NLPatVCU: CLEF 2022 ChEMU Shared Task System Description

Darshini Mahendran[1], Christina Tang[1] and Bridget T. McInnes[1]

[1] *Virginia Commonwealth University, 601 West Main Street, Richmond VA 23220, USA*

Abstract

This paper describes our team's participation in the Tracks 1a & 1b from Cheminformatics Elsevier Melbourne University (ChEMU) 2022 Challenge that focuses on information extraction in chemical patents. We discuss our systems: MedaCy, a python-based supervised multi-class entity recognition system for Named Entity Recognition (NER), and RelEx, a python-based relation extraction system for Event Extraction (EE). Our best model for Task 1a obtained an overall exact precision, recall, and F_1 score of 0.73, 0.81, and 0.77, respectively, and relaxed precision, recall, and F_1 score of 0.83, 0.92, and 0.87 respectively. Our best model for Task 1b obtained an overall exact precision, recall, and F_1 score of 0.82, 0.68, and 0.75, respectively, and a relaxed precision, recall, and F_1 score of 0.88, 0.73, and 0.79 respectively.

Keywords

Named Entity Recognition, Event Extraction, Information Extraction

1. Introduction

Chemical patents contain information about chemicals and their reactions. This information can be used to discover new chemicals and synthetic pathways [1] [2]. However, manually combing this literature is time-consuming. Therefore there is an increased need for tools to automatically extract chemicals and their reactions. The process referred to as chemical reaction detection [3] consists of two main components. The first is to identify the different components of the reaction, and the second is to identify the relation between the components.

The CLEF 2022 Cheminformatics Elsevier Melbourne University (ChEMU) Task 1a aims to create systems to perform Named Entity Recognition (NER) over chemical patents as the first step in chemical reaction detection. Specifically, the goal of this task is to automatically identify chemical compounds based on the role they play in a reaction, as well as other relevant information such as yield and temperature. The CLEF 2022 ChEMU Task 1b aims to create systems to perform Event Extraction (EE) over the entities to identify the individual steps in the reaction.

In this paper, we describe our participation in the CLEF 2022 ChEMU Task 1a and 1b Challenge. For this challenge, we used our python framework MedaCy to automatically identify the reaction components; and RelEx's GCN-BERT to automatically link the trigger words with the

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

mahendrand@vcu.edu (D. Mahendran); ctang2@vcu.edu (C. Tang); btmcinnes@vcu.edu (B. T. McInnes)
0000-0002-4950-6876 (D. Mahendran); 0000-0001-6204-0129 (C. Tang); 0000-0003-2297-6672 (B. T. McInnes)

CEUR Workshop Proceedings (CEUR-WS.org)

experimental parameters to provide the sequence of steps within the reaction. MedaCy [4] contains a number of supervised multi-label sequence classification algorithms for NER. RelEx's GCN-BERT [5] utilizes Graph Convolutional Neural Networks (GCNs). Our best model for Task 1a obtained an overall exact precision, recall, and F_1 score of 0.73, 0.81, and 0.77 respectively, and relaxed precision, recall, and F_1 score of 0.83, 0.92, and 0.87 respectively. Our best model for Task 1b obtained an overall exact precision, recall, F_1 score of 0.82, 0.68, and 0.75 respectively, and a relaxed precision, recall, and F_1 score of 0.88, 0.73, and 0.79 respectively.

2. Data

Table 1
Definitions of entity types, trigger words, and relation types of ChEMU 2022 dataset [6]

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
ARG1	The elation between an event trigger word and a chemical compound
ARGM	The relation between an event trigger word and a temperature, time, or yield entity

The ChEMU 2022 data corpus [7] includes chemical entities and events that explain the sequence of steps that leads to a chemical reaction to an end product. It contains 1500 chemical snippets sampled from 180 English document patents from the European Patent Office and the United States Patent and Trademark Office [8]. Each snippet holds a detailed description of chemical reactions.

Entities of this dataset are divided into four categories [6]: (1) chemical compounds that are involved in a chemical reaction; (2) conditions under which a chemical reaction is carried out; (3) yields obtained for the final chemical product; and (4) example labels that are associated with reaction specifications. The four categories are further divided into a total of ten entity types.

The compound category defines five roles a chemical compound can play within a chemical reaction. Conditions category and yield category each include two entity types.

A chemical reaction step involves an action and one or more chemical compounds on which the action takes effect [6]. The action is also linked to the conditions under which the action is carried out and the resultant yields from the action. Relations form between actions (trigger words) and all arguments involved in the reaction steps, such as chemical compounds, conditions, and yields. The ARG1 event label corresponds to 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 definitions of the entity types, trigger words, and relation types.

Table 2
Number of entity types and trigger words in the training data and their event relations

			Training set		Development Set	
Events	**Entities**	**Instances**	**REACTION_STEP**	**WORKUP**	**REACTION_STEP**	**WORKUP**
ARG1	EXAMPLE_LABEL	886	-	-	-	-
	REACTION_PRODUCT	2052	1101	11	719	7
	STARTING_MATERIAL	1754	1747	4	1122	1
	REAGENT_CATALYST	1281	1272	-	789	-
	SOLVENT	1140	1134	4	667	3
	OTHER_COMPOUND	4640	161	4097	105	2661
ARGM	YIELD_PERCENT	955	937	1	688	-
	YIELD_OTHER	1061	1043	2	602	-
	TIME	1059	839	81	569	46
	TEMPERATURE	1515	813	242	473	140
Triggers	REACTION_STEP	3815				
	WORKUP	3053				

3. Methods

This section describes the underlying methodology of our system for Tasks 1a and 1b.

3.1. Task 1a: Named Entity Recognition

To identify the experimental parameters and triggers from the data, we use our MedaCy NER package, which was previously trained for chemical reactions [9]. In this section, we describe the two NER algorithms we evaluated for this challenge.

BiLSTM + CRF: Long Short-Term Memory(LSTMs) units [10] are a form of Recurrent Neural Networks (RNNs). LSTMs take as their input, not just the current input example but also what they have seen in the past. This allows them to connect previous observations over arbitrarily long distances. They incorporate the functionality to identify what information should be passed to the next LSTM cell and what information should not, allowing only relevant information to be passed on. For bidirectional LSTMs (BiLSTMs), data are processed in both directions with two separate hidden layers, which are then fed forward into the same output layer. This allows the system to exploit context in both directions. In this work, we feed the result of the BiLSTM through a linear-chain Conditional Random Fields(CRF) to calculate the final class probability for each token in the sequence. Here, we use the word embeddings derived from word2vec

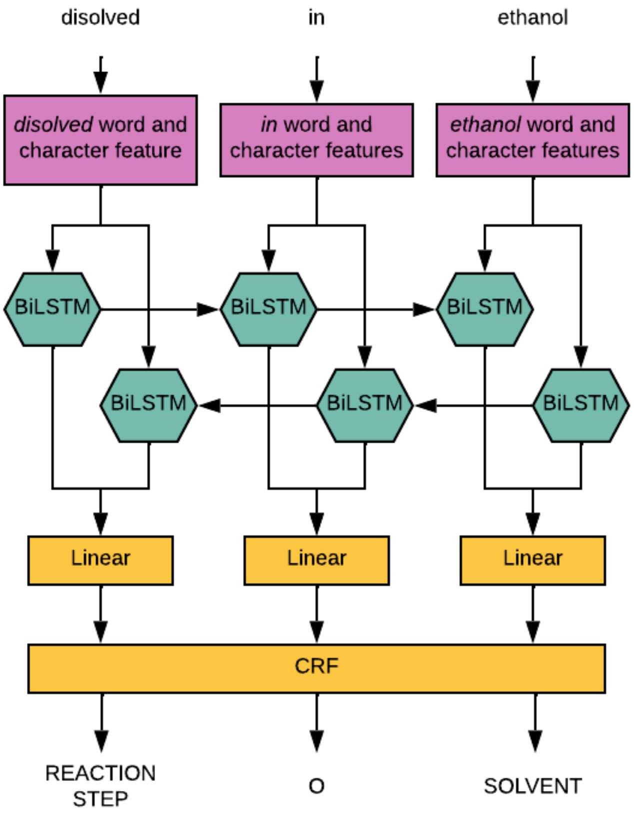
disolved
in
ethanol
disolved word and character feature
in word and character features
ethanol word and character features
BiLSTM
BiLSTM
BiLSTM
BiLSTM
BiLSTM
BiLSTM
Linear
Linear
Linear
CRF
REACTION STEP
O
SOLVENT

Figure 1: BiLSTM+CRF architecture for NER.

[11] in combination with character embeddings [12] as input into our BiLSTM+CRF model. The word2vec embeddings are derived from a neural network that learns a representation of a word-word co-occurrence matrix. At a high level, it is a neural network that learns a series of weights (a hidden layer within the neural network) that either maximizes the probability of a word given the surrounding context, referred to as the Continuous Bag Of Words (CBOW) approach, or maximizes the probability of the context given the word, referred to as the Skip-gram approach. The character embeddings are learned using a BiLSTM and concatenated onto the word2vec embeddings. Figure 1 provides a high level over view of the architecture.

BERT: The Bidirectional Encoder Representations from Transformers (BERT) is a contextualized language model trained over a large corpus for the tasks of masked language modeling and next sentence prediction. Devlin, et al. [13] showed that this pre-trained model could be fine-tuned for other NLP tasks, including NER, by adding a simple classification layer. Our system consists of and Alternate WordPiece Labeling Component, BioBERT [14] with a Linear Classification Layer, and a CRF output layer. The BERT tokenization splits tokens into “WordPieces”, creating

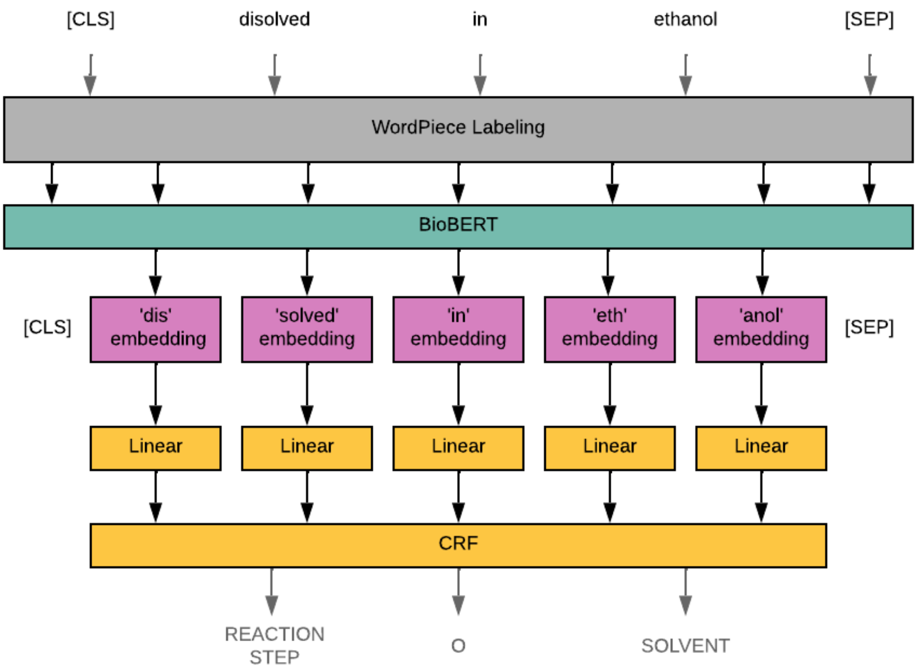

Figure 2: BERT architecture for NER.

a complication when doing token-level classification like NER. As recommended by Devlin, et al. [13], we classified the first WordPiece by masking the rest and applying an "X" label. BioBERT [14] is a BERT model that was pre-trained over PubMed abstracts and full-text articles from PubMed Central [1]. Lastly, a CRF is assigned a class probability for each subword in the sequence to incorporate the interdependence between labels into the model. Figure 2 provides a high level over view of the architecture.

3.2. Task 1b: Event Extraction

In this section, we describe our EE system, which utilizes GCN in combination with the BERT encoder.

GCN-BERT: To identify the chemical arguments between the trigger words and the entities, we use RelEx's GCN-BERT, a python-based Relation Extraction Framework developed to identify relations between two entities. BERT utilizes positional information to capture the local contextual information within a sentence, whereas GCN captures the global context information by performing convolution operations on neighbor nodes in a graph. Here, we combine BERT

[1] https://huggingface.co/monologg/biobert_v1.1_pubmed

with GCN to better represent local contextual information and global association information between words. We treat the EE 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, and (2) Negative class - there is no relation between the trigger word and the entity (no-relation).

To determine the relation between two entities, we first locate the sentence where the trigger word-entity pair is located. A sentence can have multiple such trigger word-entity pairs; therefore, we need to represent the targeted trigger word-entity pair in a distinguishable way from other pairs. Here, we replace the non-targeted trigger word-entity pairs with 'X' from the input sentence except for the targeted trigger word-entity pair.

BERT captures the contextual information within a sentence or document locally however it fails to capture the global information. On the other hand, GCN captures the global information between the nodes but may fail to capture local information. Therefore, we proposed a novel architecture that combines BERT with GCN to benefit from capturing both local and global information and allowing them to influence mutually and build together a final representation for classification.

First, we extract the sentence where the trigger word-entity pair is located. Then, we use the BERT tokenizer for word tokenization. BERT uses a hybrid of word-level and character-level tokenization to handle the Out-Of-Vocabulary (OOV) words. After sentence tokenization, we build a vocabulary map mapping the unique tokens to integers. Second, we generate a vocabulary graph G =(V, E) where 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 Point-wise Mutual Information(PMI). PMI represents a quantified measure for how likely we are to see the two words co-occur, given their individual probabilities, and relative to the case where the two words do not correlate. We calculate PMI shown in the Equation 1 using the number of observations for word x, the number of observations for word y, the probabilities for the words x and y, and the co-occurrence of x and y. A positive PMI value indicates a high semantic correlation between words, whereas a negative PMI value indicates little or no semantic correlation.

$$PMI(x,y) = \log\left(\frac{P(x,y)}{P(x)P(y)}\right) \tag{1}$$

Next, we pass the graph through a two-layer GCN, which performs multiple levels of convolution to capture the global information between the nodes that are not connected directly and generates the graph embeddings. Third, we combine the mapped word indices with the generated graph embeddings. BERT is a transformer that applies multi-head self-attention. BERT architecture initially takes a token, segment, and position embeddings of the input text. BERT converts the input token embeddings into a vector representation at first. At this point, we concatenate our graph embeddings vector with the converted vector representation. Then, BERT applies the bidirectional training, taking the previous and next tokens into account and producing a representation for the input sequence. Finally, the final embedding representation is fed into a fully connected layer for classification. Fig. 3 shows the structure of our RelEx's GCN-BERT approach.

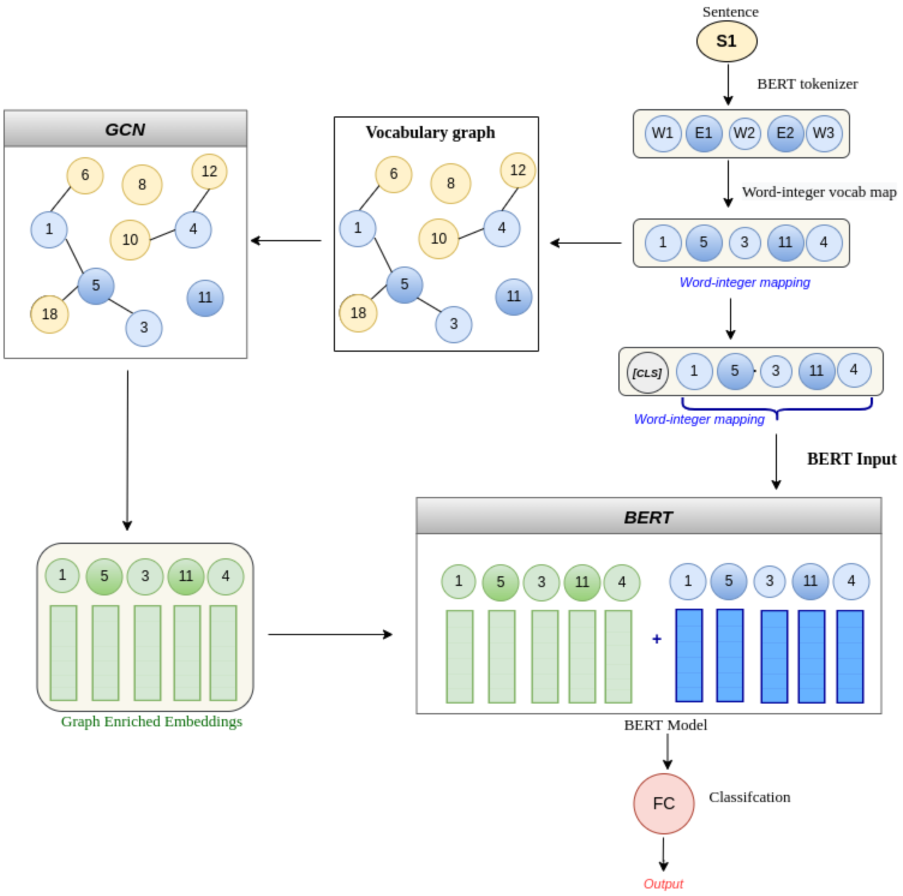

Figure 3: Structure for the RelEx's GCN-BERT approach.

3.3. Experimental Details

Word Embeddings: We utilized ChemPatent embeddings [8] trained over a collection of 84,076 full patent documents (1B tokens) in our methods.

BERT: We used BioBERT [14] as the base encoder for the transformer-based models. The transformer encoder used to further refine the BERT embeddings was implemented in PyTorch [15].

MedaCy: We used a PyTorch [15] implementation of the BiLSTM+CRF model. The models were trained for 40 epochs and optimized using stochastic gradient descent. A window size of three generated the best results. The source code is available in the MedaCy public repository [2].

RelEx: We used PyTorch [15] for the implementation of GCN-BERT. We experimented with

[2] https://github.com/NLPatVCU/medaCy

different sliding window sizes, filter sizes, loss functions for fine-tuning. PyTorch-Transformers[3] by HuggingFace Team to build the BERT model. The source code is available in the RelEx-GCN public repository [4].

3.4. Evaluation

In the results, for both Tasks 1a and 1b, we report the precision, recall, and F_1 scores. Precision is the ratio between correctly predicted mentions over the total set of predicted mentions for a specific entity; recall is the ratio of correctly predicted mentions over the actual number of mentions, and F_1 is the harmonic mean between precision and recall.

For Task 1a and 1b, we report both the exact and relaxed results. Two annotations are considered equal only if they have the same tag with exactly matching spans during the exact evaluation. In contrast, with the relaxed evaluation, two annotations are considered equal if they share the same tag, and their spans overlap each other.

4. Results and Discussion

In this section, we discuss the results for Task 1a and 1b for both development and test sets.

4.1. Task 1a: Named Entity Recognition

Table 3 shows the exact precision, recall and F_1 results over the development set. The results show that our BiLSTM+CRF consistently obtained higher precision, recall, and F_1 scores than BERT implementation. The results also show that the BiLSTM+CRF obtained higher than 95% precision and recall for all entities except for STARTING_MATERIAL.

Table 3
The exact Precision (P), Recall (R), and F_1 results for the development set using BiLSTM+CRF and BERT

	BiLSTM+CRF			**BERT**		
Entity	**P**	**R**	F_1	**P**	**R**	F_1
EXAMPLE_LABEL	0.98	0.99	0.99	0.94	0.97	0.95
OTHER_COMPOUND	0.99	0.98	0.99	0.94	0.95	0.94
REACTION_PRODUCT	0.97	0.92	0.95	0.96	0.66	0.78
REAGENT_CATALYST	0.99	0.99	0.99	0.81	0.90	0.85
SOLVENT	0.99	0.99	0.99	0.90	0.89	0.89
STARTING_MATERIAL	0.99	0.65	0.78	0.93	0.59	0.72
TEMPERATURE	0.99	1.00	0.99	0.99	0.98	0.99
TIME	0.99	1.00	0.99	1.00	0.99	0.99
YIELD_OTHER	0.96	0.99	0.98	0.91	0.97	0.94
YIELD_PERCENT	0.99	0.93	0.96	1.00	0.97	0.99
System	**0.99**	**0.93**	**0.96**	**0.94**	**0.85**	**0.89**

Tables 4 and 5 show the results over the test data for the BiLSTM+CRF and BERT respectively. Similarly, the BiLSTM+CRF obtained higher precision, recall, and F_1 scores than BERT. We

[3] https://pytorch.org/hub/huggingface_pytorch-transformers/
[4] https://github.com/NLPatVCU/RelEx-GCN

believe this is due to the embedding representations for the BiLSTM+CRF being trained on patents while our BERT implementation was trained over PubMed journal articles.

Table 4
Precision (P), Recall (R), and F_1 results for the test set using BiLSTM+CRF trained over training data with ChemPatent embeddings

	Exact			**Relaxed**		
Entity	**P**	**R**	F_1	**P**	**R**	F_1
EXAMPLE_LABEL	0.94	0.96	0.96	0.95	0.97	0.96
OTHER_COMPOUND	0.86	0.84	0.85	0.92	0.90	0.91
REACTION_PRODUCT	0.39	0.49	0.43	0.69	0.87	0.77
REAGENT_CATALYST	0.84	0.82	0.83	0.88	0.86	0.87
SOLVENT	0.90	0.93	0.91	0.92	0.94	0.93
STARTING_MATERIAL	0.44	0.72	0.55	0.53	0.88	0.66
TEMPERATURE	0.94	0.96	0.95	0.97	0.99	0.98
TIME	0.84	0.86	0.85	0.98	0.99	0.99
YIELD_OTHER	0.78	0.76	0.77	0.94	0.91	0.93
YIELD_PERCENT	0.93	0.98	0.95	0.95	0.99	0.97
System	**0.73**	**0.81**	**0.77**	**0.83**	**0.92**	**0.87**

Table 5
Precision (P), Recall (R), and F_1 results for the test set using BERT

	Exact			**Relaxed**		
Entity	**P**	**R**	F_1	**P**	**R**	F_1
EXAMPLE_LABEL	0.93	0.95	0.94	0.94	0.95	0.95
OTHER_COMPOUND	0.86	0.80	0.83	0.93	0.86	0.90
REACTION_PRODUCT	0.42	0.64	0.51	0.59	0.91	0.72
REAGENT_CATALYST	0.69	0.64	0.66	0.79	0.73	0.76
SOLVENT	0.83	0.89	0.86	0.86	0.92	0.88
STARTING_MATERIAL	0.34	0.59	0.43	0.49	0.85	0.62
TEMPERATURE	0.95	0.97	0.96	0.98	0.99	0.98
TIME	0.84	0.85	0.85	0.98	1.00	0.99
YIELD_OTHER	0.80	0.76	0.78	0.95	0.90	0.92
YIELD_PERCENT	0.92	0.99	0.96	0.93	1.00	0.96
System	**0.70**	**0.79**	**0.74**	**0.79**	**0.90**	**0.84**

Table 6 shows the true positive (tp), false positive (fp), true negative (tn) and false negative (fn) for each of the entities using the BiLSTM+CRF and BERT systems. The results show that most of the false negative and false positive issues are with the Chemical entities (OTHER_COMPOUND, STARTING_MATERIAL, REACTION_PRODUCT). These entities have a high amount of ambiguity within their mentions as a chemical can be the starting material in one experiment and the reaction product in the other.

4.2. Task 1b: Event Extraction

Table 7 shows the precision, recall, and F_1 scores obtained over the development set for EE. Here we used our RelEx's GCN-BERT trained over the ChemPatent embeddings with the trigger words identified using MedaCy's BiLSTM+CRF trained over the ChemPatent embeddings.

Table 6
Error analysis for both BiLSTM+CRF and BERT systems for the exact match evaluation over test data.

	BiLSTM+CRF			**BERT**		
Entity	tp	**fp**	**fn**	tp	**fp**	**fn**
EXAMPLE_LABEL	329	20	14	325	23	18
OTHER_COMPOUND	1384	227	270	1319	217	335
REACTION_PRODUCT	400	626	418	526	725	292
REAGENT_CATALYST	364	67	78	282	125	160
SOLVENT	360	38	29	347	69	42
STARTING_MATERIAL	469	593	178	380	738	267
TEMPERATURE	586	37	23	588	30	21
TIME	364	68	61	363	69	62
YIELD_OTHER	317	87	98	316	80	99
YIELD_PERCENT	347	25	6	350	29	3
System	4796	2015	1299	4796	2105	1299

GCN-BERT obtained an overall precision of 0.85, recall of 0.94, and F_1 score of 0.89 with the development data. The system obtained higher F_1 scores with the REACTION_STEP classes than with WORKUP classes. This is mainly because the REACTION_STEP classes have more training instances than most WORKUP classes. Also, we can see that the performance of each Trigger word-Entity pair is proportional to the number of instances in the training set. For example, classes with a higher number of instances, such as REACTION_STEP-STARTING_MATERIAL, REACTION_STEP-REAGENT_CATALYST, REACTION_STEP-YIELD_OTHER achieved higher F_1 scores. In contrast, classes that have a moderate number of instances comparatively, such as REACTION_STEP-OTHER_COMPOUND, WORKUP-TEMPERATURE, achieved moderate F_1 scores. Trigger word-Entity pairs such as WORKUP-SOLVENT and WORKUP-STARTING_MATERIAL that have very few instances obtained an F_1 score of zero.

Table 8 shows both exact and relaxed precision, recall, and F_1 scores obtained over the Test set for EE. Here we used our RelEx's GCN-BERT trained over the ChemPatent embeddings with the trigger words identified using MedaCy's BiLSTM+CRF trained over the ChemPatent embeddings.

GCN-BERT obtained an overall exact precision of 0.82, recall of 0.68, and F_1 score of 0.75 and relaxed precision of 0.88, recall of 0.73, and F_1 score of 0.79 with the test data. From the results we can see similar observations between the results of the test set and the development set. Here also the REACTION_STEP classes performed better than the WORKUP classes. Most of the classes performed better when evaluated in the relaxed mode. This is because the NER model may not always find the complete span of the entity when performing inference.

Table 9 shows a detailed error analysis of the GCN-BERT system over the test set during the exact match evaluation. Here, we report the number of true positives (tp), false positives (fp), and false negatives (fn) and also "fpm" and "fnm", two metrics that represent the number of false positives and false negatives, of which the corresponding entities are missing. The confusion matrix allows visualizing the performance of an algorithm. We can see that class imbalance played a role in the miss annotations of the events. More importantly, we can see that for the chemical entities, we have a high number of false negatives indicating that the system over generating relations and indicating relations between chemicals and trigger words

Table 7
Precision (P), Recall (R) and F_1 results for the development set using GCN-BERT system with trigger words identified using MedaCy's BiLSTM + CRF trained with ChemPatent embeddings

Argument	Trigger	Entity	# Train	P	R	F_1
ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.73	0.48	0.58
		REACTION_PRODUCT	1101	0.97	0.98	0.98
		REAGENT_CATALYST	1272	0.88	0.95	0.91
		SOLVENT	1134	0.86	0.94	0.90
		STARTING_MATERIAL	1747	0.87	0.93	0.90
	WORKUP	OTHER_COMPOUND	4097	0.89	0.97	0.93
		REACTION_PRODUCT	11	0.00	0.00	0.00
		SOLVENT	4	0.00	0.00	0.00
		STARTING_MATERIAL	4	0.00	0.00	0.00
ARGM	REACTION_STEP	TEMPERATURE	813	0.53	0.91	0.67
		TIME	839	0.75	0.92	0.83
		YIELD_OTHER	1043	0.95	0.99	0.97
		YIELD_PERCENT	937	0.95	0.99	0.97
	WORKUP	TEMPERATURE	242	0.75	0.73	0.74
		TIME	81	0 .00	0.00	0.00
System				**0.85**	**0.94**	**0.89**

Table 8
Precision (P), Recall (R) and F_1 results for the test set using GCN-BERT system with trigger words identified using MedaCy's BiLSTM + CRF trained with ChemPatent embeddings

Argument	Trigger	Entity	Strict			Relaxed		
			P	R	F_1	P	R	F_1
ARG1	REACTION_STEP	OTHER_COMPOUND	0.52	0.53	0.53	0.52	0.53	0.53
		REACTION_PRODUCT	0.77	0.54	0.63	0.91	0.65	0.76
		REAGENT_CATALYST	0.84	0.68	0.75	0.86	0.70	0.77
		SOLVENT	0.82	0.74	0.78	0.84	0.76	0.79
		STARTING_MATERIAL	0.71	0.59	0.65	0.79	0.66	0.72
	WORKUP	OTHER_COMPOUND	0.88	0.77	0.82	0.93	0.82	0.87
		REACTION_PRODUCT	0.00	0.00	0.00	0.00	0.00	0.00
		SOLVENT	0.00	0.00	0.00	0.00	0.00	0.00
		STARTING_MATERIAL	0.00	0.00	0.00	0.00	0.00	0.00
ARGM	REACTION_STEP	TEMPERATURE	0.84	0.54	0.66	0.86	0.55	0.67
		TIME	0.74	0.69	0.72	0.87	0.81	0.84
		YIELD_OTHER	0.92	0.72	0.81	0.95	0.74	0.83
		YIELD_PERCENT	0.97	0.90	0.93	0.97	0.91	0.94
	WORKUP	TEMPERATURE	0.70	0.47	0.56	0.70	0.47	0.56
		TIME	0.45	0.35	0.39	0.45	0.35	0.39
System			**0.82**	**0.68**	**0.75**	**0.88**	**0.73**	**0.79**

that are inaccurate.

5. Conclusion

In this paper, we describe our participation in the CLEF 2022 ChEMU Task 1a and 1b Challenge. For Task 1a, we evaluated two NER models to extract chemical reaction components from patents (1) a BiLSTM+CRF model over ChemPatent embeddings and (2) a BERT transformer model using BioBERT. Our results show that the BiLSTM+CRF outperformed the BERT model. We believe this is due to the embedding representations for the BiLSTM+CRF being trained on patents while our BERT implementation was trained over PubMed journal articles. In the future, further training in the BioBERT based model over patents may increase the BERT scores,

Table 9
Error analysis for the GCN-BERT system with trigger words identified using MedaCy's BiLSTM + CRF trained with ChemPatent embeddings

Argument	Trigger	Entity	tp	fp	fn	fpm	fnm
ARG1	REACTION_STEP	OTHER_COMPOUND	39	36	34	9	21
		REACTION_PRODUCT	222	68	188	67	178
		REAGENT_CATALYST	298	55	140	32	89
		SOLVENT	292	64	102	32	49
		STARTING_MATERIAL	386	161	263	120	196
	WORKUP	OTHER_COMPOUND	1124	154	332	117	246
		REACTION_PRODUCT	0	0	2	0	1
		REAGENT_CATALYST	0	0	3	0	3
		SOLVENT	0	0	1	0	1
		STARTING_MATERIAL	0	0	1	0	1
ARGM	REACTION_STEP	TEMPERATURE	262	50	226	29	45
		TIME	262	91	117	67	89
		YIELD_OTHER	294	26	115	26	104
		YIELD_PERCENT	316	11	34	10	16
	WORKUP	TEMPERATURE	62	26	71	3	36
		TIME	13	16	24	7	15
		YIELD_OTHER	0	0	2	0	1
		YIELD_PERCENT	0	0	1	0	0
System			3570	758	1656	519	1091

or utilizing ChemicalBERT rather than BioBERT.

For Task 1b, we combined BERT with GCN to integrate the local contextual and global information between the words. We replaced the non-targeted trigger word-entity pairs with 'X' except for the targeted trigger word-entity pair in the input sentences to distinguish the trigger word-entity pairs. The results showed that it performed reasonably for REACTION_STEP trigger words but less for WORKUP, partly due to class imbalance. However, we also noted that it over-generated the relations when linking trigger words to their chemical, which requires further investigation. In the future, we plan to investigate expanding our system to perform multi-class classification and benchmark against different datasets. And also to try different trigger word-entity pair representations to efficiently represent the trigger word-entity pair in a sentence.

Acknowledgments

We want to thank Jorge Vargas for the initial development of the NER figures.

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

References

[1] W. Bort, I. I. Baskin, P. Sidorov, G. Marcou, D. Horvath, T. Madzhidov, A. Varnek, T. Gimadiev, R. Nugmanov, A. Mukanov, Discovery of novel chemical reactions by deep generative recurrent neural network (2020).

[2] K. Wang, L. Wang, Q. Yuan, S. Luo, J. Yao, S. Yuan, C. Zheng, J. Brandt, Construction of a

generic reaction knowledge base by reaction data mining, Journal of Molecular Graphics and Modelling 19 (2001) 427–433.

[3] H. Yoshikawa, D. Q. Nguyen, Z. Zhai, C. Druckenbrodt, C. Thorne, S. A. Akhondi, T. Baldwin, K. Verspoor, Detecting chemical reactions in patents (2019).

[4] S. Farnsworth, G. Gurdin, J. Vargas, A. Mulyar, N. Lewinski, B. T. McInnes, Extracting experimental parameter entities from scientific articles, Journal of Biomedical Informatics (2021) 103970.

[5] D. Mahendran, C. Tang, B. McInnes, Graph convolutional networks for chemical relation extraction, Proceedings of the Semantics-enabled Biomedical Literature Analytics (SeBiLAn) (2022).

[6] J. He, D. Q. Nguyen, S. A. Akhondi, C. Druckenbrodt, C. Thorne, R. Hoessel, Z. Afzal, Z. Zhai, B. Fang, H. Yoshikawa, et al., An extended overview of the clef 2020 chemu lab, in: the Conference and Labs of the Evaluation Forum (CLEF), 22-25 September 2020, 2020.

[7] J. He, D. Q. Nguyen, S. A. Akhondi, C. Druckenbrodt, C. Thorne, R. Hoessel, Z. Afzal, Z. Zhai, B. Fang, H. Yoshikawa, et al., Chemu 2020: Natural language processing methods are effective for information extraction from chemical patents, Frontiers in Research Metrics and Analytics 6 (2021) 12.

[8] D. Q. Nguyen, Z. Zhai, H. Yoshikawa, B. Fang, C. Druckenbrodt, C. Thorne, R. Hoessel, S. A. Akhondi, T. Cohn, T. Baldwin, et al., Chemu: Named entity recognition and event extraction of chemical reactions from patents, in: European Conference on Information Retrieval, Springer, 2020, pp. 572–579.

[9] D. Mahendran, G. Gurdin, N. Lewinski, C. Tang, B. T. McInnes, Identifying chemical reactions and their associated attributes in patents, Frontiers in Research Metrics and Analytics (2021) 42.

[10] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991 (2015).

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[12] M. Gridach, Character-level neural network for biomedical named entity recognition, Journal of biomedical informatics 70 (2017) 85–91.

[13] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[14] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, CoRR abs/1901.08746 (2019). URL: http://arxiv.org/abs/1901.08746. arXiv:1901.08746.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alche-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.