word position of $q/q^{'}$ sample is $pos \in sl$. This position encoding is just added to the training/testing sample $(q/q^{'})$ and fed to the trainable self-attention block, i.e., $Attention(.)$.

The self-attention block contains nine trainable weight matrices, e.g., three for GloVe, three for FastText and three for Word2Vec. The generalized form of matrices are query ($W^{aQ} \in \mathbb{R}^{sl \times ED}$), keys ($W^{aK} \in \mathbb{R}^{sl \times ED}$) and values ($W^{FMqa} \in \mathbb{R}^{sl \times ED}$) for GloVe, FastText and Word2Vec respectively. However, the attention of q^{th} sample is calculated using Eq. 12.

$$FM_{qa}^{'} = (\frac{(FM_{qa} \times W^{aQ}) \times (FM_{qa} \times W^{aK})^T}{\sqrt{ED}}) \times (FM_{qa} \times W^{FMqa}) \quad (12)$$

Here, FM_{qa} indicates the input fused feature matrix whereas $FM_{qa}^{'}$ represent the attention-based output fused feature matrix of a^{th} best performing embedding model (i.e., $a = [1 : GloVe, 2 : FastText, 3 : Word2Vec]$). The addition and layer normalization block is combined with an attention-based and positional encoding feature, which improves the word-level correlation [43]. The layer-based normalized feature is forwarded to the feature fusion block and fuses the feature value using Eq. 7. In the training phase, the fused feature is denoted by FM_q and used in the attention-based CNN training module, whereas the testing time fused feature is represented by $FM_{q'}$ and will be used for the attention-based CNN model evaluation purpose.

5.1.2 Covid Text Identification Training

Investigate the performance of the Covid text identification task, this research investigates the performance of six deep learning-based (i.e., BiLSTM, DCNN, CNN, LSTM, VDCNN & ACNN) and four transformer-based (i.e., BERT-M, DistilBERT, ELECTRA-Bengali & IndicBERT) methods. The following paragraphs describe the training process of deep learning and transformer-based methods.

Deep Learning-based Training:

The deep learning-based methods are trained with the best performing three single embedding feature matrix $FM_{qa} \in \mathbb{R}^{sl \times ED}$ and attention-based position embedding feature fusion matrix $FM_q \in \mathbb{R}^{sl \times ED}$. Where $a \in \{GloVe, FastText \& Word2Vec\}$ and q denotes the total number of training samples, these six methods are used the tuned hyperparameters, which shows in Table 5 and produce the 36 Covid text identification models using Eq. 8 (e.g., 36: (3 single embeddings $\times$ six deep learning methods) + (3 fused embedding $\times$ six deep learning methods)). The LSTM, BiLSTM, CNN, ACNN, DCNN and VDCNN methods have tuned the hyperparameters based on *CovC* corpus and GTX 1070 single GPU[44].

Transformer-based Fine-tune Training:

The transformer-based fine-tune training module takes the training samples of *BCovC* and prepares the input feature matrix using the three multi-lingual (e.g., BERT-M, DistilBERT-M & IndicBERT) and one monolingual (e.g., ELECTRA-Bengali) pre-trained language model. Each of the input samples is encoded as a 2D input feature matrix (i.e., $2D \in \mathbb{R}^{300\times768}$) and sequentially feeds to the transformer-based fine-tune training function (i.e., $\Psi^{tr}(.)$). This function used the four tuned hyperparameters (e.g., sl, batch size, epoch & learning_rate), shown in Table 5, and the remaining hyperparameters are used as the default values. Four Covid text identification models are generated from the four transformer methods. These models are used in the Covid text testing phase.

Baseline Methods	Hyperparameters
LSTM	layer: 2, sl: 300, hidden-dim: 128, 64, batch size: 32, dropout: 0.45, 0.50, loss: categorical_crossentropy, optimizer: adam, epoch: 30.
BiLSTM	layer: 2, sl: 300, hidden-dim: 128, 64, batch size: 16, dropout: 0.30, 0.40, loss: categorical_crossentropy, optimizer: adam, epoch: 40.
DCNN	layer: 6, sl: 300, epoch: 100, learning_rate: 0.10, dropout: 0.50 activation: ReLU & softmax
CNN	CNN layer: 1, No. kernel: 3, kernel size:177, sl: 300, activation: ReLU & softmax, batch size: 64,epoch: 80, learning_rate: 0.01, dropout: 0.56, pooling:max & avg.
ACNN	CNN layer: 1, Attention layer: 2, No. kernel: 3, kernel size:177, sl: 300, activation: ReLU & softmax, batch size: 64,epoch: 80, learning_rate: 0.01, dropout: 0.56, pooling:max & avg.
VDCNN	layer: 15, Max.-len: 300, activation: ReLU & softmax, batch size: 64,epoch: 100, learning_rate: 0.01, dropout: 0.56, pooling:max & avg.
BERT-M, DistilBERT-M, ELECTRA-Bengali & IndicBERT	sl: 300, batch size: 6, epoch: 10, learning_rate: 2e-4

Table 5: Hyperparameters of deep learning & Transformer-based fine-tune methods

Due to GPU memory limitation, this research fine-tuned only a smaller number of hyperparameters for transformer models (shown in Table 5), and other parameters are used as default. The maximum batch size and sequence length are 6 and 300, respectively.

5.1.3 Covid Text Identification Testing

The CTI test phase is evaluated the different deep learning and transformer-based model performances for the unknown CTI dataset (i.e., T^e). The following paragraphs summarize the deep learning and transformer-based CTI model evaluation details.

Deep Learning-based Testing

In this phase, 36 CTI models (e.g., 36: (3 single embedding $\times$ six deep learning methods) + (3 fused embedding $\times$ six deep learning methods)) are evaluated with the test set T^e. Each of the test sample $q^{'} \in T^e$ is mapped with the best performing embedding B_a using mapping function $M(.)$ and produces two feature matrix $FM_{q'}$ and $FM_{q'a}$. The $(FM_{q'}$ & $FM_{q'a}) \in \mathbb{R}^{sl \times ED}$, ED denotes the embedding dimension and sl denotes the maximum sequence length. Now, the $k^{'}th$ deep learning method is initialized with the pre-trained CTI model wight $\Theta_{k'}$ and produces the expected output $O_{k'}$ using Eq. 9. The softmax operation normalizes the output, and the maximum softmax value index indicates the corresponding category.

Transformer-based Fine-tune Testing

The four transformer-based models' performance is verified by the $BCovC$ test set (e.g., T^e). Each test sample is produced as a 2D feature matrix (i.e., 300) and is predicted by the fine-tuned model. The fine-tuned model has generated an expected category value. The softmax operation normalizes this expected value; the maximum value index is indicated in the corresponding category.

5.1.4 Best Model Selection

This section aims to select the best performing Covid text identification model from four transformer-based and thirty-six deep learning-based models. Each classifier is trained with the training set $T^n \in BCovC$, and accuracy is measured by the test set $T^e \in BCovC$. Among the 40 model evaluation results, the maximum accuracy model is selected for the Covid text identification system (named CovTiNet). The following subsections describe the details of CovTiNet.

5.2 CovTiNet

The Proposed Covid text identification system (i.e., CovTiNet) has been built up with two significant modules, i.e., the attention-based position embedding averaging of GloVe and FastText (**APeAGF**) for text feature representation module and attention-based convolutional neural networks (**ACNN**) for Covid text identification module. Fig. 5 shows the abstract view of the proposed CovTiNet. The following subsection describes the details of the two modules.

5.2.1 APeAGF

In Figure 5, the attention-based position embedding averaging of GloVe and FastText (APeAGF) module takes input as training and testing set, i.e., $(T^n/T^e) \in BCovC$ and output is the feature matrix (e.g., $FM_q/FM_{q'}$). The $q^{th} \in T^n$ training and $q^{'th} \in T^e$ testing sample is sequentially represented the features matrix FM_{q1} and $FM_{q'1}$ for GloVe embedding, whereas FM_{q2} and $FM_{q'2}$ for FastText embedding using Eq. 11. In addition to better syntactic feature representation, position encoding (PE) is added to these feature matrices. The function of $Attention(.)$ calculates the attention value of each word in the feature matrix and improves the contextual representation of training/testing samples (i.e., $q/q^{'}$) using Eq. 12. The attention value normalization

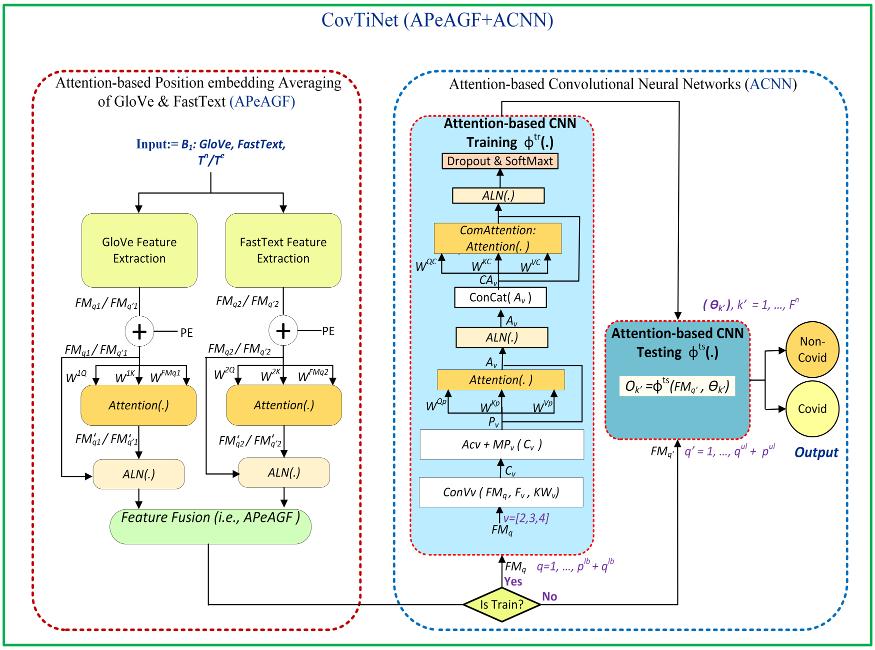


Fig. 5: High-level View of CovTiNet

functions $ALN(.)$ take the attention-based feature matrix and original feature matrix (i.e., take from skip-connection). The $ALN(.)$ function normalized the attention value and forwarded it to the feature fusion module. The feature fusion module is just averaging the attention-based feature matrix of GloVe and FastText. Finally, the APeAGF module output FM_q for q^{th} training sample attention-based feature matrix and $FM_{q'^{th}}$ testing sample attention-based feature matrix. The FM_q will be used for training purposes, and $FM_{q'}$ will be used for testing purposes.

5.2.2 ACNN

The Attention-based Convolutional Neural Networks (ACNN) module works in two steps. The ACNN module training with the training set T^n generates a Covid text identification model in the first step. In the second step, the Covid text identification model is evaluated by the testing set T^e and calculates the performance of the ACNN module. The following paragraphs describe the details of the two steps.

Attention-based CNN Training:

The training function $\Psi(.)^{tr}$ takes the training samples fused feature matrices $FM_q : FM_m \in \mathbb{R}^{sl \times ED}$ and outputs a Covid text identification model $\Theta_{k'}$. Initially, a convolution operation is applied to the single CNN layer with three different kernel sizes (i.e., $v = [2,3,4]$). The v^{th} kernel conducted the convolution operation ($ConV_v$) using Eq. 13.

$$C_v[1:F_v] = KW_v[v:ED] \otimes FM_q + bias_v, v = [2,3,4];\ q = 1,...,p^{lb} + q^{lb} \quad (13)$$

Here, v^{th} trainable kernel indicates KW_v and the convolution output is represented by C_v. The three kernels' convolution output is stored in C_v and is forwarded to the second layer. The second layer applies kernel-wise activation and max-pooling operations. The v^{th} kernel activation and max-pooling operation is conducted by Eq. 14.

$$P_v[1:len(F_v)] = MaxPooling(ReLU(C_v[1:F_1])), v = [2,3,4] \quad (14)$$

The $ReLU(.)$ activation function ($AC_v(.)$) normalized the sentence-level convoluted features and the max-pooling function $MP_v(.)$ returns a single maximum value from the trainable convolution output (i.e., C_v). The output of the max-pooling operation is stored in P_v and forwarded to the third layer (i.e., $Attention(.)$). The attention layer calculated the sentence-level attention using Eq. 12 and concatenated the three kernels (e.g., $v = [2,3,4]$) attention-based feature (A_v). This concatenated feature is passed to another attention-based encoding layer ($ComAttention$), and the dropout operation is applied. The dropout operation randomly blocks some neuron values, which helps overcome the overfitting issues. Finally, the dropout features are forwarded to the softmax layer, predicting the Covid identification score. The error value is calculated from the predicted and ground-truth value and adjusts the error using the backpropagation operation. At the end of the training, the attention-based CNN saves a Covid text identification model ($\Theta_{k'}$), which is used in the next phase (i.e., the attention-based CNN testing phase).

Attention-based CNN Testing:

The attention-based CNN test function calculates the model's ability to perform the task ($\Psi^{ts}(.)$). The function takes the Covid text identification model ($\Theta_{k'}$) and sequentially predicts the test set samples (T^e). The $q^{'}$ test sample fused feature matrix $FM_{q'} : FM_{q'} \in \mathbb{R}^{sl \times ED}$. The fused feature matrix is fed to the pre-trained mode ($\Theta_{k'}$) and calculates the expected value using Eq. 15.

$$E_{k'}[q^{'}] = \Theta_{k'} \times FM_{q'}, q^{'} = 1,...,p^{ul} + q^{ul} \quad (15)$$

The $E_{k'}[q^{'}]$ denotes the expected value of $q^{'th}$ test sample (i.e., T^e). Now, the expected value is normalized by Eq. 16.

$$O_{k'}[q^{'}] = max(\frac{e^{(E_{k'}[q^{'}])}}{\sum_{z=1}^{z=p^{ul}+q^{ul}} e^{(E_{k'}[z])}}), q^{'} = 1,...,p^{ul} + q^{ul} \quad (16)$$

Here, $O_{k'}[q']$ indicates normalized expected value of q' test sample. All of the statistical measures will use this outcome of $(O_{k'}[q'])$ to evaluate the performance of the model.

6 Experiments

The CovTiNet framework is implemented using the Pytorch: 1.9.0, Pandas, and Sklearn libraries, Python3 (version: 3.6), Numpy, Transformer (version:4.9.0) and Tensor-flow (version: 2.0). The Hardware configurations are a multi-core processor (core-i7) with NVIDIA GTX 1070 GPU (Internal GPU memory 8GB) and 32GB physical memory. The following subsection describes the intrinsic (i.e., embedding) and extrinsic (i.e., Covid text identification) evaluation of the models.

6.1 Intrinsic Evaluators

The intrinsic evaluators evaluate each word embedding model's word-level semantic, syntactic, relatedness or analogy tasks performance. This evaluation helps to decide the best-suited embedding model for the downstream task (CTI) that requires a minimum time and memory usage (based on Algorithm 3). The semantic (C_{S_m}), syntactic (C_{S_y}) and relatedness (C_{S_r}) similarity measure is calculated using Eq. 17.

$$C_{cs}(A_w, B_w) = \frac{\overrightarrow{A_w} \cdot \overrightarrow{B_w}}{\overrightarrow{A_w} \times \overrightarrow{B_w}}, cs = [S_m, S_y, S_r] \tag{17}$$

Here, A_w and B_w denote the semantic, syntactic or relatedness first and second word of the intrinsic datasets, respectively. The feature vector of word A_w and B_w represented by $\overrightarrow{A_w}$ and $\overrightarrow{B_w}$ respectively. C_{cs} presents the Cosine similarity score of $cs \in \{S_m, S_y, S_r\}$. The average Cosine similarity score of semantic, syntactic and relatedness datasets are calculated using Cosine similarity score C_{cs}, which are represented by $\overline{C}_{S_m}$, $\overline{C}_{S_y}$ and $\overline{C}_{S_r}$ respectively. In this study, we also measure the Spearman (ρ), and Pearson (δ) correlations [45] using the Cosine similarity and human judgement scores.

The word analogy also measures the embedding model performance using the pair-wise word alikeness, such as: if word A_w is to be word B_w and word C_w is to be word D_w then pair (A_w:B_w) is alike (C_w:D_w). The word alikeness problem is solved by the 3COSADD [46], and 3COSMULL arithmetic formulations [47]. For this purpose, given this (A_w:B_w :: C_w : $-$) then find the best match word for the blank $-$ (i.e., D_w) such that (A_w : B_w) is alike (C_w : D_w). To solve this problem, the 3COSADD finds the best matching word D_w using Eq. 18.

$$D_w = \max_{D_w \in V}(C_{cs}(D_w, C_w) - C_{cs}(D_w, A_w) + C_{cs}(D_w, B_w)), cs = [a_t] \tag{18}$$

Here V is the total number of vocabularies in the embedding model. Another variation of this solution is 3COSMULL to find the best-matching word D_w using Eq. 19.

$$D_w = \max_{D_w \in V} \frac{C_{cs}(D_w, C_w) \times C_{cs}(D_w, B_w)}{C_{cs}(D_w, A_a) + \epsilon}, cs = [a_t] \tag{19}$$

Here, ϵ is a small (i.e., 0.000001) value used for overcoming the division by zero. For calculating the arithmetic correlation of D_w with other three words, Eq. 18 or 19 is used, whereas Eq. 17 is used to compute Cosine similarity. The word analogy task performance is calculated by the ratio of $\frac{Acc}{len(a_t)}$, where Acc indicates the total number of deserted words D_w found and $len(a_t)$ represents the length of the analogy task.

6.2 Extrinsic Evaluators

The extrinsic evaluators assess the CTI task performance of the models. The accuracy and error of the proposed CovTiNet is estimated by several statistical metrics such as accuracy (A_c), precision (P_c), recall (R_c), micro f1 score (F_1), macro average (M_a), weighted average (W_a), and confusion matrix.

6.2.1 Ablation Analysis

An ablation analysis is carried out for selecting features extraction method and text identification method from a set of methods [48]. For this analysis, three best-performed single embeddings (i.e., GloVe, FastText, and Word2Vec) and three best-performed attention-based feature fusion embeddings (i.e., AeCGF, AeCPGF and AeAGF) are evaluated for feature extraction methods. In contrast, ten text identification methods (i.e., CNN, ACNN, VDCNN, CNN, LSTM, BiLSTM, BERT-M, DistilBERT, ELECTRA-Bengali, and IndicBERT) are evaluated for Covid text identification system. The final CovTiNeT system comprises the best-performing feature extraction and text identification methods.

7 Results

The developed CovTiNet is evaluated in two ways: feature extraction performance evaluation (i.e., intrinsic version) and CTI performance evaluation (i.e., extrinsic version).

7.1 Intrinsic Evaluation

The intrinsic evaluation is carried out on a word-level semantic/syntactic performance. Therefore, the position encoding value can not be used in attention calculation. Only the attention and fusion operations are employed to represent word semantics. Table 6 shows the performance of Spearman (ρ), Pearson (δ) and Cosine similarity of semantic (S_m), syntactic (S_y) and relatedness (S_r) datasets. The embedding parameter identification (EPI) Algorithm selects three embedding dimensions (EDs) (e.g., $ED \in \{200, 250, 300\}$) and two contextual windows (e.g., 12 and 13) for GloVe, FastText and Word2Vec methods. These three methods yield 18 single embedding models using *CovEC* corpus. The best-performed embedding models are used to generate the attention-based feature fusion model using Concatenation ($ConCat$), Averaging (Average), and Concatenation with principal component analysis ($ConCat - PCA$) methods [49]. The $ConCat$ method produced four fused embedding feature matrices (e.g., GloVe+FastText, GloVe+Word2Vec, FastText+Word2Vec, GloVe+FastText+Word2Vec). The other two methods also generated eight fused embedding feature matrices. Among these 18 single and 12 fused embedding models, top-performed three single (e.g., one from GloVe, one from FastText and one from Word2Vec) embedding and three fused embedding (e.g., AeCGF: Attention-based embedding with ConCat (GloVe, FastText), AeAGF: Attention-based embedding with Averaging (GloVe, FastText), AeCPGF: Attention-based embedding with ConCat-PCA (GloVe, FastText)) models are selected for the downstream task (i.e., CTI). Table 6 illustrates the summary of the best-performed single and fusion-based embedding models.

Models	*Semantic* S_m(%)			*Syntactic* S_y(%)			*Relatedness* S_r(%)		
	ρ_m	δ_m	$\overrightarrow{C}_{S_m}$	ρ_y	δ_y	$\overrightarrow{C}_{S_y}$	ρ_r	δ_r	$\overrightarrow{C}_{S_r}$
GloVe	65.97	67.10	79.13	70.93	76.33	80.41	81.67	81.89	88.10
Word2Vec	49.74	52.07	56.92	51.50	54.29	60.80	60.11	63.19	66.28
FastText	56.29	63.48	67.03	66.11	67.16	67.20	68.84	72.59	74.31
AeAGF	**68.20**	**69.10**	**81.78**	**73.68**	**79.27**	**82.41**	83.01	**84.70**	88.59
AeCGF	65.83	67.04	78.90	72.93	77.46	81.18	82.21	83.57	87.02
AeCPGF	66.70	67.96	79.02	73.05	77.53	80.11	**83.79**	83.72	**88.52**

Table 6: Intrinsic performance of the best-performed embedding models

The maximum Spearman (ρ_m), Pearson (δ_m), and average cosine similarity ($\overrightarrow{C}_{S_m}$) of semantic dataset are 68.20%, 69.10% and 81.78% respectively achieved by AeAGF. Similarly, the syntactic dataset obtained the maximum accuracy of 73.68%, 79.27% and 82.27% by AeAGF. In contrast, the relatedness dataset obtained the maximum value for Spearman (ρ_r) and Pearson (δ_r) from AeCPGF. Overall, Pearson (δ_y) performance has an improvement of 2.94% for the syntactic dataset using the attention-based feature fusion embedding model compared to the single embedding (i.e., GloVe, FastText & Word2Vec). The attention operation improves the word-word correlations, whereas the feature fusion operation combines the unique features of semantic, syntactic and relatedness from the single embedding. Thus, it is confirmed that attention-based feature fusion is better than single embedding for extracting textual features.

Table 7 shows the performance of analogy tasks for single and attention-based feature fusion embedding models. In most cases, the intrinsic evaluation

Models	Semantic a_t(%)		Syntactic a_t(%)		Relatedness a_t(%)	
	Add	Mull	Add	Mull	Add	Mull
GloVe	46	52	50	60	64	66
FastText	42	44	42	50	60	64
Word2Vec	38	42	40	48	60	62
AeAGF	**50**	**56**	**54**	**64**	62	**68**
AeCGF	48	54	52	60	62	66
AeCPGF	**50**	50	52	62	**66**	**68**

Table 7: Performance of the best-performed embedding models for analogy tasks regarding 50 semantic, 50 syntactic & 50 relatedness word-pairs.

revealed that the attention-based average feature fusion (AeAGF) with GloVe+FastText achieved the highest performance for semantic, syntactic, and relatedness datasets. The attention-based fused feature combines the morphologically significant variations of the Bengali language. The maximum semantic (50% & 56%) and syntactic (54% & 64%) analogy accuracies have been achieved from AeAGF feature fusion, whereas the relatedness dataset obtained a maximum accuracy of 68% for AeCPGF and AeAGF feature fusions.

Among 30 embedding models (18 for single and 12 for fusion), the intrinsic evaluators select the best three models to perform the downstream task (CTI). Thus, instead of sending all 30 models for training, the system can use only the best models, reducing the downstream task's time and storage complexity. Due to intrinsic evaluation, 90% (i.e., only the top three models can be used instead of 30 for CTI task evaluation) of training time was saved to perform CTI tasks. For better clarification, we investigate one best single embedding model (GloVe) and two fused embedding models (e.g., AeAGF and AeCPGF) for CTI tasks. The following section describes the performance of the various models for CTI tasks.

7.2 Extrinsic Evaluation

The six deep learning baseline methods, the proposed CovTiNet method, and the four transformer-based fine-tuning methods produced 40 models (where deep learning + CovTiNet contributed 36 models and the transformer-based technique contained four models). Among these 40 models, Table 8 shows the performance of 17 models (six best-performed models, six worst-performed, and four transformer-based fine-tuned models), including the proposed CovTiNet for CTI tasks. The extrinsic evaluation reported the CTI task performance based on the learning ability and intelligence of the model.

Results revealed that the proposed model (CovTiNet) achieved the maximum accuracy of $96.61 \pm 0.001\%$, whereas GloVe+LibSVM achieved the minimum accuracy ($82.26 \pm 0.001\%$). The proposed attention-based fusion and position encoding improved the accuracy of $14.35 \pm 0.001\%$ compared to GloVe+LibSVM, $5.72 \pm 0.001\%$ from GloVe+LSTM and $4.92 \pm 0.001\%$ from CNN. There are two critical reasons for improving the proposed CovTiNet performance compared to other models: (i) the proposed position encoding

Models	A_c(%)	M_a (%)			W_a (%)		
		P_c	R_c	F_1	P_c	R_c	F_1
GloVe+LibSVM	82.26±.001	82	82	82	82	82	82
GloVe+LSTM	90.89±.001	91	91	91	91	91	91
GloVe+BiLSTM	92.54±.001	93	93	93	93	93	93
GloVe+VDCNN	93.17±.001	93	93	93	93	93	93
GloVe+DCNN	92.32±.001	92	92	92	92	92	92
GloVe+CNN	91.69±.001	92	92	92	92	92	92
APeAGF+LibSVM	84.75±.001	85	85	85	85	85	85
APeAGF+LSTM	92.64±.001	93	93	93	93	93	93
APeAGF+BiLSTM	95.14±.001	95	95	95	95	95	95
APeAGF+VDCNN	93.65±.001	92	91	92	94	94	94
APeAGF+DCNN	92.97±.001	93	93	93	93	93	93
APeAGF+CNN	94.13±.001	94	94	94	94	94	94
BERT-M	95.88±.001	96	96	96	96	96	96
DistilBERT-M	95.01±.001	95	95	95	95	95	95
IndicBERT	93.13±.001	93	93	93	93	93	93
ELECTRA-Bengali	96.19±.001	96	96	96	96	96	96
CovTiNet (Proposed)	**96.61±.001**	**97**	**97**	**97**	**97**	**97**	**97**

Table 8: CTI task performance of the proposed (CovTiNet) and baseline models. The M_a and W_a values are round up to two decimal point

extracts the word-level syntactic information, and (ii) the attention-based fusion enhances the quality of the semantic features representation. Thus, the combined attention and position encoding improve linguistic understanding concerning Bengali. In contrast, the statistical classifier (e.g., LibSVM), the sequential classifier (e.g., LSTM), and the Convolutional classifier (e.g., CNN) with non-contextual embedding (e.g., GloVe, FastText and Word2Vec) can not adequately represent the Bengali textual features based on semantic and syntactic meaning.

7.3 Comparison with Previous Research

According to this work exploration, no significant research has been done to identify or classify Covid text in Bengali, including corpus development. Thus, this study embraced several contemporary methods that have been examined on similar tasks in other language datasets. For consistency, a few past techniques [5, 50–55] have been implemented on the developed dataset (i.e., *BCovC*) and compared their performance with the proposed approach (CovTiNet). Table 9 shows the comparison among various techniques in terms of accuracy (A_c), training time in hours (TTH) and GPU memory consumption in GB (GMCG) to perform CTI tasks.

The transformer-based fine-tuned models (BERT-M, IndicBERT and DistilBERT) consumed too much GPU memory and training time compared to CovTiNet. However, their accuracy is significantly lower than the CovTiNet. Because of the smaller vocabularies in the language model and significant morphological variation (semantic and syntactic) of the Bengali language, the transformer-based model showed inferior performance. The ELECTRA-Bengali is a monolingual language model whose accuracy (96.19%) is much

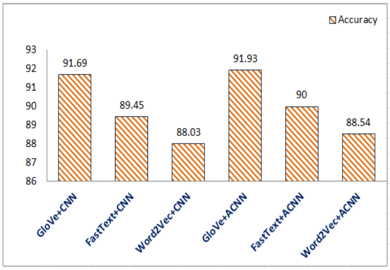
Methods	A_c(%)	TTH	GMCG
BiLSTM+FastText [50]	91.47	0.53	6.5
CNN+FastText [51]	89.45	**0.43**	**3.8**
VDCNN+Word2Vec [52]	90.68	0.62	5.6
ELECTRA-Bengali [53]	96.19	0.68	6.2
BERT-M [5]	95.88	3.03	7.9
DistilBERT [54]	94.88	0.70	6.01
IndicBERT [55]	93.13	2.33	7.6
CovTiNet (Proposed method)	**96.61**	0.51	4.5

Table 9: Comparison between the proposed and recent techniques in terms of A_c, TTH and $GMCG$ on $BCovC$

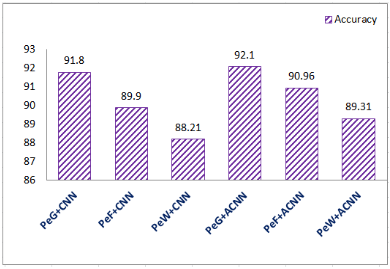
better than the BERT-M (95.88%), IndicBERT (93.13%), and DistilBERT (94.88%) due to monolingual effect due to the single language model gained much attention for semantic and syntactic representations than multilingual models [56].

7.4 Impact of Attention-based Positional Embedding Feature Fusion on CTI Task

This section demonstrates how the CovTiNet gained better performance than other models due to incorporating attention-based positional embedding feature fusion and attention operation on the CNN method. Fig. 6a illustrates the impact of attention-based CNN (ACNN) embedding on the single embedding models (e.g., GloVe, FastText and Word2Vec).



(a) B_a+(CNN,ACNN)



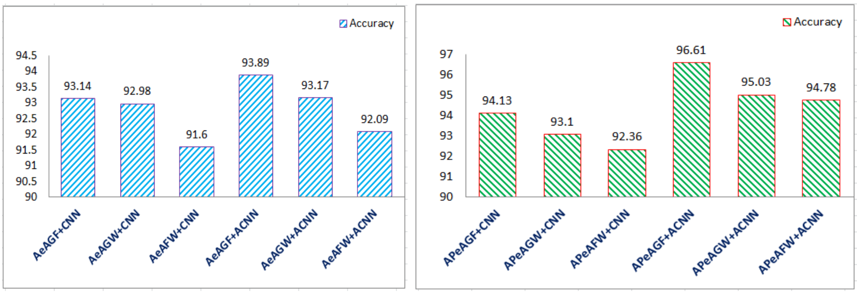
(b) (B_a, P_e)+(CNN,ACNN)

Fig. 6: Impact of position encoding (P_e) on embedding models for CTI task performance with CNN and ACNN

Due to attention operation on the CNN method, the document-level semantic and syntactic feature extraction has an accuracy improvement of about 0.55% by FastText+CNN (from 89.45% to 90.00%). Fig. 6b depicts the impact of position encoding operation with the three single embedding models: position encoding with GloVe (PeG), position encoding with FastText (PeF) and position encoding with Word2Vec (PeW). Figure 6b illustrated that the combination of position encoding on embedding models and attention operation on CNN achieved a notable performance improvement in the CTI tasks. The position encoding and attention operation have improved by about 0.17% accuracy of GloVe (i.e., 91.93% for GloVe+ACNN, 92.1% for PeG+ACNN), 0.96% accuracy improvement of FastText (i.e., 90.00% for FastText+ACNN, 90.96% for PeF+ACNN) and 0.77% improvement achieved for Word2Vec embedding (i.e., 88.54% for Word2Vec+ACNN, 89.31% for PeW + ACNN). Fig. 6 depicts the overall performance of ACNN and the position encoding with embedding models, which are better than CNN with single embedding models.

The intrinsic evaluation results (in Sec. 7.1) showed enhanced performance on CTI tasks due to the *attention-based average feature fusion*. Therefore, we analyzed the impact of attention-based average feature fusion and position encoding operation on CNN and ACNN on CTI (Fig. 7a). In particular, we investigate three operations: (i) attention-based average feature fusion of GloVe+FastText (AeAGF), (ii) attention-based average feature fusion of GloVe+Word2Vec (AeAGW) and (iii) attention-based average feature fusion of FastText+Word2Vec (AeAFW).

It is revealed that the attention-based average feature fusion (AeAGF+ACNN) has enhanced the maximum accuracy of 0.75% compared to AeAGF+CNN (Fig. 7a). Fig. 7b shows the attention-based position encoding average feature fusion GloVe+FastText (APeAGF) and Attention operation on CNN (ACNN). The CovTiNet system achieved the best accuracy of 96.61%. Regarding attention operation on CNN, the maximum accuracy of 2.42% is improved compared to APeAGF+CNN (94.13%). Thus, it is confirmed that the attention-based position encoding average feature fusion and attention operation on CNN has a significant performance improvement in performing CTI tasks in Bengali.



(a) attention-based average feature fusions (AeAGF, AeAGW, AeAFW)

(b) positional embedding average feature fusions (APeAGF, APeAGW, APeAFW)

Fig. 7: Impact of attention-based and positional embedding-based average feature fusions on CTI task performance with CNN and ACNN

Figs. 6 and 7 showed that the attention-based position encoding feature fusion is better than the single embeddings. The attention operation on CNN has significantly improved the semantic and syntactic features representation at sentence and paragraph levels, whereas the position encoding operation improved the contextual features representation. Therefore, the combination of attention, feature fusion and position encoding showed the enhanced CTI task performance by CovTiNet.

7.5 Ablation Evaluation

In the text-to-feature extraction module, the three best-performed non-contextual embedding methods, i.e. Word2Vec, GloVe, and FastText, as well as the three best-performed attention-based feature fusion embeddings (i.e., AeCGF, AeCPGF, and AEAGF) are used for Bengali text-to-feature extraction purposes. However, the word-level performance analysis (i.e., intrinsic evaluators) is summarized in Table 6 and Table 7. These results drastically drop the single embedding performance compared to the attention-based feature fusion performance. For example, the best performing attention-based averaging of GloVe and FastText-based features fusion (i.e., AeAGF) improved the Spearman correlation of 11.91%, 18.46%, and 2.23% for single embedding FastText, Word2Vec and GloVe respectively for Semantic similarity dataset (i.e., S_m). Similarly, the syntactic, relatedness and analogy task dataset performs better using AeAGF embedding than other embeddings. From this ablation analysis, the text-to-features extraction module removed the single embedding methods (i.e., GloVe, FastText & Word2Vec) and removed the other two attention-based feature fusion embeddings (i.e., AeCGF and AeCPGF). The position-based information significantly impacts text identification performance, as depicted in Figure 6. This study included the position information with AeAGF and named an attention-based position embedding averaging of GloVe and FastText (APeAGF). Finally, the APeAGF is selected for the part of the CovTiNet module (Figure 5).

In the Covid text identification module, the ablation analysis initially considers six deep learning methods (i.e., CNN, VDCNN, DCNN, ACNN, LSTM and BiLSTM) and four transformer-based language model fine-tuning methods (i.e., BERT-M, DistilBERT-M, ELECTRA-Bengali and IndicBERT). Among these ten methods, the ablation analysis carried the attention-based CNN (i.e., ACNN) achieved a better performance in terms of accuracy in the Bengali Covid text corpus (i.e., BCovC). The ten text identification methods' performance is summarized in Table 8, where Covid text identification performance is evaluated using the different combinations of single embeddings and attention-based feature fusion embeddings with ten text identification methods. So, the ablation analysis concludes the proposed CovTiNet, i.e., a combination of attention-based position embedding averaging of GloVe and FastText (APeAGF) and attention-based CNN (ACNN) achieved the best performance in BCovC text identification corpus and word level intrinsic evaluation dataset (i.e., IEDs)

7.6 Error Analysis

The error analysis provides in-depth insights into the proposed model's performance regarding qualitative and quantitative strengths and weaknesses. Fig. 8 shows a quantitative analysis of the CovTiNet system using the confusion matrix.

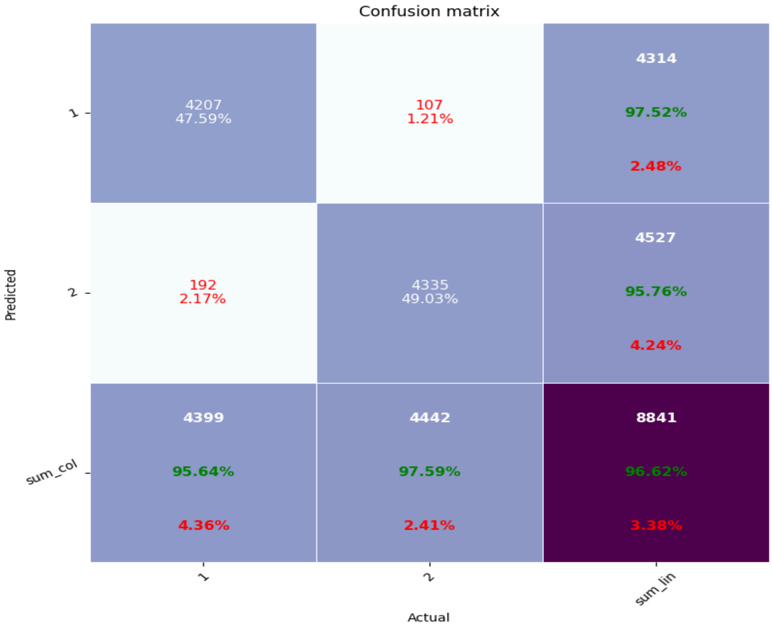


Fig. 8: Confusion matrix of the proposed model (CovTiNet) on of test samples

A total of 107 out of 4,314 misidentifications occurred in the Covid test samples, whereas 192 out of 4,527 misidentifications occurred in the non-Covid test samples due to joint feature distribution presented in both categories. For example, *Accident* and *Health*-related samples of non-Covid categories contain death-related frequent words, which are also available in the Covid test samples. As a result, the standard typical word distribution obtained some extra attention, and the model failed to detect the actual category. Overall, a 2.41% error was obtained from the non-Covid category, whereas a 4.36% error occurred in the Covid class with an average error of 3.38%.

Fig. 9 shows some test set samples with the actual and predicted labels. The first two Covid test samples (# 1 and #2) are taken from the Newspaper domain. The CovTiNet and ELECTRA-Bengali models correctly predicted the S# 1 text sample, whereas the other baseline methods failed to predict the correct labels due to the limitations of feature extraction methods (e.g., shortage of word semantics and context information). The proposed and baseline models cannot predict sample # 2 text samples owing to a shortage of aspect information (e.g., Covid-related word and semantic information).

S#.	Input: Translate	Actual Labelled	Correctly Predicted Models	Wrongly Predicted Models	Domain (URL)
1.	ক্ষুধার্ত মানুষের ... মোনেম লিমিটেড: Monem Limited ... for hungry people	Covid	CovTiNet & ELECTRA-Bengali	M-BERT, DistilBERT, IndicBERT, BiLSTM, CNN, VDCNN & DCNN	Newspaper (/https://tinyurl.com/2y6puj8v)
2.	কৃষক খামারিদের ... এখনই প্রণোদনা: Incentives for ... farmers now	Covid	Non of Them	All baselines & CovTiNet	Newspaper (https://tinyurl.com/yv2j7k2r)
3.	কুবির উন্নয়নের ... লক্ষে চীনা: Chinese for the ...development of Kubir	Non-Covid	Non of Them	All baselines & CovTiNet	Social Media (https://tinyurl.com/5yarm8br)
4.	শিশুর কিডনিতে ... সার্জারি প্রয়োজন: The baby's kidney ... needs surgery	Non-Covid	CovTiNet	ELECTRA-Bengali, M-BERT, DistilBERT, IndicBERT, BiLSTM, CNN, VDCNN & DCNN	Newspaper (https://tinyurl.com/2c6vvv3n)

Fig. 9: Actual and predicted test samples

In Fig. 9, the third and fourth Non-Covid samples are taken from social media and newspapers, respectively. The baseline and proposed systems do not correctly detect the third sample (i.e., #3) because a large number of words are semantically and syntactically similar to the Covid category [57, 58] whereas the context information is not similar to Covid category. Thus, the proposed (CovTiNet) and baseline methods cannot capture the context information correctly. The proposed model can successfully detect sample #4 text samples that express non-Covid health text samples. The proposed system correctly predicts this sample, but baseline methods failed to detect it. In this sample (#4), most of the words are related to the health category and, like with Covid category words, but the aspect is different (i.e., non-Covid). The proposed system position encoding and attention-based fusion properly extract the semantic, syntactic and context information, whereas the other methods do not adequately extract that information. As a result, the proposed CovTiNet is better for semantic, syntactic and aspect-based information retrieval purposes.

8 Discussion

The CTI is an essential prerequisite task (e.g., controlling the Covid related fake news, misinformation and disinformation identification) in social media and the World Wide Web. Another reason for CTI is post-Covid information retrieval and mining for topics or queries. Bengali is the 7^{th} most widely spoken language globally, it has been considered one of the crucial low-resource languages [5]. To the best of our knowledge, none of the past studies focused on identifying or classifying Bengali text related to Covid-19 using deep learning techniques. For this reason, this research motivated us to develop an automatic

Covid-19 text identification system in Bengali with a newly developed covid text corpus (BCovC). This work used attention-based position embedding feature fusion with Attention-based Convolutional Neural Networks (ACNN) called CovTiNet to perform the task.

Some key findings of this research are highlighted in the following:

- In this research (i.e., Sec. 4), Algorithms 1 and 2 explained detailed guidelines of corpus development, including data collection, pre-processing, annotation and quality measurements. Based on these algorithms, this work developed a new corpus (*BCovC*) for identifying Covid text in Bengali. To the best of our knowledge, *BCovC* is the first corpus in Bengali for Covid text identification. The process described in this research can be utilized to build any text corpora for other zero or low-resource languages.
- Morphological variations of a language significantly impact the semantic, syntactic and contextual meaning of words. In Sec. 7.1, Tables 6 and 7 confirmed that the attention-based feature fusion embedding is better than the single embedding for extracting textual features. *Bengali* is a morphologically rich language that consists of three linguistic variants in written forms: Sadhu-bhasha, Cholito-bhasha and Sanskrit-bhasha. As a result, a single embedding method cannot represent words or sentences' semantic and syntactic meanings well. In contrast, the attention and feature fusion operations can represent text's better semantic and syntactic meanings. Thus, the CovTiNet model achieved superior performance than baseline models for Covid text identification [59].
- The combinations of word embeddings and classification methods generate 40 classifier models. It is very arduous and time-consuming to evaluate all modes. We can reduce the evaluation burden by reducing the number of embedding models selected for the downstream task (CTI). In particular, in this work, three embedding models and six deep learning methods produce 18 classifier models only for a single hyperparameter combination. There were 40 CTI models, i.e., 36 for deep learning models and 4 for transformers models. It is possible to select only the best embedding models and use them to perform the classification task for better outcomes [60]. This work introduced an intrinsic evaluation method (see Algorithm 3) to evaluate the embedding models (Sec. 5.1.1). We selected the best-performed embedding models based on intrinsic evaluation, and only these modes are used for the CTI tasks. This process will help generate fewer classifier models (due to the reduced number of combinations of embedding and classification methods), reducing the training and evaluation time. The technique proposed in this work may be used for other low-resource languages.
- Table 8 showed the performance of baselines and the proposed model (CovTiNet) to perform the CTI task in Bengali (Sec. 7.2). Although the transformers-based fine-tuning models have achieved state-of-the-art results for text classification tasks in high-resource languages (like English), these models cannot show better performance due to large morphological variations in Bengali. At the same time, the performance of non-contextual word

embedding models has improved due to the integration of attention-based feature fusion and position encoding schemes. It is evident from Table 8 that the tokenization operation of transformer-based language models had degraded the classification performance, position encoding improved the contextual information, and attention-based feature fusion improved the semantic and syntactic feature representations.

- The non-contextual embedding methods (i.e., Word2Vec, FastText, GloVe) cannot extract the context-aware and semantically or syntactically correlated features due to their methodological limitations. To overcome the non-contextual embedding issues, this research introduces an attention-based position embedding feature fusion. Three additional operations have been added with the non-contextual embeddings, such as (i) word-position information, which improves the context-aware feature representations, (ii) fusion of multiple non-contextual embeddings, that combine multiple embedding features and enhances the semantic/syntactic correlations and (iii) finally applied the attention operation for improving the holistic feature representation. To the best of our knowledge, this is the first attempt to develop the attention-based position embedding feature fusion for a resource-constrained (i.e., Bengali) language using non-contextual embeddings.
- Due to morphological variation and lack of impactful global features, the existing single-layer multi-kernel CNN has not adequately extracted the sentence and document-level semantics of Bengali texts. In this regard, the attention operation is applied after the CNN operation. This attention operation improves the word-word correlation and extracts better sentence-level features. These sentence-level features also improve the document-level semantics and overcome the existing CNN shortcomings. We developed a network called CovTiNet by combining APeAGF and attention-based CNN (ACNN). We have tuned this network on the developed dataset with optimized hyperparameters (Table 8).
- In this research, the text pre-processing and expert-level annotation operations have overcome the data-level uncertainty, whereas the model uncertainty is partially overcome by the expected and soft-max probability values. The developed CovTiNet is a neural network-based supervised classification method where a set of non-linear equations (i.e., Eqs. 1-14) have been applied for text-to-expected category tagging purposes. The CovTiNet output layer contains two probability-related equations (concerning uncertainty), such as the expected category selection equation (Eq. 15) and the soft-max probability distribution equation (Eq. 16). The Covid text identification is a binary text classification task. Eq. 16 is forced to assign a category name based on the maximum probability value, and subtracted value is partially considered as an uncertainty or error value of the corresponding category (i.e., ground-truth maximum probability). Thus, if the input contains an out-of-distribution (OOD), then the soft-max value must belong to any category

(Covid or non-Covid). However, an uncertain situation is when a text contains OOD value and equally distributed information, and both categories contain equal probability value. The uncertainty can be solved using a multi-label text classification task, but the current research's primary concern is to develop a multi-class text classification task. A future research task will consider a more depth analysis of the uncertainty in the deep learning model. The developed CovTiNeT system is generalized interims of language, i.e., CovTiNet is generalized for Bengali text classification tasks, such as sentiment analysis, emotion classification and other Bengali text classification domains. The proposed CovTiNet can be applied to similar applications in other low-resource languages. This system can be applied straight away to other resource-constrained languages (e.g., Urdu, Arabic, Hindi, and others) by simply tuning the hyperparameters if the corpus is available for the respective language.

- If a sample text belongs to the Covid category or non-Covid category with a specific ratio at the same time, the uncertainty of this kind is resolved by the CovTiNet model (i.e., employing Eqs. 15-16), where the decision is made in favour of the category based on the maximum expected value. Although uncertainty related to the text classification task described in this research is not reasonably related to the methods explained by Lotfi et al. [2] and Kropat et al. [61], we will explore uncertainty issue in future.
- Future uncertainty in the text classification domain relates to the difficulty of predicting the exact nature of future data sets and the types of text classification problems that may arise [62]. There is also uncertainty around the availability and effectiveness of new technologies and algorithms that may be used for text classification, as well as the potential for changes in the field as new research and data become available. Additionally, there is a need to understand the potential risks associated with text classification, such as the potential for incorrect or biased classifications and data leakage and privacy violations. The development of more effective techniques for handling uncertainty in text classification is a critical research area that has the potential to improve the accuracy and efficiency of these systems significantly. Future research in this field will likely focus on developing more advanced ensemble techniques, such as stacking and boosting, as well as exploring the potential implications of new methods and technologies. Additionally, researchers must consider the potential risks associated with text classification, such as incorrect or biased classifications, data leakage and privacy violations. Finally, to ensure the reliability of text classification systems, it is crucial to assess the potential for future uncertainty and develop methods to mitigate it.
- The CovTiNet does not work for short text (when two or three words exist in a document). The attention-based feature fusion may incorrectly change the semantic/syntactic meaning due to biased attention operation. On the other hand, the ACNN required more training due to additional attention parameters.

9 Conclusion

This research presented an intelligent text processing framework (CovTiNet) to identify Covid-related texts in Bengali using an attention-based positional embedding feature fusion with ACNN. The data-driven position encoding and attention-based feature fusion overcame the OOV issues of single embeddings and improved the contextual semantic/syntactic features representation. The attention operation enhanced the Bengali feature correlations of word-level and sentence-level, whereas the position encoding and feature fusion improved the contextual representation. Additionally, due to the unavailability of Covid-related datasets, this study developed a couple of corpora: Bengali Covid text corpus (*BCovC*) and Covid embedding corpus (*CovEC*) for covid text identification and classification. The intrinsic evaluation has reduced the burden of evaluating classification models for the downstream task (CTI). Moreover, the proposed CovTiNet framework has achieved an accuracy of 96.61 ± 0.001, which is the maximum based on deep learning and transformer-based baseline methods.

Although the CovTiNet framework has achieved the highest performance, further improvement can be obtained using another pre-trained transformer-based language model in Bengali (e.g., RoBERTa, ELECTRA and BERT). Improving the sub-word feature representation and dynamic feature fusion methods can enhance the performance of the CTI task.

Conflict of interest

The authors declare that they have no conflict of interest.

Data availability

The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

References

[1] Alsinglawi, B., Mubin, O., Alnajjar, F., Kheirallah, K., Elkhodr, M., Zobbi, M.A., Novoa, M., Arsalan, M., Poly, T.N., Gochoo, M., Khan, G., Dev, K.: A simulated measurement for covid-19 pandemic using the effective reproductive number on an empirical portion of population: epidemiological models. Neural Computing and Applications, 1–9 (2021). https://doi.org/10.1007/s00521-021-06579-2

[2] Lotfi, R., Kheiri, K., Sadeghi, A., Tirkolaee, E.B.: An extended robust mathematical model to project the course of covid-19 epidemic in iran. Annals of Operations Research (2022). https://doi.org/10.1007/s10479-021-04490-6

[3] Hasni, S., Faiz, S.: Word embeddings and deep learning for location prediction: tracking coronavirus from british and american tweets. Social Network Analysis and Mining **11**(1), 66 (2021). https://doi.org/10.1007/s13278-021-00777-5

[4] DAngelo, G., Palmieri, F.: Enhancing covid-19 tracking apps with human activity recognition using a deep convolutional neural network and har-images. Neural Computing and Applications (2021). https://doi.org/10.1007/s00521-021-05913-y

[5] Hossain, M.R., Hoque, M.M., Siddique, N., Sarker, I.H.: Bengali text document categorization based on very deep convolution neural network. Expert Systems with Applications **184**, 115394 (2021). https://doi.org/10.1016/j.eswa.2021.115394

[6] Huan, J.L., Sekh, A.A., Quek, C., Prasad, D.K.: Emotionally charged text classification with deep learning and sentiment semantic, vol. 34, pp. 2341–2351 (2022). https://doi.org/10.1007/s00521-021-06542-1. https://doi.org/10.1007/s00521-021-06542-1

[7] Afroze, S., Hoque, M.M.: Sntiemd: Sentiment specific embedding model generation and evaluation for a resource constraint language. In: Intelligent Computing & Optimization, pp. 242–252. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-19958-5_23

[8] Hossain, M.R., Hoque, M.M.: Automatic bengali document categorization based on word embedding and statistical learning approaches. In: Proc. IC4ME2, Rajshahi, Bangladesh, pp. 1–6 (2018). https://doi.org/10.1109/IC4ME2.2018.8465632

[9] Hossain, M.R., Hoque, M.M.: Automatic bengali document categorization based on deep convolution nets. In: Proc. ERCICA, Bangalore, India, pp. 513–525 (2019). https://doi.org/10.1007/978-981-13-5953-8_43

[10] Asim, M.N., Ghani, M.U., Ibrahim, M.A., Mahmood, W., Dengel, A., Ahmed, S.: Correction to: Benchmarking performance of machine and deep learning-based methodologies for urdu text document classification. Neural Computing and Applications **33**(6), 2157–2157 (2021). https://doi.org/10.1007/s00521-020-05435-z

[11] Abiodun, E.O., Alabdulatif, A., Abiodun, O.I., Alawida, M., Alabdulatif, A., Alkhawaldeh, R.S.: A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities. Neural Computing and Applications **33**(22), 15091–15118 (2021). https://doi.org/10.1007/s00521-021-06406-8

[12] Rahimi, I., Chen, F., Gandomi, A.H.: A review on covid-19 forecasting models. (2021). https://doi.org/10.1007/s00521-020-05626-8. https://doi.org/10.1007/s00521-020-05626-8

[13] Hossain, M.R., Hoque, M.M.: Covtexminer: Covid text mining using cnn with domain-specific glove embedding. In: Intelligent Computing & Optimization, pp. 65–74. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-19958-5_7

[14] Kolluri, N.L., Murthy, D.: Coverifi: A covid-19 news verification system. Online Social Networks and Media **22**, 100123 (2021). https://doi.org/10.1016/j.osnem.2021.100123

[15] Ng, R., Chow, T.Y.J., Yang, W.: News media narratives of covid-19 across 20 countries: Early global convergence and later regional divergence. PLOS ONE **16**(9), 1–12 (2021). https://doi.org/10.1371/journal.pone.0256358

[16] Miao, L., Last, M., Litvak, M.: Tracking social media during the covid-19 pandemic: The case study of lockdown in new york state. Expert Systems with Applications **187**, 115797 (2022). https://doi.org/10.1016/j.eswa.2021.115797

[17] Koh, J.X., Liew, T.M.: How loneliness is talked about in social media during covid-19 pandemic: Text mining of 4,492 twitter feeds. Journal of Psychiatric Research **145**, 317–324 (2022). https://doi.org/10.1016/j.jpsychires.2020.11.015

[18] Paka, W.S., Bansal, R., Kaushik, A., Sengupta, S., Chakraborty, T.: Cross-sean: A cross-stitch semi-supervised neural attention model for covid-19 fake news detection. Applied Soft Computing **107**, 107393 (2021). https://doi.org/10.1016/j.asoc.2021.107393

[19] Elhadad, M.K., Li, K.F., Gebali, F.: Detecting misleading information on covid-19. IEEE Access **8**, 165201–165215 (2020). https://doi.org/10.1109/ACCESS.2020.3022867

[20] Song, X., Petrak, J., Jiang, Y., Singh, I., Maynard, D., Bontcheva, K.: Classification aware neural topic model for covid-19 disinformation categorisation. PLOS ONE **16**(2), 1–22 (2021). https://doi.org/10.1371/journal.pone.0247086

[21] Ghasiya, P., Okamura, K.: Investigating covid-19 news across four nations: A topic modeling and sentiment analysis approach. IEEE Access **9**, 36645–36656 (2021). https://doi.org/10.1109/ACCESS.2021.3062875

[22] Nassif, A.B., Elnagar, A., Elgendy, O., Afadar, Y.: Arabic fake news detection based on deep contextualized embedding models. Neural Computing and Applications (2022). https://doi.org/10.1007/s00521-022-07206-4

[23] Patwa, P., Bhardwaj, M., Guptha, V., Kumari, G., Sharma, S., PYKL, S., Das, A., Ekbal, A., Akhtar, M.S., Chakraborty, T., Shu, K., Bernard, H.R., Liu, H.: Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In: Combating Online Hostile Posts in Regional Languages During Emergency Situation, pp. 42–53. Springer, Cham (2021)

[24] Hussein, A., Ghneim, N., Joukhadar, A.: DamascusTeam at NLP4IF2021: Fighting the Arabic COVID-19 infodemic on Twitter using AraBERT. In: Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, pp. 93–98. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.nlp4if-1.13. https://aclanthology.org/2021.nlp4if-1.13

[25] Mattern, J., Qiao, Y., Kerz, E., Wiechmann, D., Strohmaier, M.: FANG-COVID: A new large-scale benchmark dataset for fake news detection in German. In: Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER), pp. 78–91. Association for Computational Linguistics, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.fever-1.9. https://aclanthology.org/2021.fever-1.9

[26] Saghayan, M.H., Ebrahimi, S.F., Bahrani, M.: Exploring the impact of machine translation on fake news detection: A case study on persian tweets about covid-19. In: 2021 29th Iranian Conference on Electrical Engineering (ICEE), pp. 540–544 (2021). https://doi.org/10.1109/ICEE52715.2021.9544409

[27] Harakawa, R., Iwahashi, M.: Ranking of importance measures of tweet communities: Application to keyword extraction from covid-19 tweets in japan. IEEE Transactions on Computational Social Systems **8**(4), 1030–1041 (2021). https://doi.org/10.1109/TCSS.2021.3063820

[28] Hajek, P., Barushka, A., Munk, M.: Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. Neural Computing and Applications **32**(23), 17259–17274 (2020). https://doi.org/10.1007/s00521-020-04757-2

[29] Paul, S., Saha, S., Singh, J.P.: Covid-19 and cyberbullying: deep ensemble model to identify cyberbullying from code-switched languages during the pandemic. Multimedia Tools and Applications **9**, 1573–7721 (2022). https://doi.org/10.1007/s11042-021-11601-9

[30] Dhar, A., Mukherjee, H., Dash, N.S., Roy, K.: Text categorization: past

and present. Artificial Intelligence Review **54**, 3007–3054 (2021). https://doi.org/10.1007/s10462-020-09919-1

[31] Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement **20**(1), 37–46 (1960). https://doi.org/10.1177/001316446002000104

[32] Alissa, M., Lones, M.A., Cosgrove, J., Alty, J.E., Jamieson, S., Smith, S.L., Vallejo, M.: Parkinson's disease diagnosis using convolutional neural networks and figure-copying tasks. Neural Computing and Applications **34**(2), 1433–1453 (2022). https://doi.org/10.1007/s00521-021-06469-7

[33] Dasari, S.K., Cheddad, A., Palmquist, J., Lundberg, L.: Clustering-based adaptive data augmentation for class-imbalance in machine learning (cada): additive manufacturing use case. (2022). https://doi.org/10.1007/s00521-022-07347-6. https://doi.org/10.1007/s00521-022-07347-6

[34] Jadoon, N.K., Anwar, W., Bajwa, U.I., Ahmad, F.: Statistical machine translation of indian languages: a survey, vol. 31, pp. 2455–2467 (2019). https://doi.org/10.1007/s00521-017-3206-2. https://doi.org/10.1007/s00521-017-3206-2

[35] Hossain, M.R., Hoque, M.M.: Towards Bengali word embedding: Corpus creation, intrinsic and extrinsic evaluations. In: Proceedings of the 17th International Conference on Natural Language Processing (ICON), pp. 453–459. NLP Association of India (NLPAI), Indian Institute of Technology Patna, Patna, India (2020)

[36] Li, J., Hu, R., Liu, X., Tiwari, P., Pandey, H.M., Chen, W., Wang, B., Jin, Y., Yang, K.: A distant supervision method based on paradigmatic relations for learning word embeddings. Neural Computing and Applications **32**(12), 7759–7768 (2020). https://doi.org/10.1007/s00521-019-04071-6

[37] Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Commun. ACM **8**(10), 627–633 (1965). https://doi.org/10.1145/365628.365657

[38] Hill, F., Reichart, R., Korhonen, A.: SimLex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics **41**(4), 665–695 (2015). https://doi.org/10.1162/COLI_a_00237

[39] Jeffrey Pennington, C.M. Richard Socher: Glove: Global vectors for word representation. In: Proc. EMNLP, pp. 1532–1543. ACL, Doha, Qatar (2014). https://doi.org/10.3115/v1/D14-1162

[40] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Tran. ACL **5**, 135–146 (2017). https://doi.org/

10.1162/tacl_a_00051

[41] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proc. ICLR, Scottsdale, Arizona, USA, pp. 1–12 (2013)

[42] Wang, B., Wang, A., Chen, F., Wang, Y., Kuo, C.-C.J.: Evaluating word embedding models: methods and experimental results. APSIPA Transactions on Signal and Information Processing **8**, 19 (2019). https://doi.org/10.1017/ATSIP.2019.12

[43] Potamias, R.A., Siolas, G., Stafylopatis, A.G.: A transformer-based approach to irony and sarcasm detection, vol. 32, pp. 17309–17320 (2020). https://doi.org/10.1007/s00521-020-05102-3. https://doi.org/10.1007/s00521-020-05102-3

[44] Hossain, M.R., Hoque, M.M., Dewan, M.A.A., Siddique, N., Islam, N., Sarker, I.H.: Authorship classification in a resource constraint language using convolutional neural networks. IEEE Access **9**, 100319–100338 (2021). https://doi.org/10.1109/ACCESS.2021.3095967

[45] Cadoni, M., Lagorio, A., Khellat-Kihel, S., Grosso, E.: On the correlation between human fixations, handcrafted and cnn features. Neural Computing and Applications **33**(18), 11905–11922 (2021). https://doi.org/10.1007/s00521-021-05863-5

[46] Mikolov, T., Yih, W.-t., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751. Association for Computational Linguistics, Atlanta, Georgia (2013)

[47] Levy, O., Goldberg, Y.: Linguistic regularities in sparse and explicit word representations. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning, pp. 171–180. Association for Computational Linguistics, Ann Arbor, Michigan (2014). https://doi.org/10.3115/v1/W14-1618

[48] Bi, J., Wang, F., Yan, X., Ping, J., Wen, Y.: Multi-domain fusion deep graph convolution neural network for eeg emotion recognition. Neural Computing and Applications (2022). https://doi.org/10.1007/s00521-022-07643-1

[49] Williams, J., Comanescu, R., Radu, O., Tian, L.: DNN multimodal fusion techniques for predicting video sentiment. In: Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML), pp. 64–72. Association for Computational Linguistics,

Melbourne, Australia (2018). https://doi.org/10.18653/v1/W18-3309. https://aclanthology.org/W18-3309

[50] Islam, K.I., Kar, S., Islam, M.S., Amin, M.R.: SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 3265–3271. Association for Computational Linguistics, Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.findings-emnlp.278. https://aclanthology.org/2021.findings-emnlp.278

[51] Hossain, M.R., Hoque, M.M., Sarker, I.H.: Text classification using convolution neural networks with fasttext embedding. In: Proc. HIS, pp. 103–113. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73050-5_11

[52] Bhowmik, N.R., Arifuzzaman, M., Mondal, M.R.H.: Sentiment analysis on bangla text using extended lexicon dictionary and deep learning algorithms. Array **13**, 100123 (2022). https://doi.org/10.1016/j.array.2021.100123

[53] Rahman, M.M., Pramanik, M.A., Sadik, R., Roy, M., Chakraborty, P.: Bangla documents classification using transformer based deep learning models. In: 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), pp. 1–5 (2020). https://doi.org/10.1109/STI50764.2020.9350394

[54] Kula, S., Kozik, R., Choras, M.: Implementation of the bert-derived architectures to tackle disinformation challenges. Neural Computing and Applications (2021). https://doi.org/10.1007/s00521-021-06276-0

[55] Kakwani, D., Kunchukuttan, A., Golla, S., N.C., G., Bhattacharyya, A., Khapra, M.M., Kumar, P.: IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4948–4961. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.445. https://aclanthology.org/2020.findings-emnlp.445

[56] Bhowmick, R.S., Ganguli, I., Sil, J.: Character-level inclusive transformer architecture for information gain in low resource code-mixed language. Neural Computing and Applications (2022). https://doi.org/10.1007/s00521-022-06983-2

[57] Singh, S.M., Singh, T.D.: An empirical study of low-resource neural machine translation of manipuri in multilingual settings. Neural Computing and Applications (2022). https://doi.org/10.1007/s00521-022-07337-8

[58] Sphaier, P.B., Paes, A.: User intent classification in noisy texts: an investigation on neural language models. Neural Computing and Applications (2022). https://doi.org/10.1007/s00521-022-07383-2

[59] Song, S., Sun, Y., Di, Q.: Multiple order semantic relation extraction. Neural Computing and Applications **31**(9), 4563–4576 (2019). https://doi.org/10.1007/s00521-018-3453-x

[60] Huang, J., Zhang, T., Zhu, J., Yu, W., Tang, Y., He, Y.: A deep embedding model for knowledge graph completion based on attention mechanism. Neural Computing and Applications **33**(15), 9751–9760 (2021). https://doi.org/10.1007/s00521-021-05742-z

[61] Kropat, E., Meyer-Nieberg, S.: Slime mold inspired evolving networks under uncertainty (slimo). In: 2014 47th Hawaii International Conference on System Sciences, pp. 1153–1161 (2014). https://doi.org/10.1109/HICSS.2014.149

[62] Özmen, A., Weber, G.W., İnci Batmaz, Kropat, E.: Rcmars: Robustification of cmars with different scenarios under polyhedral uncertainty set. Communications in Nonlinear Science and Numerical Simulation **16**(12), 4780–4787 (2011). https://doi.org/10.1016/j.cnsns.2011.04.001. SI:Complex Systems and Chaos with Fractionality, Discontinuity, and Nonlinearity