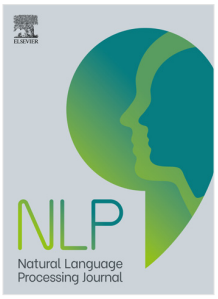
Journal Pre-proof



Leveraging contextual representations with BiLSTM-based regressor for lexical complexity prediction

Abdul Aziz, Md. Akram Hossain, Abu Nowshed Chy, Md. Zia Ullah, Masaki Aono

PII: S2949-7191(23)00036-5
DOI: https://doi.org/10.1016/j.nlp.2023.100039
Reference: NLP 100039

To appear in: *Natural Language Processing Journal*

Received date : 16 February 2023
Revised date : 17 September 2023
Accepted date : 26 October 2023

Please cite this article as: A. Aziz, Md.A. Hossain, A.N. Chy et al., Leveraging contextual representations with BiLSTM-based regressor for lexical complexity prediction. *Natural Language Processing Journal* (2023), doi: https://doi.org/10.1016/j.nlp.2023.100039.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2023 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Journal Pre-proof

Highlights

Leveraging Contextual Representations with BiLSTM-based Regressor for Lexical Complexity Prediction

Abdul Aziz,Md. Akram Hossain,Abu Nowshed Chy,Md Zia Ullah,Masaki Aono

- Lexical complexity prediction is a subtask of text simplification
- Fusion of transformer models provides more meaningful representations of the inputs
- The BiLSTM-Regressor improve pairwise learning between sentence and target word
- Contextual features from transformers effective for lexical complexity estimation

Journal Pre-proof

Journal Pre-proof

Leveraging Contextual Representations with BiLSTM-based Regressor for Lexical Complexity Prediction

Abdul Aziz[a,1], Md. Akram Hossain[a,*,1], Abu Nowshed Chy[a], Md Zia Ullah[b] and Masaki Aono[c]

[a]Department of Computer Science and Engineering, University of Chittagong, Chattogram-4331, Bangladesh
[b]School of Computing, Engineering, and the Built Environment, Edinburgh Napier University, Edinburgh, UK
[c]Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Aichi, 441-8580, Japan

ARTICLE INFO

Keywords:
Lexical complexity prediction
Lexical simplification
Sentence-pair regression
Transformer models.

ABSTRACT

Lexical complexity prediction (LCP) determines the complexity level of words or phrases in a sentence. LCP has a significant impact on the enhancement of language translations, readability assessment, and text generation. However, the domain-specific technical word, the complex grammatical structure, the polysemy problem, the inter-word relationship, and dependencies make it challenging to determine the complexity of words or phrases. In this paper, we propose an integrated transformer regressor model named ITRM-LCP to estimate the lexical complexity of words and phrases where diverse contextual features are extracted from various transformer models. The transformer models are fine-tuned using the text-pair data. Then, a bidirectional LSTM-based regressor module is plugged on top of each transformer to learn the long-term dependencies and estimate the complexity scores. The predicted scores of each module are then aggregated to determine the final complexity score. We assess our proposed model using two benchmark datasets from shared tasks. Experimental findings demonstrate that our ITRM-LCP model obtains 10.2% and 8.2% improvement on the news and Wikipedia corpus of the CWI-2018 dataset, compared to the top-performing systems (DAT, CAMB, and TMU). Additionally, our ITRM-LCP model surpasses state-of-the-art LCP systems (DeepBlueAI, JUST-BLUE) by 1.5% and 1.34% for single and multi-word LCP tasks defined in the SemEval LCP-2021 task.

1. Introduction

Text simplification is the procedure of transforming a complex sentence using simple and familiar words to improve its readability (Nisioi et al. (2017)). It is beneficial for improving reading aids for children, people with reading disabilities like dyslexia, aphasia, non-native speakers, and people with a low literacy rate (Watanabe et al. (2009); Saggion (2017)). It is also beneficial for other natural language processing (NLP) applications including text summarization (Vanderwende et al. (2007); Zaman et al. (2020)), machine translation, and text generation. Lexical simplification (LS) is a subtask of simplification of texts (Vanderwende et al. (2007)) that focuses on identifying the complex words for a target audience and replacing them with their simpler alternatives of equivalent meaning (Shardlow, Matthew (2014)). It follows the four important steps including synonym ranking, word sense disambiguation, substitution generation, and complex word identification (CWI) (Shardlow, Matthew (2014)). CWI focuses on determining the complex words of a sentence whereas substitution generation is the process of finding words or expressions to replace these complex words. The polysemy problem of the substitute candidates is addressed through the word sense disambiguation task (Li and Suzuki (2021); Kwon et al. (2021)) and the synonym ranking task ranks the remaining substitutes according to the simplicity score. The CWI and synonym ranking tasks in LS are highly identical as they both address the notion of lexical complexity.

Lexical complexity prediction (LCP) is the process of precisely determining the complexity of words or phrases in a sentence (Shardlow et al. (2021a)). It has a significant impact on selecting complex words and their alternative simpler words in the lexical simplification task. However, it is challenging to estimate the complexity of a word or phrase due to the use of domain-specific technical words and the complex grammatical structure. The contextual information of a word is also important because the complexity of a word might be context-dependent. Recently, researchers are

*Corresponding author
aziz.abdul.cu@gmail.com (A. Aziz); akram.hossain.cse.cu@gmail.com (Md.A. Hossain); nowshed@cu.ac.bd (A.N. Chy); m.ullah@napier.ac.uk (M.Z. Ullah); aono@tut.jp (M. Aono)
ORCID(s): 0000-0002-4022-7344 (M.Z. Ullah)
[1]The first two authors have equal contributions.

Journal Pre-proof

increasingly interested in the LCP task because of its promising applications. Following this trend, a couple of shared tasks are introduced to address the challenges (Yimam et al. (2018); Shardlow et al. (2021a)).

Most of the LCP systems (Gooding and Kochmar (2018); Gooding, Sian and Kochmar, Ekaterina (2019); Islam et al. (2021)) used a vast amount of features through exploiting morphological, lexical, semantic, collocational, nominal, syntactical, and psycho-linguistic characteristics. These features are mostly word-based and do not capture the context properly. To address this limitation most of the top-performing systems (Pan et al. (2021); Rivas Rojas and Alva-Manchego (2021); Yuan et al. (2021)) approached to leverage various transformers models for learning better contextual representation. We want to shed light on some of those systems. First, DeepBlueAI (Pan et al. (2021)) the top performing team of LCP-2021 fuse four transformer models prediction with different additional training strategies including multi-sample dropout, pseudo-labeling, and data augmentation. Second, (Rivas Rojas and Alva-Manchego (2021)) fuse four transformer models prediction but they use light gradient-boosting machine (LightGBM) Bayesian optimizer to fuse them whereas (Yuan et al. (2021)) performed a fusion of three transformer models prediction and a random forest regressor models prediction for the LCP-2021 task. The findings from these recent works motivate us to make a hypothesis that the fusion of various transformer models may learn better contextual representation than other settings. However, those systems still have some limitations to learn pair-wise features effectively. Those models predict the complexity score from the transformer model's final layer without adding any neural network architecture that affects the model's performance in distilling the relationship of the token-sentence pair effectively.

In this work, the benefits of contextual representation of the sentence pair settings from different transformer models are investigated for diverse corpus. Moreover, we also study the effectiveness of different integration techniques to fuse the outcome of the transformers models. This facilitates an investigation into how the inclusion of a deep neural network (DNN) architecture improves pairwise learning. Finally, an investigation is performed to determine whether the domain-specific contexts of words can be extracted effectively from the text that is useful for the LCP task.

The key contributions of our work are listed as follows:

1. We introduce an integrated transformer regressor model (ITRM) for the lexical complexity prediction (LCP) task named as ITRM-LCP. It employs various fine-tuned transformer models with token-sentence pairs to capture the diversity of contextual features.
2. To address the long-term dependencies problem and estimate the complexity scores effectively, a BiLSTM-based regressor is plugged on top of each transformer model.
3. We analyze and present the experimental findings of various integration strategies to select the effective one for the lexical complexity prediction (LCP) task.
4. Rigorous experimental findings and the comparative performance analysis against the state-of-the-art approaches are presented based on the two benchmark datasets. Our research findings provide some useful insights of the lexical complexity prediction (LCP) task.

The structure of the remaining contents is as follows: Section 2 includes a summary of prior research that ignites us to contribute in this problem domain. Later, we introduce our proposed lexical complexity prediction method in Section 3. Section 4 includes the detailed experiments and evaluation as well as performance comparison with related approaches. Some insightful discussions are provided in Section 4.5. Finally, we conclude our work and draw a set of future notions in Section 5.

2. Related Work

Complex word identification (CWI) is a crucial part of the automatic lexical simplification (LS) task (Zaman et al. (2020); Shardlow, Matthew (2014)). Several methods have been proposed to identify complex words or phrases in a sentence (Gooding and Kochmar (2018); De Hertog and Tack (2018)). Most of the earlier CWI systems are low-level feature-based (Kajiwara and Komachi (2018); Hartmann and Dos Santos (2018); Gooding and Kochmar (2018)) that exploited various hand-designed features to tackle the task challenges (see Section 2.1). Later, various methods used embedding features and deep learning-based approaches (De Hertog and Tack (2018); Saggion et al. (2018)) (see Section 2.2). Along with this direction, recently, transformer models are studied in the CWI task (Pan et al. (2021); Aziz et al. (2021)) (see Section 2.3).

2.1. Hand-crafted Features (HCFs) based Machine Learning Approaches for the LCP Task

To identify complex words, most of the CWI-2016 participants used different types of hand-crafted features (HCFs) including bag of n-grams, word length (Malmasi and Zampieri (2016); Malmasi et al. (2016)), psycholinguistic,

Journal Pre-proof

morphological (Paetzold and Specia (2016b); Sanjay et al. (2016)), syntactic, semantic (Paetzold and Specia (2016b); Sanjay et al. (2016)), Zipfian frequency distribution (Zampieri et al. (2016)), and word probability (Malmasi and Zampieri (2016)). They have utilized different well-established classification approaches including support vector machine (SVM), tree-based classifiers, and maximum entropy classifiers. Later, in the CWI-2018 shared task, most of the top-performing systems also employed several feature-based approaches including n-grams, word length, inter-word relationship, WordNet and parts of speech (POS) based features, syntactic, lexical, and psycholinguistic features (De Hertog and Tack (2018); Gooding and Kochmar (2018); Kajiwara and Komachi (2018)). Recently, in the LCP-2021 shared task (Shardlow et al. (2021a)), some of the HCF-based systems (Rotaru (2021)) also performed well. However, these systems are either good for single-word instances (SWIs) (Rotaru (2021); Islam et al. (2021)) or multi-word expressions (MWEs) (Bestgen (2021)) tasks. Since HCFs are mostly task and domain-specific, formulating an efficient generalized model for LCP is difficult. Besides, designing and exploiting hand-crafted features is arduous and time-consuming and these features are unable to represent the contextual dimension effectively.

2.2. Word-embedding and Deep Learning based Approaches for the LCP Task

A number of prior systems (Kuru (2016); Sanjay et al. (2016); Saggion et al. (2018)) used word or sentence embedding based features to train their learning models built on support vector machine (SVM), linear regression, and deep learning-based models. Some other systems concatenate embedding features with other HCFs. To extract the embedding-based features, Kuru (2016) used Glove embedding (Pennington et al. (2014)) whereas Sanjay et al. (2016) and Saggion et al. (2018) used the Gensim Word2Vec skip-gram and continuous bag-of-words (CBOW) models trained with Google News dataset (Mikolov et al. (2013)). In the LCP-2021 task, some participants used word embeddings including global vectors (Glove), Word2Vec, and embeddings from language models (ELMo) (Rotaru (2021); Islam et al. (2021); Rozi et al. (2021)) where they used these embeddings to initiate their neural models or concatenates those features with other HCFs.

Deep learning-based approaches achieved competitive results in lexical complexity prediction. De Hertog and Tack (2018) used a deep neural network architecture with three types of feature sets including character embeddings with convolution neural network (CNN), HCFs, and word embeddings. Aroyehun et al. (2018) also employed a deep learning approach where word embeddings passed into the convolution layer. Recently, transformer (Vaswani et al. (2017)) based approaches are widely employed to tackle the LCP problem. Zaharia et al. (2020) utilized several transformer models and reported their experimental results on the CWI-2018 dataset where they obtained competitive performances but lower than the top-performing CWI-2018 system. In LCP-2021, most of the top-performing systems (Bani Yaseen et al. (2021); Pan et al. (2021)) used transformer models including bidirectional encoder representations from transformers (BERT), robustly optimized BERT approach (RoBERTa), and a lite BERT (ALBERT) and obtained notable performances. Most of them followed the sentence-pair training strategy to train the transformer model. Some systems (Bani Yaseen et al. (2021); Pan et al. (2021)) used different training strategies and employed various ensemble approaches. Pan et al. (2021) used data augmentation, pseudo labeling, ensemble training, and multi-sample dropout. (Bani Yaseen et al. (2021) used the weighted average of the prediction from different settings of BERT and RoBERTa models. Few systems applied diverse kinds of training strategies including adversarial training, multi-task learning (Islam et al. (2021)), and dummy annotation generation (Shirude et al. (2021)).

2.3. Comprehensive Analysis of the Transformer Models for the LCP Task

Observing most of the lexical complexity prediction (LCP) systems that were based on handcrafted features (HCF) and word embeddings with the deep neural networks. These systems do not capture the semantic orientation of the texts properly to estimate the lexical complexity. They usually suffer from generalization problems as systems perform well either for single-word or multi-word token data. Alternatively, transformer-based systems perform better than HCF and word embedding-based systems. Transformers overcome the limitations of learning long-term dependency through processing the sentence as a whole rather than word by word. Here, the multi-head attention and positional embedding mechanisms provide the necessary information about the relationship between words. These properties of transformers are important to tackle the challenges of the LCP task. Among the various transformers model, bidirectional encoder representations from transformers (BERT) (Devlin et al. (2019)) is very popular and perform well for various natural language processing (NLP) tasks including sentence classification, question-answering, text tagging, and text-pair regression. The general transformer (Vaswani et al. (2017)) model uses an encoder and a decoder network, whereas BERT uses only the encoder network to learn the latent representation of the input text. BERT employs two different types of training objectives; one is a masked language model and another is next sentence prediction (NSP). The NSP

Journal Pre-proof

idea focuses on learning whether the next text portion in a pair of texts is either true next or not. It helps BERT to distill the relationships between sentences which is crucial for the LCP task.

After BERT, several augmented and revised architectures are proposed by the researchers to improve the learning ability of the contextual information as well as reduce the resource overhead. For instance, the robustly optimized BERT pre-training approach, RoBERTa (Liu et al. (2019)) uses the same architecture as BERT but eliminates the next sentence prediction (NSP) objectives of BERT in pre-training. It focuses mainly on the key hyper-parameter choices through exploiting model training with longer sequences and larger mini-batches. Besides, RoBERTa utilizes dynamic masking that is performed every time a sequence is fed to the model. Therefore, the model encodes the different versions of the same sentence with masks on different positions which is critical for the LCP task to learn the inherent complexity of a token in context. The distilled version of BERT (DistilBERT) (Sanh et al. (2019)) model is another variant of BERT (Devlin et al. (2019)) that focuses on the speed-up of training through exploiting knowledge distillation to BERT. It discards the token-type embeddings and the NSP objective used in BERT as well as lessens the layers by a factor of two thus making it 60 percent faster and smaller than BERT. But following the RoBERTa (Liu et al. (2019)) model, DistilBERT is also trained on large batches using gradient accumulation with dynamic masking and retains the 97 percent performance of BERT. Extension of the Transformer-XL model (XLNet) (Yang et al. (2019)) is an another variant of BERT (Devlin et al. (2019)). It is a large bidirectional transformer that exploits advanced training strategy, greater data sets and more processing power. During the training phase, XLNet used permutation language modeling to combine the advantages of auto-regressive (AR) language modeling and auto-encoding (AE) methods, where tokens are predicted in random order. This aids the model to distill and learn bidirectional relationships between words which is beneficial for the LCP task.

The transformer-based systems distill the contextual information effectively and perform well in both single-word and multi-word token data. However, employing pre-trained transformer models without following appropriate neural architecture may not encode the pair-wise relations of the texts effectively. Therefore, an effective ensemble approach to fuse those diverse transformer models may capture the contextual dimension as well as pair-wise representation of the texts effectively.

3. Proposed Method

Given a text pair as the input and generating a continuous value as the output is called the text-pair regression task. We employed this approach to predict the lexical complexity of given target words in a sentence. The overview of our proposed integrated transformer regressor model for lexical complexity prediction (ITRM-LCP) framework is shown in Figure 1.

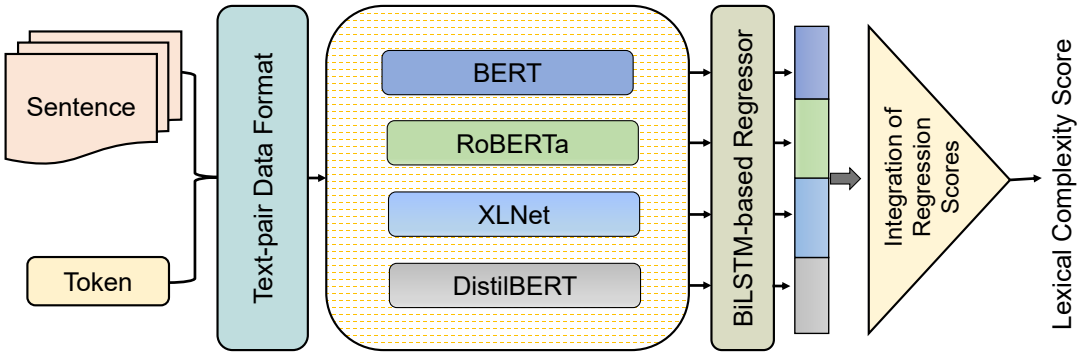


Figure 1: Schematic diagram of our proposed ITRM-LCP system. Transformer models are tuned on pairwise settings of sentences and tokens to generate the contextualized vectors. A BiLSTM-regressor module is plugged on the top of each transformer to enhance the feature learning representations. Finally, regression scores of each module are fused to get the final prediction.

Given an input token and sentence, we convert them to the corresponding data format where they are represented as a single packed sequence. To extract the contextualized embedding features, we utilize various transformers including

BERT, DistilBERT, RoBERTa, and XLNet. We fine-tune these models to capture the task-specific information of lexical complexity prediction explicitly. Later, we apply a bidirectional long short-term memory (BiLSTM) based regressor layer on top of each transformer model to predict the lexical complexity scores. We explore two different integration strategies including mean-based and blending integration-based approaches to fuse these prediction scores and estimate the final complexity prediction score.

3.1. Fine-tuning Transformer Models

To extract the effective contextual representations, we leverage four pre-trained transformers including BERT, DistilBERT, RoBERTa, and XLNet in our LCP method (ITRM-LCP) motivated from the prior works as described in Section 2.2 and 2.3. We fine-tune these models to make them specialized in the LCP task to learn the task-specific information effectively.

Fine-tuning the transformer models can lead to performance enhancement because it helps to fit the model with the domain-specific task. Hence, we fine-tune each transformer model based on the domain-specific datasets. We just add a single linear output layer on top of the core model and use the BERT's text-pair training approach for the LCP task as illustrated in Figure 2. Here, the target word and sentence pairs are represented as a single sequence where a special classification [CLS] token is appended at the starting position of the first sentence and a [SEP] tag is added in between sentences to separate them. A learned embedding is also added where every token indicates whether it belongs to sentence A or sentence B which is very crucial to learning contextual dependency between sentence and token in the LCP task. Finally, at the end of the input sequence, a [SEP] token is added. Therefore, the text-pair input sequence is like- *[CLS] sentence [SEP] target word [SEP].*

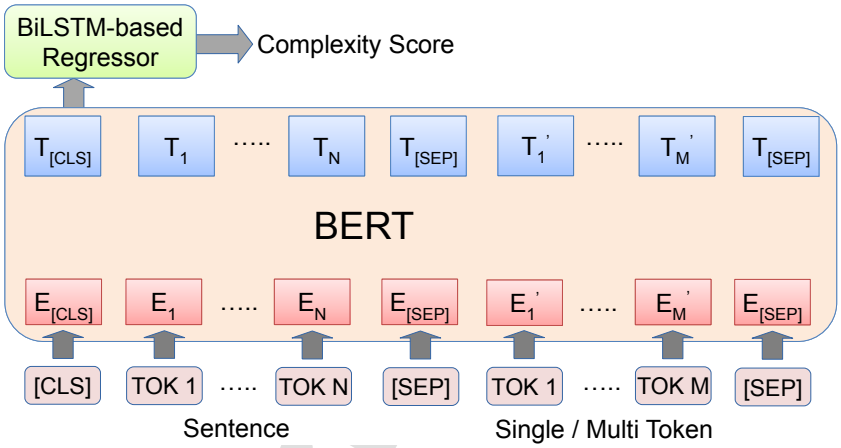


Figure 2: To employ pairwise settings of an input representation, the tokenizer packed the sentence and token as a single sequence. Then, this input sequence ([CLS] Sentence [SEP] Token [SEP]) feeds into the BERT model. The last layer's hidden states vector output of the BERT is passed to the BiLSTM-regressor and finally predicts the complexity score.

During the fine-tuning phase, we learn one new parameter (as compared to the pre-train BERT model) which is the complexity score (i.e. label). In our fine-tuning procedure, all other hyper-parameters stay the same as in transformer model training except for learning rates, batch size, epochs, and dropout. We use the last hidden state to get a fixed-dimensional pooling representation of the input sequence and add a BiLSTM-based regressor layer on top to get the predicted complexity score of each model.

3.2. BiLSTM-based Regressor Architecture

Bi-directional long short-term memory (BiLSTM) (Brueckner and Schulter (2014)) is a state-of-the-art variant of recurrent neural network (RNN). It has the ability to selectively assign weights to the words considering varying contexts. BiLSTM ensures the context-based semantic association information used to impressively make up for the shortfall of deep neural networks in obtaining local features and highlighting the importance of specific words to the whole context. To learn the long-term dependencies effectively and distill the required contextual information for predicting the complexity score of a token-sentence pair, we exploit a BiLSTM-based regressor model on top of the

Journal Pre-proof

transformer model for the LCP task as shown in Figure 3. We used this architecture for each transformer model to predict the complexity score.

The sequence output of each transformer model is used as the input features of the BiLSTM layer. Let, $f = (f_1, f_2, ..., f_T)$ be the sequential input to the BiLSTM. For capturing the past and future contexts, the BiLSTM model operates the sequential input in both forward and reverse directions. The forward hidden vector sequence $\overrightarrow{h} = (h_1, h_2, ..., h_T)$ and the backward hidden vector sequence $\overleftarrow{h} = (h_T, h_{T-1}, ..., h_1)$ operates the input in standard and reverse order, respectively (Yu et al. (2015)). Here, W represent the weight matrices, σ is the activation function to set the gating values in [0,1], b is the bias vector, and the following operations presented in equation (1) are iterating from $t = 1$ *to* T time steps to produce the output $y = (y_1, y_2, ..., y_T)$. This output vector y conveys the context of the input features.

$$\begin{aligned}
\overrightarrow{h_t} &= \sigma(W_{f\overrightarrow{h}} f_t + W_{\overrightarrow{hh}} \overrightarrow{h}_{t+1} + b_{\overrightarrow{h}}) \\
\overleftarrow{h_t} &= \sigma(W_{f\overleftarrow{h}} f_t + W_{\overleftarrow{hh}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \\
y_t &= W_{\overrightarrow{hy}} \overrightarrow{h_t} + W_{\overleftarrow{hy}} \overleftarrow{h_t} + b_y
\end{aligned} \tag{1}$$

The BiLSTM model uses two LSTMs on the input where the first and second LSTMs are the reversed copy of one another. It helps the model to take full advantage of the forward and backward input features to learn the inter and intra-relational structure of the tokens and sentences. The pooling layer lessens the dimensions of the feature maps by selecting a part of the input matrix. Hence, it lessens the learning parameters and the amount of computation in the network. In this work, we have utilized a one-dimensional max-pooling layer that summarizes the features of a particular region of the feature map produced by the BiLSTM layer as shown in Figure 3.

To avoid over-fitting, we use the dropout layer (Srivastava et al. (2014)) after both the max-pooling and dense layer. Following this strategy, some neurons are ignored at random during training. That means, on the forward pass their contribution to the activation of downstream neurons is discarded and on the backward pass, corresponding weight updates are not considered. Hence, it reduces the model complexity and enhances the generalization ability. Later, a final activation layer predicts the output complexity score. We use the mean squared error as a loss function and employ the AdamW (Loshchilov and Hutter (2017)) optimizer.

3.3. Fusion of Transformer Models Scores

Performing the models' integration is an efficient strategy that may produce better prediction accuracy and robustness than individual models. To capture the benefit of the diversity of predictions, we integrate the predicted scores of different transformer models to estimate the final complexity score. We explore two integration techniques to do this as described below.

3.3.1. Integration with Arithmetic Mean

To capture the benefits of diverse models, we integrate the estimated complexity scores of four models to determine a unified score in our ITRM-LCP model. Here, we utilize the arithmetic mean to aggregate the predicted complexity scores of all four models to determine the final score as shown in equation (2).

$$C_i = \frac{a_i + b_i + c_i + d_i}{4} \tag{2}$$

where C_i is the final complexity score. a_i, b_i, c_i, and d_i correspond to the complexity score obtained from four transformer models followed by BiLSTM module, respectively as shown in Figure 1.

3.3.2. Blending Integration

Blending integration is an extension of the stacked generalization integration technique. It uses the predictions of the validation set obtained from different base models as features to train the meta-model. Then, the predictions of the test set are passed to the trained meta-model to generate the final prediction (Chatzimparmpas et al. (2021)). The term *blending* was first introduced by the winning team of the Netflix Prize data competition[2], where they improved existing algorithm performance by a margin of 10% using the blending integration technique (Koren (2009)) which motivates

[2]https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

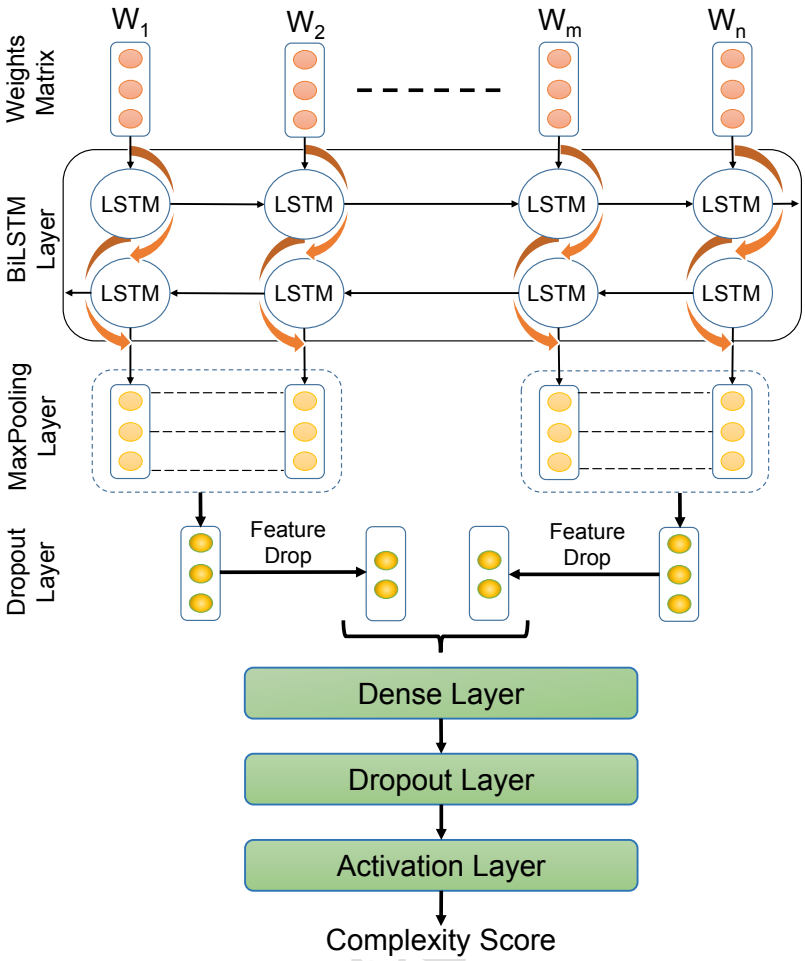


Figure 3: Regressor architecture of our ITRM-LCP model. The features vector of the transformer passes to the BiLSTM layer as an input to learn context-based semantic association information. The MaxPooling layer filters the top features from the features vector. Later, two Dropout layers and a Linear layer uses for better feature selection and learning.

us to investigate this technique for the LCP task. In Figure 4, we demonstrate the blending integration technique that we employ in our method. As the base models, we use the four transformer models employed in our method, including BERT, RoBERTa, XLNet, and DistilBERT with a BiLSTM regressor on top of each. Then, we take predictions on both validation and test data using these trained base models. To perform blending integration, we blend the validation predictions of four different base models as features and train the meta-model using these predictions. We choose various regressors as the meta-model, including decision tree, passive-aggressive, linear regression, support vector regressor, Theil Sen, Bayesian ridge, and automatic relevance determination (ARD) regressor. Finally, we stacked the test set predictions of four base models and pass this to the trained meta-model for predicting the final lexical complexity score.

4. Experiments and Evaluation

4.1. Dataset

To demonstrate the efficacy of our ITRM-LCP model, we evaluate our model on two benchmark datasets including CWIG3G2 (Yimam et al. (2018)) and CompLex (Shardlow et al. (2021b,a)). The CWIG3G2 dataset is used in NAACL-HLT-2018 CWI (Yimam et al. (2018)) shared task and the CompLex dataset is used in SemEval-2021 LCP (Shardlow et al. (2021a)) shared task.

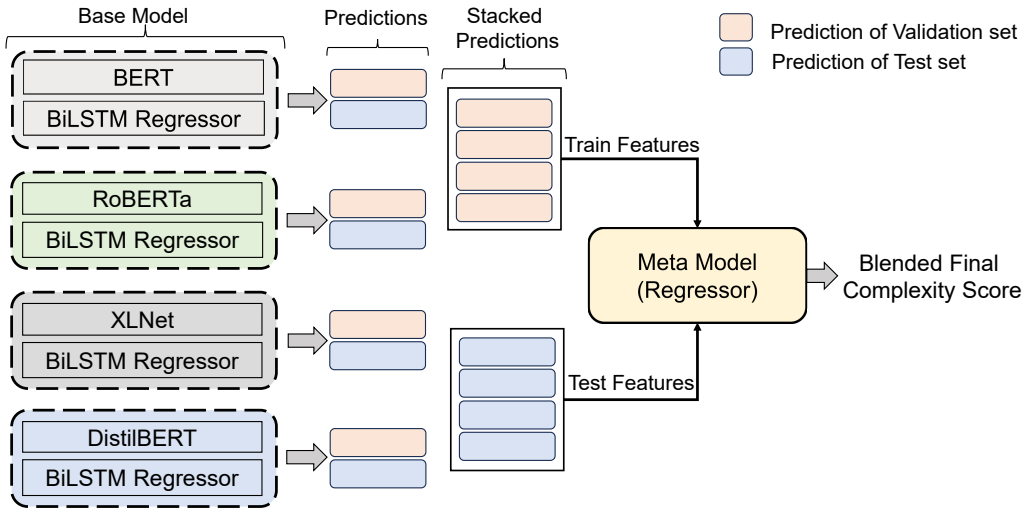


Figure 4: Oveview of our blending integration strategy.

Table 1
The statistics of the datasets used in this work.

CWI-2018 English dataset (CWIG3G2) (Yimam et al. (2018))				LCP-2021 dataset (CompLex) (Shardlow et al. (2021a))			
Corpus-Genres	Train	Dev	Test	Corpus-Genres	Train	Dev	Test
News (SWIs)	11949	1502	1813	Bible (SWIs)	2574	143	283
News (MWEs)	2053	262	282	Bible (MWEs)	505	29	66
Wikinews (SWIs)	6780	776	1138	Biomed (SWIs)	2576	135	289
Wikinews (MWEs)	966	94	149	Biomed (MWEs)	514	33	53
Wikipedia (SWIs)	4833	606	750	Europarl (SWIs)	2512	143	345
Wikipedia (MWEs)	718	88	120	Europarl (MWEs)	498	37	65
Total	27299	3328	4252	Total	9179	520	1101

NAACL-HLT-2018 Complex Word Identification (CWI) Task (Yimam et al. (2018)). According to the benchmark of CWI-2016 (Paetzold and Specia (2016a)) shared task, a system needs to identify whether a given word in a sentence is complex. Later, the CWI-2018 (Yimam et al. (2018)) task focused on both binary and probabilistic classification tasks. The probabilistic task focused on predicting the complexity score of a given target word in a particular context. CWI-2018 organizers provided a multilingual and multi-domain dataset. The English dataset of CWI-2018 contains texts from three different corpora including News (professionally written), Wikinews, and Wikipedia contents. The model assessments were performed per domain. To annotate the English dataset, they employed both native and non-native English speakers. CWI-2018 dataset contains an amalgam of single-word tokens and multi-word tokens annotation. The statistics for the single-word instances (SWIs) and the multi-word expressions (MWEs) in the train-dev-test segments of the News, Wikinews, and Wikipedia corpora are shown in Table 1. We performed some minor preprocessing here. To do this, we removed all of the noisy hashtags with numbers (e.g. Wikinews section id) that appeared before every sentence in Wikinews corpus and also removed the unnecessary white-space character beginning of the sentence.

SemEval-2021 Lexical Complexity Prediction (LCP) Task (Shardlow et al. (2021a)). CWI-2018 probabilistic classification task (Yimam et al. (2018)) was an impressive inclusion in the LCP domain. However, measuring the binary judgments of complexity based on the continuous value is still challenging. To address the challenges, Shardlow et al. (2021a) introduced a task named LCP at the SemEval-2021 where they focused on the continuous label of the lexical complexity estimation for a single and multi-word expression. SemEval-2021 LCP used a multiple domain-based English benchmark dataset CompLex (Shardlow et al. (2021b)). The dataset consists of three different genres including Bible (World English Bible translation), Biomed (articles from the CRAFT (Colorado richly annotated full

Journal Pre-proof

Table 2
Examples sentences with single and multi-word instances with their complexity scores.

Sentence	Token	Complexity Score
Examples from CWI-2018 Task		
The poll has been marred by widespread allegations of vote-rigging.	marred	.4000
	poll	.0000
	widespread	.0500
	widespread allegations	.1500
	widespread allegations of vote-rigging	.0000
	allegations	.5500
	vote-rigging.	.6500
Examples from LCP-2021 Task		
Yahweh's blessing brings wealth, and he adds no trouble to it.	blessing	.1718
That is why the European Council will also look at these issues.	European Council	.2205
Results of the European Council (Brussels, 13-14 March 2008) (debate)	European Council	.3611

text) corpus: a collection of open-access biomedical articles), and Europearl (a portion of the European parliament proceedings) domain. The task is composed of two sub-tasks, where Sub-task-1 and Sub-task-2 focused on estimating the complexity of a single-word instances (SWIs) and multi-word expressions (MWEs). Here, the length of the MWEs token is limited to two words. The LCP-2021 dataset statistics are shown in Table 1.

In Table 2, we articulate the instances of single-word and multi-word expressions (MWEs) of CWI-2018 probabilistic and LCP-2021 shared tasks. The examples of the CWI-2018 probabilistic task show that participants needed to predict the complexity score of almost every word (as single or MWEs) within a sentence. This makes it one of the most challenging tasks. The LCP-2021 shared task also addressed the problem in the same way where participants were asked to predict the complexity score on different genres of data and fixed the MWEs token length two. It also tried to explore the challenges of estimating the complexity score of the same token in diverse contexts. For example, from the last two rows of Table 2, we see that although both samples contain the identical phrase *European Council,* the complexity scores of both cases are not the same.

4.2. Model Configuration

In this section, we illustrate the strategy to set the optimal settings of the hyper-parameters of our ITRM-LCP model. To design and implement our model we used PyTorch and performed train-test on a GPU to take advantage of the effectiveness of tensors' parallel computation. All the experiments were performed on the Google Colaboratory (Bisong and Bisong (2019)) platform.

In our ITRM-LCP model, we used four pre-trained transformers (Wolf et al. (2020)) models including BERT[3], RoBERTa[4], XLNet[5], and DistilBERT[6]. Prior studies (Chy et al. (2021); Aziz et al. (2022)) suggest that fine-tuning the hyper-parameters of those models always outperforms the pre-trained models for downstream tasks. We fine-tuned some hyperparameters including training batch size, learning rate, and epochs. The optimal settings of these hyper-parameters are illustrated in Table 3. We used a grid search technique based on the development dataset, a kind of brute-force approach to select the optimal hyperparameters. After constructing a grid of potential discrete hyperparameter values considering the literature, we fit the model using every possible combination. Later, the combination that provided the best performance is then chosen.

A BiLSTM-based regressor layer plugged at the end of each of the transformers models as described in Section 3.2. To reduce the noisy features and avoid over-fitting, we fine-tuned some hyper-parameters of this module including the output dimension of the BiLSTM layer, dropout rate, and hidden units sizes of a dense layer. We employed various settings for the output dimension of the BiLSTM layer and dropout rate. The best settings of these hyper-parameters

[3]https://huggingface.co/bert-base-uncased
[4]https://huggingface.co/roberta-base
[5]https://huggingface.co/xlnet-base-cased
[6]https://huggingface.co/distilbert-base-uncased

Journal Pre-proof

Table 3
The optimal hyper-parameters settings for CWI-2018 (Yimam et al. (2018)) and LCP-2021 (Shardlow et al. (2021a)) datasets.

	BERT	RoBERTa	XLNet	DistilBERT	Corpus
	Hyper-parameters (Batch size, Learning rate, Epochs, Dropout#1, Dropout#2, BiLSTM output size)				
CWI-2018	16, 2.9e-5, 5, 0.7, 0.3, 256	8, 2.5e-5, 7, 0.7, 0.3, 256	8, 2.5e-5, 7, 0.7, 0.3, 256	8, 2.5e-5, 7, 0.7, 0.3, 256	News
	8, 2.8e-5, 5, 0.7, 0.3, 256	8, 3.0e-5, 7, 0.7, 0.3, 256	16, 2.8e-5, 7, 0.7, 0.3, 256	8, 2.6e-5, 7, 0.7, 0.3, 256	WikiNews
	8, 2.8e-5, 5, 0.7, 0.3, 256	8, 2.5e-5, 7, 0.7, 0.3, 256	16, 2.5e-5, 7, 0.7, 0.3, 256	8, 2.5e-5, 7, 0.7, 0.3, 256	Wikipedia
LCP-2021	8, 1.0e-5, 5, 0.7, 0.3, 256	8, 2.5e-5, 5, 0.5, 0.2, 256	8, 2.5e-5, 5, 0.7, 0.3, 256	8, 2.5e-5, 5, 0.7, 0.3, 256	SWIs
	16, 2.8e-5, 5, 0.7, 0.3, 128	16, 3.0e-5, 10, 0.5, 0.2, 128	8, 2.8e-5, 7, 0.5, 0.2, 256	8, 2.5e-5, 10, 0.5, 0.2, 256	MWEs

are reported in Table 3. Besides, we used a 1-dimensional max-pooling layer with kernel size 2 and the hidden units size of the dense layer set to 512. Moreover, we used the Sigmoid activation function at the end of our BiLSTM-based regressor architecture to predict the complexity score. During training, we saved our model based on the best Pearson correlation score by evaluating on the validation set. We set up a torch seed using the torch.manual_seed (5) in order to acquire a consistent and reproducible performance. The rest of the parameters were set to their default values unless otherwise mentioned. Since the nature of the Wikipedia and Wikinews corpus of the CWI-2018 dataset is similar, we have aggregated them during the training phase of our model. We followed the default dataset setting for the rest of the cases.

4.3. Evaluation Metrics

In this experiment, we considered various standard evaluation measures including the Pearson correlation (R) (Virtanen et al. (2020)), the Spearman correlation (Rho) (Virtanen et al. (2020)), the mean absolute error (MAE) (Pedregosa et al. (2011)), the mean squared error (MSE) (Pedregosa et al. (2011)), and the R-squared (R^2) (Pedregosa et al. (2011)). Following the benchmark of CWI-2018 (Yimam et al. (2018)) and LCP-2021 (Shardlow et al. (2021a)) shared tasks, mean absolute error and Pearson correlation are used as the primary evaluation measure for these tasks, respectively. We also report the results based on other mentioned evaluation measures.

Pearson correlation score measures the context learning efficacy of the lexical complexity prediction (LCP) task. Spearman's correlation coefficient measures the strength of a monotonic relationship that shows the data has to be monotonically related. We use MAE to measure how close the system can predict the gold scores. MSE is a measure that determines a fitted line is how close to the data points where the squaring is critical to reducing the complexity with negative signs. To determine how well the model fits the data we use the R-squared measure. A higher score is better for Pearson and Spearman correlation and R-squared measures whereas a lower score is better for error-related measures including MAE and MSE.

4.4. Experimental Setup, Results, and Analysis

In this section, we now assess the performance of our ITRM-LCP approach. Thus, we shed light on the following research questions (RQs) related to the lexical complexity prediction (LCP) tasks from the text.

- *RQ1:* What is the effect of different integration strategies on LCP performance?
  — We fuse the scores from individual models using the arithmetic mean and blending integration strategies to improve the overall system performance. The findings are available in the following Section 4.4.1.

4.4.1. Performance of Integration Techniques (RQ1)

We utilize two integration strategies including arithmetic mean and blending as described in Section 3.3. To select the effective integration strategies (*RQ1*), we evaluate the performance of these two integration strategies using primary evaluation measure Pearson correlation on the LCP-2021 MWEs dataset as reported in Table 4. The findings show

Journal Pre-proof

Table 4
Performance (Pearson scores (higher is better)) of our used integration strategies on LCP-2021 multi-word expressions (MWEs) (Shardlow et al. (2021a)) dataset.

Model with Regression	Pearson Score
Decision Tree Regressor	.8111
Passive Aggressive Regressor	.8523
Linear Regression	.8613
Support Vector Machine Regressor	.8618
Theil Sen Regressor	.8632
Bayesian Ridge Regressor	.8633
Automatic Relevance Determination Regressor	.8643
Mean-based Integration	**.8727**

Table 5
Performance (Pearson, Spearman, and R^2: higher is better; MAE and MSE: lower is better) of our baseline models and ITRM-LCP for SemEval LCP-2021 datasets.

Model	Pearson	Spearman	MAE	MSE	R^2
Performance of baseline models on LCP-2021 Dataset SWIs corpus					
HCF	.7363	.6976	.0671	.0075	.5422
Transformer	.7525	.7075	.0667	.0073	.5663
ITRM-LCP	**.8003**	**.7491**	**.0612**	**.0060**	**.6365**
Performance of baseline models on LCP-2021 Dataset MWEs corpus					
HCF	.7861	.7674	.0771	.0093	.6179
Transformer	.8361	.8291	.0719	.0082	.6874
ITRM-LCP	**.8727**	**.8538**	**.0587**	**.0059**	**.7617**

that the mean-based integration performance using equation (2) is ∼1% higher than the other blending integration techniques for the LCP-2021 MWEs corpus. It validates the efficacy of the mean-based integration strategy. Therefore, we choose this integration strategy for our ITRM-LCP system and the rest of the results are reported following this setting.

- *RQ2:* How much does our proposed approach improve the performance in comparison to other state-of-the-art lexical complexity prediction methods?
  — To validate our ITRM-LCP method effectiveness we compare it with other SOTA LCP methods. The corresponding details are available in the following Sections 4.4.2, 4.4.3, and 4.4.4.

4.4.2. Baseline Systems Design and Performance Analysis (RQ2)

The earlier lexical complexity prediction (LCP) systems are mostly low-level hand-crafted features based whereas in recent times researchers mostly applied the embedding and transformer-based deep learning features to design their LCP systems. Therefore, to design the standard baseline systems we consider both the hand-crafted features (HCF) based method and the state-of-the-art (SOTA) transformer-based method.

For our HCF-based baseline system, we extend the work reported by Nandy et al. (2021) [7]. Here, we incorporate 14 unique HCF features including transformer (BERT, RoBERTa, and DistilBERT) probability, word length, word frequency, number of syllables, corpus features (Bible, Biomed, and Europerl), WordNet (Fellbaum (2010)) features (synsets, hyponyms, and hypernyms), and Glove 50 and 100 dimension features and the total feature dimension is 162. For the transformer's probability feature, we calculate the probability score of a token using the transformer mask language modeling feature whereas, for the multi-word expressions (MWEs) task, we multiply both token probabilities to calculate as a feature. However, to calculate the WordNet features we utilize the natural language toolkit (NLTK) (Bird (2006)) library where we calculate the mean average score of the token's synsets, hyponyms,

[7] https://github.com/abhi1nandy2/CS60075-Team-2-Task-1.git

Journal Pre-proof

Table 6
Performance (Pearson, Spearman, and R^2: higher is better; MAE and MSE: lower is better) of our ITRM-LCP model for NAACL-HLT CWI-2018 and SemEval LCP-2021 datasets.

Corpus	Pearson	Spearman	MAE	MSE	R^2
CWI-2018 Dataset Performance					
News	.8970	.7822	.0404	.0067	.8046
Wikinews	.8073	.7393	.0540	.0105	.6516
Wikipedia	.7774	.7534	.0616	.0125	.6044
Average	.8273	.7583	.0520	.0099	.6868
LCP-2021 Dataset Performance					
SWIs	.8003	.7491	.0612	.0060	.6365
MWEs	.8727	.8538	.0587	.0059	.7617
Average	.8365	.8015	.0599	.0059	.6991

and hypernyms frequencies. We also utilize two different dimensions i.e. 50 and 100 of Glove feature since it is one of the widely used embedding features in natural language processing (NLP) and we want to exploit its efficacy on the LCP task. Finally, based on the extracted features, we use the boosting-based regression model XGBoost (eXtreme Gradient Boosting) (Chen and Guestrin (2016)) to estimate the complexity score.

Besides, recent studies (Bani Yaseen et al. (2021); Pan et al. (2021)) suggested that the transformer-based model learns contextual information effectively. Therefore, in our transformer-based baseline, we used the BERT transformer model with sentence-pair setting to represent contextual information of the lexical complexity prediction (LCP) task. To capture the task-specific information of LCP explicitly, we fine-tune the BERT model. We exploit a BiLSTM layer on top of the BERT transformer model for the LCP task to learn the long-term dependencies and capture the contextual information crucial for predicting the complexity score.

The results of our two baseline methods considering two corpora of the LCP-2021 dataset are presented in Table 5. The HCF-based baseline model achieves 0.7363 and 0.7861 Pearson scores on the single-word instances (SWIs) and the multi-word expressions (MWEs) corpora, whereas the transformer-based baseline model achieves 0.7525 and 0.8361 Pearson scores, respectively. This means that the transformer-based baseline model performs 2.16% and 5.98% higher than the HCF-based baseline model in terms of primary evaluation measure Pearson correlation on the SWIs and MWEs corpora, respectively. This deduced the importance of the transformer-based model in the LCP task. In contrast, our ITRM-LCP method, where we exploited the ensemble of transformer models to determine the complexity score achieved substantial improvement over both the HCF and transformer-based baselines. For the SWIs and MWEs corpora, our ITRM-LCP model outperformed the HCF baseline by 8.7% and 11% as well as outperformed the transformer-based baseline by 6.4% and 4.4%, respectively. This deduced the effectiveness of our ITRM-LCP to estimate the complexity score effectively. We also report the results based on other evaluation metrics.

4.4.3. Overall Performance Across Two Benchmark Datasets (RQ2)

The summarized results of our ITRM-LCP method considering different corpora are articulated in Table 6. The overall performance for the CWI-2018 dataset is 0.8273 and 0.0520 based on Pearson correlation and MAE, respectively. Here, our proposed method performs better for News corpora compared to the Wikinews and Wikipedia corpus. On the LCP-2021 dataset, the overall results of our ITRM-LCP method based on the Pearson score and MAE score are 0.8365 and 0.0599, respectively. Here, our proposed method performs better for MWEs corpora than the SWIs corpora. This is because MWEs contain more words in the token and, therefore, contain diverse contextual information that aids the model for better estimation of complexity compared to the SWIs.

4.4.4. Comparative Analysis with Related Methods (RQ2)

We compared the performance of our ITRM-LCP method against the current state-of-the-art methods to validate its effectiveness (*RQ2*). The top-performing systems on CWI-2018 dataset (Yimam et al. (2018)) includes DAT (Zaharia et al. (2022)), Camb (Gooding and Kochmar (2018)), TMU (Kajiwara and Komachi (2018)), ITEC (De Hertog and Tack (2018)), NILC (Hartmann and Dos Santos (2018)), and SB@GU (Alfter and Pilán (2018)). The comparative

Table 7
Comparative performance (MAE; lower is better) of ITRM-LCP model against the state-of-the-art on CWI-2018 (Yimam et al. (2018)) test set. We highlighted the best results in boldface.

Method	News	WikiNews	Wikipedia
ITRM-LCP	**.0404**	.0540	**.0616**
Top Performing System on CWI-2018 Dataset			
DAT (Zaharia et al. (2022))	.0450	**.0513**	.0671
Camb (Gooding and Kochmar (2018))	.0558	.0674	.0739
TMU (Kajiwara and Komachi (2018))	.0510	.0704	.0931
ITEC (De Hertog and Tack (2018))	.0539	.0707	.0809
NILC (Hartmann and Dos Santos (2018))	.0588	.0733	.0819
SB@GU (Alfter and Pilán (2018))	.1526	.1650	.1750
Baseline (Yimam et al. (2018))	.1127	.1053	.1112

Table 8
Comparative performance of our method against the state-of-the-art on SemEval-2021 LCP (Shardlow et al. (2021a)) test set. We highlighted the best results in boldface.

Method	Single Word Instances (SWIs)					Multi Word Expressions (MWEs)				
	Pearson	Spearman	MAE	MSE	R^2	Pearson	Spearman	MAE	MSE	R^2
ITRM-LCP	**.8003**	**.7491**	.0612	**.0060**	**.6365**	**.8727**	**.8538**	**.0587**	**.0059**	**.7617**
Top Performing Systems on LCP-2021 Dataset										
JUST-BLUE (Bani Yaseen et al. (2021))	.7886	.7369	**.0609**	.0062	.6172	-	-	-	-	-
DeepBlueAI (Pan et al. (2021))	.7882	.7425	.0610	.0061	.6210	.8612	.8526	.0616	.0063	.7389
Andi (Rotaru (2021))	.7782	.7287	.0637	.0064	.6036	.8506	.8381	.0667	.0070	.7107
DAT (Zaharia et al. (2022))	.7744	-	.0652	-	-	.8285	-	.0693	-	-
CSECU-DSG (Aziz et al. (2021))	.7716	.7326	.0632	.0066	.5909	.8311	.8153	.0678	.0077	.6825
BigGreen (Islam et al. (2021))	.7749	.7294	.0629	.0065	.5983	.7898	.7769	.0903	.0124	.4858
LAST (Bestgen (2021))	.7534	.6988	.0652	.0070	.5652	.8417	.8299	.0677	.0072	.7030
Baseline (Shardlow et al. (2021a))	.5287	.5263	.0870	.0136	.2779	.6571	.6345	.0924	.0140	.4030

results are articulated in Table 7 based on MAE which is the official measure of the NAACL-HLT CWI-2018 shared task. It shows that our ITRM-LCP model achieved 10.2% improvement on the news corpus and 8.2% improvement on the Wikipedia corpus compared to the top-performing system DAT (Zaharia et al. (2022)) domain adaptation-based transformer model.

In addition, we also evaluate the performance of our ITRM-LCP model on the CompLex dataset against the current top-performing methods. The findings are articulated in Table 8. It demonstrated that our ITRM-LCP model outperformed all other participants' systems in both the SWIs and MWEs subtasks of the LCP-2021 task. In the SWIs subtask, our ITRM-LCP obtained a 1.5% higher score compared to the top-performing system JUST-BLUE (Bani Yaseen et al. (2021)). Similarly, in the MWEs subtask, our ITRM-LCP achieved a 1.34% higher score compared to the top-performing system DeepBlueAI (Pan et al. (2021)). However, in comparison to the task baseline, our ITRM-LCP obtained a 51.37% and 32.81% performance improvement. The baseline used the log frequency from the Google Web1T corpus with linear regression. The comparative performance analysis confirms that an approach which integrates several transformer models with deep neural network (DNN) can achieve good performance for lexical complexity estimation from text across different datasets (*RQ2*). This validates the effectiveness of our method of lexical complexity estimation.

To analyze the performance of the above-mentioned related methods, we articulate the description of their system in Table 9. Top-performing participants' (De Hertog and Tack (2018); Kajiwara and Komachi (2018); Gooding and Kochmar (2018)) of CWI-2018 used various handcrafted features (HCF) including word length, WordNet-based

Journal Pre-proof

Table 9
Feature description and regression approaches used in top performing participants systems at the CWI-2018 and LCP-2021 (Shardlow et al. (2021a)) shared tasks.

Team Name	Features Description	Regression Approach
CWI-2018 Participants' systems features and regression approaches		
Camb (Gooding and Kochmar (2018))	Bag of N-grams, POS tags, dependency parsing relations, WordNet and psycholinguistic features.	Linear regression
TMU (Kajiwara and Komachi (2018))	token length, token frequency, and probability features	Random forest regressors
ITEC (De Hertog and Tack (2018))	Word length and frequency, word and character embeddings, psycholinguistics features.	LSTM
NILC (Hartmann and Dos Santos (2018))	N-grams, word length, number of syllables,average embedding of target words, psycholinguistic features, WordNet-based features.	LSTM
SB@GU (Alfter and Pilán (2018))	Word length, number of syllables, hypernyms, n-grams, frequency distribution, and POS tags.	Extra trees
LCP-2021 Participants' systems features and regression approaches		
JUST-BLUE (Bani Yaseen et al. (2021))	Sentence and token encoded using BERT and RoBERTa	Weighted averaging
DeepBlueAI (Pan et al. (2021))	Sentence and token encoded using BERT, ALBERT, RoBERTa, and ERNIE with Data augmentation	Linear regression
Andi (Rotaru (2021))	Psycholinguistic features, Glove embeddings, Word2Vec embeddings, ConceptNet NumberBatch, and ensemble features of language models	Ridge regression, gradient boosted regression
CSECU-DSG (Aziz et al. (2021))	Sentence and token encoded using BERT, RoBERTa	Arithmetic mean
BigGreen (Islam et al. (2021))	Word length, semantic, phonetic, word frequency, N-gram, syntactic, and Glove, Elmo, InferSent embeddings, BERT	Gradient boosted regression, linear regression
LAST (Bestgen (2021))	Word frequency, lexical norms, sentence length, bi-gram association	Gradient boosted regression

features, and N-grams to extract the contextual features which reduce the automaticity of the systems and a huge amount of features trouble the model to learn the contextual patterns. Besides, these systems also explored various types of regression approaches including linear regression, random forest-based regressor, long short-term memory (LSTM), and extra tree. However, these traditional regression approaches have limitations to predict the complexity scores effectively. In LCP-2021, most of the top-performing systems (Pan et al. (2021); Bani Yaseen et al. (2021)) employed transformer-based systems, though some used the HCFs based systems (Islam et al. (2021); Bestgen (2021)).

Recently (Zaharia et al. (2022)) proposed a domain adaptation-based transformer model named DAT where they used character-level BiLSTM for target word representation and transformers model for context representation. However, this model performed well in learning contextual information but was limited to learning the pair-wise dependencies between sentence and token that hurt the performance.

To overcome the ineptness of the aforementioned systems, we utilize four transformer models including BERT, RoBERTa, XLNet, and DistilBERT. Thus, our system effectively captures the diversity of contextual features compared to the HCF-based approaches. We employ a BiLSTM-based regressor on top of each transformer model that helps our ITRM-LCP model to learn the long-term dependencies as well as capture the pair-wise dependencies between sentence and token effectively. Besides, we fused the predicted complexity predictions of these four models to estimate the unified score that improves the performance of our system.

- *RQ3:* Can diverse transformer models capture better contextual features from different corpus data?
  — We incorporate several state-of-the-art (SOTA) transformer models which provide diverse contextual representations and improve the generalization ability of our proposed method. The corresponding details are available in the following Section 4.4.5.

Journal Pre-proof

Table 10
Performance (Pearson correlation, Spearman correlation, MAE, MSE, and R^2) of different experimental settings on SWIs and MWEs dataset of SemEval-2021 LCP shared task (Shardlow et al. (2021a)). We highlighted the best results in boldface.

Method	Single Word Instances (SWIs)					Multi Word Expressions (MWEs)				
	Pearson	Spearman	MAE	MSE	R^2	Pearson	Spearman	MAE	MSE	R^2
ITRM-LCP	**.8003**	**.7491**	**.0612**	**.0060**	**.6365**	**.8727**	**.8538**	**.0587**	**.0059**	**.7617**
Performance on Individual Component										
BERT†	.7842	.7305	.0654	.0068	.6149	.8511	.8448	.0636	.0068	.7244
RoBERTa†	.7673	.7251	.0671	.0074	.5887	.8500	.8179	.0672	.0072	.7223
XLNet†	.7583	.7168	.0696	.0078	.5749	.8481	.8122	.0632	.0069	.7192
DistilBERT†	.7613	.7101	.0645	.0069	.5796	.8363	.8126	.0712	.0078	.6994

Table 11
Statistical significant testing on SemEval LCP-2021 shared task's (Shardlow et al. (2021a)) SWIs and MWEs datasets using paired t-test. † indicates the statistically significant difference between ITRM-LCP and each method at (p-value < 0.05).

Method	Mean Score (p-value) (SWIs)	Mean Score (p-value) (MWEs)
ITRM-LCP	-	-
Individual Component p-value Against ITRM-LCP		
BERT	2.7972e-20†	2.1899e-3†
RoBERTa	2.9089e-42†	4.6587e-1
XLNet	5.0967e-26†	1.2217e-2†
DistilBERT	1.3825e-42†	1.0005e-6†

4.4.5. Impact of Individual Transformer Models (RQ3)

Here, we further examine the performance of our ITRM-LCP model through evaluating the performance of individual transformer models. To do this, we only keep one transformer model at a time and remove the other three models. The BiLSTM-based regressor head is added on top as usual. We evaluated the performance on LCP-2021 shared tasks (i.e. SWIs and MWEs) and the results are reported in Table 10.

To capture diverse contextual representation, we incorporate four transformer models including BERT, RoBERTa, XLNet, and DistilBERT. Such integration is crucial for learning semantic information from various domain specific corpora including News, Wikinews, Wikipedia, BioMed, European parliament proceedings, and English bible translation (*RQ3*). It demonstrated that the performance of our ITRM-LCP model on the SWIs task is 0.8003 which is a maximum of 5.54% and a minimum of 2.05% higher than the individual transformer model performances according to the primary evaluation measure Pearson score. Similarly, in the MWEs task, ITRM-LCPs performance is 0.8727 and that is a maximum of 4.35% and a minimum of 2.54% higher compared to the performances of other individual models. We also noticed that among all the four transformer models, BERT based model performed better than others for both tasks. Moreover, we report the result based on the Spearman correlation, MAE, MSE, and R^2 measures. From the results, we have seen that our model outperformed each of the individual models in all the evaluation metrics. This deduced the effectiveness of our different model integration strategies in the ITRM-LCP model to capture the benefits of the individual model.

Additionally, we perform statistical significance testing with a two-sided paired t-test at a 95% confidence level based on the performances between our ITRM-LCP and individual component variations as shown in Table 10. The findings of our significance testing are presented in Table 11. Here, † represents the statistically significant at ($p < 0.05$). It shows that our ITRM-LCP model significantly outperforms all the other variations in the SWIs task. However,

Journal Pre-proof

Table 12
Impact of other alternatives rather than BiLSTM layer into our ITRM-LCP model using MWEs dataset of LCP-2021.

Model: ITRM-LCP with Other Top Layers	Pearson	Spearman	MAE	MSE	R^2
– BiLSTM	**.8727**	**.8538**	**.0587**	**.0059**	**.7617**
– RNN	.8465	.8282	.0669	.0069	.7165
– GRU	.8434	.8277	.0653	.0070	.7114
– Linear regression	.8389	.8250	.0666	.0072	.7039
– LSTM	.8348	.8238	.0689	.0077	.6968

for the MWEs task, our ITRM-LCP model significantly outperforms all the other methods except the RoBERTa-based approach. Though our approach obtained better results than the RoBERTa-based approach but the difference in performance is not significant.

- *RQ4:* Does the deep neural network architecture with transformer models improve the learning of pairwise features of texts for the lexical complexity prediction tasks?
  — We exploit a BiLSTM-based neural network architecture on top of the transformer model for the LCP task to learn the long-term dependencies effectively and distill the required contextual information for predicting the complexity score of a token-sentence pair. The corresponding details are available in the following Section 4.4.6.

4.4.6. Impact of BiLSTM-based Regressor (RQ4)

To learn the semantic information effectively, we pass the sequence of hidden states for the whole input sequence obtained from each transformer model to a BiLSTM-based regressor layer on top as discussed in Section 3.2. BiLSTM incorporates two LSTM layer to process the input sentence forward and backward sequentially which help it learn better semantic dependency than the transformer models on the task-specific dataset. In order to enhance the learning of task-specific knowledge, we incorporate a BiLSTM layer on top of each transformer model. To validate our selection, we conduct extensive experiments on our ITRM-LCP model utilizing other popular feature learning algorithms including recurrent neural network (RNN), long short-term memory (LSTM) network, gated recurrent unit (GRU), and linear regression. Table 12 shows the experimental results which demonstrate that BiLSTM based ITRM-LCP model outperformed the other settings thus validating our selection of BiLSTM module.

Besides, to demonstrate the impact of BiLSTM-based regressor on our ITRM-LCP model, we present the comparative performances of each transformer model with and without using the BiLSTM-based regressor layer based on the LCP-2021 MWEs dataset in Figure 5. When removing the BiLSTM regressor, a fully connected layer is added on top of the transformer model to estimate the complexity score. Experimental findings demonstrate that the performance of the BERT, RoBERTa, XLNet, and DistilBERT models improved by 5.36%, 3.53%, 2.75%, and 3.45%, respectively in terms of the Pearson correlation score (*RQ4*). This deduced the importance of adding a BiLSTM-based regressor layer on top of each transformer model.

To validate the significance of adding a BiLSTM-based regressor we also perform the token-based analysis in our ITRM-LCP model (*RQ4*). In this regard, in Figure 6 we show the importance of every token in predicting the final complexity score where we focus on three crucial breaking points of our ITRM-LCP model. We analyze the output (I) after the BERT layer, (II) after the BERT with BiLSTM layer, and (III) after the final layer (i.e BERT with BiLSTM and max-pooling layer). We visualize the output of these model variants using the Captum library[8] where the input tokens are mapped to their corresponding scores and color gradients are used to visualize them. To calculate the attribution score we use the Integrated Gradients (Sundararajan et al. (2017)) algorithm. In Figure 6, we can see that the attribution score of the first model (I) is significantly lower than the others which indicates the minimal contribution of only BERT embedding into the final prediction. Moreover, the first model (I) is limited to focusing on crucial words but model (II) and model (III) improve the attribution score and focus on the crucial words which deduce the effectiveness of adding the BiLSTM-based regressor on top of the transformer model. Here, in example A, we present the findings for predicting the complexity score of the *EU competitiveness* token. Considering this context, one of the crucial words is *Europe*. We see that model (II) and model (III) addressed this token importance whereas model (I) i.e. only BERT embeddings failed to capture this context. Similarly, in examples B and C crucial words

[8]https://github.com/pytorch/captum

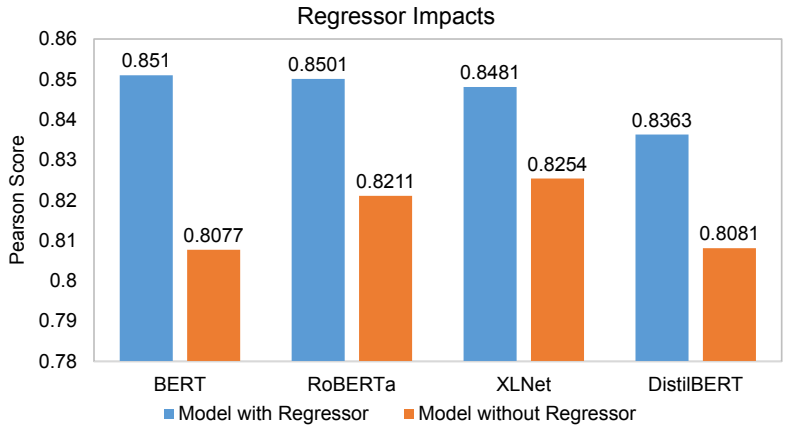


Figure 5: Impact of BiLSTM-based regressor on transformer models using MWEs dataset of LCP-2021.

Figure 6: Visualization of word contributions based on different settings and our proposed ITRM-LCP method for predicting complexity scores.

are *bound before* and *conspired*, respectively, to predict the complexity score of token *chief priests*. Here, also we see that BERT with BiLSTM-based regressor (model (III)) layer provides comparatively better attention on crucial words rather than the other two variants. Hence, this visual analysis validates the selection of adding the BiLSTM-based regressor layer on top of each transformers in our ITRM-LCP model.

- *RQ5:* Can our proposed approach detect domain-specific inherent meanings of tokens in phrasal texts?
  — For effective adaptation to domain-specific words, we fine-tune diverse transformer models. The corresponding details are available in the following Section 4.4.7.

4.4.7. Genre Based Comparison (RQ5)

In the LCP task, it is important to consider data from different domains including scientific, biomedical, political, and religious to ensure the generalizability of an LCP system (Shardlow et al. (2021b)). Hence, genre-based analysis is important for this task (*RQ5*). Considering this, CWI-2018 shared task employed the genre-wise evaluation strategy. Since the LCP-2021 shared task focused on single and multi-word evaluation, we conducted a genre-wise comparative analysis using the LCP-2021 MWEs dataset. The experimental findings are illustrated in Figure 7. It shows that our system obtained the top performance for the Biomed genre though we didn't employ any domain-specific methodologies for this genre. The performance for the Bible genre is also satisfactory. However, our model obtained comparatively poor performance for the Europarl genre. Further observation revealed that in the Europarl genre, some target tokens are chosen from the short form of the word(s) like EU (European Union). Besides, in some cases, context is based on the speaker's talk time in the European Parliament and the law number of the European Union constitution. Thus, our model failed to capture such context or was sometimes misled during the training phase.

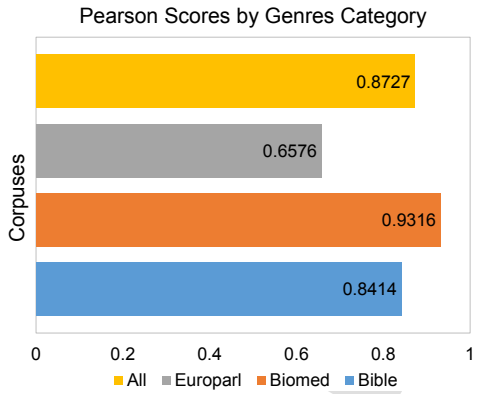


Figure 7: Comparative analysis among different genres of SemEval-2021 LCP MWEs dataset based on Pearson score.

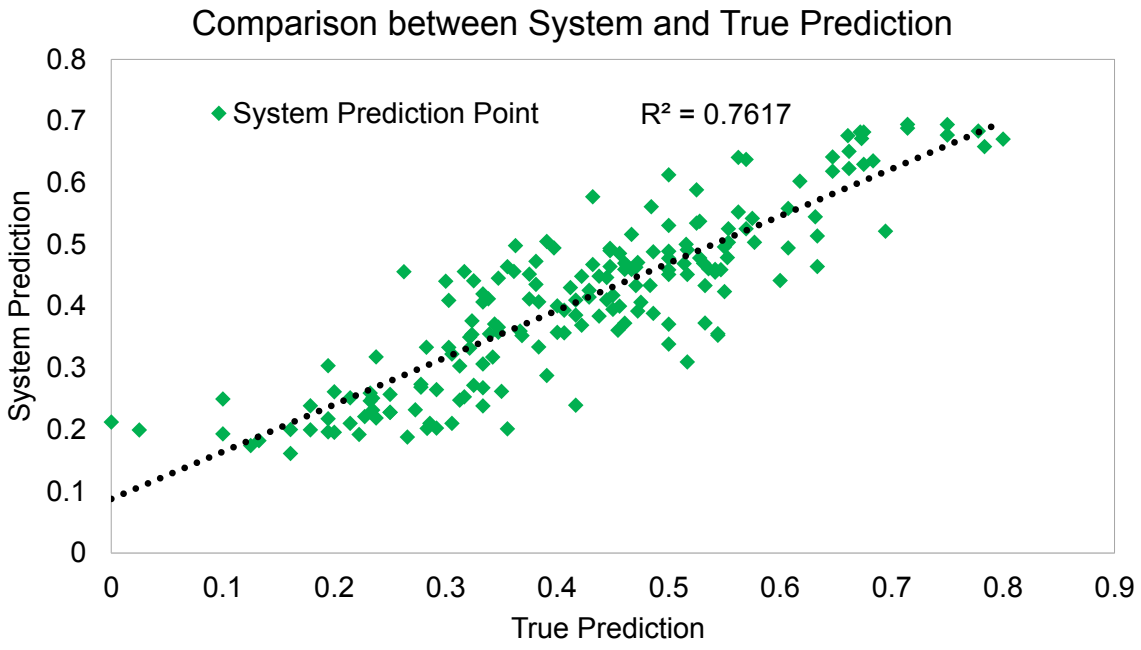


Figure 8: Correlation score between system and true prediction using MWEs dataset of SemEval-2021.

Journal Pre-proof

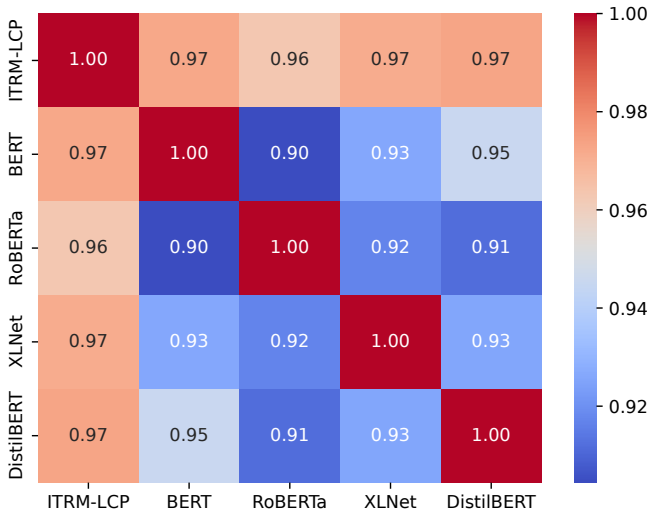


Figure 9: Correlation matrix among the predictions of our proposed ITRM-LCP model and BERT, RoBERTa, XLNet, and DistilBERT individual transformer-based models on the MWEs task of LCP-2021 dataset.

4.5. Discussion

To estimate the effectiveness of our ITRM-LCP model, we present a scatter plot diagram between the true predictions and the predictions of our system using the LCP-2021 MWEs dataset in Figure 8. It indicates that our system strongly fits the LCP task because of a higher R^2 value of 0.7617. Besides, from the scatter plot, we can observe that the predicted data points of our system are placed nearly to the linear line which indicates the usefulness of our system in the LCP task. In Figure 9, we depict the correlation among the BERT, RoBERTa, XLNet, DistilBERT, and our proposed ITRM-LCP model's predictions of the MWEs task. Here, the ITRM-LCP model highly correlates with other individual transformer-based models whereas those models' individual predictions are not so closely correlated. This deduces the effectiveness of our proposed ensembling approach.

To further examine the aptness of our ITRM-LCP model, we perform the computation time analysis, feature analysis of the baseline system, research analysis, and error analysis. We use the LCP-2021 MWEs corpora dataset to perform these comparisons.

4.5.1. Computation Time

Now, we discuss the computation time for training and testing of our proposed ITRM-LCP method. We used Google Colab's (Bisong and Bisong (2019)) GPU machine to implement our method. The total training time for the LCP-2021 MWEs corpus of our ITRM-LCP method is 23.23 minutes. This indicates that our method is able to learn from the training data and optimize its parameters in a considerable amount of time. The prediction time for a single instance and loading base models (transformers with BiLSTM-based regressors), ITRM-LCP requires 45.94 seconds. The prediction time for a single instance, when base models are already loaded into memory, is only 0.11 seconds. This indicates that the computational cost of making predictions with the ITRM-LCP is relatively low once base models are loaded.

4.5.2. Feature Analysis of the Baseline System

For our HCF-based baseline system, we have extracted 162 features. The detailed description of these features is already described in section 4.4.2. To analyze the contribution of these hand-engineered features on the LCP task, we depict a feature importance graph in Figure 10, where we plot the most important 15 out of 162 features according to their F-score. We conducted this experiment using the LCP-2021 multi-word expressions (MWEs) dataset. In Figure 10, we have seen that the contextual transformer features have the highest contribution to tackling the challenges of the LCP task. Besides, this plot also deduces that HCF-based features including the number of syllables, corpus features, word frequency, and WordNet features have limited contributions to model performance.

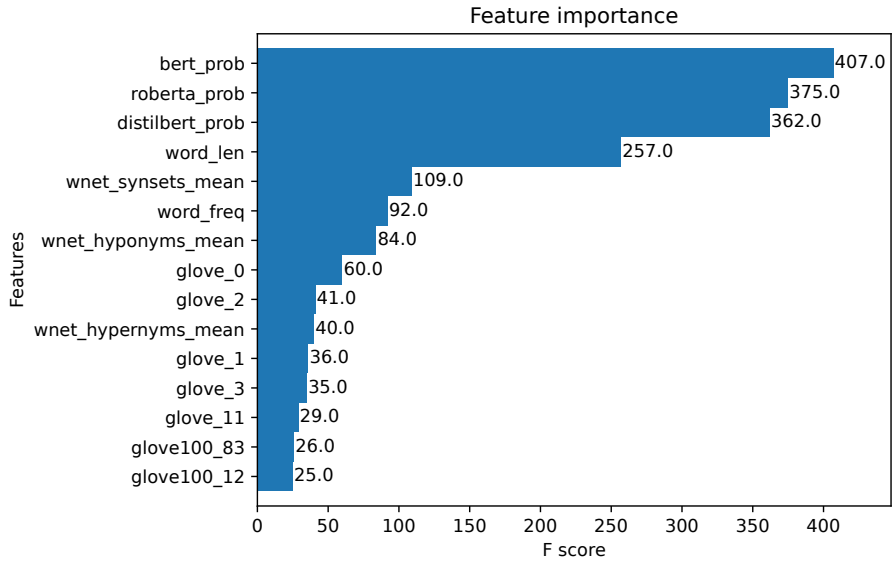


Figure 10: Feature importance graph of HCFs based baseline model on MWEs train dataset.

These findings actually motivate us to shift from HCF feature-based baseline model and develop an effective LCP model through exploiting an ensemble of the transformers model.

4.5.3. Research Analysis

In Section 4.4, we presented five research questions that drove this work. In the first question (*RQ1*), we focused on the effective integration techniques for leveraging four transformer models including BERT, RoBERTa, XLNet, and DistilBERT to determine the final prediction. We showed the comparative performance between the arithmetic mean and linear regression techniques in Section 4.4.1 where the arithmetic mean Pearson score is on average ∼1% higher than the various blending integration techniques on the LCP-2021 dataset for our proposed method. Next, in the second question (*RQ2*), we provided a performance comparison of our ITRM-LCP method with other state-of-the-art methods (i.e. DAT, CAMB, DeepBlueAI etc.) as described in Sections 4.4.2 and 4.4.4. The comparative analysis based on the CWI-2018 dataset showed that our proposed ITRM-LCP method achieved 10.2% improvement on the News corpus and 8.2% improvement on the Wikipedia corpus compared to the top-performing method. Besides, compared with the top-performing method on the LCP-2021 dataset our proposed ITRM-LCP method led up to 1.5% and 1.34% improvement for single and multi-word complexity prediction tasks, respectively. The third question (*RQ3*) was concerned with capturing better contextual features from different corpus data. In this regard, we incorporated four transformer models for extracting diverse contextual features. The analysis results of Section 4.4.5 demonstrated the efficacy of exploiting diverse transformer models in the LCP task. Some prior LCP methods leveraged multiple transformers models and performed well to learn contextual information but are limited to learning the pair-wise dependencies between sentence and token that hurt the performance. Considering this, we concentrate on the further improvement of the performance of the individual transformer model. We placed a BiLSTM-based regressor cap on top of each transformer model which helped the model to take full advantage of the forward and backward input features to learn the inter and intra-relational structure of the token-sentence pair (*RQ4*). The impact of BiLSTM-regressor is presented in Section 4.4.6. Finally, the fifth question (*RQ5*) pertained to identifying domain-specific meaning from text. The LCP-2021 dataset contains data from three different domains including Biomedical text, European parliament proceedings, and English bible translation where it is challenging for a method to equally address domain-specific contexts of tokens and sentences. In Section 4.4.7, we presented a genre-based analysis that demonstrated the efficacy of our proposed ITRM-LCP method for capturing domain-specific contexts from token-sentence pairs.

Journal Pre-proof

Table 13
Comparative analysis of the predicted complexity score of the highlighted single word/multi-words in the corresponding sentences using BERT, RoBERTa, XLNet, DistilBERT, and our proposed ITRM-LCP system against the gold complexity score for some samples from both datasets (CWI-2018 and LCP-2021).

Sentence	Genre	Gold	BERT	RoBERTa	XLNet	DistilBERT	ITRM-LCP
SWIs Example							
E#1 However I will not tear away all the kingdom; but I will give one tribe to your son, for David my servant's sake, and for Jerusalem's sake which I have chosen.	Bible	.1875	.1703	.1995	.1944	.1955	.1899
E#2 loxP sites with different sequences were generated to overcome this problem, but these sites also underwent intramolecular recombination, making RMCE efficient only if the replacement cassette contained a marker enabling selection of the desired recombinant (7,9â€12).	Biomed	.2236	.2154	.1857	.2306	.2626	.2236
E#3 I should like, on behalf of the European Parliament, to express our sympathy to the parents and families of the victims.	Europarl	.1875	.2033	.1546	.1971	.1987	.1884
E#4 police say, they were met by gunfire and a standoff ensued.	Wikinews	.55	.6782	.0657	.7490	.6485	.5353
MWEs Example							
E#5 It was planted in a good soil by many waters, that it might bring forth branches, and that it might bear fruit, that it might be a goodly vine.	Bible	.4375	.4757	.4949	.3299	.4939	.4486
E#6 Single strains on an HG background were created for each chromosomal region outside of MMU2, while a comprehensive panel of overlapping strains with identical donor regions on both B6 and HG backgrounds were developed for MMU2.	Biomed	.4	.3271	.4978	.4117	.3655	.4005
E#7 Developing innovation policy is crucial to EU competitiveness and our ability to keep good jobs in Europe.	Europarl	.4285	.4530	.4515	.4023	.3961	.4257

4.5.4. Error Analysis

To investigate the efficacy of our ITRM-LCP model, we articulate some examples in Table 13. The examples are taken from all the used corpora and from both the single and multi-word expressions (MWEs). We then presented the estimated complexity score of the highlighted single word/multi-words in the corresponding sentences using BERT, RoBERTa, XLNet, DistilBERT, and our proposed ITRM-LCP system. The comparative analysis of these estimated complexity scores against the gold score shows that the prediction from our proposed ITRM-LCP model is the closest to the gold compared to its other component variants.

From the Table 13 illustration, we observe a few reasons behind the erroneous prediction of the complexity score by individual components where our proposed ITRM-LCP model has predicted nearly the gold score. For instance, it is difficult to extract the contextual dimension from a short sentence as presented in E#4 and E#7. Besides, Biomed

Journal Pre-proof

corpora contain highly domain-specific words in the context thus making it difficult to estimate the perfect complexity score. However, our ITRM-LCP system overcome this limitation and predicted the complexity score of nearly the gold as shown in E#2 and E#6. It is also difficult to determine the complexity of such a token with a short form (i.e. EU). Though the predicted scores of individual transformers including BERT, RoBERTa, XLNet, and DistilBERT are shaky in comparison with the gold score, our proposed system overcomes this limitation by incorporating an effective integration strategy as discussed in Section 3.3.

5. Conclusions

In this paper, we have proposed a model for the lexical complexity prediction task where we integrated four transformer models, including BERT, RoBERTa, XLNet, and DistilBERT. Using pairwise learning of those transformer models, we have exploited the contextual relation between sentence-word pairs. We have also added a BiLSTM-based regressor layer on top of each transformer model, which improves the feature learning of each model in text-pair settings. Besides, we have applied a simple mean-based integration of the prediction of these transformer models that improved the overall system performance. Experimental findings on benchmark datasets showed that our ITRM-LCP method surpassed all state-of-the-art LCP methods. To validate the aptness of our system we discussed our proposed system from various views of angles, including BiLSTM-based regressor impact, integration impact, and genre-based comparisons. Our experimental results demonstrated that BiLSTM with a deep neural network (DNN) cap on top of diverse transformer models provide more relevant and insightful representations of the given inputs and ensures the robustness of the new representations. It is crucial for the LCP task to improve results by encouraging the models to extract more general features.

We intend to investigate two more strategies in the future. The first one is task-adaptive pre-training on the transformers models, where we need to feed relevant genre sentences into the pre-trained language models. It may help to learn genre-based information effectively to enhance the model's efficiency. The second one is to examine and design a graph neural network (GNN) model to extract effective feature embeddings. The transformer models are relatively limited in capturing the global information of large linguistic vocabulary. However, the GNN architecture overcomes this limitation by encoding the topological information and is also promising to apply in the middle of various transformer models. Moreover, we intend to employ our ITRM-LCP model on related tasks, including lexical simplification, translation, and text generation.

References

Alfter, D., Pilán, I., 2018. SB@ GU at the Complex Word Identification 2018 Shared Task, in: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 315–321.

Aroyehun, S.T., Angel, J., Alvarez, D.A.P., Gelbukh, A., 2018. Complex Word Identification: Convolutional Neural Network vs. Feature Engineering, in: Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications, pp. 322–327.

Aziz, A., Hossain, M.A., Chy, A.N., 2021. CSECU-DSG at SemEval-2021 Task 1: Fusion of Transformer Models for Lexical Complexity Prediction, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 627–631.

Aziz, A., Hossain, M.A., Chy, A.N., 2022. Enhancing the deberta transformers model for classifying sentences from biomedical abstracts, in: Proceedings of the The 20th Annual Workshop of the Australasian Language Technology Association, pp. 156–160.

Bani Yaseen, T., Ismail, Q., Al-Omari, S., Al-Sobh, E., Abdullah, M., 2021. JUST-BLUE at SemEval-2021 Task 1: Predicting Lexical Complexity using BERT and RoBERTa Pre-trained Language Models, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 661–666.

Bestgen, Y., 2021. LAST at SemEval-2021 Task 1: Improving Multi-Word Complexity Prediction Using Bigram Association Measures, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 571–577.

Bird, S., 2006. Nltk: the natural language toolkit, in: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, pp. 69–72.

Bisong, E., Bisong, E., 2019. Google colaboratory. Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners , 59–64.

Brueckner, R., Schulter, B., 2014. Social Signal Classification Using Deep BLSTM Recurrent Neural Networks, in: 2014 IEEE International Conference on Coustics, Speech and Signal Processing (ICASSP), IEEE. pp. 4823–4827.

Chatzimparmpas, A., Martins, R.M., Kucher, K., Kerren, A., 2021. Empirical study: visual analytics for comparing stacking to blending ensemble learning, in: 2021 23rd International Conference on Control Systems and Computer Science (CSCS), IEEE. pp. 1–8.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794.

Chy, A.N., Siddiqua, U.A., Aono, M., 2021. Exploiting transfer learning and hand-crafted features in a unified neural model for identifying actionable informative tweets. Journal of Information Processing 29, 16–29.

De Hertog, D., Tack, A., 2018. Deep Learning Architecture for Complex Word Identification, in: Thirteenth Workshop of Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics (ACL); New Orleans, Louisiana. pp. 328–334.

Journal Pre-proof

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL:HLT), pp. 4171–4186.

Fellbaum, C., 2010. Wordnet, in: Theory and applications of ontology: computer applications. Springer, pp. 231–243.

Gooding, S., Kochmar, E., 2018. CAMB at CWI Shared Task 2018: Complex Word Identification with Ensemble-based Voting, in: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 184–194.

Gooding, Sian and Kochmar, Ekaterina, 2019. Complex Word Identification as a Sequence Labelling Task, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1148–1153.

Hartmann, N., Dos Santos, L.B., 2018. Nilc at CWI 2018: Exploring Feature Engineering and Feature Learning, in: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 335–340.

Islam, A., Ma, W., Vosoughi, S., 2021. BigGreen at SemEval-2021 Task 1: Lexical Complexity Prediction with Assembly Models, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 667–677.

Kajiwara, T., Komachi, M., 2018. Complex Word Identification Based on Frequency in a Learner Corpus, in: Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications, pp. 195–199.

Koren, Y., 2009. The bellkor solution to the netflix grand prize. Netflix prize documentation 81, 1–10.

Kuru, O., 2016. Ai-ku at SemEval-2016 Task 11: Word Embeddings and Substring Features for Complex Word identification, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 1042–1046.

Kwon, S., Oh, D., Ko, Y., 2021. Word sense disambiguation based on context selection using knowledge-based word similarity. Information Processing & Management 58, 102551.

Li, W., Suzuki, E., 2021. Adaptive and hybrid context-aware fine-grained word sense disambiguation in topic modeling based document representation. Information Processing & Management 58, 102592.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 .

Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 .

Malmasi, S., Dras, M., Zampieri, M., 2016. Ltg at SemEval-2016 Task 11: Complex Word Identification with Classifier Ensembles, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 996–1000.

Malmasi, S., Zampieri, M., 2016. Maza at semeval-2016 task 11: Detecting Lexical Complexity Using a Decision Stump Meta-classifier, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 991–995.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 .

Nandy, A., Adak, S., Halder, T., Pokala, S.M., 2021. cs60075_team2 at SemEval-2021 task 1 : Lexical complexity prediction using transformer-based language models pre-trained on various text corpora, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Association for Computational Linguistics, Online. pp. 678–682. URL: https://aclanthology.org/2021.semeval-1.87, doi:10.18653/v1/2021.semeval-1.87.

Nisioi, S., Štajner, S., Ponzetto, S.P., Dinu, L.P., 2017. Exploring neural text simplification models, in: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers), pp. 85–91.

Paetzold, G., Specia, L., 2016a. Semeval 2016 task 11: Complex Word Identification, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 560–569.

Paetzold, G., Specia, L., 2016b. Sv000gg at SemEval-2016 Task 11: Heavy Gauge Complex Word Identification with System Voting, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 969–974.

Pan, C., Song, B., Wang, S., Luo, Z., 2021. DeepBlueAI at SemEval-2021 Task 1: Lexical Complexity Prediction with A Deep Ensemble Approach, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 578–584.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python . Journal of Machine Learning Research 12, 2825–2830.

Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

Rivas Rojas, K., Alva-Manchego, F., 2021. Iapucp at semeval-2021 task 1: Stacking fine-tuned transformers is almost all you need for lexical complexity prediction, Association for Computational Linguistics.

Rotaru, A., 2021. ANDI at SemEval-2021 Task 1: Predicting Complexity in Context Using Distributional Models, Behavioural Norms, and Lexical Resources, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 655–660.

Rozi, E., Iyer, N., Chi, G., Choe, E., Lee, K.J., Liu, K., Liu, P., Lack, Z., Tang, J., Chi, E.A., 2021. Stanford MLab at SemEval-2021 Task 1: Tree-Based Modelling of Lexical Complexity using Word Embeddings, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 688–693.

Saggion, H., 2017. Automatic Text Simplification. Synthesis Lectures on Human Language Technologies 10, 1–137.

Saggion, H., et al., 2018. LaSTUS/TALN at Complex Word Identification (CWI) 2018 Shared Task, in: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 159–165.

Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2019. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. arXiv preprint arXiv:1910.01108 .

Sanjay, S., Soman, K., et al., 2016. Amritacen at SemEval-2016 Task 11: Complex Word Identification Using Word Embedding, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 1022–1027.

Shardlow, M., Evans, R., Paetzold, G.H., Zampieri, M., 2021a. SemEval-2021 Task 1: Lexical Complexity Prediction, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 1–16.

Shardlow, M., Evans, R., Zampieri, M., 2021b. Predicting Lexical Complexity in English Texts. arXiv preprint arXiv:2102.08773 .

Journal Pre-proof

Shardlow, Matthew, 2014. Out in the Open: Finding and Categorising Errors in the Lexical Simplification Pipeline, in: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pp. 1583–1590.

Shirude, N., Mukherjee, S., Shandhilya, T., Mukherjee, A., Modi, A., 2021. IITK@LCP at SemEval-2021 Task 1: Classification for Lexical Complexity Regression Task, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 541–547.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15, 1929–1958.

Sundararajan, M., Taly, A., Yan, Q., 2017. Axiomatic attribution for deep networks, in: International conference on machine learning, PMLR. pp. 3319–3328.

Vanderwende, L., Suzuki, H., Brockett, C., Nenkova, A., 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. Information Processing & Management 43, 1606–1618.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is All you Need, in: Advances in neural information processing systems, pp. 5998–6008.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, 261–272. doi:10.1038/s41592-019-0686-2.

Watanabe, W.M., Junior, A.C., Uzêda, V.R., Fortes, R.P.d.M., Pardo, T.A.S., Aluísio, S.M., 2009. Facilita: Reading Assistance for Low-literacy Readers, in: Proceedings of the 27th ACM international conference on Design of communication, pp. 29–36.

Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S., et al., 2020. Transformers: State-of-the-art Natural Language Processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V., 2019. Xlnet: Generalized Autoregressive Pretraining for Language Understanding. Advances in neural information processing systems 32.

Yimam, S.M., Biemann, C., Malmasi, S., Paetzold, G., Specia, L., Štajner, S., Tack, A., Zampieri, M., 2018. A Report on the Complex Word Identification Shared Task 2018, in: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 66–78.

Yu, Z., Ramanarayanan, V., Suendermann-Oeft, D., Wang, X., Zechner, K., Chen, L., Tao, J., Ivanou, A., Qian, Y., 2015. Using bidirectional lstm recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech, in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), IEEE, Arizona, USA. pp. 338–345.

Yuan, Z., Tyen, G., Strohmaier, D., 2021. Cambridge at semeval-2021 task 1: An ensemble of feature-based and neural models for lexical complexity prediction, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 590–597.

Zaharia, G.E., Cercel, D.C., Dascalu, M., 2020. Cross-Lingual Transfer Learning for Complex Word Identification, in: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), IEEE. pp. 384–390.

Zaharia, G.E., Smădu, R.A., Cercel, D., Dascalu, M., 2022. Domain adaptation in multilingual and multi-domain monolingual settings for complex word identification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 70–80.

Zaman, F., Shardlow, M., Hassan, S.U., Aljohani, N.R., Nawaz, R., 2020. Htss: A novel hybrid text summarisation and simplification architecture. Information Processing & Management 57, 102351.

Zampieri, M., Tan, L., van Genabith, J., 2016. Macsaar at SemEval-2016 Task 11: Zipfian and Character Features for Complex Word Identification, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 1001–1005.

Abdul Aziz has recently completed his B.Sc. (Engg.) degree from the Department of Computer Science and Engineering, University of Chittagong, Bangladesh. His research interests include multimodal NLP, multilingual and low-resource NLP, text simplification, relation extraction, opinion mining, social computing, lexical semantics, computational linguistics, and natural language processing.

Md. Akram Hossain received his B.Sc. (Engg.) degree from the Department of Computer Science and Engineering at the University of Chittagong, Bangladesh, in 2023. His research interests include text simplification, information extraction, opinion mining, multimodal NLP, multilingual and low-resource NLP, social computing, lexical semantics, deep learning, and natural language processing.

Journal Pre-proof

Abu Nowshed Chy received his B.Sc. (Hons.) degree from the University of Chittagong, Chittagong, Bangladesh in 2012 and the M.Eng. and Ph.D. degrees from the Toyohashi University of Technology, Toyohashi, Japan in 2016 and 2019. He is currently working as an Assistant Professor at the Graduate School of Computer Science and Engineering Department, University of Chittagong, Bangladesh. His research interests include multimodal microblog information retrieval, opinion mining, natural language processing, computational social science, crisis informatics, and deep learning.

Md Zia Ullah is a Lecturer at the school of computing, engineering, and the built environment at Edinburgh Napier University, UK. He received his B.Sc. (Hons.) degree from the University of Chittagong, Bangladesh, in 2010 and his M.Eng. and PhD from the Toyohashi University of Technology, Japan, in 2013 and 2016, respectively. He was a Post-doc researcher in Information retrieval and Machine learning at the Universite de Toulouse, France. His research interests include Adaptive information retrieval, Query performance prediction, Intent mining, Natural language processing, Applied machine learning, and Deep learning.

Masaki Aono received the BS and MS degrees from the Department of Information Science from the University of Tokyo, Tokyo, Japan, and the PhD degree from the Department of Computer Science at Rensselaer Polytechnic Institute, New York. He was with the IBM Tokyo Research Laboratory from 1984 to 2003. He is currently a professor at the Graduate School of Computer Science and Engineering Department, Toyohashi University of Technology. His research interests include text and data mining for massive streaming data, and information retrieval for multimedia including 2D images, videos, and 3D shape models. He is a member of the ACM and IEEE Computer Society. He has been a Japanese delegate of the ISO/IEC JTC1 SC24 Standard Committee since 1996.

Journal Pre-proof

Journal Pre-proof

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Journal Pre-proof