Graph Convolutional Networks for Chemical Relation Extraction

Darshini Mahendran
Virginia Commonwealth University
Richmond, Virginia, USA
mahendrand@vcu.edu

Christina Tang
Virginia Commonwealth University
Richmond, Virginia, USA
ctang@vcu.edu

Bridget T. McInnes
Virginia Commonwealth University
Richmond, Virginia, USA
btmcinnes@vcu.edu

ABSTRACT

Extracting information regarding novel chemicals and chemical reactions from chemical patents plays a vital role in the chemical and pharmaceutical industry. Due to the increasing volume of chemical patents, there is an urgent need for automated solutions to extract relations between chemical compounds. Several studies have used models that apply attention mechanisms such as Bidirectional Encoder Representations from Transformers (BERT) to capture the contextual information within a text. However, these models do not capture the global information about a specific vocabulary. On the other hand, Graph Convolutional Networks (GCNs) capture global dependencies between terms within a corpus but not the local contextual information. In this work, we propose two novel approaches, GCN-Vanilla and GCN-BERT, for chemical relation extraction. GCN-Vanilla approach builds a single graph for the whole corpus based on word co-occurrence and sentence-word relations. Then, we model the graph with GCN to capture the global information and classify the sentence nodes. GCN-BERT approach combines GCN and BERT to capture both global and local information, and build together a final representation for relation extraction. We evaluate our approaches on the CLEF-2020 dataset. Our results show the combined GCN-BERT approach outperforms standalone BERT and GCN models, and achieves a higher F_1 than that reported in our previous studies.

CCS CONCEPTS

• **Computing methodologies** → **Feature selection**; **Information extraction**.

KEYWORDS

relation extraction, chemical natural language processing, graph convolutional neural networks, BERT

ACM Reference Format:
Darshini Mahendran, Christina Tang, and Bridget T. McInnes. 2022. Graph Convolutional Networks for Chemical Relation Extraction. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3487553.3524702

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04.
https://doi.org/10.1145/3487553.3524702

1 INTRODUCTION

Chemical patents offer exclusive rights to use specific chemicals, molecules, compounds to the scientists who obtained them [5]. Chemical patents include information about novel chemicals and chemical reactions; therefore, they play a vital role in the chemical and pharmaceutical industry. Due to the exponential growth of chemical patents in recent years, it is difficult for researchers to keep up with the current state of the art. Manual extraction of information is almost impossible; therefore, there is an urgent need to find automated solutions to extract relations between chemical compounds. Chemical Relation Extraction (RE) is a task of extracting semantic relations between chemical entities from raw texts. RE has been applied to extract relation between various chemical domain entities such as chemicals–genes [12], chemicals-diseases [11, 30], chemicals-proteins [20].

RE approaches extensively use techniques based on neural networks such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and self-attention based models such as Bidirectional Encoder Representations from Transformers (BERT) [3]. These techniques capture the local contextual information within a sentence or document well by embedding both semantic and syntactic information in a learned representation, especially BERT [14]. The position and order of the words play a vital role in determining the context of the words in a sentence. Most of the deep learning approaches rely on the position embeddings to model the dependency between elements at different positions in a sentence to generate a contextualized representation [14]. However, their ability to capture the long-range dependency global information in a text is limited. Utilizing the global association information between words outside the sentence boundaries can help generate better representations. For example, in RE, we learn the representation of the words and entities in a sentence to classify whether the entity pair has a relation between them or not. In some instances, the associations between co-occurring words in the corpus except the targeted sentence can have the relevant information to determine the relation than the words/phrases within the sentence. In this instance, generating the representation based on the local information within the sentence does not help. Therefore, we need to utilize approaches that also capture the global association information between the co-occurring words.

BERT is a self-attention-based model that captures the given token information and the information of the surrounding tokens but it does not capture the global association information within language well, as it only considers the positional information of the words. Graph Neural Networks (GNNs) are deep learning models that operate in the graph domain and capture global information between words/ phrases. Recent research based on the graph has

been receiving more attention due to the great expressive power of graphs [31]. The variant of GNN that gained popularity recently is Graph Convolutional Networks (GCNs) [10]. GCN captures the global context information by performing convolution operations on neighbor nodes in a graph and incorporating information from neighbors. GCNs can preserve global structure information of a graph in graph embeddings [26]. GCNs are also explored in several NLP tasks over the years, such as text classification [14, 26], relational reasoning [33], and sentiment analysis [24]. However, little or no work has been done on using GCNs for chemical RE. To generate a better representation capturing both local and global information is essential. Models that use attention mechanisms such as BERT capture local contextual information better, whereas graph-based models such as GCN capture global information better.

In this work, we explore how to effectively capture the global dependencies between terms within a corpus and how to combine the capability of BERT with a GCN and benefit from the combination. We propose two novel approaches to extract relations between chemical entities: GCN-Vanilla and GCN-BERT. In the GCN-Vanilla approach, we construct a single graph based on word co-occurrence and sentence-word relations and model the graph with GCN to capture the global information. In the GCN-BERT approach, we combine BERT and GCN to complement the local information captured by BERT with the global information captured by GCN and allow both types of information to influence mutually build a final representation for RE. We evaluated our methods on the CLEF ChEMU-2020 dataset [19]. Our experimental results demonstrate that the GCN-BERT approach outperforms BERT and GCN alone and achieve a higher F_1 than that reported in our previous studies.

The remainder of this paper is structured as follows. First, discuss the algorithms behind our approaches. Second, we discuss the related works done in this area of research. Third, we describe the dataset we use to evaluate our system. Fourth we discuss our approaches in detail. Fifth, we present and analyze the results. Finally, we end by stating the conclusions we derive from this work and what we plan to do in the future.

2 BACKGROUND

Here, we discuss the algorithms we used in this paper.

2.1 Graph Convolutional Networks

Neural networks gained popularity in the past decade, and different variants of simple neural networks achieved success in many research fields. However, most of these variants deal with euclidean data, while many real-world data are non-euclidean. These data have led to the recent invention of variants – GNNs. GNNs are a deep learning-based method that extends existing neural network methods to operate on the data represented in graph domains [32]. GNNs deal with non-euclidean graph data that contains rich relational information between elements. The following highlight the advantages of GNNs over CNNs [32]:

- Traditional neural networks such as CNNs and RNNs operate on regular euclidean data like images (2D grid) and do not handle non-euclidean types data well because they stack features by a specific order. The graph data do not have a natural order of nodes, and nodes can be traversed in different orders.
- The dependency between two nodes in the graphs is represented by an edge in GNN, whereas they are considered just another feature in the traditional networks.
- Traditional networks learn by the distribution of the data, whereas GNNs generate graphs from non-structural and learn the reasoning, which can be helpful in high-level AI-related research.

GCNs [10] are a recent variant of the basic GNN architectures that are designed to perform inference over data described using a graph. Given a graph $G = (V, E)$, a GCN takes the following as the input [10]: an input feature matrix $N * F$, where N is the number of nodes and F is the number of input features for each node, a feature matrix X, and an $N * N$ matrix representation of the graph structure such as the adjacency matrix A of G. GCNs utilize the "message passing" mechanism, which is performed through matrix operations where the information is passed from one node to another. Each layer of the GCN defines a propagation rule in the form of a matrix, which determines how inputs will be transformed before being sent to the next layer. In this layer, the incoming feature matrix is multiplied by the adjacency matrix as shown in the Equation 1:

$$f(H^i, A) = \sigma(A\,H^i\,W^i) \tag{1}$$

where W^i is the weight matrix for layer i, σ is a non-linear activation function such as the ReLU function, H^i is a hidden layer, f is a propagation rule, and $H^0 = X$, the feature matrix. This helps the features to become increasingly more abstract at each consecutive layer. The basic operations of the GCNs are similar to CNNs. The convolution is applied in CNNs by multiplying the input neurons with weights commonly known as filters or kernels. GCNs perform a similar operation to learn the features of the neighboring nodes. However, the difference is that the nodes in a GCN are unordered, and the connections between nodes are not uniform (irregular non-euclidean data), whereas CNNs operate on regular euclidean data.

Kipf, et al. [10] presented GCNs in their pioneering work showing it achieved state-of-the-art classification results on several benchmark graph datasets including Stanford Sentiment Treebank (SST-2) [23], Corpus of Linguistic Acceptability (CoLA) [25], and ArangoHate [1].

2.2 Bidirectional Encoder Representations from Transformers (BERT)

In 2018, Google introduced BERT [3], a language model that utilizes an attention mechanism to model semantic relations between words of a text. BERT is the first bidirectionally trained language model, models before that train left-to-right or vice versa. In addition, BERT produces contextual embedding representations of a token. These representations can be fine-tuned for specific domains.

To do this, BERT utilizes a transformer that consists of an encoder to read the input. The language model generation takes part in the encoder, which reads the input. The input representation is the sum of the token, segmentation, and position embeddings. The token embedding transforms the tokens into vector representations of fixed dimensions. The segment embedding adds a marker to indicate which sentence the word token is from and checks whether the

input came from one sentence only. Positional embeddings indicate the position of the input token in the sentence.

3 RELATED WORK

GCN-based models have been gaining attention recently among the NLP community. However, GCN alone or combined with BERT has not been applied to chemical RE before. Here, we discuss works related to RE and works that inspired us to propose our approaches using GNN/GCN.

Relational reasoning tries to reason about entities and their relations, which are of great importance in many NLP tasks, including RE [33]. Zhu, et al. [33] proposed to generate the parameters of GNNs (GP-GNNs) according to natural language sentences, which enabled GNNs to process relational reasoning on unstructured text inputs. GP-GNN is constructed with entities in the sequence of the text followed by three modules that encode rich information from natural languages, propagate relational information among various nodes, and classify. Joint entity and relation extraction is an essential task in information extraction, which aims to extract all relational triples from unstructured text [29]. Zhao, et al. [29] proposed a representation, iterative fusion based on heterogeneous GNN for RE (RIFRE). They modeled relations and words as nodes on the graph and updated them through a message-passing mechanism to perform RE. This fuses the semantic information of the relations nodes to the word nodes associated with them, which helps to extract the entities that form valid relations. Inter-sentence RE deals with complex semantic relations in documents [22]. Sahu, et al. [22] presented a novel inter-sentence RE model that builds a labeled edge GCN model on a document-level graph. The graph was constructed using various inter and intra-sentence dependencies, and they utilized multi-instance learning with bi-affine pairwise scoring to predict the relation of an entity pair.

One of the major applications of the GNN is node classification, where we train the graph nodes with labels and try and predict the label for a node without ground truth. This has been adapted to perform text classification using graph structures. Yao, et al. [26] utilized a GCN for text classification. First, they built a single text graph based on word co-occurrence and document word relations, then learned a Text GCN for the corpus. Text GCN jointly learns the embeddings for both words and documents, as supervised by the known classes for documents. Huang, et al. [8] proposed a different GNN based method. Instead of building a single corpus level graph, they built a graph for each input text. They connected the word nodes within a relatively small text window rather than all. The representations of the same nodes and weights of edges are shared globally and updated in the text level through a message passing mechanism, where a node takes in the information from neighboring nodes to update its representation. They claimed this removed the dependency burden between a single input text and the entire corpus. Zhang, et al. [27] proposed a novel method for "*INductive word representations*" via GNN, termed TextING. They built individual graphs for each document first, then used GNN to learn the fine-grained word representations based on their local structures, effectively producing embeddings for unseen words in the new document. Finally, the word nodes are incorporated as the document embedding. Lu, et al. [14] proposed a model which combines the strengths of BERT with a Vocabulary VGCN in the same model. The word embedding and graph embedding interacted through the self-attention mechanism while learning the classifier.

In our work, we build two models that utilize GCN alone first and then combine GCN with BERT for chemical RE. Our models using GCN are inspired by the works of Yao, et al. [26] and Lu, et al. [14] for text classification.

4 DATA

In 2020, the Cheminformatics Elsevier Melbourne University (ChEMU) evaluation lab, which is part of the Conference and Labs of the Evaluation Forum (CLEF-2020) introduced the CLEF-2020 dataset to identify chemical entities and events that explain the sequence of steps that lead from a chemical reaction to an end product [7]. The dataset contains chemical snippets sampled from chemical patents and includes ten entity classes under four categories and two classes of trigger words: REACTION_STEP, WORKUP. Fig 4 in the appendix shows the hierarchical structure of the entity labels, and the table 4 in the appendix shows the definitions of each entity type. Relations are divided into two classes: ARG1 and ARGM. The ARG1 includes relations between a trigger word and chemical compound entities. The ARGM event label corresponds to the relations between a trigger word and temperature, time, or yield entities. Table 1 shows the statistics of the training dataset.

Table 1: Number of entity types and trigger words in the training data and their event relations

Events	Entities	Instances	REACTION_STEP	WORKUP
ARG1	EXAMPLE_LABEL	886	-	-
	REACTION_PRODUCT	2052	1101	11
	STARTING_MATERIAL	1754	1747	4
	REAGENT_CATALYST	1281	1272	-
	SOLVENT	1140	1134	4
	OTHER_COMPOUND	4640	161	4097
ARGM	YIELD_PERCENT	955	937	1
	YIELD_OTHER	1061	1043	2
	TIME	1059	839	81
	TEMPERATURE	1515	813	242
Triggers	REACTION_STEP	3815		
	WORKUP	3053		

Each chemical snippet in the dataset is annotated by the Brat Rapid Annotation Tool (BRAT), a web-based tool for text annotation. This helps to identify the entities, their types, and relations between them. Fig 5 in the appendix section shows an example of a BRAT annotated sentence from the dataset containing chemical entities and relations. The entities except for the trigger words are gold standard entities. We use our Named Entity Recognition (NER) system [15]. Our NER system utilizes a combined model of Bidirectional Long Short Term Memory (BiLSTM) and Conditional Random Fields (CRF) trained with ChemPatent embeddings [27].

5 METHODS

Here, we propose two approaches for RE: GCN-Vanilla and GCN-BERT. We treat the RE task as a binary classification task building a separate model for each trigger word-entity type to determine whether a relation exists between them: 1) Positive class - there is a relation between the trigger word and the entity, 2) Negative

class - there is no relation between the trigger word and the entity (no-relation).

5.1 GCN-Vanilla Approach

In this approach, we first build one single graph with word and sentence nodes over the entire corpus. The number of nodes V in the graph equals the number of sentences and the number of unique words in the corpus.

Second, we measure the weight of the edge between two word nodes (word-word nodes) using Point-wise Mutual Information (PMI) [2]. The occurrence of two words together can be just by chance or because there is an above-chance frequency of two words in that particular order. For example, the term 'disturbed sleep' has different independent meanings, but together, they express a precise, unique concept. PMI is a measure that quantifies the likelihood of the co-occurrence of two words. Equation 2 shows how PMI is computed between two word nodes. If x and y are independent, their joint probability equals the product of their marginal probabilities that result in a log equal to 0, which means the words occurred by chance. A positive PMI value indicates a semantic correlation between words in a sentence, whereas a negative value indicates no correlation. Therefore, we consider edges between word nodes where the PMI value is positive when the graph is generated.

$$PMI(x, y) = \log\left(\frac{P(x, y)}{P(x)P(y)}\right) \tag{2}$$

Third, we measure the weight of the edge between the node and sentence (word-sentence nodes) using Term Frequency - Inverse Document Frequency (TF-IDF) [9]. With multiple documents inside a corpus, TF-IDF takes into account how frequently tokens appear across multiple documents [6]. TF-IDF is calculated by multiplying two metrics: Term Frequency (TF) and Inverse Document Frequency (IDF). TF measures the raw count of a word in a document and IDF measures how common the word is across multiple documents. Higher the score, higher the relevancy of the word in a specific document. Here, TF is measured by the number of times the word appears in the sentence, and IDF is measured by the logarithmically scaled inverse fraction of the number of sentences that contain the word.

Fourth, we utilize pre-trained word embeddings to generate the initial word vectors for the word nodes. We average the word vectors of the word nodes connected to a sentence node to create an embedding representation for the sentence. Fig 1 shows the structure of the graph we build for this approach. Nodes that begin with S are sentence nodes (green), and the rest are unique word nodes (yellow). Black bold edges between sentence nodes and word nodes are sentence-word edges, and the thin black edges between word nodes are word-word edges. We consider this approach as our baseline.

Fifth, we model the graph with a multi-layer GCN to capture the high-order neighborhoods information. A multi-layer GCN allows message passing between nodes that are not connected directly but a few levels away. A two-layer GCN passes messages from the nodes that are at a maximum of two steps away [26]. There are no direct sentence-sentence nodes in our graph, but they are connected through the word nodes; therefore, a two-layer GCN allows information passing from one sentence node to another. Initially, the weight vectors of the nodes are randomly initialized and then jointly learn the embeddings for both words and sentences. Finally, the output of the second-layer nodes is fed into a softmax layer for classification. This turns the relation classification problem into a node classification problem. Softmax is calculated as shown in Equation 3:

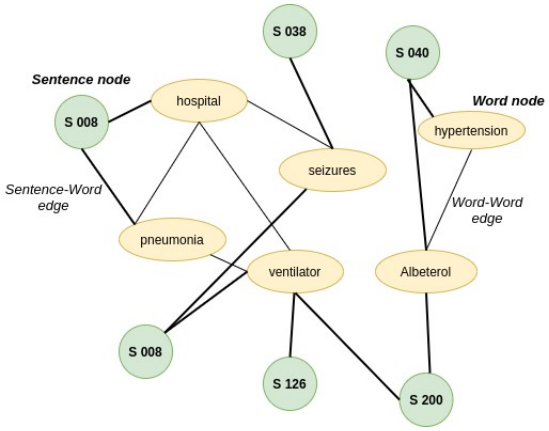

Figure 1: Structure of the graph for the GCN-Vanilla approach.

$$Z = \text{softmax}(\tilde{A}\,\text{ReLU}(\tilde{A}\,X\,W_0)W_1), \tag{3}$$

where $\tilde{A} = D^{-1/2}AD^{-1/2}$ [26]. The cross entropy error is calculated over all sentences.

5.2 GCN-BERT Approach

From our previous works [17] we found the BERT-based approaches outperformed other supervised deep learning approaches. BERT utilizes positional information to capture the local contextual information within a sentence or document. On the other hand, GCN captures the global context information by performing convolution operations on neighbor nodes in the graph. To generate a better representation capturing both local contextual information and global association information between words in the input is essential. Therefore, we propose to combine BERT with GCN to benefit from capturing local and global information. In this approach, we first generate a vocabulary graph based on the word association information, which is passed through GCN to capture the global information of the language. Then we combine the graph embedding and word embedding together to a self-attention encoder in BERT [14]. Both embeddings interact with each other and build together a final representation for classification [14].

First, we extract the sentence where the entity pair is located. We use the BERT tokenizer to tokenize the sentence into words. Since BERT is a pre-trained model, input data needs to be in a specific format, and the BERT tokenizer carries out specific operations to generate the format. First, the words are split into subwords and characters. BERT handles the Out-of-vocabulary (OOV) words by tokenizing them to the character level. They utilize the '##' sign to indicate they are part of a larger word, distinguishing a subword

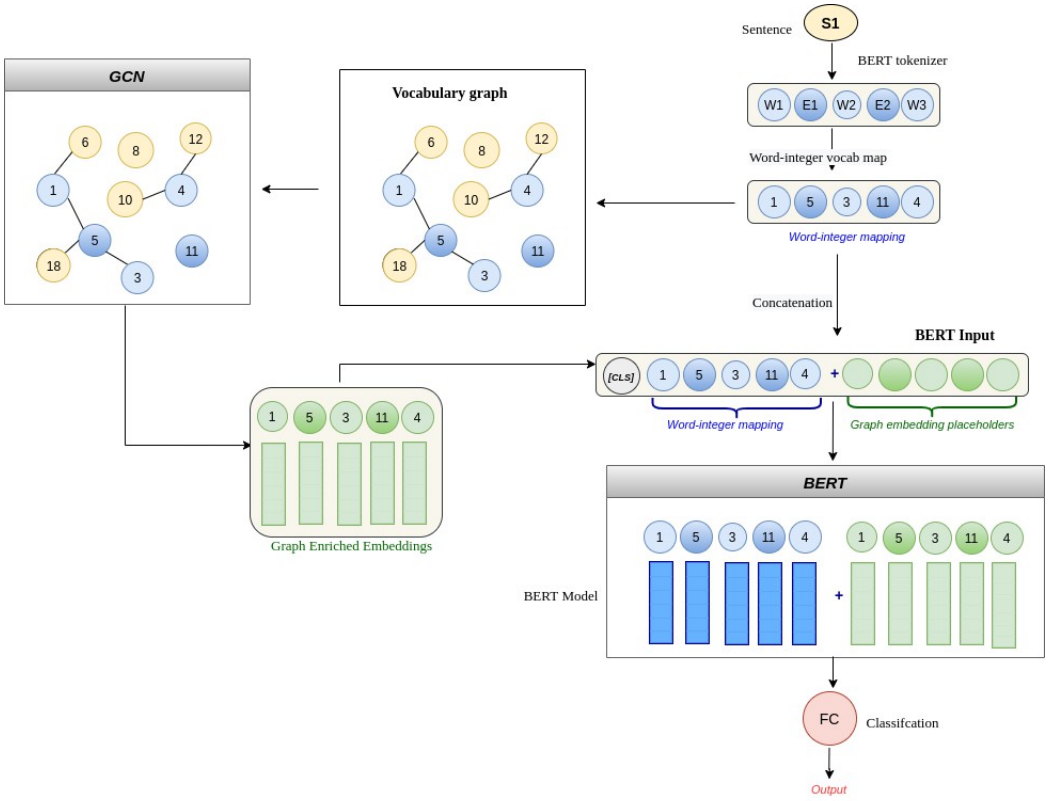

Figure 2: Structure for the GCN-BERT combined approach.

token from a word token when generating word embeddings. Subword vectors are averaged to generate an approximate vector for the original word. After splitting the sentence into the tokens, we build a vocabulary map mapping the unique tokens to integers.

Second, we generate a vocabulary graph G = (V, E) where the number of nodes equals the number of unique words in the corpus. We denote the word nodes in the graph by the mapped integers, and we measure the weight of the edge between two word nodes (word-word nodes) using PMI as shown in Equation 2. Here, the PMI values are normalized (NPMI) between the range of [-1, 1]. A positive NPMI value indicates a semantic correlation between words, whereas a negative NPMI value indicates little or no semantic correlation. Edges exist between word nodes from the training set when PMI > 0. Next, we pass the graph through a two-layer GCN to generate the graph embeddings based on the properties of their neighborhoods. GCN performs two layers of convolution to capture the information between the nodes that are not connected directly. We use ReLU activation function in the GCN [14] described in Equation 4:

$$\text{VGCN} = \text{ReLU}(X_{mv}\,\tilde{A}_{vv}\,W_{vh})W_{hc} \tag{4}$$

where m is the batch size, v is the vocabulary size, h is the size of the hidden layer, c is the size of the sentence embedding.

Third, we combine the mapped word indices with the generated graph embeddings before passing them into BERT, which helps capture the order of the words in the sentence and the global information captured by the graph. BERT uses a transformer, an attention mechanism that learns contextual relations between words. BERT applies multi-layer multi-head self-attention on the concatenated input. The input text sequence travels through a stack of 12 encoders at each level and a feed-forward neural network and outputs a sentence embedding for classification. Usually, BERT architecture takes a token, segment, and position embeddings of the input text and a [CLS] token. However, here we combine word token embeddings and placeholders for the graph embeddings before passing them into the BERT. When the token embeddings layer converts each word piece token into a vector representation, we combine the graph embeddings vector. Next, BERT applies the bidirectional training, which simultaneously takes the previous and next tokens into account and represents the input sequence. Finally, the final embedding representation is fed into a fully connected layer for classification.

Fig 2 shows the overall structure of this combined GCN-BERT approach and illustrates how an input sentence is passed through this architecture. The input sentence $S1$ is tokenized and the word nodes are denoted in blue. In the vocabulary graph, words nodes from the input sentence $S1$ are shown in blue whereas the word

nodes from other sentences are shown in yellow. This approach captures and combines the input text's local and global information.

5.3 Entity representations

To determine the relation between two chemical entities, we first locate the sentence where the entity pair is located. A sentence can have multiple such entity pairs therefore we need to represent the targeted entity pair in a distinguishable way from other entity pairs. Here, we explored three variations of input sentence entity representations from our previous work[18]:

(1) Representation A - we input the entire sentence where the entity pair is located. Both the targeted and non-targeted entity pairs are represented as it is.
(2) Representation B - we remove the non-targeted entity pairs from the input sentence. Targeted entity pairs are represented as it is.
(3) Representation C - we replace the targeted entity pair with its semantic type in the input sentence. Non-targeted entity pairs are represented as it is.

Figure 3 shows an example of an input sentence from the CLEF-2020 dataset and how a targeted entity pair is represented differently in each representation.

6 EXPERIMENTAL DESIGN

Word embeddings. For our GCN-Vanilla approach, we use GloVe [21] embedding representations. The GloVe is trained over Wikipedia (2014) and Gigaword 5. For our GCN_BERT approach, we use BERT embeddings. BERT is pre-trained on the whole of the English Wikipedia and Brown Corpus originally and is fine-tuned on downstream NLP tasks.

Hyper-parameters. We define our model training hyper-parameters by adjusting the batch size, learning rate, regularization, and the number of epochs. We used the batch size of 512, Adam optimizer with the learning rate of 0.01, and train for 20 epochs with an early stopping of 15 epochs for our GCN-Vanilla approach. We use the batch size of 512, Adam optimizer with the learning rate of 0.0001, and train for 10-20 epochs for our GCN-BERT approach. We use the PyTorch-Transformers[1] by HuggingFace Team to build the BERT model.

Reproducability. The source code of this paper is available in the following public repository: https://github.com/NLPatVCU/RelEx-GCN

7 EVALUATION CRITERIA

We evaluate our approaches using Precision (P), Recall (R), and F_1 score (F). Precision calculates out of all instances how many instances are predicted correctly, and Recall calculates out of all the correct instances that should have been predicted how many instances are correctly predicted. F_1 score is the harmonic mean of Precision and Recall. We also report the micro averages of the system performance. Micro average calculates metrics globally by counting the total true positives, false negatives, and false positives.

[1] https://pytorch.org/hub/huggingface$_p ytorch - transformers$/

8 RESULTS AND DISCUSSION

In this section, we present and discuss the results of our two approaches, and conduct a comparison with previous work.

8.1 Test set results

Table 2 shows precision (P), recall (R), and F_1 (F) scores on the test set of the CLEF-2020 dataset for each of our architectures across the three input representations described in Section 5.3. The overall results show that the GCN-BERT approach outperformed the baseline GCN-vanilla approach for all three input representations except Representation A. Also, GCN-BERT approach obtained the highest precision, recall, and F_1 scores for all the relations of both REACTION_STEP and WORKUP classes. The notable increase in the performance of the GCN-BERT shows the advantage of combining BERT with GCN allowing interactions between the local contextual and global association information. Comparing the results of the REACTION_STEP classes than WORKUP classes, we see that both approaches obtained higher F_1 scores with the REACTION_STEP classes than WORKUP classes. This is because the REACTION_STEP classes have more training instances than most of the WORKUP classes, therefore they can differentiate themselves when training than the WORKUP classes. Despite the lower number of training instances in the WORKUP classes, GCN-BERT comparatively obtained higher F_1 scores than the GCN-Vanilla. This indicates combining the local information of the text with global information provides additional information for the classification layer than just considering the global information only, especially for the classes with fewer training instances in a class.

We use various input entity representations to distinguish the targeted entity pairs. When multiple entity pairs are present in an input sentence, Representation B removes the non-targeted entity pairs, and Representation C replaces the entity pair is with its semantic type, whereas Representation A passes sentence as it is. The overall analysis of the various input entity representations showed that Representation B outperformed the other two input representations by obtaining a higher F_1 score, which shows that masking non-targeted entities helped extract essential information to identify the classes better. All representations obtained similar precision, recall, and F_1 score with the GCN-vanilla but Representations B and C obtained comparatively higher precision, recall, and F_1 score with the GCN-BERT. This indicates that the performance increases when the targeted entities are distinguished from the non-targeted entities. Since most of the input sentences in the test set are quite long, we can find multiple entity pairs in one sentence. Therefore, we believe masking non-targeted entities (Representations B) or replacing targeted entities by their semantic types (Representations C) provides a better classification representation.

8.2 Comparison with previous work

Table 3 shows a comparison between the top results reported by the CLEF ChEMU-2020 challenge using the CLEF-2020 dataset, the co-occurrence baseline provided by the organizers of the challenge, best overall results of our previous approaches [16] and best results of our current approaches. Bold terms show the best results in each category.

Table 2: Precision (P), Recall (R), and F_1 (F) score on the test set with trigger words identified using our previous NER model [16] (BiLSTM+CRF trained with ChEMU patent embeddings)

Method	Relation	Trigger	Entity	# Train	Representation A			Representation B			Representation C		
					P	R	F	P	R	F	P	R	F
GCN-Vanilla	ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			REACTION_PRODUCT	1101	0.85	0.96	0.90	0.85	0.96	0.90	0.85	0.95	0.91
			REAGENT_CATALYST	1272	0.58	0.73	0.65	0.61	0.71	0.65	0.59	0.68	0.63
			SOLVENT	1134	0.58	0.70	0.64	0.58	0.75	0.65	0.58	0.69	0.63
			STARTING_MATERIAL	1747	0.61	0.76	0.68	0.61	0.77	0.68	0.61	0.76	0.68
			Average		*0.52*	*0.47*	*0.51*	*0.87*	*0.79*	*0.82*	*0.64*	*0.65*	*0.63*
		WORKUP	OTHER_COMPOUND	4097	0.59	0.68	0.63	0.62	0.75	0.68	0.63	0.67	0.65
			REACTION_PRODUCT	11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			REAGENT_CATALYST	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			SOLVENT	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			STARTING_MATERIAL	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.18*	*0.09*	*0.12*	*0.19*	*0.17*	*0.18*	*0.19*	*0.18*	*0.18*
	ARGM	REACTION_STEP	TEMPERATURE	813	0.55	0.38	0.45	0.56	0.38	0.45	0.61	0.34	0.44
			TIME	839	0.56	0.63	0.59	0.61	0.64	0.62	0.60	0.58	0.59
			YIELD_OTHER	1043	0.85	0.97	0.91	0.85	0.97	0.91	0.85	0.97	0.91
			YIELD_PERCENT	937	0.85	0.96	0.90	0.85	0.96	0.91	0.86	0.95	0.90
			Average		*0.70*	*0.74*	*0.71*	*0.72*	*0.74*	*0.72*	*0.73*	*0.71*	*0.71*
		WORKUP	TEMPERATURE	242	0.00	0.00	0.00	0.62	0.21	0.31	0.56	0.13	0.21
			TIME	81	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.00*	*0.00*	*0.00*	*0.31*	*0.11*	*0.16*	*0.28*	*0.07*	*0.11*
	System				**0.65**	**0.70**	**0.67**	**0.66**	**0.73**	**0.69**	**0.67**	**0.69**	**0.68**
GCN-BERT	ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.43	0.59	0.50	0.60	0.48	0.53	0.00	0.00	0.00
			REACTION_PRODUCT	1101	0.97	0.85	0.90	0.93	0.90	0.91	0.89	0.90	0.89
			REAGENT_CATALYST	1272	0.00	0.00	0.00	0.94	0.87	0.90	0.55	0.78	0.64
			SOLVENT	1134	0.87	0.40	0.54	0.93	0.85	0.89	0.82	0.70	0.73
			STARTING_MATERIAL	1747	0.78	0.50	0.61	0.95	0.84	0.89	0.96	0.88	0.91
			Average		*0.61*	*0.47*	*0.51*	*0.87*	*0.79*	*0.82*	*0.64*	*0.65*	*0.63*
		WORKUP	OTHER_COMPOUND	4097	0.88	0.43	0.58	0.95	0.85	0.89	0.94	0.88	0.91
			REACTION_PRODUCT	11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			REAGENT_CATALYST	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			SOLVENT	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			STARTING_MATERIAL	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.44*	*0.22*	*0.29*	*0.48*	*0.43*	*0.45*	*0.47*	*0.44*	*0.45*
	ARGM	REACTION_STEP	TEMPERATURE	813	0.55	0.30	0.39	0.90	0.50	0.64	0.89	0.55	0.68
			TIME	839	0.72	0.41	0.52	0.88	0.72	0.79	0.90	0.77	0.83
			YIELD_OTHER	1043	0.88	0.94	0.91	0.99	0.97	0.98	0.94	0.89	0.91
			YIELD_PERCENT	937	0.85	0.96	0.90	0.99	0.92	0.96	0.87	0.93	0.90
			Average		*0.75*	*0.65*	*0.68*	*0.94*	*0.78*	*0.84*	*0.90*	*0.79*	*0.83*
		WORKUP	TEMPERATURE	242	0.00	0.00	0.00	0.87	0.61	0.72	0.00	0.00	0.00
			TIME	81	0.00	0.00	0.00	0.00	0.00	0.00	0.72	0.49	0.58
			Average		*0.00*	*0.00*	*0.00*	*0.44*	*0.31*	*0.36*	*0.36*	*0.25*	*0.29*
	System				**0.82**	**0.48**	**0.61**	**0.94**	**0.81**	**0.87**	**0.87**	**0.75**	**0.81**

Table 3: Our best results in comparison with our previous results and the top results of the ChEMU-2020 competition. Baseline is provided by the organizers of the ChEMU-2020 challenge

		P	R	F
Our current methods	GCN-Vanilla	0.66	0.73	0.69
	GCN-BERT	**0.94**	**0.81**	**0.87**
Our previous methods	Rule-based	0.51	**0.72**	0.60
	CNN-based	**0.81**	0.54	**0.65**
	BERT-based	0.58	0.59	0.58
	BioBERT-based	0.62	0.50	0.55
ChEMU_2020 teams	Melaxtech [28]	**0.96**	**0.95**	**0.95**
	NextMove/Minesoft [13]	0.94	0.86	0.90
	BOUN_REX [4]	0.76	0.69	0.72
Baseline	ChEMU organizers [19]	0.24	0.89	0.38

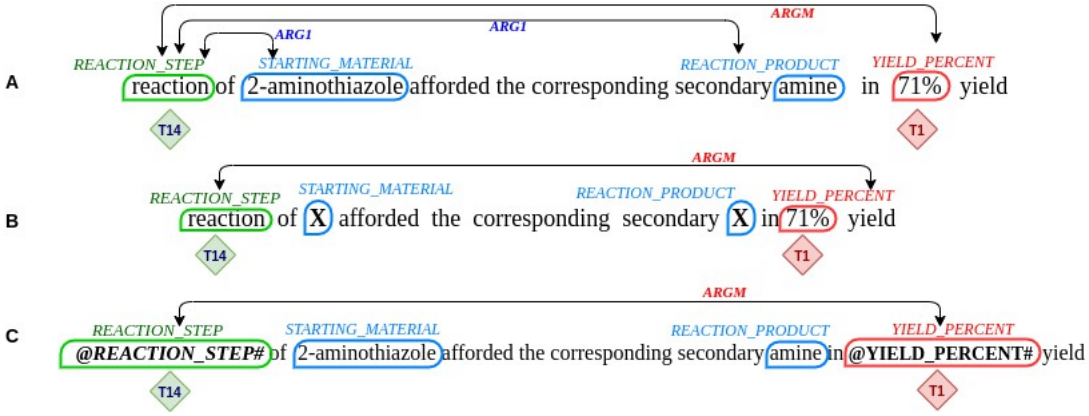

Figure 3: Illustration of the various entity representations of a sample input sentence

Comparison of the results between our current and previous approaches shows that the GCN-BERT outperformed all other approaches and obtained the highest overall precision, recall, and F_1 score. In particular, it outperformed both GCN and BERT alone, which confirmed the advantage of combining them. Among the models that only use local information, CNN performed better than BERT. If we compare the CNN-based approach and the GCN-Vanilla based approach, we can see the GCN-Vanilla based approach obtained higher recall and F_1 scores but not precision. CNN and BERT capture the local information between words better, whereas GCN captures the global information better. This shows that capturing the global information is beneficial for classifying the relations in the CLEF-2020 dataset. The superior performance of GCN-BERT shows that combining both BERT and GCN and allowing interactions between the two types of information is beneficial. The baseline provided by the organizers of the ChEMU-2020 challenge obtained a higher recall than our current approaches. Since the baseline is a rule-based approach based on the co-occurrence information, it obtains a high recall but low precision. Our approaches outperformed the baseline in terms of precision and F_1 score.

In the CLEF ChEMU-2020 challenge, Melaxtech [28] used a hybrid approach combining a deep learning model with pattern matching rules and obtained the overall highest F_1 score. First they retrained the BioBERT patent data to generate a new language model of Patent_BioBERT and utilized a binary classifier to recognize relations between event triggers and semantic roles in the same sentence. They also applied post-processing rules to recover relations in long complex sentences. NextMove/Minesoft [13] utilized parsing information with grammar rules, and BOUN_REX [4] utilized a set of rules to identify the relations. Both Melaxtech [28] and NextMove/Minesoft [13] obtained higher F_1 scores than our approaches. In the future, we plan to explore integrating rule-based information into our GCN-BERT based approach.

9 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed two approaches for relation extraction: GCN-Vanilla and GCN-BERT approaches. GCN-Vanilla utilizes GCN to capture the global structure information in graph embeddings. In contrast, GCN-BERT combines GCN and BERT to integrate the local contextual and global information between the words. We also explored three input entity representations with both approaches. We evaluated our approaches on the CLEF-2020 chemical patent dataset. From the results, we can conclude that combining GCN and BERT and allowing both types of information to interact through the layers of attention mechanism is beneficial compared to using BERT and GCN alone. We also found that replacing the targeted entities with their semantic types or masking the non-targeted entities in a sentence effectively provides unique entity representations of an input sentence.

In the future, we plan to investigate expanding both approaches to perform multi-class classification and benchmark against different datasets. We also plan to build a model that trains GCN and BERT separately and then concatenate the graph and BERT embeddings before feeding them through the final classification layer. We utilized a custom-built word-integer mapping to represent a word node in the vocabulary graph in the GCN-BERT approach. Also, we used random weight vectors initially to generate the graph embeddings. In the future, we would like to use an external pre-built vocabulary and explore the performance of the graph embeddings with various external pre-trained word embeddings.

ACKNOWLEDGMENTS

This work was funded by the National Science Foundation (NSF) under Grant No. CMMI 1651957.

REFERENCES

[1] Aymé Arango, Jorge Pérez, and Barbara Poblete. 2019. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 45–54.
[2] Kenneth W. Church and Patrick Hanks. 1989. Word association norms, mutual information, and Lexicography. In *Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Vancouver, B.C., 76–83.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[4] Hilal Dönmez, Abdullatif Köksal, Elif Ozkirimli, and Arzucan Özgür. 2020. BOUN-REX at CLEF-2020 ChEMU Task 2: Evaluating Pretrained Transformers for Event Extraction. (2020).
[5] Geoffrey M Downs and John M Barnard. 2011. Chemical patent information systems. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1, 5

(2011), 727–741.
[6] Zhou Guodong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 427–434.
[7] Jiayuan He, Dat Quoc Nguyen, Saber A Akhondi, Christian Druckenbrodt, Camilo Thorne, Ralph Hoessel, Zubair Afzal, Zenan Zhai, Biaoyan Fang, Hiyori Yoshikawa, et al. 2020. An extended overview of the CLEF 2020 ChEMU Lab. In *the Conference and Labs of the Evaluation Forum (CLEF)*. 22-25 September 2020.
[8] Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text level graph neural network for text classification. *arXiv preprint arXiv:1910.02356* (2019).
[9] Karen Spärck Jones. 2004. IDF term weighting and IR research lessons. *Journal of Documentation* (2004).
[10] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[11] Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database* 2016 (2016).
[12] Sangrak Lim and Jaewoo Kang. 2018. Chemical–gene relation extraction using recursive neural network. *Database* 2018 (2018).
[13] Daniel Lowe and John Mayfield. 2020. Extraction of reactions from patents using grammars. In *Central Europe Workshop Proceedings (CEUR-WS)*.
[14] Zhibin Lu, Pan Du, and Jian-Yun Nie. 2020. VGCN-BERT: Augmenting BERT with Graph Embedding for Text Classification. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12035)*. Springer, 369–382.
[15] Darshini Mahendran, Gabrielle Gurdin, Nastassja Lewinski, Christina Tang, and Bridget T McInnes. 2020. NLPatVCU CLEF 2020 ChEMU Shared Task System Description. In *Conference and Labs of the Evaluation Forum (CLEF) 2020 Working Notes*.
[16] Darshini Mahendran, Gabrielle Gurdin, Nastassja Lewinski, Christina Tang, and Bridget T McInnes. 2021. Identifying chemical reactions and their associated attributes in patents. *Frontiers in Research Metrics and Analytics* 6 (2021).
[17] Darshini Mahendran and Bridget T McInnes. 2021. Extracting Adverse Drug Events from Clinical Notes. In *AMIA Annual Symposium Proceedings*, Vol. 2021. American Medical Informatics Association, 420.
[18] Darshini Mahendran, Sudhanshu Ranjan, Jiawei Tang, Mai H Nguyen, and Bridget T McInnes. 2021. BioCreative VII-Track 1: A BERT-based System for Relation Extraction in Biomedical Text. In *Proceedings of the Biocreative VII*.
[19] Dat Quoc Nguyen, Zenan Zhai, Hiyori Yoshikawa, Biaoyan Fang, Christian Druckenbrodt, Camilo Thorne, Ralph Hoessel, Saber A Akhondi, Trevor Cohn, Timothy Baldwin, et al. 2020. ChEMU: Named Entity Recognition and Event Extraction of Chemical Reactions from Patents. In *European Conference on Information Retrieval*. Springer, 572–579.
[20] Yifan Peng, Anthony Rios, Ramakanth Kavuluru, and Zhiyong Lu. 2018. Chemical-protein relation extraction with ensembles of SVM, CNN, and RNN models. *arXiv preprint arXiv:1802.01255* (2018).
[21] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[22] Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence Relation Extraction with Document-level Graph Convolutional Neural Network. *arXiv preprint arXiv:1906.04684* (2019).
[23] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.
[24] Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5679–5688.
[25] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics* 7 (2019), 625–641.
[26] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7370–7377.
[27] Zenan Zhai, Dat Quoc Nguyen, Saber A Akhondi, Camilo Thorne, Christian Druckenbrodt, Trevor Cohn, Michelle Gregory, and Karin Verspoor. 2019. Improving chemical named entity recognition in patents with contextualized word embeddings. *arXiv preprint arXiv:1907.02679* (2019).
[28] Jingqi Wang1 Yuankai Ren2 Zhi Zhang and Yaoyun Zhang. 2020. Melaxtech: A report for CLEF 2020–ChEMU Task of Chemical Reaction Extraction from Patent. (2020).
[29] Kang Zhao, Hua Xu, Yue Cheng, Xiaoteng Li, and Kai Gao. 2021. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems* 219 (2021), 106888.
[30] Huiwei Zhou, Huijie Deng, Long Chen, Yunlong Yang, Chen Jia, and Degen Huang. 2016. Exploiting syntactic and semantics information for chemical–disease relation extraction. *Database* 2016 (2016).
[31] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.
[32] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434* (2018).
[33] Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun. 2019. Graph neural networks with generated parameters for relation extraction. *arXiv preprint arXiv:1902.00756* (2019).

A CLEF-2020 DATASET

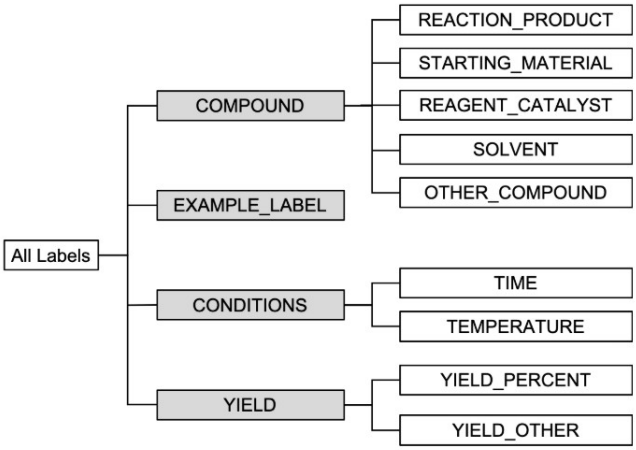

Figure 4: An illustration of the hierarchical structure of the entity labels of the CLEF-2020 dataset [7]

Table 4: Definitions of entity types of CLEF-2020 dataset [7]

Entity Type	Definition
REACTION_PRODUCT (R.P.)	A product is a substance that is formed during a chemical reaction.
STARTING_MATERIAL (S.M.)	A substance that is consumed in the course of a chemical reaction providing atoms to products is considered as starting material.
REAGENT_CATALYST (R.C.)	A reagent is a compound added to a system to cause or help with a chemical reaction. Compounds like catalysts, bases to remove protons or acids to add protons must be also annotated with this tag.
SOLVENT (S)	A solvent is a chemical entity that dissolves a solute resulting in a solution.
OTHER_COMPOUND (O.C.)	Other chemical compounds that are not the products, starting materials, reagents, catalysts and solvents.
TIME	The reaction time of the reaction.
TEMPERATURE (Temp)	The temperature of the reaction.
YIELD_PERCENT (Y.P.)	Yields given in percent values.
YIELD_OTHER (Y.O.)	Yields provided in other units than %.
WORKUP	A manipulation required to isolate and purify the product of a chemical reaction.
REACTION STEP	An event that converts starting materials into a product.

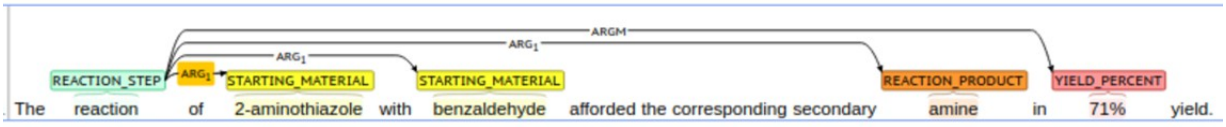

Figure 5: An example of a BRAT annotated sentence from the CLEF-2020 dataset