Detecting Chemical Reactions in Patents

Hiyori Yoshikawa[1,3], **Dat Quoc Nguyen**[1], **Zenan Zhai**[1], **Christian Druckenbrodt**[2], **Camilo Thorne**[2], **Saber A. Akhondi**[2], **Timothy Baldwin**[1], **Karin Verspoor**[1]
[1]The University of Melbourne, Australia; [2]Elsevier; [3]Fujitsu Laboratories Ltd.
[1]{hiyori.yoshikawa,dqnguyen,zenan.zhai,tbaldwin,karin.verspoor}@unimelb.edu.au
[2]{c.druckenbrodt,c.thorne.1,s.akhondi}@elsevier.com

Abstract

Extracting chemical reactions from patents is a crucial task for chemists working on chemical exploration. In this paper we introduce the novel task of detecting the textual spans that describe or refer to chemical reactions within patents. We formulate this task as a paragraph-level sequence tagging problem, where the system is required to return a sequence of paragraphs that contain a description of a reaction. To address this new task, we construct an annotated dataset from an existing proprietary database of chemical reactions manually extracted from patents. We introduce several baseline methods for the task and evaluate them over our dataset. Through error analysis, we discuss what makes the task complex and challenging, and suggest possible directions for future research.

1 Introduction

Chemical patents are a crucial resource for chemical research and development activities. In fact, many compounds are reported first in patents and only a small fraction of them appears in the chemical literature after 1 to 3 years (Senger et al., 2015), meaning that chemists habitually search over both academic papers and patent databases. Moreover, as the number of chemical patents awarded each year is ever-increasing, there is an increasing urgency to perform patent searches to establish the novelty of chemical compounds (Akhondi et al., 2014). Text mining methods are a vital tool in this process, enabling a significant reduction in associated time and effort.

Most previous research in text mining of chemical information has focused on named entity recognition (NER) of chemical concepts, and several publicly-available NER corpora have been derived from both scientific literature (Kim et al., 2003; Corbett et al., 2007; Krallinger et al., 2015) and chemical patents (Akhondi et al., 2014). Some studies have also addressed relation extraction between chemical entities and other concepts such as protein and diseases (Wei et al., 2015; Krallinger et al., 2017).

However, there has been limited work on automatically extracting chemical reactions from patents. A chemical patent usually contains a description of chemical reactions that are relevant to its claims. Figure 1 shows an example of a chemical reaction description. Generally speaking, a chemical reaction is a process where a set of chemical compounds is transformed into another set of chemical compounds. A reaction description may include the source chemical compounds, solvents and reagents involved in the reaction, reaction conditions, and materials obtained as a result of the reaction. Despite the fact that such information is crucial for a comprehensive understanding of chemical patents, there are — to the best of our knowledge — few methods or annotated resources that can be used for this purpose.

As a first step to extracting chemical reactions, a filtering step must take place to determine where reactions are described in a patent. In this paper, we introduce this new task of **chemical reaction detection**. The output of this task can be used as the input to (more complex) downstream tasks. For example, consider an event extraction system that extracts every step of a reaction as an individual event. Events in the first reaction of Figure 1 would be: (1) heating 2-pyridine-ethanol, triphenyl phosphine and carbon tetrachloride; (2) the addition of triphenyl phosphine; (3) heating them again; and so on. The application would also include estimating the relevance of chemical compounds to a given reaction, based on their role in the reaction (Akhondi et al., 2019). Such downstream tasks require as input a paragraph sequence corresponding to a reaction, in a repre-

P : *Example 1: Preparation of 2-(4-benzyloxybutyl)pyridine*
P : *2-(4-Benzyloxybutyl)pyridine is prepared in 5 steps according to the following reaction route.*
...
P : *(1) Preparation of 2-(2-chloroethyl)pyridine*
P : *2-Pyridine-ethanol (15.00 g, 122 mmol), triphenyl phosphine (38.40 g, 146 mmol) and carbon tetrachloride (100 mL) were put in a 500-mL flask, and heated under reflux. After 1.5 hours, triphenyl phosphine (9.60 g, 36.6 mmol) was added thereto and further heated for 30 minutes under reflux.*
P : *The reaction liquid was cooled down to room temperature, and then pentane (200 mL) was added thereto, and filtered using a Kiriyama funnel. The resultant filtrate was concentrated to give a crude product (17.07 g). This was distilled under reduced pressure to give 11.16 g (yield 65.8%, purity 97.1%) of 2-(2-chloroethyl)pyridine.* } reaction
P : *(2) Preparation of dimethyl 2-(2-pyridyl)ethylmalonate*
P : *2-(2-Chloroethyl)pyridine (10.35 g, 73.1 mmol), N,N-dimethylformamide (100 mL), dimethyl malonate (14.48 g, 110 mmol) and potassium carbonate (18.18 g, 132 mmol) were put in a 300-mL flask, and stirred. ...* } reaction
...
...

Figure 1: An example of chemical reaction description in a patent document (US20180072670A1). The symbol "P :" stands for the beginning of a paragraph.

sentation that preserves the order of the reaction substeps. Indeed, reactions are complex processes composed of sequentially ordered steps (much like recipes in cooking) and tend to be described sequentially. Thus, we formulate this task as a paragraph-level sequence tagging problem, where the output is a set of paragraphs containing the description of a reaction. Although the task formulation is simple, it is not straightforward to automate as it requires document-level understanding of the patent text, which tends to be highly ambiguous.

To measure the feasibility and identify the key challenges of this new task, we created an annotated dataset and established benchmark results over it, in the form of rule-based and machine learning methods.[1] The dataset is based on the chemical structure data from Reaxys®.[2] The database contains chemical reactions manually extracted from a very large number of patents. The reactions are associated with the patent from which they are extracted. As the primary purpose of the extraction is to populate a database of chemical reactions (some of which are extracted from non-textual sources) and not provide training data for NLP, it is not always possible to completely map back from the data to the source text. However, the large database enabled us to automatically create a potentially very large amount of training data that can be used to train state-of-the-art deep learning methods. For the experiments presented in this paper, we created an annotated corpus from a subset of Reaxys® reactions and filtered out documents with low mapping coverage from the database to text, in an attempt to boost the fidelity of evaluation over that data. This culminated in training, development, and test sets consisting of 143 documents made up of >39,000 paragraphs in total.

2 Related Work

Patents are regarded as an important resource for chemical information, and a large volume of NLP research has focused on them (Fujii et al., 2007; Tseng et al., 2007; Gurulingappa et al., 2013; Southan, 2015; Rodriguez-Esteban and Bundschus, 2016; Akhondi et al., 2019). However, most previous work on chemical information extraction has focused on the NER task of extracting chemical names or chemistry-related concepts from literature (Kim et al., 2003; Corbett et al., 2007; Böhme et al., 2014; Akhondi et al., 2014; Krallinger et al., 2015). Some previous work has attempted to extract not only chemical names but also reaction procedures from the literature (Lawson et al., 2011; Wei et al., 2015; Mayfield et al., 2017; Krallinger et al., 2017). Among them, Lowe (2012) presents an integrated system that detects reaction text from chemical patents, and extracts chemicals and their roles in the corresponding reaction. The system is heavily rule-based and incorporates existing NLP libraries, and was reported to detect reactions with high accuracy. However, evaluation was limited to a small number of good-quality reaction texts, and the performance of the reaction detection sub-task was not evaluated in isolation.

Another line of work with an extensive litera-

[1] Contact the authors for data requests.

[2] Copyright ©2019 Elsevier Limited, except certain content provided by third parties. Reaxys is a trademark of Elsevier Limited.

ture is patent retrieval, where the task is to retrieve patent documents or passages given a query in the form of keywords, a sentence, or a document; Shalaby and Zadrozny (2019) survey this task extensively. A relevant shared task was organized as part of CLEF-IP 2012 (Piroi et al., 2012). In the sub-task titled "Passage Retrieval Starting From Claims", participants were required to extract passages from chemical patents that are relevant to a given claim. The difference between our task and theirs is that the output of our task is *all* chemical reactions mentioned in a given patent, independent of any claim. In addition, the CLEF-IP task does not require the identification of reaction spans. That is, they deal with each passage independently, ignoring ordering.

The proposed task can also be viewed as a text segmentation problem. Koshorek et al. (2018) formulated the text segmentation task on general domain corpora such as Wikipedia as a supervised learning problem, and proposed a two-level bidirectional LSTM model to learn to detect text spans. In particular, they used a softmax layer on top of a standard BiLTSM architecture for segmentation prediction. We experiment with a BiLTSM-CRF architecture as a document-level training method as described in Section 4, i.e. we use a CRF layer to obtain the document-level label sequence, instead of applying a softmax classifier on top of the BiLSTM.

3 Task and Dataset

3.1 Task Formulation

A patent document usually describes the reactions to produce the relevant compounds as part of its claims. As shown in Figure 1, a reaction may involve several steps to obtain the target compounds.[3] In our example, multiple contiguous paragraphs are used to describe a single reaction, and multiple reactions are necessary to obtain the final compounds.

As a reaction consists of a series of sub-steps executed over time, it is important to detect the beginning and the end of each reaction text accurately. Therefore, we define the task as a span detection problem rather than the simpler task of binary classification (i.e., classifying each paragraph as describing (part of) a reaction or not), which would not be able to detect reactions as a whole or capture reaction substructure.

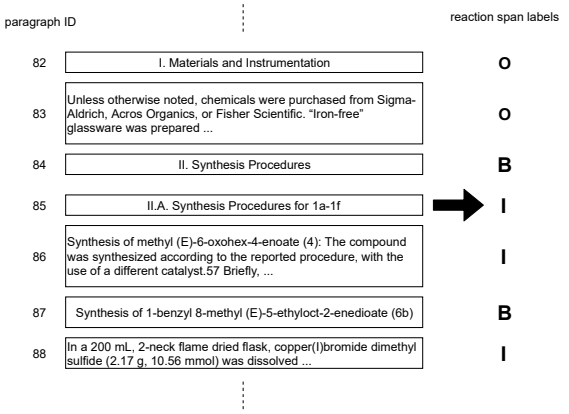

Figure 2: Illustration of our reaction span detection task.

Figure 2 shows an example of an input and gold-standard output of the reaction span detection task. A patent document is given as a sequence of paragraphs. The task is to detect a span of contiguous paragraphs that describe a single chemical reaction. In our corpus, we provide paragraph-level label sequences over paragraphs in patent documents, following the IOB2 tagging scheme (Tjong et al., 1999).

The definition of "reaction spans" in our dataset follows the extraction rules of the original database. In principle, a reaction is extracted from a patent if the requisite information about the reaction (e.g., starting materials, reaction conditions and target compounds) is provided within the patent document and there is no obvious error or inconsistency in the description. Typically a reaction constitutes an example section or a subsection beginning with a title paragraph such as *Example 1*, *Step 1* and *Preparation of [product name]*, as shown in Figure 1. However, it is also commonly the case that an example section contains multiple reactions, in which case they have no title paragraph.

3.2 Data Preparation

Our corpus contains patents from the European Patent Office and the United States Patent and Trademark Office, all of which are written in English and freely available in a digital format. The corpus is based on the Reaxys® database, which contains reaction entries for each patent document manually created by experts in chemistry. A reaction entry has "locations" of the reaction in the corresponding patent document, mostly in terms

[3] In this study we only focus on text data, although information of reactions can also be present in images and tables.

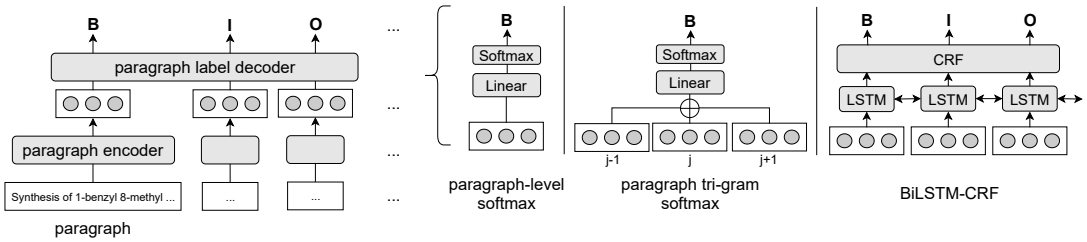

Figure 3: Our model architecture. The left figure illustrates the general architecture of the whole model, while the right figure details the decoder component.

of paragraph IDs (e.g., the reaction entry of *synthesis of methyl (E)-6-oxohex-4-enoate* in Figure 2 has a location property with value 84, 85, 86). We used this location information to automatically label the reaction text spans in the patent text. As the reaction data available in the database is extracted and curated from text, images, and tables based on specific guidelines and hence not directly aligned with NLP requirements, it is not always possible to completely map all the locations from Reaxys® database to text. First, the annotation was performed by a single expert worker for each patent document, without redundancy or explicit post-checking of the extraction. Second, some locations are missing in the original data. For example, as the manual extraction process is at the document level, a reaction is sometimes extracted only once regardless of how many times it is mentioned in the patent. As part of the mapping process, we filtered out potentially-incorrect paragraph spans using a set of rules. For instance, we discarded paragraph spans in which we could not find any of the chemical compounds or related information associated with the corresponding reaction in the database.

For evaluation, we applied the mapping process to a part of the database and selected patent documents with 100% mapping coverage (i.e. all reaction records in the database can be mapped to the text) and split them into training, development, and test partitions. As a result, we obtained training, development, and test sets consisting of 143 documents with >39,000 paragraphs in total. Although the test set is singly-annotated and no inter-annotator agreement is available, we manually checked a small subset to confirm that the annotation quality is sufficiently good to support high-fidelity evaluation. Table 1 presents a breakdown of the dataset.

A patent document consists of three main parts: title/abstract, claims, and description. We extracted the text of the description part, where chemical reactions are described. For simplicity, we only use textual information, and ignore other types of data such as images describing chemical structures. Paragraphs that do not contain text (e.g., tables or references to images) are also discarded.

	Train	Dev	Test
# Documents	86	29	28
# Paragraphs	24,402	7,194	7,481
# Reaction spans	1,787	638	567
# Tokens / Paragraph	72.9	74.5	75.7

Table 1: Composition of the evaluation dataset. "# Tokens / Paragraph" stands for the average number of tokens in a paragraph based on OSCAR4 tokenization (Jessop et al., 2011).

4 Our modeling approach

In this section, we describe our neural approach to reaction span detection. As illustrated in Figure 3, our general model architecture is composed of two main parts: a paragraph encoder and a paragraph label decoder. The encoder represents each paragraph as a vector that is then fed into the label decoder to determine the corresponding `B`/`I`/`O` label of the paragraph.

4.1 Paragraph encoder

We use a paragraph encoder to encode each paragraph p into vector $\boldsymbol{v}_p$. Assume that the paragraph p consists of n tokens $w_1, w_2, \ldots, w_n$. We create a vector $\boldsymbol{e}_i$ to represent the ith word token w_i by concatenating its pre-trained word embedding $\boldsymbol{e}^{\mathrm{WE}}_{w_i}$, contextualized embedding $\boldsymbol{e}^{\mathrm{CW}}_{w_i|p}$, and an optional embedding $\boldsymbol{e}^{\mathrm{FT}}_{f_i}$ representing additional features f_i associated with w_i:

$$\boldsymbol{e}_i = \boldsymbol{e}^{\mathrm{WE}}_{w_i} \oplus \boldsymbol{e}^{\mathrm{CW}}_{w_i|p} \oplus \boldsymbol{e}^{\mathrm{FT}}_{f_i} \quad (1)$$

We then use a BiLSTM paragraph encoder to

learn the paragraph vector $\boldsymbol{v}_p$ from a sequence $\boldsymbol{e}_{1:n}$ of vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$. We compute the hidden states of the LSTMs corresponding to the ith token ($i \in \{1, \ldots, n\}$) as follows:

$$\overrightarrow{\boldsymbol{r}_i} = \overrightarrow{\text{LSTM}_e}(\boldsymbol{e}_{1:i}) \quad (2)$$

$$\overleftarrow{\boldsymbol{r}_i} = \overleftarrow{\text{LSTM}_e}(\boldsymbol{e}_{i:n}) \quad (3)$$

where $\overrightarrow{\text{LSTM}_e}$ and $\overleftarrow{\text{LSTM}_e}$ denote forward and backward LSTMs in the encoder, respectively. We then concatenate the final states of these two LSTMs to obtain the paragraph vector $\boldsymbol{v}_p$:

$$\boldsymbol{v}_p = \overrightarrow{\boldsymbol{r}_n} \oplus \overleftarrow{\boldsymbol{r}_1} \quad (4)$$

4.2 Paragraph label decoder

Assume that we have an input document consisting of m paragraphs $p_{(1)}, p_{(2)}, ..., p_{(m)}$. The decoder will assign a `B/I/O` label to the jth paragraph $p_{(j)}$ based on the input paragraph vector representation(s) $\boldsymbol{v}_{p_{(j)}}$ produced by the encoder as in Equation (4). We explore the following settings.

Paragraph-level softmax classifier: In this setting, we feed each vector $\boldsymbol{v}_{p_{(j)}}$ into a softmax classifier for paragraph label prediction:

$$\mathbf{P}_{(j)} = \text{Softmax}(\mathbf{W}^{\text{PS}} \boldsymbol{v}_{p_{(j)}} + \boldsymbol{b}^{\text{PS}}) \quad (5)$$

where $\mathbf{P}_{(j)} \in \mathbb{R}^3$ is the final output of the network, and $\mathbf{W}^{\text{PS}} \in \mathbb{R}^{3 \times 2k}$ and $\boldsymbol{b}^{\text{PS}} \in \mathbb{R}^3$ are a transformation weight matrix and a bias factor, respectively (here, k is the dimensionality of the $\overrightarrow{\text{LSTM}_e}$ and $\overleftarrow{\text{LSTM}_e}$ hidden states).

Paragraph-trigram softmax classifier: The paragraph-trigram softmax decoder extends the paragraph-level softmax decoder by taking the previous and next paragraphs of $p_{(j)}$ into account.[4] In particular, it is formalized as:

$$\boldsymbol{u}_{p_{(j)}} = \boldsymbol{v}_{p_{(j-1)}} \oplus \boldsymbol{v}_{p_{(j)}} \oplus \boldsymbol{v}_{p_{(j+1)}} \quad (6)$$

$$\mathbf{P}_{(j)} = \text{Softmax}(\mathbf{W}^{\text{PT}} \boldsymbol{u}_{p_{(j)}} + \boldsymbol{b}^{\text{PT}}) \quad (7)$$

where $\mathbf{W}^{\text{PT}} \in \mathbb{R}^{3 \times 6k}$ and $\boldsymbol{b}^{\text{PT}} \in \mathbb{R}^3$ are a transformation weight matrix and a bias factor, respectively.

We train each of the two softmax classifiers by minimizing the model negative log likelihood (i.e. cross-entropy loss). At inference time, we calculate the label probabilities for every paragraph using the learned classifier, and construct the label sequence with the highest joint probability score under the constraint of a valid IOB2 output (i.e. an `I` label must not come right after `O`).

BiLSTM-CRF classifier: In this setting, we use a BiLSTM-CRF architecture (Huang et al., 2015) to capture contextual information across paragraphs as well as label transitions. We first use another BiLSTM to learn latent feature vectors representing input paragraphs from a sequence $\boldsymbol{v}_{p_{(1)}:p_{(m)}}$ of vectors $\boldsymbol{v}_{p_{(1)}}, \boldsymbol{v}_{p_{(2)}}, ..., \boldsymbol{v}_{p_{(m)}}$, and then perform a linear transformation over each latent feature vector. Then output vector $\boldsymbol{h}_j$ for the jth paragraph ($j \in \{1, \ldots, m\}$) is computed as:

$$\overrightarrow{\boldsymbol{r}_j} = \overrightarrow{\text{LSTM}_d}(\boldsymbol{v}_{p_{(1)}:p_{(j)}}) \quad (8)$$

$$\overleftarrow{\boldsymbol{r}_j} = \overleftarrow{\text{LSTM}_d}(\boldsymbol{v}_{p_{(j)}:p_{(m)}}) \quad (9)$$

$$\boldsymbol{r}_j = \overrightarrow{\boldsymbol{r}_j} \oplus \overleftarrow{\boldsymbol{r}_j} \quad (10)$$

$$\boldsymbol{h}_j = \mathbf{W}^{\text{BC}} \boldsymbol{r}_j + \boldsymbol{b}^{\text{BC}} \quad (11)$$

where $\overrightarrow{\text{LSTM}_d}$ and $\overleftarrow{\text{LSTM}_d}$ denote forward and backward LSTMs in the decoder, respectively. $\mathbf{W}^{\text{BC}} \in \mathbb{R}^{3 \times 2l}$ and $\boldsymbol{b}^{\text{BC}} \in \mathbb{R}^3$ are a transformation weight matrix and a bias factor, respectively (here, l is the dimensionality of the $\overrightarrow{\text{LSTM}_d}$ and $\overleftarrow{\text{LSTM}_d}$ hidden states).

Output vectors $\boldsymbol{h}_j$ are fed into a linear-chain CRF layer (Lafferty et al., 2001) for final `B/I/O` paragraph label prediction. A negative joint log likelihood loss is minimized when training, while the Viterbi algorithm is used for decoding.

5 Experimental Settings

5.1 Evaluation Metrics

We evaluate the model perfomance by using span-based metrics as described below.

We find that calculating micro-averaged scores over documents (i.e. the scores over the all spans in the datasets) leads to biased results. This is because the development and test sets consist of a small number of documents and style is consistent within a document, meaning that errors caused by the same writing style tend to accumulate and be overestimated. To mitigate this effect, we evaluate based on *document-level macro-averaged* recall, precision, and F-score, i.e. we compute the scores for each document, and use the average of document-level scores for model selection and final evaluation.

For model selection we use the span-based

[4] When $j = 1$ and $j = m$ we use paragraphs $p_{(0)}$ and $p_{(m+1)}$, each of which consist of two special symbols ⟨`S`⟩ and ⟨/`S`⟩.

scores based on a *strict match* strategy, where an output span is regarded as correct if the beginning and ending paragraphs strictly match those of the gold span. In some practical applications, it also makes sense to understand if the model can identify the *approximate* region where a reaction is described. Thus, for evaluation, we also compute the scores based on a *fuzzy match* strategy, where we calculate the number of matches by counting the number of gold spans that have at least one corresponding predicted output span whose beginning and ending paragraph indices are at most 1 paragraph away from the gold ones.

5.2 Implementation Details

5.2.1 Input Text

We use the text of each paragraph as input, with a maximum length of 128 tokens.[5] For tokenization, we used the OSCAR4 tokenizer (Jessop et al., 2011), as it is customized to chemical text mining.

Equation (1) formulates the input token-level representation for the BiLSTM paragraph encoder in the form of (context-insensitive) word embeddings, contextualized word embeddings, and feature embeddings. For the word embeddings $\boldsymbol{e}^{\mathrm{WE}}_{w_i}$ and contextualized embeddings $\boldsymbol{e}^{\mathrm{CW}}_{w_i|p}$, we employ Word2Vec (Mikolov et al., 2013) and ELMo (Peters et al., 2018), respectively, both pre-trained on chemical patent documents from Zhai et al. (2019). These embeddings are fixed during training. We denote our encoder employing only the pre-trained word and contextualized embeddings (i.e. $\boldsymbol{e}_i = \boldsymbol{e}^{\mathrm{WE}}_{w_i} \oplus \boldsymbol{e}^{\mathrm{CW}}_{w_i|p}$) as **W2V +ELMO**.

We also explore additional learnable feature embeddings $\boldsymbol{e}^{\mathrm{FT}}_{f_i}$ (in Equation 1) based on the output of a chemical named entity recognizer (Zhai et al., 2019). This named entity recognizer was trained on a patent corpus named Reaxys® Gold data (Akhondi et al., 2019). For self-containment purpose we show the entity label set of Reaxys® Gold data in Table 4 in the Appendix. As the label set has two levels of granularity, we use the output in two different ways: coarse-grained (left-hand side of the table) and fine-grained (right-hand side). We first obtain a token-level label sequence in the IOB2 format such as `[O, O, B-chemClass, I-chemClass, O]` for each paragraph and then embed the labels into 5-dimensional vectors $\boldsymbol{e}^{\mathrm{FT}}_{f_i}$. We refer to the paragraph encoder with additional input of coarse-grained NER labels as **W2V +ELMO +NER$_{\mathrm{COARSE}}$**, and the one with the fine-grained labels as **W2V +ELMO +NER$_{\mathrm{FINE}}$**.

[5] We also explored another option where we use only the first sentence of each paragraph. However, the experimental results show better performance when we use the entire paragraph in all cases.

5.2.2 Model Optimization

Our neural models are implemented using the AllenNLP framework (Gardner et al., 2018). With the paragraph-level and paragraph-trigram softmax classifiers, we train model parameters using the training set for 20 epochs and apply early stopping if no improvement in the loss over the development set is observed for 3 continuous epochs. With the BiLSTM-CRF classifier, we train model parameters for 30 epochs, and early stopping is applied after 10 epochs of no improvement. Experimental results with the decoder of paragraph-level softmax on the development set show the highest score when using W2V +ELMO +NER$_{\mathrm{FINE}}$ input representation. Thus, for models with the paragraph-trigram softmax and BiLSTM-CRF decoders, we only explore the use of the W2V +ELMO +NER$_{\mathrm{FINE}}$ input representation.

We use Adam (Kingma and Ba, 2015) as our optimizer for all experiments. We apply a grid search to select optimal hyper-parameters based on the document-level macro F-score over the development set. Table 5 in the Appendix shows the model hyper-parameters used for evaluation. For the model with the BiLSTM-CRF decoder, we initialize parameters of its paragraph encoder with those from the model trained with the paragraph-level softmax classifier, and fine-tune them together with the decoder parameters.

5.3 Baselines

5.3.1 Rule-based baseline

We additionally implement a rule-based baseline, based on common patterns in the first paragraphs of chemical reactions. For example, in the case where a chemical reaction description constitutes an Example part in a patent document, the first paragraph begins with phrases such as *Example 1*, *Step 1*, and *Preparation of [product name]*. We use a list of frequent patterns in the first paragraphs of chemical reaction descriptions to distinguish `B` paragraphs from `I` or `O`. Once a `B` paragraph is detected, we label succeeding paragraphs as `I` until a new `B` paragraph is detected.

Decoder	Input token representation	Strict match			Fuzzy match		
		$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
Rule-based		.205	.381	.241	.278	.482	.319
Logistic		.421	.380	.376	.521	.462	.461
Paragraph-level softmax	W2V +ELMO	.352	.365	.336	.475	.457	.437
	W2V +ELMO +$\text{NER}_{\text{COARSE}}$	.340	.389	.337	.446	.468	.415
	W2V +ELMO +NER_{FINE}	.345	.383	.341	.479	.485	.447
Paragraph-trigram softmax	W2V +ELMO +NER_{FINE}	.513	.488	.482	.643	.573	.574
BiLSTM-CRF	W2V +ELMO +NER_{FINE}	**.658**	**.653**	**.640**	**.718**	**.708**	**.696**

Table 2: Performance of the baseline methods on reaction span detection, in terms of document-level macro-averaged precision ("$\mathcal{P}$"), recall ("$\mathcal{R}$"), and F-score ("$\mathcal{F}_1$"). "$\text{NER}_{\text{COARSE}}$" and "$\text{NER}_{\text{FINE}}$" indicate NER embeddings based on coarse- and fine-grained entity types, respectively.

5.3.2 Feature-based logistic regression

Another baseline we explore is the logistic regression classifier, where the output is calculated as:

$$\mathbf{P}_{(j)} = \text{Softmax}(\mathbf{W}^{\text{L}}\boldsymbol{\phi}_{p_{(j)}} + \boldsymbol{b}^{\text{L}}) \qquad (12)$$

where $\mathbf{P}_{(j)} \in \mathbb{R}^3$ is the final output of the network, $\mathbf{W}^{\text{L}} \in \mathbb{R}^{3\times d}$ and $\boldsymbol{b}^{\text{L}} \in \mathbb{R}^3$ are a transformation weight matrix and a bias factor, and $\boldsymbol{\phi}_{p_{(j)}} \in \mathbb{R}^d$ is the concatenation of the following features:

- word count vectors of $p_{(j-1)}$, $p_{(j)}$ and $p_{(j+1)}$, where the ith entry of each vector is the number of times the ith token appears in the corresponding paragraph, and;
- 2-dimensional one-hot vectors for $p_{(j-1)}$, $p_{(j)}$ and $p_{(j+1)}$ indicating if the paragraph is a heading or a body paragraph.[6]

As preprocessing, we apply lowercasing and lemmatization using the NLTK WordNet Lemmatizer.[7] We also replace all numeric characters with a special character. We further apply the named entity recognizer presented in Section 5.2.1 with the fine-grained label set to replace all chemical names with special tokens corresponding to their entity types. The vocabulary consists of tokens that appear at least three times in the training set.

6 Results and Discussion

6.1 Overall Results

Table 2 shows the overall results on the test set. On the left-hand side of the table we show the results in terms of strict match. The BiLSTM-CRF classifier achieves by far the best score in terms of both precision and recall, indicating that contextual information over paragraphs is key in this task. While the rule-based baseline achieves an exact-match recall of nearly 0.4, the precision is half this value, indicating a high number of false positives. Comparing the three different input features for paragraph-level softmax, we can see that the named entity features improve the recall very slightly, but overall have little impact on results.

The right-hand side of the table shows the results in terms of fuzzy match, where the BiLSTM-CRF model achieves an F-score around 70%.

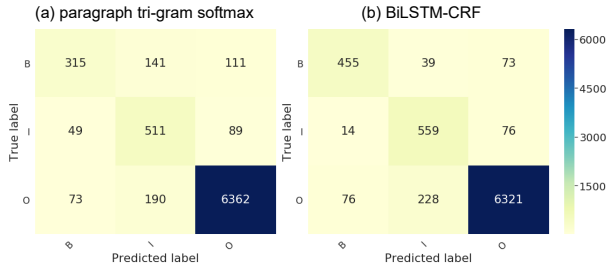

Figure 4: Confusion matrices of the paragraph-trigram softmax vs. BiLSTM-CRF model outputs, based on paragraph-level B/I/O labels.

6.2 Error Analysis

Figure 4 shows the confusion matrices of the model output based on the paragraph-level B/I/O labels. We compare the paragraph-trigram softmax and BiLSTM-CRF models, both with the W2V +ELMO +NER_{FINE} input representation. We can observe that the BiLSTM-CRF output shows a large improvement in distinguishing be-

[6] The paragraph type information is also available in the original database.

[7] https://www.nltk.org/

Ex.	Gold	ParSoftmax	Trigram	BiLSTM-CRF	Text
1	B	B	B	B	*SYNTHETIC EXAMPLE 2*
	I	B	I	I	*Compound A2*
	I	I	I	I	*1,4-dibromonaphthalene (7 g, 24.48 mmol) and 4-cyclohexyl-N-(4-isopropylphenyl)aniline ...*
	I	I	I	I	*1H NMR (400 MHz, CDCl3, δ):*
	I	I	I	I	*δ 8.04-7.985 (dd, 5H); 7.345-7.275 (m, 10H); 7.086-7.028 (m, 23H); 7.028-6.958 (m, 20H).*
2	B	B	B	B	*Example 5. Preparation of SM-5: 2-Hydroxy-4'-trifluoromethylacetophenone*
	I	I	I	I	*Step A. A 200 mL flask was charged with 4'-trifluoromethylacetophenone ...*
	B	I	I	B	*Step B. A 200 mL flask was charged with the crude of 2-Bromo-4'...*
3	B	B	O	B	*Example 5*
	I	I	O	I	*The following Example illustrates a method for producing (R,Z)-dodec-5-ene-1,3-diol ...*
	I	I	O	I	*To a -78 C. solution of oct-1-yne (1.8 equiv.) in THF (0.5M) will be added ...*
	B	I	B	B	*(R)-1-(benzyloxy)dodec-5-yn-3-ol will then taken up in hexane to generate ...*
	B	I	I	B	*(R,Z)-1-(benzyloxy)dodec-5-en-3-ol will be taken up in CH2Cl2 to generate ...*
	B	O	B	B	*Alternatively, the secondary alcohol of (R,Z)-1-(benzyloxy)dodec-5-en-3-ol may be ...*
	B	B	I	B	*(R,Z)-((1-(benzyloxy)dodec-5-en-3-yl)oxy)(tert-butyl)dimethylsilane will be taken up ...*
	B	I	I	B	*To a stirring 2.0M solution of (R,Z)-3-((tert-butyldimethylsilyl)oxy)dodec-5-en-1-ol ...*
4	O	I	B	B	*7-(2-Methylphenylethyl)-Sancycline*
	O	I	I	I	*7-(2-Methylphenylethynyl)-sancycline (1 mmol) was taken in saturated ...*
	B	I	B	B	*9-(4'-Acetyl phenyl) Minocycline*
	I	I	I	I	*In a clean, dry reaction vessel, was placed 9-iodominocycline (0.762 mmoles) ...*
	B	I	B	I	*7-(n-Propyl)-Sancycline*
	I	I	I	I	*7-propynyl sancycline was dissolved in a saturated methanol hydrochloric acid solvent. ...*
5	B	B	B	B	*Example 25: Synthesis of Compound 15-Br-Boc*
	I	I	I	I	*In a three-necked flask, compound 13-Br (3.4 g, 5.675 mmol), DMAP (0.139 g, ...*
	O	B	B	B	*Example 26: Synthesis of Compound 16-B-Boc*
	O	I	I	I	*In a three-necked flask, compound 14-B (2.726 g, 5.675 mmol), DMAP (0.139 g, ...*

Table 3: Output examples. Columns "ParSoftmax", "Trigram", and "BiLSTM-CRF" show the output of paragraph-level softmax, paragraph-trigram softmax and BiLSTM-CRF decoders, respectively, with the W2V +ELMO +NER_{FINE} encoder.

tween B and I labels, indicating that long-term contextual information is crucial to correctly detecting the beginning of the reaction spans.

Table 3 shows system output examples from the test set. In Example 1, the first two paragraphs of the reaction span are its title and subtitle. The paragraph-level softmax model labels both the title and the subtitle as B, while the paragraph-trigram and BiLSTM-CRF model successfully classify the subtitle paragraph as I. In Example 2 and 3, there are multiple independent reactions in an Example part. In this case, the paragraph-level and paragraph-trigram classifiers often regard several reaction steps as one single reaction, while the BiLSTM-CRF model correctly separates out the individual reactions in both cases. Without context, it would be hard to distinguish the independent reactions from reaction steps inside a single reaction, as they have a common writing style (an example where a single reaction has several reaction steps can be found in Figure 1). As shown in Example 4, it is often the case that the title of a reaction span is the name of a chemical compound. All baseline classifiers often fail to detect such spans, even when chemical named entities are used as input features. Presumably, the fact that the paragraph beginning with a compound name can occur at the beginning, within or outside a reaction span, makes it hard to leverage such patterns for span detection. In Example 5, the text span beginning with *Example 26* is written in exactly the same way as the previous *Example 25* except that *compound 14-B* is used instead of *compound 13-Br*. However, *Example 26* is not extracted as a reaction by the gold annotation while *Example 25* is extracted. This would be because the reaction was regarded as incorrect for technical reasons (e.g., the compound used in the paragraph is clearly incorrect), or was just discarded by the human expert because it is not an important reaction. Such cases are hard for the system to pick up on, as they require deep understanding of the context and background knowledge.

7 Conclusions

In this paper we introduced the chemical reaction detection task and formulated this task as a paragraph-level sequence tagging problem. We proposed heuristic and machine-learning based baseline methods to measure the feasibility of the task as well as to identify the key challenges. We also created an annotated dataset by map-

ping back reactions from the Reaxys® database to their source patents. We used this corpus to train and evaluate our baseline methods. The experimental results show that this task requires a deep understanding of patent document context, as well as chemical background knowledge. Indeed, the BiLSTM-CRF model trained at the document-level performed much better than the paragraph-level classification methods.

The performance of the baseline methods presented in this paper is still not satisfactory considering the complex downstream tasks such as event extraction. We believe that both the models and the corpus have potential to be improved. As future work, we plan to explore more efficient document-level training methods, and, in particular, methods that work well on noisy training sets. For instance, techniques successfully used for distant supervision (Mintz et al., 2009) may be effective. Furthermore, although we used only textual information, patent documents contain substantial visual information (e.g., images of compounds, or tables) that may be helpful to properly understand a reaction description. Longer term, we will also tackle finer-grained information extraction for chemical reactions utilizing the output of this task. This step involves extracting the details of the detected reactions, that is, inferring the underlying structure of the reactions themselves.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was supported by an Australian Research Council Linkage Project grant (LP160101469). The computations in this paper were performed using the Spartan HPC-Cloud Hybrid (Lafayette et al., 2017) at the University of Melbourne.

References

Saber A. Akhondi, Alexander G. Klenner, Christian Tyrchan, Anil K. Manchala, Kiran Boppana, Daniel Lowe, Marc Zimmermann, Sarma A. R. P. Jagarlapudi, Roger Sayle, Jan A. Kors, and Sorel Muresan. 2014. Annotated Chemical Patent Corpus: A Gold Standard for Text Mining. *PLOS ONE*, 9(9):e107477.

Saber A. Akhondi, Hinnerk Rey, Markus Schwörer, Michael Maier, John Toomey, Heike Nau, Gabriele Ilchmann, Mark Sheehan, Matthias Irmer, Claudia Bobach, Marius Doornenbal, Michelle Gregory, and Jan A. Kors. 2019. Automatic identification of relevant chemical compounds from patents. *Database*, 2019.

Timo Böhme, Matthias Irmer, Anett Püschel, Claudia Bobach, Ulf Laube, and Lutz Weber. 2014. OCMiner: Text processing, annotation and relation extraction for the life sciences. In *SWAT4LS*.

Peter Corbett, Colin Batchelor, and Simone Teufel. 2007. Annotation of Chemical Named Entities. In *Proceedings of the Workshop on BioNLP 2007*, BioNLP '07, pages 57–64.

Atsushi Fujii, Makoto Iwayama, and Noriko Kando. 2007. Introduction to the special issue on patent processing. *Information Processing & Management*, 43(5):1149–1153.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Harsha Gurulingappa, Anirban Mudi, Luca Toldo, Martin Hofmann-Apitius, and Jignesh Bhate. 2013. Challenges in mining the literature for chemical information. *RSC Advances*, 3(37):16194–16211.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv:1508.01991 [cs]*.

David M. Jessop, Sam E. Adams, Egon L. Willighagen, Lezan Hawizy, and Peter Murray-Rust. 2011. OSCAR4: A flexible architecture for chemical text-mining. *Journal of Cheminformatics*, 3(1):41.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text Segmentation as a Supervised Learning Task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 469–473.

Martin Krallinger, Obdulia Rabal, Saber A Akhondi, et al. 2017. Overview of the BioCreative VI chemical-protein interaction Track. In *Proceedings of the Sixth BioCreative Challenge Evaluation Workshop*, volume 1, pages 141–146.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, SV Ramanan, Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A. Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usié, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzabal, and Alfonso Valencia. 2015. The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(1):S2.

Lev Lafayette, Greg Sauter, Linh Vu, and Bernard Meade. 2017. Spartan HPC-Cloud Hybrid: Delivering Performance and Flexibility.

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc.

Alexander Johnston Lawson, Stefan Roller, Helmut Grotz, Janusz L. Wisniewski, and Libuse Goebels. 2011. Method and software for extracting chemical data.

Daniel M. Lowe. 2012. *Extraction of Chemical Structures and Reactions from the Literature*. Ph.D. thesis, University of Cambridge.

John Mayfield, Daniel Lowe, and Roger Sayle. 2017. CINF 13: Pistachio - Search and Faceting of Large Reaction Databases.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 2, pages 1003–1011.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Florina Piroi, Mihai Lupu, Allan Hanbury, Alan P. Sexton, Friedemann Magdy, and Igor V. Filippov. 2012. CLEF-IP 2012: Retrieval Experiments in the Intellectual Property Domain. In *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes*.

Raul Rodriguez-Esteban and Markus Bundschus. 2016. Text mining patents for biomedical knowledge. *Drug Discovery Today*, 21(6):997–1002.

Stefan Senger, Luca Bartek, George Papadatos, and Anna Gaulton. 2015. Managing expectations: Assessment of chemistry databases generated by automated extraction of chemical structures from patents. *Journal of Cheminformatics*, 7(1):49.

Walid Shalaby and Wlodek Zadrozny. 2019. Patent Retrieval: A Literature Review. *Knowledge and Information Systems*, 61(2):631–660.

Christopher Southan. 2015. Expanding opportunities for mining bioactive chemistry from patents. *Drug Discovery Today: Technologies*, 14:3–9.

Erik F. Tjong, Kim Sang, and Jorn Veenstra. 1999. Representing Text Chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*.

Yuen-Hsien Tseng, Chi-Jen Lin, and Yu-I Lin. 2007. Text mining techniques for patent analysis. *Information Processing & Management*, 43(5):1216–1247.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. 2015. Overview of the BioCreative V chemical disease relation (CDR) task. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, volume 14.

Zenan Zhai, Dat Quoc Nguyen, Saber Akhondi, Camilo Thorne, Christian Druckenbrodt, Trevor Cohn, Michelle Gregory, and Karin Verspoor. 2019. Improving Chemical Named Entity Recognition in Patents with Contextualized Word Embeddings. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 328–338.

Decoder	Input token representation	Layers (enc)	Dim (enc)	LR
Paragraph-level softmax	W2V +ELMO	1	200	5×10^{-5}
	W2V +ELMO +$\text{NER}_{\text{COARSE}}$	2	200	5×10^{-5}
	W2V +ELMO +NER_{FINE}	2	100	5×10^{-5}
Decoder	Input token representation	Layers (dec)	Dim (dec)	LR
BiLSTM-CRF	W2V +ELMO +NER_{FINE}	2	100	1×10^{-3}

Table 5: The best hyperparameters on the development set. "LR" is the initial learning rate, "Layers (enc/dec)" is the number of LSTM layers; and "Dim (enc/dec)" is the dimensionality of the LSTM hidden states, where "enc/dec" indicate the LSTMs in the paragraph encoder and paragraph label decoder, respectively. "$\text{NER}_{\text{COARSE}}$" and "$\text{NER}_{\text{FINE}}$" mean NER tag embeddings based on coarse- and fine-grained entity types, respectively. The structure of the paragraph encoder used for the paragraph-trigram softmax and BiLSTM-CRF classifier is the same as the one for the paragraph-level softmax classifier, thus omitted from the table.

Coarse-grained	Fine-grained
chemClass	chemClass
	$\texttt{chemClass}_{\texttt{biomolecule}}$
	$\texttt{chemClass}_{\texttt{markush}}$
	$\texttt{chemClass}_{\texttt{mixture}}$
	$\texttt{chemClass}_{\texttt{mixture-part}}$
	$\texttt{chemClass}_{\texttt{polymer}}$
chemCompound	chemCompound
	$\texttt{chemCompound}_{\texttt{mixture-part}}$
	$\texttt{chemCompound}_{\texttt{prophetics}}$

Table 4: Entity types from Reaxys® gold-standard data.

Appendix

NER label sets

Table 4 shows the NER label sets that we used as additional features to include in the input representation as described in Section 5.2.1.

Hyper-parameters

Table 5 shows the optimal hyper-parameters we used for the final evaluation.