IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 22, 2021, accepted February 19, 2022, date of publication February 25, 2022, date of current version March 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3154820

Attention Retrieval Model for Entity Relation Extraction From Biological Literature

PRASHANT SRIVASTAVA[1], SAPTARSHI BEJ[1,2], KRISTIAN SCHULTZ[1], KRISTINA YORDANOVA[1], AND OLAF WOLKENHAUER[1,2]

[1]Department of Systems Biology and Bioinformatics, Institute of Computer Science, University of Rostock, 18057 Rostock, Germany
[2]Leibniz-Institute for Food Systems Biology, Technical University of Munich, 85354 Freising, Germany

Corresponding author: Olaf Wolkenhauer (olaf.wolkenhauer@uni-rostock.de)

ABSTRACT Natural Language Processing (NLP) has contributed to extracting relationships among biological entities, such as genes, their mutations, proteins, diseases, processes, phenotypes, and drugs, for a comprehensive and concise understanding of information in the literature. Self-attention-based models for Relationship Extraction (RE) have played an increasingly important role in NLP. However, self-attention models for RE are framed as a classification problem, which limits its practical usability in several ways. We present an alternative framework called the Attention Retrieval Model (ARM), which enhances the applicability of attention-based models compared to the regular classification approach, for RE. Given a text sequence containing related entities/keywords, ARM learns the association between a chosen entity/keyword with the other entities present in the sequence, using an underlying self-attention mechanism. ARM provides a flexible framework for a modeller to customise their model, facilitate data integration, and integrate expert knowledge to provide a more practical approach for RE. ARM can extract unseen relationships that are not annotated in the training data, analogous to *zero-shot learning*. To sum up, ARM provides an alternative self-attention-based deep learning framework for RE, that can capture directed entity relationships.

INDEX TERMS Attention models, biological literature mining, deep learning, knowledge graphs.

I. INTRODUCTION

A. BACKGROUND

Modelling molecular and cellular processes considers an increasingly large number of interacting molecules that are relevant to the questions investigated. For a more comprehensive understanding of molecular networks, it is thus necessary to extract relationships among large numbers of biological entities, including genes, proteins, diseases, processes, phenotypes, where the details of interactions are usually extracted from literature. The rise of research studying large scale molecular interaction networks has subsequently renewed an interest in text mining of biological literature.

Text mining and Natural Language Processing (NLP)-based techniques are finding their applications in reducing the efforts of biologists to mine the biological literature for tasks like curation of large-scale models and keeping databases updated, handling problems such as Named Entity Recognition (NER) and Relationship Extraction (RE). Both rule-based and machine learning (ML) NLP approaches have been popular in this context, with multiple research and review articles examining the scope of such models in Biological Literature Mining (BLM) [15], [21]–[23].

Sequence-to-sequence models typically receive a sequence as input and generate a sequence as output. Input and output sequences can be numerical, time-dependent data or string data. Recurrent Neural Network (RNN) is a deep learning based model designed for learning from sequence data. At every learning step, RNNs take elements of a sequence as input and generate an output for that time step and updates a hidden state, that can be associated with the ''memory'' of the network. For text-based data, RNNs once used to be the state-of-the-art models. However, RNNs proved to be less effective to learn from longer sequences, that is to create associations among elements of long sequences. This means that if there is a long sequence of text (say, a long sentence) and there is an association between two words, one located very early in the sentence and the other very late, RNNs are unlikely to capture that information. LSTMs and GRUs were designed to mitigate this ''memory'' problem. The extremely popular LSTM model, for example, is designed to retain or forget information that is stored in the hidden state sequentially.

The associate editor coordinating the review of this manuscript and approving it for publication was Ravinesh C. Deo.

Since the introduction of the attention model by Bahdanau *et al.* for Machine Translation in 2015, it has found applications in a wide range of Neural Network-based architectures [16], while it received more recognition after the introduction of transformer models in 2017 [13]. Moreover, even more advanced domain specific pre-trained models like BERT and BioBERT based on the principle of self attention, are finding their application in biomedical relationship extraction problems. The ARM approach that we propose is also a self-attention-based model designed to address the issues of the common classification based RE, as discussed in Section II-C.

B. RELATED WORK

The general problem of relation extraction from text is a popular and active NLP research field. For example, Neural Relation Extraction with Edge-oriented Graphs [24], Relation Extraction from Wikipedia Using Subtree Mining [26], Relation Extraction using Sequential and Tree-structured LSTM with attention [27], etc. Now, we discuss some state-of-the-art self-attention-based models that have been proposed on using the classification approach for biological RE over the last two years.

Elangovan *et al.* draw their motivation from the fact that popular PPI databases such as IntAct contain a large amount of data on PPIs, but only 4% of these interactions are functionally annotated [9]. Such functional annotations can, however, often be found in relevant publications. Functional annotations can be very important to understand causal aspects driving biological processes. Elangovan *et al.* [9] focus on extracting functional annotations of interacting proteins, providing relevant information from text data (e.g., abstracts of publications). Conventional string matching is used to search for co-occurrences of entities (gene or protein names) in a sentence. However, this can result in the inclusion of noisy data curation.

A contrastive learning approach is implemented by Su *et al.* (2021) to improve performance of pre-trained models [12]. The training process for such models is designed such that similar input instances have ''positive'' labels whereas, dissimilar input instances are labelled as ''negative'' instances. The goal is to learn a text representation by maximising the agreement between inputs from positive pairs via a contrastive loss in the latent space, and the learned representation can then be used for relation extraction [12].

The architecture of the model proposed by Wang *et al.* takes advantage of multitask (main and auxiliary tasks) learning strategy as proposed by the authors [11]. The authors use BERT and BioBERT model, to create a meaningful vector representation of the input text adding the main RC task and auxiliary Document Triage task, a downstream Text CNN model to the model. Moreover, BiLSTM layers are also used as a downstream layer for the gene recognition auxiliary task. According to this research, introduction of the auxiliary learning tasks improves the classification performance of the main RE task [11]. Zhou *et al.* propose the Knowledge-aware Attention Network (KAN) for PPI extraction. The motivation of this work, published in 2019, is the fact that pre-existing methods needed extensive feature engineering and could not make full use of the prior knowledge available in the form of knowledge bases. This work integrates external knowledge with a deep learning framework for RE [8].

According to Giles *et al.*, for PPI extraction from biological text, about 75 % of the sentences containing co-occurring names of possibly interacting proteins do not describe any causal relationship between them [10]. The authors, thus, investigate the possibility of using fine-tuned BioBERT to analyse these co-occurrences and thereby to accurately determine the functional association between the co-occurring proteins in a given sentence [10]. An experiment conducted by the authors during the data preparation is the investigation of inter-annotator agreement, is worth mentioning. Three independent expert curators curated PPIs from 925 sentences identified by NER tagging within papers drawn from MEDLINE. Surprisingly, concordance between all three curators was observed in only 48.8 % of the cases, which demonstrates the complexity of the problem [10]. This is a significant experiment in the sense that even manual annotations can be subjective, demonstrating the complexity of the problem.

C. EXISTING RESEARCH GAPS AND OUR CONTRIBUTIONS

We observe from the existing literature that it is a popular trend to frame RE as a classification problem. However, framing RE as a classification problem has its limitations [28].First, prediction of a classification model for RE is limited to the labels used in the training dataset. Secondly, the classification framework does not directly preserve the sense of directionality among interacting entities. Thirdly, for integration across different datasets, the classification approach can lead to complex multi-label classification problems [28]. The details of these are discussed in Section II-C and Table 3. In this article, we provide a practical alternative to the classification approach for biological RE such that modellers can customise RE easily as per their modelling requirements, maintaining the sense of directed interactions, wherever applicable.

Biological information can be represented in its most general form as knowledge graphs. The nodes of the knowledge graph represent the entities, and edges are annotations of directed or undirected relations among the entities. Customised edge annotations, as per the interest of the modeller, can be fed into the ARM model, making the model adaptable to the need of the modeller. Given the positional information on a word representing an interaction type (e.g. activation, repression, phosphorylation) in a sentence, ARM can predict the positions of the source and target node entities (e.g. source gene, target gene) for that particular edge annotation. We will henceforth refer to an inter-entity interaction, where the type of the interaction is of interest, as a typed interaction. In the cases where this information is not relevant, we will call the inter-entity interaction as untyped.

We developed the ARM, which can be used to curate and represent new literature, entities such as genes, proteins, phenotypes, etc. and their relationships. ARM provides a flexible framework for a modeller to customise their model, facilitate data integration, and integrate expert knowledge to provide a more practical approach for RE. The general architecture of ARM was reused for different tasks such as retrieving a new related entity from the query entity or interaction keyword. In case a modeller is not interested to model a particular relationship and is simply interested in modelling whether there is an association between two entities (gene-phenotype association), ARM can account for this by learning the position of one entity given the position of the related entity in a sentence. Moreover, ARM is not affected by imbalance and is capable of zero-shot learning. We used the dataset from Elangovan *et al.* having unseen interaction keywords to validate ARM's zero-shot learning capability. In contrast, classification models cannot predict unseen interaction keywords without architectural modifications. Even though the research field of RE from biological texts is still dynamic, with model-based publication arising so often, some research gaps that we address are uniquely addressed by ARM.

II. ATTENTION RETRIEVAL MODEL

In this section, starting with the applicability of ARM to model biological information in terms of knowledge graphs, we formalise the architectural framework of ARM and thereby discuss the aspects of the framework associated to its practical use.

A. ARM IN THE CONTEXT OF MODELLING KNOWLEDGE GRAPHS

A graph $H(\mathcal{N}, \mathcal{E})$, is consisted of a set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$. A KG, $G(\mathcal{N}, \mathcal{E})$ can be realised as encoding of context-specific knowledge using graphs. Biomedical knowledge such as protein interactions, gene to ontology associations, chemical to gene relations, disease to drug associations can be represented as KGs in its most general form. Biological entities such as genes, chemicals, diseases, or pathways can be identified as the node set of the KG. The set of nodes of a KG can have several categories as well. In other words, the set of nodes can consist of several types of entities such as genes, proteins, chemicals or diseases. In the most general representation of KG, one can assume a mapping $\tau_{\mathcal{N}} : \mathcal{N} \rightarrow \mathcal{T}_{\mathcal{N}}$, where $\mathcal{T}_{\mathcal{N}}$ is the set of types of node entities. Edge set $\mathcal{E}$ of G encode relations between a pair of nodes. Similar to the set of nodes, in the most general representation of KG, the edges in the edge set $\mathcal{E}$ can also have several types. For instance, given that the nodes of a certain KG are proteins, an edge among any two proteins can encode the type of interactions among the protein pair. Given that the interaction can be an activation, inhibition, binding, or even more specific such as phosphorylation, the interactions are considered as typed. Note that, some of the typed interactions for example, activation are directed while some, for instance, binding are undirected. In the other cases, such as gene to disease associations, where interaction type is not specified, the interactions are considered as untyped. Similar to the map $\tau_{\mathcal{N}}$, in this case, we can define a mapping $\tau_{\mathcal{E}} : \mathcal{N} \times \mathcal{N} \supseteq \mathcal{E} \rightarrow \mathcal{T}_{\mathcal{E}}$, where $\mathcal{T}_{\mathcal{E}}$ is the set of types of edge entities. For untyped interactions, we can assume that $|\mathcal{T}_{\mathcal{E}}| = 1$ and for typed interactions $|\mathcal{T}_{\mathcal{E}}| > 1$.

Suppose that, a sentence contains information about a directed interaction: "Gene-A activates Gene-B". Here, Gene-A and Gene-B can be called the source entity and the target entity respectively, to indicate the directionality. If we consider the KG, G, having directed interactions, we can assume: $\tau_{\mathcal{E}} : \mathcal{N}_1 \times \mathcal{N}_2 \supseteq \mathcal{E} \rightarrow \mathcal{T}_{\mathcal{E}}$, where $\mathcal{N}_1 \subseteq \mathcal{N}$ and $\mathcal{N}_2 \subseteq \mathcal{N}$ are the set of source nodes and the set of target nodes, respectively. Note that, modelling a knowledge graph is equivalent to realising the map $\tau_{\mathcal{E}}$. Consequently, this requires realising the sets $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{T}_{\mathcal{E}}$.

Let S be a set of sentences such that, $s \in S$ consists of a directed set of words $W^s = \{w_1^s, \ldots, w_k^s\}$. Note that, for convenience of explanation, we can assume that the sentences in S have the same sequence length. In practice, sentences are padded so that every sentence is represented by the same sequence length k. We can call S a knowledge-annotated set if, for every $s \in S$, we have information about the sets $W^s \cap \mathcal{N}_1^S$, $W^s \cap \mathcal{N}_2^S$ and $W^s \cap \mathcal{T}_{\mathcal{E}}^S$, where $\mathcal{N}_1^S$, $\mathcal{N}_2^S$ and $\mathcal{T}_{\mathcal{E}}^S$ are the sets of source nodes, target nodes and interaction types with respect to S. Given a set of annotated sentences S, ARM is designed to learn positional associations among the sets $W^s \cap \mathcal{N}_1^S$, $W^s \cap \mathcal{N}_2^S$ and $W^s \cap \mathcal{T}_{\mathcal{E}}^S$ from all sentences in S, using multi-headed self attention. Given a set of new sentences T, if we have information about any of the sets $W^s \cap \mathcal{N}_1^T$, $W^s \cap \mathcal{N}_2^T$ and $W^s \cap \mathcal{T}_{\mathcal{E}}^T$ but not all of them, ARM predicts the unknown sets by retrieving the attention distribution predicted by a trained model over each sentence. Retrieving the unknown sets can help us realise the map $\mathcal{T}_{\mathcal{E}}^S$ and thereby model the knowledge from the literature, in the form of a knowledge graph.

B. ARCHITECTURE OF ARM

Depending on what information is available in the external i.e. validation set unseen by the model of annotated sentences P, the training of ARM can be customised by a modeller. Given a set of annotated sentences S, let $I = \{\mathcal{N}_1^S, \mathcal{T}_{\mathcal{E}}^S, \mathcal{N}_2^S\}$, be the set of information that is required for a KG.

1) FOR TYPED INTERACTIONS

Let $I_T \subseteq I$, be the set of typed interactions. Given a set of knowledge-annotated sentences, we can assume $D \in I_T$ as the input set of information and $R = R_T \neq D, R = R_T \in I_T$ as the output set of information. In this case, the tuples, $(n_1 \in \mathcal{N}_1^S, t \in \mathcal{T}_{\mathcal{E}}^S, n_2 \in \mathcal{N}_2^S)$ and $(n_2 \in \mathcal{N}_2^S, t \in \mathcal{T}_{\mathcal{E}}^S, n_1 \in \mathcal{N}_1^S)$ may or may not be equivalent, signifying that there may or may not be a sense of direction among the entities and thereby separate source and target entities are preserved.

2) FOR UNTYPED INTERACTIONS

Let $I_U \subseteq I$, be the set of untyped interactions. Given a set of knowledge-annotated sentences, we can assume $D \in I_U$ as

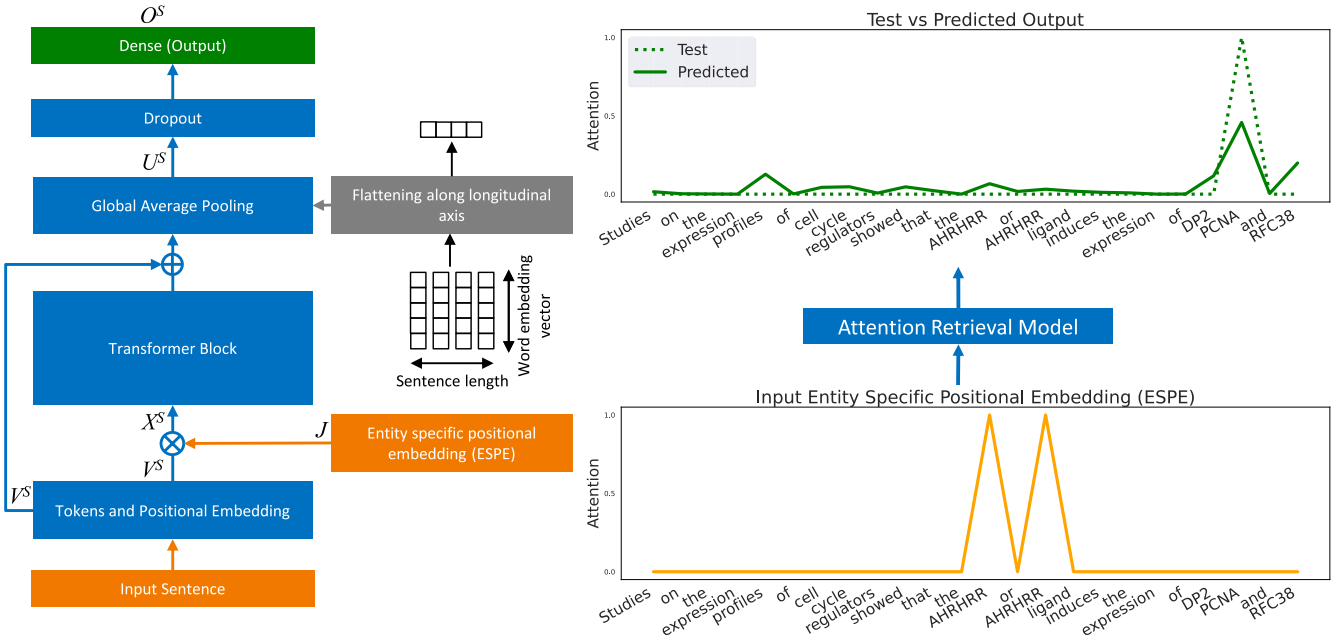

FIGURE 1. Figure showing architecture of ARM model (left) and a schematic visualisation of the ESPE for the input and output entity (right).

the input set of information and $R = R_U \neq D, R = R_U \in I_T$ as the output set of information. In this case, the tuples, $(n_1 \in \mathcal{N}_1^S, t \in \mathcal{T}_\mathcal{E}^S, n_2 \in \mathcal{N}_2^S)$ and $(n_2 \in \mathcal{N}_2^S, t \in \mathcal{T}_\mathcal{E}^S, n_1 \in \mathcal{N}_1^S)$ are equivalent, signifying that there is no sense of direction among the entities.

Traditional attention-based models such as transformers include positional encoding for word sequences as a part of the architecture. The formal expression for positional encoding is given by a pair of equations:

$$\mathcal{P}_{(\text{pos},2i)} = \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d}}}\right) \tag{1}$$

$$\mathcal{P}_{(\text{pos},2i+1)} = \cos\left(\frac{\text{pos}}{10000^{\frac{2i}{d}}}\right) \tag{2}$$

In Equations (1) and (2), the expression pos is used to denote the position of a word in a sentence and d denotes the user-defined dimensions for the word-embedding vectors. That is, each word is essentially perceived by the model as a d-dimensional vector. The index i runs over the dimensions of these word-embeddings and thus can take values in the range $[1, d]$. Note that, Equations (1) and (2) propose two different functions over the vector sequence, depending on whether one is calculating an odd index or even index of the word-embedding vector. The ARM model adapts the notion of positional word-embeddings. However, in addition to word-embeddings, training ARM requires introduction of entity-specific positional embedding (ESPE) J. Given a sentence $s \in S$, with a sequence of words $W^s = \{w_1^s, \ldots, w_k^s\}$, for entity $E \in I$, we can define $J(E)$ as a vector of dimension k, such that the j-th component $(1 \leq j \leq k)$ of the vector $J(E)$ is:

$$J(E)_j = \begin{cases} 1 & : \text{if } w_j^s \in E \\ 0 & : \text{otherwise} \end{cases} \tag{3}$$

For a sentence $s \in S$, ARM receives as input a tokenised version W_t^s of the list of words W^s as well as its input ESPE $J(W^s \cap D)$. Tokenised word vectors W_t^s are used to create a word-wise vector embedding of dimension d, where d is a hyper-parameter of ARM. The traditional positional encoding is added to the word-wise vector embedding to finally obtain a position-encoded vectorised version V_t^s of the words in S. The dimension of V_t^s is thus $d \times k$, where k is the constant sentence length with the assumption of proper padding. The information on the input ESPE, $J(W^s \cap D)$ is now integrated to the V_t^s through a simple product as:

$$X_t^s = V_t^s \otimes J(W^s \cap D) \tag{4}$$

where $\otimes$ represents the component-wise product operation. The obtained vector whose dimension is still $d \times k$, is then passed to a transformer block as input. The transformer block consists of two layers, the first is a self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network [13]. In the first sub-layer of the transformer block, the self-attention mechanism, three copies of this component wise product are passed as Query (Q), Key (K) and Value (V) triplet.

$$Q = K = V = X_t^s \otimes J(W^s \cap D) \tag{5}$$

The equation governing the attention mechanism is given by [13]:

$$\mathcal{A}(K, Q, V) = \text{softmax}\left(\frac{K \cdot Q^T}{\sqrt{d'}}\right)V \tag{6}$$

In practicality, a multi-headed attention mechanism with h heads (h being a hyper-parameter of ARM) is used. The idea of the multi-head attention mechanism can be compared to the use of different filters in Convolutional Neural Networks (CNNs), where each filter learns different latent features from the input. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions [13]. The information from all attention heads is later integrated by a concatenation operation. The multi-headed attention can be written as [13]:

$$MHA(K, Q, V) = Concat(head_1, head_2, \ldots, head_h)Y^O \tag{7}$$

where $head_i = \mathcal{A}(KY_i^K, QY_i^Q, VY_i^V)$. Y_i^K, Y_i^Q, Y_i^V, and Y^O are projection parameters with dimensions $d \times d'$, $d \times d'$, $d \times d'$, $d \times d$ respectively, and $d' = d/h$ for all $h > 0$. A residual connection is then added around, *MHA* followed by layer normalization:

$$U_t^s = X_t^s \oplus MHA(Q, K, V) \tag{8}$$

to retain the entire textual information of the input sentence. The output U_t^s from the first sub-layer of the transformer block is then passed as input to the second sub-layer, the feed-forward network with dimension d_{ff}. The feed-forward network is applied to all positions separately and similar to the first sub-layer, a residual connection is added to the feed-forward network followed by layer normalisation. The output of the second sub-layer is the final output of the transformer block with dimension $d \times k$.

At this point, V_t^s is added to the output of the transformer block. This is possible since both V_t^s and transformer block output are of the same dimension $d \times k$. Finally, we employ global average pooling [20] over the second dimension of U_t^s, to obtain a flattened k-dimensional vector U^s, which is then fed into a dropout layer for regularisation and a fully connected dense layer with sigmoid activation to obtain the k-dimensional output O^s:

$$O^s = \text{Sigmoid}\Big(\text{Dropout}(U^s)\Big) \tag{9}$$

The loss of the model is then estimated as:

$$\mathcal{L} = \text{MSE}\Big(J(W^s \cap R), O^s\Big) \tag{10}$$

where, MSE is the mean-squared error loss. $\mathcal{L}$ is minimised using a suitable optimiser, in our case the ADAM optimiser. Adam optimisation is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments [19].

In Figure 1 (right), the given sentence has the information about the AHRHRR gene inducing the expression of DP2, PCNA, and RFC38 genes. ARM is trained to take as input the sentence and entity-specific position encoding of candidate gene AHRHRR and output the ESPE for one of the possible related entities, in this case, the PCNA gene. The highest three important words in the output ESPE of ARM are DP2, PCNA, and RFC38 genes. This example shows the ARMs capability to understand the relation between two related entities or between an interaction word and a related entity in a text and retrieve them, rather than classifying the text based on predefined annotations.

C. JUSTIFICATION BEHIND THE USABILITY OF THE MODEL

In simple terms, ARM can model knowledge graphs with both typed and untyped interactions. For modelling typed interactions, ARM uses the positions of words indicating interaction types in a sentence and predicts the corresponding source and target entities. For typed interactions, ARM can also be trained to predict the target entity given the source entity as input and vice-versa. For modelling untyped interactions, ARM uses the position of either of the related entity pairs or the interaction word and retrieves the position of the other entity in the sentence that is annotated in a dataset.

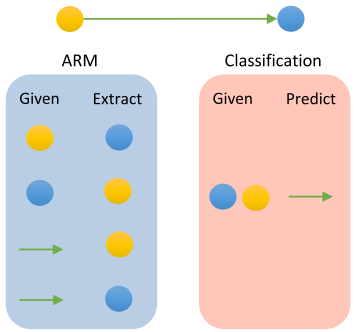

FIGURE 2. An interaction in a KG has two nodes (biological entities) and an edge (interaction). The classification model takes two interacting entities as input and predicts the type of iteraction. Whereas, the ARM model can take as input the interaction and predict either of the interacting entities or take as input one of the interacting entities and predict the other entity.

Note that the approach for RE in ARM is opposite to that of a classification approach, in the sense that, in the classification approach, the goal is to predict the type of interaction from an input text with or without the information about the related entities (e.g. in PPI, a pair of interacting proteins). As a justification to this, we discuss certain limitations of framing RE as a classification problem that can hinder the practical applicability of such classification-based approaches for knowledge extraction from new literature.

- **Directionality in interacting triplets is not preserved:** Modelling directed relationships among biological entities such as genes, proteins, chemicals, diseases, and ontologies require distinction between the source and the target entity. RE problems have also been approached as triplet extraction problems, where interacting entities

are extracted from the text as an ordered tuple of source and target entities and the corresponding interaction. This approach would be effective in preserving the sense of directionality in the interaction. Most classification models for RE lack the framework of preserving the sense of directionality among the interacting entities. Some classification models, such as KAN, take the interacting entities as part of the input, without any associated sense of directionality [8]. Even if we assume that a classification framework provides such a scope, this implies relation extraction from unseen literature requires prior knowledge of whether the entities are sources or targets, as they would be necessary for the input. This can increase annotation effort for unseen datasets. This would require the model to integrate a triplet finding algorithm for the entire knowledge extraction pipeline to work. Thus, the cumulative performance and yield of the RE approach would depend on the efficiency of the triplet finding algorithm.

- **Difficulty in extraction of unseen interactions:** Existing classification models proposed for RE, are usually trained and tested on datasets extracted from publicly available databases such as ChemProt, BioGRID and IntAct [9], [11], [12]. The classification models in such cases are useful to detect interactions that are annotated in the training data. To classify labels that are not present in training data, a classification model requires re-training with new data having the new labels. For example, a classification model trained to classify up-regulation and down-regulation among pairs of genes, cannot predict an interaction such as ''phosphorylation''. For the model to predict ''phosphorylation'', it would need re-training with new training data that has ''phosphorylation'' label.
- **Increasing complexity with data integration:** A classification model that is trained as per the annotations of the BioGRID dataset, for example, cannot be used for extracting knowledge from a differently annotated dataset such as IntAct. Combining knowledge across datasets can increase the number of classes in the classification problem, making the model more complex. For example, the BioGRID and ChemProt have 16 and 22 annotated interaction classes. If we are to combine knowledge across these datasets, we have to integrate them into a 38-class classification problem. In addition, the databases are also likely to be annotated by different experts. Moreover, training such models are likely to be affected by inherent imbalances present in such datasets.

We now discuss how the ARM approach addresses the above-mentioned limitations of the classification approach.

- ARM can be trained for typed RE, where a modeler can provide a bag of interaction words as input to extract the corresponding source or target entities. Since the source and target entities can be extracted individually, it preserves the sense of directionality for the interacting triplet.
- Since ARM predicts the positions of entities in a sentence, instead of classes, it can be used to predict entities related to unseen interaction types, that is, interaction types that are absent in the training dataset. This implies that ARM provides a *zero shot learning* framework for RE. We demonstrate this in Case Study 2 of our experiments.
- ARM learns positional associations between related entities or between a word describing a relationship and its corresponding entities in a sentence. For this reason, it can integrate data with varied grammatical structure of sentences. For example, a single model can be used to learn protein-protein as well as gene-disease associations. Within a specific language style, ARM is less affected by imbalance. We mention the phrase ''within a language style'', to indicate the scenario when ARM deals with data containing consistent information. This means that the data is relevant to a problem with a particular entity type, say proteins, and a fixed bag of keywords. The relative indifference to imbalance makes ARM more suitable for learning from integrated data.

Moreover, as per the investigation of Giles *et al.* [10], there can be considerable ambiguity and disagreements even among expert manual annotators. Thus, varying annotations across datasets, hinders the integration and transfer of knowledge across datasets. With a choice of customized bag-of-words, ARM can integrate the domain knowledge of a modeler in the decision-making process.

III. EXPERIMENTAL SETUP

A. DATASET CURATION

Several popular databases exist, documenting information on entity interactions. For our investigations, we curated data from six different databases/datasets: Atlas of Inflammation Resolution (AIR) [2], TRRUST [1], BioGRID [3], ChemProt [4], Genetic association database (GAD) [5], and a publicly available dataset curated in the work by Elangovan *et al.* [9]. AIR, TRRUST and BioGRID contain protein-protein interactions, ChemProt contains chemical-protein interactions, and GAD contains information on gene-disease association. Note that we chose datasets with multiple entity types (protein, chemical, gene, disease) for our investigation. We chose particularly these datasets considering the following broader motivation:

- We chose the TRRUST dataset, since the annotation of this dataset is the same as AIR. Both AIR and TRRUST consist of typed annotations. The annotations correspond to positive, negative, and unknown type of interactions among proteins. Directed interactions such as positive and negative regulation play a crucial role in building large scale boolean models to explain biological processes.
- AIR is an expert- curated knowledge base focusing on inflammation resolution. We thus chose this dataset to support this research further [2].

TABLE 1. The table shows information on the two curated datasets for typed and untyped relation extraction.

Curated Dataset	Databases	Entity Types	No of Entities	No of relations
Typed Interactions (Training and testing)	TRRUST AIR	Genes	2913	14153
Typed Interactions (Independent validation)	Elangovan dataset	Genes	258	494
Untyped Interactions	BioGRID ChemProt GAD	Genes Chemicals Diseases	20523	334560

- We chose BioGRID and ChemProt datasets that have been used in multiple related publications. Both these databases contain a large volume of data to learn from.
- We chose the dataset by Elangovan *et al.*, since it contains relationship annotations present in no other databases, such as phosphorylation, methylation and acetylation, etc. Using this dataset, we design some interesting case studies. We will henceforth refer to this dataset as Elangovan dataset.

1) CURATION OF DATASETS

We curated two datasets, D_T consisting of typed interactions and D_U consisting of untyped interactions. D_T is curated selectively from AIR, TRRUST and Elangovan databases/datasets. On the other hand, D_U is curated by integration of ChemProt, BioGRID and GAD databases. In accordance to our experimental setup, D_T is further divided into two parts, D_T^* and D_T^{val}. D_T^* is the part of D_T curated from TRRUST and AIR databases and D_T^{val} is the part of D_T curated from the Elangovan dataset.

We emphasise here that, even though, D_U comprises annotated datasets like ChemProt and BioGRID, these annotated datasets are not specifically aimed towards positive and negative regulations among genes. Moreover, both BioGRID and ChemProt use many annotations that are not likely to be present in curated sentences from publication abstracts (e.g. ''synthetic lethality'', ''dosage growth defect'', ''synthetic haploinsufficiency'' etc. for BioGRID and ''Antagonist'', ''Modulator'', ''Agonist-activator'' etc. for ChemProt). Moreover, the majority of interactions in these datasets are undirected. On the other hand, AIR and TRRUST uses simple annotations focused on interactions such as activation and inhibitions among genes. For these annotations, it is possible to construct an expert annotated bag of interaction words. We thus use the large volume of data present in ChemProt, BioGRID and GAD to investigate the potential of ARM to extract untyped interactions, assuming the data from these databases to be untyped, whereas selected instances from AIR and TRRUST are used for typed RE.

RM focuses on customisable knowledge extraction and relies on a bag of interaction words that can be provided by a modeller. For an interaction mentioned in a PubMed abstract, the sentences present in the abstracts are considered. The entity names and their synonyms present in the sentence are replaced with standardised Entrez [6] and PubChem [7] names. The obtained entity-normalised sentences are then searched for entity and interaction keys. For typed interactions, if both interacting entities and interaction keys are mentioned in a sentence, it is assumed that the sentence has information about the typed interaction. In the case of untyped interactions, if both interacting entities are mentioned in a sentence, it has information about the untyped interaction. For simplicity, sentences having more than five occurrences of the same entity are not considered.

It is well known that at a large scale, logic-based models and molecular interaction maps are popular in explaining biological processes. For protein interactions, such models are particularly interested in positive and negative regulations between protein pairs and information on protein pair binding to form complexes are some key interactions [2], [14]. Thinking of practical applicability, we therefore considered three types of interactions: Positive, Negative, and Physical/Unspecified interactions. Positive and Negative interactions are of course directed typed interactions, while physical/unspecified interactions can be considered as undirected typed interactions. We consider two keywords for positive interactions: 'Activate' and 'Up-regulate' and extract sentences containing these keywords and source-target entity names from the correspondingly annotated positive class in the dataset from the AIR and TRRUST databases. Similarly, the keywords we considered for the negative classes 'Repress', 'Down-regulate', and 'Inhibit' and for Unspecified classes, 'Bind' and 'Interact' keys are considered as keywords for sentence extraction.

We curate D_T^{val} from the Elangovan dataset as an independent validation dataset. This is to account for one of our case studies, where we investigate whether ARM can extract relations from data that concerns different interaction types than the data which the model was trained on. The Elangovan dataset uses a completely different set of annotations for interactions compared to that of AIR or TRRUST. These annotations are, namely: phosphorylation, dephosphorylation, methylation, demethylation, ubiquitination, deubiquitination, and acetylation. From these, we chose the interactions that are annotated as phosphorylation, since it was by far the largest class with 800 instances. In one of our case studies, we train ARM on D_T^*, which does not contain instances for the annotation ''phosphorylation'', and validate it on D_T^{val}. The ARM model thus, allows a framework for *zero-shot learning*, which is not possible with the classical classification approach for RE.

B. TRAINING AND EVALUATION PROCEDURE

We perform several case studies to demonstrate the diverse applications of ARM. Here, we provide an overview of these case studies:

- **Case Study 1:** The first case study is designed to investigate how ARM performs when trained and tested on data with the same bag of interaction words. For this, we use the curated dataset D_T^*, where our specified bag of interaction words corresponds to the interactions: Activation, Up-regulation, Repression, Down-regulation, Inhibition, binding and Interaction. The ARM model takes selected sentences and the position of these interaction words in the respective sentences (through the function P) as input and retrieves the positions of the source and target entities corresponding to these interaction words. The ARM in this case, is validated using a 10-fold cross validation procedure.
- **Case Study 2:** The second case study is designed to investigate how ARM performs when validated on unseen labels. As discussed before, this cannot be done using the classification approach. We train the ARM model in this case using D_T^* and test it using D_T^{val}. Recall, that for D_T^{val}, the interaction bag of words contain only phosphorylation, that is not present in the bag of words chosen to curate D_T^*. The training process is similar to that of Case Study I. While testing, the ARM takes as input the selected sentences in D_T^{val} and the position of the interaction word corresponding to phosphorylation in the respective sentences.
- **Case Study 3:** The third case study is to investigate the performance of ARM to take selected sentences and the position of the source or target entity in the respective sentences (through the function P) as input, and retrieves the positions of the target or source entity corresponding to these interaction words from the typed dataset D_T, respectively. Note that, in this case, we still maintain a sense of directionality in the interactions since D_T largely consists of directed interactions, and, therefore, consider the source and target entities separately. ARM, in this case, is trained using a 10-fold cross validation procedure.
- **Case Study 4:** The fourth case study is to investigate the performance of ARM on the untyped dataset D_U. Recall that, even though interactions in some databases used for curating D_U are typed, most of the interactions are undirected, and the interaction word is unlikely to be present in a sentence and therefore cannot be used as a keyword for the bag of interaction words necessary to train the typed version of the ARM model. Thus, we adapt D_U for our "untyped" case study, by assuming the data to be untyped. By this assumption, there is no distinction between the source and the target entity. So for this case, we randomly select 50% of data from D_U and interchange the source and target entities as annotated in the dataset. We simply refer to two related entities as "Entity 1" and "Entity 2", and investigate whether given the position information of "Entity 1", ARM can retrieve the positional information of "Entity 2". The ARM in this case, is validated using a 10-fold cross validation procedure.

As an output, ARM generates a signal vector with values in the range [0, 1], over the words of the input sentence. The word in the input sentence corresponding to the highest peak of this signal can be considered as the predicted entity by the trained ARM model. We demonstrate this in Figure 1. To quantify the performance of ARM, we define the following performance measure.

Definition 1 k-Exact Entity Match Accuracy (k-EEMA): If the labelled output entity for a test data point d is present in the set of words corresponding to the k-highest peaks of the output signal $O^s(d)$ generated by a trained ARM, then the prediction for d is assumed to be correct. The percentage of correct predictions on a validation set thus gives the k-EEMA.

This means that a 1-EEMA Score corresponds to the percentage of test instances where the highest peak of the predicted signal corresponds exactly to the output entity and a 2-EEMA Score corresponds to the percentage of test instances where any of the two highest peaks of the predicted signal correspond exactly to the output entity and so on. We will refer to 1-EEMA simply as EEMA.

IV. RESULTS AND DISCUSSION

A. CASE STUDY 1

From the first case study, we observe that, given the interaction word, ARM can predict the corresponding source entities related to the word with 1-EEMA, 2-EEMA, and 3-EEMA scores of 77.0%, 90.4%, and 95.2% respectively. It can predict target entities related to the word with 1-EEMA, 2-EEMA, and 3-EEMA scores of 79.3%, 88.7%, and 92.7% respectively. This shows that if we train and test ARM on the data which has the same label, the ARM can detect the source and target entities corresponding to a customised bag of words.

B. CASE STUDY 2

From the results of the second case study, we observe that, given an interaction word that is absent in the training data, ARM can still detect source entities with 1-EEMA, 2-EEMA, and 3-EEMA scores of 30.2%, 49.3%, and 62.4% respectively. Similarly, for an interaction word that is absent in the training data, ARM can detect target entities with 1-EEMA, 2-EEMA, and 3-EEMA scores of 30.3%, 47.3%, and 60.0% respectively. Note that, this example is analogous to *zero shot learning*, since the test dataset contains sentences from a completely different source and with completely different annotations. A classification based approach would not be able to perform an RE task in such a case, since such an approach can only learn annotations/labels that are present in the training data.

TABLE 2. Table showing 1,2, and 3 EEMA scores for different case studies. The types of tasks are shown in the format Input Entity → Output Entity. SRC, TRGT, INT, ENT refers to source, target, interaction, and entity respectively.

Case Study	Task	Training	Validation	1-EEMA	2-EEMA	3-EEMA
1	INT → SRC	D_T^*	10-fold CV	77.0%	90.4%	95.2%
1	INT → TRGT	D_T^*	10-fold CV	79.3%	88.7%	92.7%
2	INT → SRC	D_T^*	D_T^{val}	30.2%	49.3%	62.4%
2	INT → TRGT	D_T^*	D_T^{val}	30.3%	47.3%	60.0%
3	SRC → TRGT	D_T^*	10-fold CV	85.1%	92.3%	95.0%
3	SRC → TRGT	D_T^*	D_T^{val}	40.7%	59.1%	70.3%
3	TRGT → SRC	D_T^*	10-fold CV	83.4%	93.2%	96.5%
3	TRGT → SRC	D_T^*	D_T^{val}	40.0%	60.0%	71.0%
4	$\text{ENT}_1 \rightarrow \text{ENT}_2$	D_U	10-fold CV	72.6%	82.4%	89.3%

C. CASE STUDY 3

The third case study consists of two parts. For the first part, we use 10-fold cross validation for training and validation on D_T^* and the second part, we train ARM on D_T^* and validate on D_T^{val}. For the first part, we observe that given the source entity, ARM can predict the target entity with 1-EEMA, 2-EEMA, and 3-EEMA scores of 85.1%, 92.3%, 95.0% respectively. Given the target entity, ARM can predict the source entity with 1-EEMA, 2-EEMA, and 3-EEMA scores of 83.4%, 93.2%, 95.0% respectively. For the second part, where we validate ARM on an independent validation set with a different type of interaction, ARM can detect the source entity, given the target entity, with 1-EEMA, 2-EEMA, and 3-EEMA scores of 40.0%, 60.0%, and 71.0%. Given the source entity, it can detect the target entity with an 1-EEMA, 2-EEMA, 3-EEMA scores of 40.7%, 59.1%, and 70.3%. The reduced instance in the third case can be attributed to the fact that, D_T^{val} being annotated differently, contains sentences with different structures than that of D_T^*.

D. CASE STUDY 4

Recall that this case study is for the untyped case, for which there is no particular notion of a source and a target entity. From the results of the fourth case study, we observe that, given an entity, ARM can predict a related entity with an 1-EEMA, 2-EEMA, and 3-EEMA scores of 72.6%, 82.4%, and 89.3% respectively.

A summary of the results is provided in Table 2. The key philosophy behind the ARM approach is to train a model to understand the linguistic context between two related entities in a text, rather than classifying the text based on predefined annotations. Our results demonstrate some advantages of the ARM approach for RE over the standard classification approach. As ARM uses a customised bag of interaction words as per the choice of the modeller, one can customise the model as per their requirements.

While a more desirable scenario could be a comprehensive comparison among the models discussed in Section I, this turns out to be difficult for different reasons. To make an unbiased comparison between these models, BLM sub-tasks like NER, Triplet finding, and Document triage are required, which are not addressed in many of the discussed models, as the general norm in this domain is to address the RE task singularly. Although most of the models are publicly available, almost all have a very elaborate pre-processing protocol, which are difficult to reproduce exactly even from the provided coding resources. Moreover, ARM also has its own distinct performance measure. Such factors make it challenging to perform a comparative study and infer superiority or inferiority of the ARM model from such experiments. While the classification approach is well established, even without going into quantitative comparisons, from the construct of the models themselves, we can observe that ARM has some scope beyond the classification approach. We summarise some of these in Table 3.

TABLE 3. Key differences between the classification and the ARM approach.

Requirements	Classification	ARM
Named entity recognition required	✓	✓
Not affected by imbalance	✗	✓
Sense of direction among interacting entities	✗	✓
Zero-shot learning	✗	✓
Availability of pre-trained networks	✓	✗
Convenience of data integration	✗	✓
Model flexibility	✗	✓
Improved explainability	✗	✓

Since the position of the associated/interacting entities and interaction words are used, not the labels, the ARM is not affected by imbalance. Furthermore, the ARM can retrieve entities related to interaction words, which are not present in the training data. To summarise, we have demonstrated the effectiveness of ARM with D_U, which contains diverse entity types or with D_T^{val}, where the interaction word is different from that of the training set.

ARM provides a modeller with the opportunity to integrate their knowledge in the decision-making for RE. Suppose that there is a sentence containing information about the interaction among two entities, where one protein positively regulates the other. When an expert annotates the sentence, and they base their decision precisely upon two keywords describing the relationship, say activation or up-regulation. When a model learns from such a data, its knowledge is essentially influenced by the knowledge of the annotator. However, the

perception of positive regulation can be different for another independent second modeller. For instance, it might be possible that the modeller considers methylation of one protein by another, also as activation. Therefore, a second modeller is likely to get unsatisfactory results if the modeller uses the data curated by the first expert for his model training. Giles *et al.* already points out about annotation-related discrepancies among experts [10].

Moreover, the ARM approach can be used for RE even when an annotation is absent in the dataset. As we have demonstrated in Case Study 2, we use a dataset with completely different annotations for validation and yet, ARM can retrieve the source and target entities from it. This task, analogous to *zero shot learning*, is impossible to achieve with the classification approach, as the output of such a model would be dependent on the annotations used for training. It is worth noting that the RE in the pre-Neural Network era used to be in the form of triplet extraction. A triplet encodes a relationship between two entities as a tuple (Entity1, interaction type, Entity2) [17], [18]. It was an effective method for RE, since it preserved a sense of direction in the interaction, given the interaction word. Existing deep-learning-based models for RE rarely address the issue of retaining this directionality. ARM proposes a framework that accounts for the direction of the interaction. For typed interactions, given the interaction word, ARM retrieves the source and target entities separately. This is demonstrated in the Case Studies 1 and 2. Since the ARM predicts the source and target entity, the unseen data does not need to be annotated with source and target entities. If one is not interested in the type of interaction, but simply chooses to extract related entities in a directed or undirected manner, ARM provides a framework for both of these. For the first purpose, the framework for Case Study 3 can be employed, while for the second purpose, the framework for Case Study 4 can be used. For Case Study 4, we have also demonstrated that even if there are diverse entities in the dataset, ARM can learn the associations between them. In the curated dataset D_U used for Case Study 4, entities can be of different types such as genes, chemicals, and disease. To the best of our knowledge, there is also no evidence of self-attention-based frameworks that base their study on RE with mixed entity types. This provides the ARM framework a unique advantage for data integration across datasets across diverse literature. Note, that the ARM framework thus can also be easily adapted to targeted relationship extraction. For example, researchers may be interested in the investigation of a particular entity and its corresponding interactions to understand their role in a biological process (e.g., E2F1) [14]. A framework of ARM can be trained to retrieve the position of an entity given the position of a related entity as input, or vice versa. This can therefore be used to extract the molecular interactions by providing a query molecule as a source entity or a target entity.

Many sentences that we have used for training are complex in structure, having more than one source/target entities and interaction words. So, the 1-EEMA score alone would not be a precise evaluation of the model performances. However, the consistent increase in performance from 1-EEMA to 3-EEMA scores in all case studies coupled with the fact, that in most cases, the labelled output entity lies within the top three retrievals (3-EEMA score) of ARM, provides us with ample evidence that the ARM model is capable of understanding the grammatical structure of a sentence by successfully associating and thereby extracting interacting entities.

For practical use, ARM can be integrated with tasks such as NER to normalise entities and document triage in a pipeline. The NER task can extract named entities from a given text. The document triage task, popularised by the BioCreative VI challenge, determines whether a piece of text contains information relevant to an interaction triplet [11]. The classification approach for RE also uses these as pre-processing approaches. An auxiliary advantage of ARM is that the output itself can be visualised and interpreted. ARM extracts the positional distribution of attention over the input sentence, given an input entity in the form of a signal. This provides an easy interpretation of the output. In classification-oriented attention-based models, this can be achieved by extracting and visualising the attention matrix for each data point. In addition to this, ARM is much faster compared to pre-trained attention based models such as BioBERT, which is often employed for the classification task.

V. CONCLUSION

The ARM approach for RE provides an alternative to the usual deep-learning based classification approach for RE. The key philosophy behind the ARM approach is to train a model to understand the linguistic context between two related entities in a text, rather than classifying the text based on predefined annotations. ARM provides a flexible framework for a modeller to customise their model, with the opportunity to integrate expert knowledge on interaction keywords. This enables modellers to build their models as per their choice of annotations rather than using predefined annotations, which can be evidently ambiguous, even among expert annotators. ARM provides an opportunity to learn from integrated data with diverse entity types and contents. This facilitates data integration across different datasets. Furthermore, unlike its classification-based counterpart, ARM can extract relationships, that might be unannotated in the training data.

AVAILABILITY OF CODE

The computations were performed on Intel(R) Xeon(R) Gold 6142 CPU @ 2.60GHz with 8 Nvidia TU102 graphic processors. The study was conducted using Python3 and Jupyter-Notebooks. We provide the codes here.

ACKNOWLEDGMENT

Prashant Srivastava and Saptarshi Bej designed, drafted and revised the manuscript and contributed equally to

the work; Kristian Schultz assisted in several computational aspects; Kristina Yordanova and Olaf Wolkenhauer critically discussed and revised the approach and content of the manuscript. All the authors have read and agreed to the contents of the manuscript.

REFERENCES

[1] H. Han, H. Shim, D. Shin, J. E. Shim, Y. Ko, J. Shin, and H. Kim, "TRRUST: A reference database of human transcriptional regulatory interactions," *Sci. Rep.*, vol. 5, no. 1, Sep. 2015, Art. no. 11432, doi: 10.1038/srep11432.

[2] C. N. Serhan, S. K. Gupta, M. Perretti, C. Godson, E. Brennan, Y. Li, and O. Soehnlein, "The atlas of inflammation resolution (AIR)," *Mol. Aspects Med.*, vol. 74, Aug. 2020, Art. no. 100894, doi: 10.1016/j.mam.2020.100894.

[3] R. Oughtred, J. Rust, C. Chang, B. Breitkreutz, C. Stark, A. Willems, L. Boucher, G. Leung, N. Kolas, F. Zhang, S. Dolma, J. Coulombe-Huntington, A. Chatr-Aryamontri, K. Dolinski, and M. Tyers, "TheBioGRIDdatabase: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions," *Protein Sci.*, vol. 30, no. 1, pp. 187–200, Jan. 2021, doi: 10.1002/pro.3978.

[4] O. Taboureau, S. K. Nielsen, K. Audouze, N. Weinhold, D. Edsgard, F. S. Roque, I. Kouskoumvekaki, A. Bora, R. Curpan, T. S. Jensen, S. Brunak, and T. I. Oprea, "ChemProt: A disease chemical biology database," *Nucl. Acids Res.*, vol. 39, pp. D367–D372, Jan. 2011, doi: 10.1093/nar/gkq906.

[5] K. G. Becker, K. C. Barnes, T. J. Bright, and S. A. Wang, "The genetic association database," *Nature Genet.*, vol. 36, no. 5, pp. 431–432, May 2004, doi: 10.1038/ng0504-431.

[6] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez gene: Gene-centered information at NCBI," *Nucl. Acids Res.*, vol. 39, pp. D52–D57, Jan. 2011, doi: 10.1093/nar/gkq1237.

[7] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, and S. H. Bryant, "PubChem substance and compound databases," *Nucl. Acids Res.*, vol. 44, no. D1, pp. D1202–D1213, Jan. 2016, doi: 10.1093/nar/gkv951.

[8] H. Zhou, Z. Liu, S. Ning, C. Lang, Y. Lin, and L. Du, "Knowledge-aware attention network for protein-protein interaction extraction," *J. Biomed. Informat.*, vol. 96, Aug. 2019, Art. no. 103234, doi: 10.1016/j.jbi.2019.103234.

[9] A. Elangovan, M. Davis, and K. Verspoor, "Assigning function to protein-protein interactions: A weakly supervised BioBERT based approach using PubMed abstracts," 2020, *arXiv:2008.08727*.

[10] O. Giles, A. Karlsson, S. Masiala, S. White, G. Cesareni, L. Perfetto, J. Mullen, M. Hughes, L. Harland, and J. Malone, "Optimising biomedical relationship extraction with BioBERT," 2020, doi: 10.1101/2020.09.01.277277.

[11] Y. Wang, S. Zhang, Y. Zhang, J. Wang, and H. Lin, "Extracting protein-protein interactions affected by mutations via auxiliary task and domain pre-trained model," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Seoul, South Korea, Dec. 2020, pp. 495–498, doi: 10.1109/BIBM49941.2020.9313120.

[12] P. Su, Y. Peng, and K. Vijay-Shanker, "Improving BERT model using contrastive learning for biomedical relation extraction," 2021, *arXiv:2104.13913*.

[13] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000—6010. [Online]. Available: https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[14] F. M. Khan, S. Marquardt, S. K. Gupta, S. Knoll, U. Schmitz, A. Spitschak, D. Engelmann, J. Vera, O. Wolkenhauer, and B. M. Pützer, "Unraveling a tumor type-specific regulatory core underlying E2F1-mediated epithelial-mesenchymal transition to predict receptor protein signatures," *Nature Commun.*, vol. 8, no. 1, p. 198, Dec. 2017, doi: 10.1038/s41467-017-00268-2.

[15] S. Zhao, C. Su, Z. Lu, and F. Wang, "Recent advances in biomedical literature mining," *Briefings Bioinf.*, vol. 22, no. 3, May 2021, Art. no. bbaa057.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[17] K. Raja, S. Subramani, and J. Natarajan, "PPInterFinder—A mining tool for extracting causal relations on human proteins from literature," *Database*, vol. 2013, Jan. 2013, Art. no. bas052, doi: 10.1093/database/bas052.

[18] L. Tari, S. Anwar, S. Liang, J. Cai, and C. Baral, "Discovering drug-drug interactions: A text-mining and reasoning approach based on properties of drug metabolism," *Bioinformatics*, vol. 26, no. 18, pp. i547–i553, Sep. 2010, doi: 10.1093/bioinformatics/btq382.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[20] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.

[21] P.-Y. Lung, Z. He, T. Zhao, D. Yu, and J. Zhang, "Extracting chemical–protein interactions from literature using sentence structure analysis and feature engineering," *Database*, vol. 2019, Jan. 2019, Art. no. bay138, doi: 10.1093/database/bay138.

[22] J. Ren, G. Li, K. Ross, C. Arighi, P. McGarvey, S. Rao, J. Cowart, S. Madhavan, K. Vijay-Shanker, and C. H. Wu, "iTextMine: Integrated text-mining system for large-scale knowledge extraction from the literature," *Database*, vol. 2018, Jan. 2018, Art. no. bay128.

[23] K. Raja *et al.*, "Automated extraction and visualization of protein–protein interaction networks and beyond: A text-mining protocol," in *Protein-Protein Interaction Networks* (Methods in Molecular Biology), vol. 2074, S. Canzar and F. Ringeling, Eds. New York, NY, USA: Humana, 2020, doi: 10.1007/978-1-4939-9873-9_2.

[24] F. Christopoulou, M. Miwa, and S. Ananiadou, "Connecting the dots: Document-level neural relation extraction with edge-oriented graphs," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1–12.

[25] M. Ishizuka, "Exploiting macro and micro relations toward web intelligence," in *Proc. PRICAI*, 2010, pp. 4–7.

[26] D. P. Nguyen, Y. Matsuo, and M. Ishizuka, "Relation extraction from wikipedia using subtree mining," in *Proc. Nat. Conf. Artif. Intell.*, 2007, pp. 1–7.

[27] Z. Geng, G. Chen, Y. Han, G. Lu, and F. Li, "Semantic relation extraction using sequential and tree-structured LSTM with attention," *Inf. Sci.*, vol. 509, pp. 183–192, Jan. 2020.

[28] P. Srivastava, S. Bej, K. Yordanova, and O. Wolkenhauer, "Self-attention-based models for the extraction of molecular interactions from biological texts," *Biomolecules*, vol. 11, no. 11, p. 1591, Oct. 2021.

PRASHANT SRIVASTAVA received the master's degree in physics from the Indian Institute of Science Education and Research, Kolkata, and the master's degree in data science and analytics from the Department of Computer Science, Royal Holloway, University of London. He is currently a Research Assistant with the Department of Systems Biology and Bioinformatics, Institute of Computer Science, University of Rostock, Germany. His current research interests include algorithm development and their applications in literature mining.

SAPTARSHI BEJ received the master's degree in mathematics from the Indian Institute of Science Education and Research, Kolkata, in 2014. He is currently a Research Associate at the Institute of Computer Science, University of Rostock, Germany, and a Guest Scientist with the Leibniz-Institute for Food Systems Biology, Technical University of Munich, Germany. His research interests include graph theory, specifically the Barnette's conjecture, the development of machine learning algorithms and their applications in the life sciences, machine learning on small and imbalanced datasets, and literature mining.

KRISTIAN SCHULTZ received the degrees in mathematics and computer science, in 2014. After that, he worked for two years in the field of discrete mathematics on Sperner families with the Department of Mathematics, University in Rostock, Germany. In between, he extended his software developing skills in industry. Since 2020, he returned to the Department of Systems Biology and Bioinformatics, University of Rostock, where he focuses on the correctness, efficiency, and implementation of algorithms.

KRISTINA YORDANOVA received the bachelor's degree in computer engineering from the University of Duisburg-Essen, Germany, in 2008, the master's degree in artificial intelligence from Maastricht University, The Netherlands, in 2009, and the Ph.D. degree in ubiquitous computing from the University of Rostock, Germany, in 2014. She is currently the Head of the Junior Research Group "Cognitive Methods for Situation-Aware Assistive Systems," University of Rostock, and a Research Associate with the University of Bristol, U.K. Her research interests include natural language processing, machine learning, and symbolic and probabilistic modeling with applications in assistive systems, social sciences, healthcare, and medicine.

OLAF WOLKENHAUER received the degrees in systems and control engineering and the Ph.D. degree for research in possibility theory with applications to data analysis. He spent over ten years at the University of Manchester Institute of Science and Technology, U.K. In 2003, he was appointed as a Professor of systems biology and bioinformatics at the University of Rostock, Germany. In 2005, he became a fellow of the Stellenbosch Institute for Advanced Study and holds professorships at Case Western Reserve University, USA, and Chhattisgarh Swami Vivekanand Technical University, India. In 2015, he was elected as a member of the Foundations in Medicine and Biology Review Panel of the German Research Foundation (DFG). His research interests include data-driven modelling with model-driven experimentation, using a wide range of approaches, including machine learning and systems theory.