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

Name: Bibina M T
Roll No: 16
Reg No: VDA22MCA-2016
Guide: Ms. Rajalekshmy K D

ABSTRACT

- Lexical Complexity Prediction (LCP) determines how complex words or phrases are in a sentence.
- LCP helps improve language translation, readability assessment, and text generation.
- Challenges include technical words, grammar complexity, polysemy (multiple meanings), and word dependencies.
- The paper proposes ITRM-LCP, a model that integrates multiple transformer models for better complexity prediction.
- Fine-tunes transformer models with text-pair data to extract diverse contextual features.
- Uses a bidirectional LSTM-based regressor to capture long-term dependencies.
- Aggregates predictions from multiple models to get a final complexity score.
- Evaluated on CWI-2018 and SemEval LCP-2021 datasets.

INTRODUCTION

- Text Simplification makes complex sentences easier to read using simpler words.
- It helps children, non-native speakers, and people with reading disabilities.
- Also useful for text summarization, machine translation, and text generation.

Lexical Simplification (LS)

- A part of text simplification that replaces complex words with simpler alternatives.

Follows four steps:

1. Complex Word Identification (CWI) – Detects difficult words.
2. Substitution Generation – Finds simpler alternatives.
3. Word Sense Disambiguation – Ensures correct word meaning.
4. Synonym Ranking – Ranks substitutes by simplicity.

Lexical Complexity Prediction (LCP)

- Determines how complex a word or phrase is in a sentence.
- Helps in choosing simpler words for text simplification.

Challenges:

- Domain-specific words.
- Complex grammar.
- Context-dependent word meanings.

Existing Approaches in LCP

- Early systems used morphological, lexical, and semantic features, but lacked context awareness.
- Recent transformer-based models (DeepBlueAI, Rivas Rojas, Yuan et al.) improved performance by capturing better context.

Limitations:

1. Most models only use transformer final layers without extra neural networks.
2. Pairwise relationships (word-sentence context) are not well captured.

Proposed Model: ITRM-LCP

- Integrated Transformer Regressor Model (ITRM) for LCP using multiple fine-tuned transformers.
- BiLSTM-based Regressor added on top of transformers to capture long-term dependencies.
- Various integration strategies tested to find the most effective method.
- Experimental evaluation using benchmark datasets shows improved performance over state-of-the-art methods.

RELATED TOPIC

1. Complex Word Identification (CWI):
   - A crucial part of Lexical Simplification (LS).
   - Early methods relied on handcrafted features (e.g., word length, frequency).
   - Later methods used word embeddings and deep learning.
   - Recent approaches leverage transformer models for better results.
2. Handcrafted Features (HCF) & Machine Learning Approaches:
   - Used in early CWI models (e.g., SVM, decision trees).
   - Features included n-grams, word length, syntax, and semantics.
   - Hard to generalize across different tasks and domains.
3. Word Embeddings & Deep Learning Approaches:
   - Used embeddings like Word2Vec, GloVe, and ELMo.
   - Deep learning methods like CNNs and LSTMs improved performance.
   - Still struggled with capturing full contextual meaning.

4. Transformer Models for LCP:

- BERT and its variants (RoBERTa, ALBERT, DistilBERT, XLNet) performed well.
- These models capture contextual meaning and long-term dependencies.
- Some systems used data augmentation and ensemble learning for better accuracy.

5. Limitations & Motivation for Current Work:

- Many models focus on either single-word or multi-word tokens but not both.
- Pre-trained transformers alone may not fully capture pairwise relationships.
- Effective ensemble strategies can improve contextual learning for better LCP results.

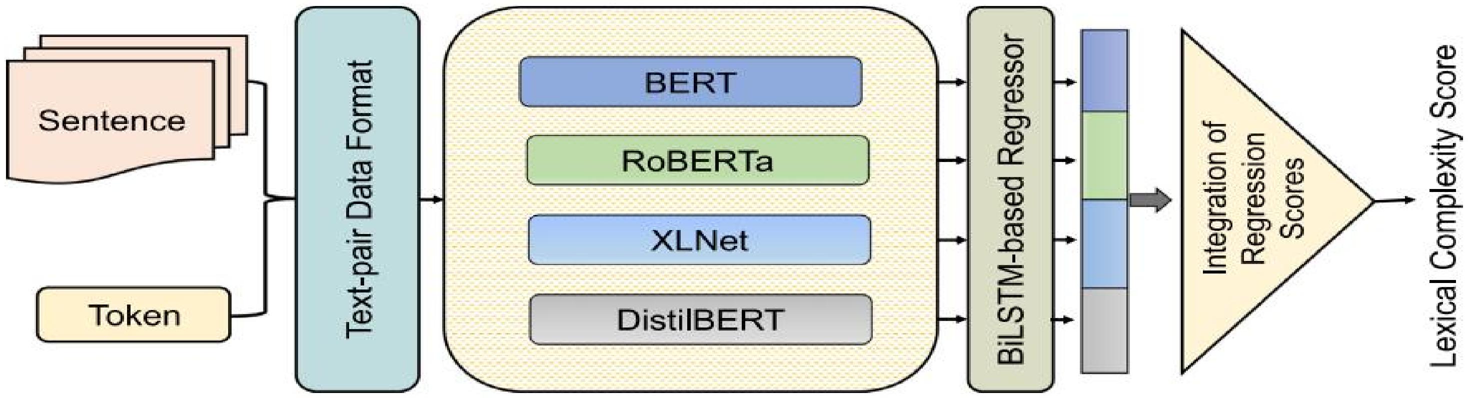

Fig. 1. Schematic diagram of our proposed ITRM-LCP system. Transformer models are tuned on pairwise settings of sentences and tokens to generate the contextualized vectors. A BiLSTM-regressor module is plugged on the top of each transformer to enhance the feature learning representations. Finally, regression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

PROPOSED METHOD

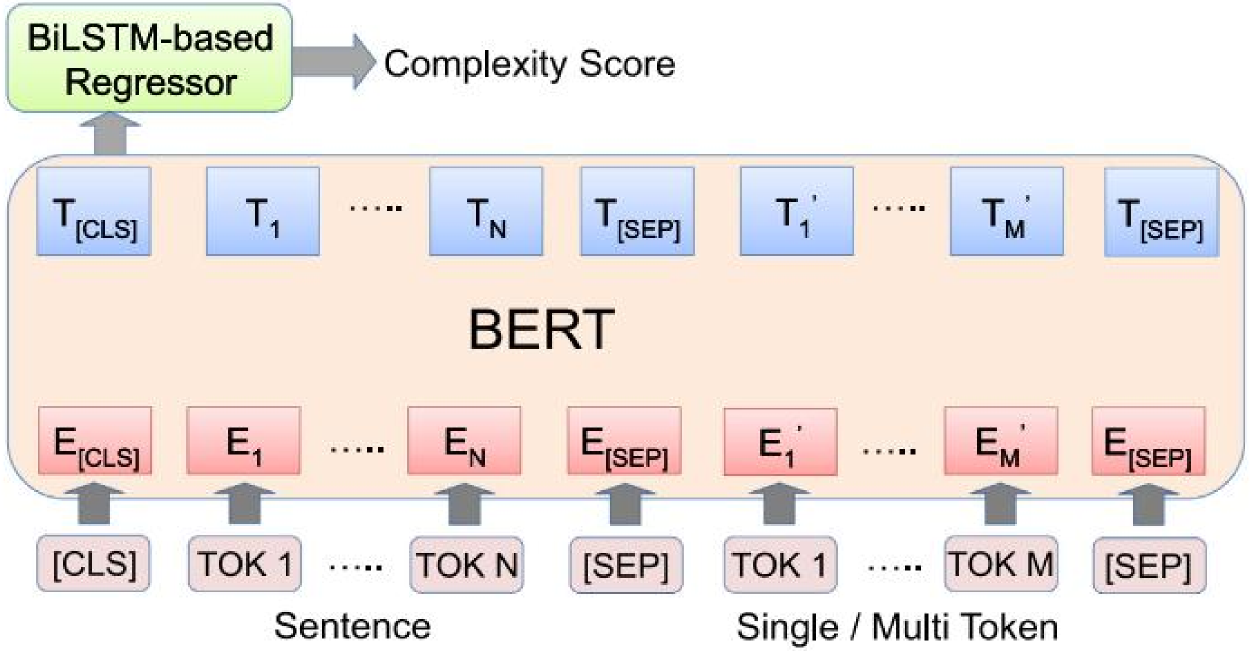

Fig. 2. To employ pairwise settings of an input representation, the tokenizer packed the sentence and token as a single sequence. Then, this input sequence ([CLS] Sentence [SEP] Token [SEP]) feeds into the BERT model. The last layer's hidden states vector output of the BERT is passed to the BiLSTM-regressor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

1. Text-Pair Regression Task
   - Input: Target word and sentence as a single packed sequence
   - Output: Continuous lexical complexity score.

2. Transformer-Transformer Models for Contextual Features:
    - Uses BERT, RoBERTa, XLNet, and DistilBERT for contextual embeddings.
    - Fine-tuned on domain-specific datasets
3. BiLSTM-Based Regressor
    - Applied on top of transformers to capture sequential dependencies.
    - Uses forward and backward LSTMs for context-aware predictions.
    - Includes max-pooling, dropout, and AdamW optimizer to enhance performance.
4. Integration of Model Predictions:
    - Mean-based Fusion: Computes the arithmetic mean of predicted scores.
    - Blending Integration: Uses base models' predictions as features for a meta-model (e.g., decision tree, linear regression, SVR, Bayesian ridge, etc.) to refine final scores.
5. Training Process:
    - BERT's text-pair training approach is used.
    - Loss function: Mean Squared Error (MSE).
    - Dropout prevents overfitting.
    - AdamW optimizer improves weight updates.

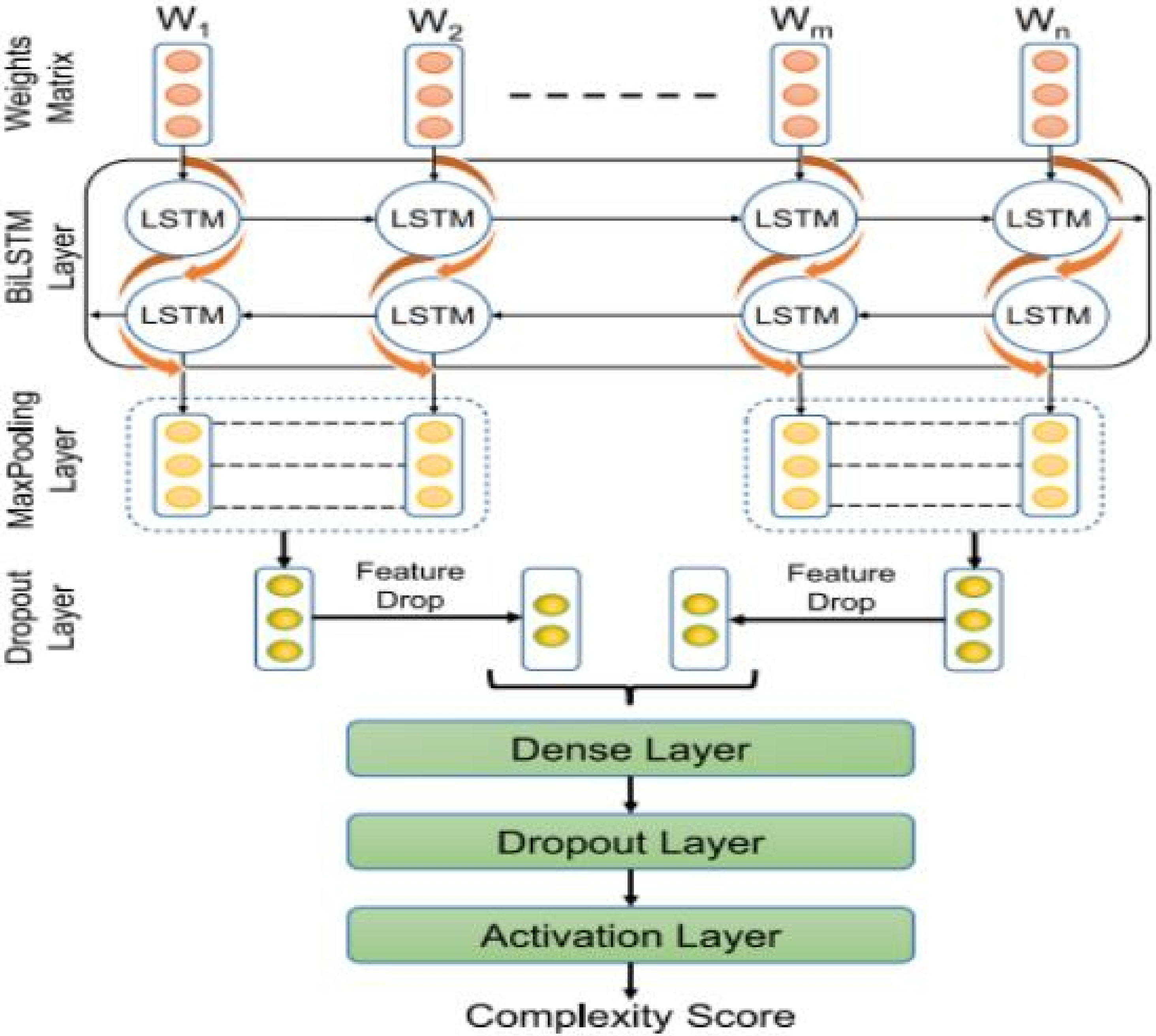

Fig. 3. Regressor architecture of our ITRM-LCP model. The features vector of the transformer passes to the BiLSTM layer as an input to learn context-based semantic association information. The MaxPooling layer filters the top features from the features vector. Later, two Dropout layers and a Linear layer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

EXPERIMENTS AND EVALUATION

4.1. Dataset

- CWIG3G2 Dataset: Used for the NAACL-HLT-2018 CWI task. It includes texts from News, Wikinews, and Wikipedia, annotated by native and non-native English speakers. The dataset contains single-word (SWIs) and multi-word expressions (MWEs).
- CompLex Dataset: Used for the SemEval-2021 LCP task. It includes texts from three domains: Bible, Biomed, and Europarl. The task involves predicting the complexity of SWIs and MWEs (limited to two words).

4.2. Model Configuration

- ITRM-LCP Model: Uses four pre-trained transformer models (BERT, RoBERTa, XLNet, DistilBERT) fine-tuned for the task. A BiLSTM-based regressor is added to each transformer to capture long-term dependencies and reduce overfitting.
- Training: Done on Google Colab using GPU. Hyperparameters like batch size, learning rate, and epochs were fine-tuned using grid search.

4.3. Evaluation Metrics

- Metrics used: Pearson correlation (R), Spearman correlation (Rho), Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2).
- Primary metrics: Pearson correlation for LCP-2021 and MAE for CWI-2018.

4.4. Experimental Results

- Integration Strategies (RQ1): Arithmetic mean integration performed better than blending, improving performance by ~1% on the LCP-2021 MWEs dataset.
- Baseline Systems (RQ2):
  - HCF-based baseline (handcrafted features) achieved Pearson scores of 0.7363 (SWIs) and 0.7861 (MWEs).
  - Transformer-based baseline (BERT) achieved 0.7525 (SWIs) and 0.8361 (MWEs).
  - ITRM-LCP outperformed both baselines, achieving 8.7% and 11% improvement over HCF, and 6.4% and 4.4% over the transformer baseline.

- Overall Performance (RQ2):
  - CWI-2018: Pearson correlation of 0.8273, MAE of 0.0520.
  - LCP-2021: Pearson correlation of 0.8365, MAE of 0.0599.
- Comparative Analysis (RQ2): ITRM-LCP outperformed state-of-the-art methods on both CWI-2018 and LCP-2021 datasets.
- Impact of Individual Transformers (RQ3): ITRM-LCP performed better than individual transformer models (BERT, RoBERTa, XLNet, DistilBERT) by 2.05% to 5.54% on SWIs and 2.54% to 4.35% on MWEs.
- Impact of BiLSTM (RQ4): Adding a BiLSTM-based regressor improved performance by 2.75% to 5.36% across transformer models.
- Genre-Based Comparison (RQ5): ITRM-LCP performed best on Biomed and Bible genres but struggled with Europarl due to domain-specific challenges like abbreviations (e.g., EU).

4.5. Discussion

- Scatter Plot Analysis: ITRM-LCP predictions closely matched true values, with a high R^2 value of 0.7617.
- Computation Time: Training took 23.23 minutes, and prediction time was 0.11 seconds per instance after loading models.
- Feature Analysis: Transformer-based features contributed more to performance than handcrafted features like word frequency or syllables.
- Error Analysis: ITRM-LCP performed well on short sentences and domain-specific terms, though challenges remained with abbreviations and highly specialized vocabulary.

CONCLUSION

1. Proposed Model:
   - Integrated BERT, RoBERTa, XLNet, and DistilBERT for Lexical Complexity Prediction (LCP).
   - Used pairwise learning to exploit sentence–word contextual relations.
2. BiLSTM-Based Regressor:
   - Added on top of each transformer model to enhance feature learning.
3. Integration Strategy:
   - Applied mean-based fusion of transformer predictions, improving performance.
4. Experimental Results:
   - ITRM-LCP outperformed state-of-the-art LCP models.
   - Analyzed model impact from multiple perspectives (BiLSTM, integration, genre-based performance).

5. Key Findings:

- BiLSTM + DNN on transformers provides better representations.
- Extracting general features is crucial for LCP tasks.

6. Future Work:

- Task-adaptive pre-training with genre-based sentences to enhance efficiency.
- Graph Neural Networks (GNNs) to capture global linguistic information.
- Expanding ITRM-LCP for lexical simplification, translation, and text generation.