Knowledge Base Construction of Organic Compound Synthesis*

Wei Li[1], Zhewen Song[2], Xiuyuan He[3]

Abstract— **Fonduer is a knowledge base construction (KBC) framework for richly formatted information extraction (RFIE). Here we extend Fonduer framework to understand visual information of images and extract multi-modal (texts and images) entity relationship in document level. Our work focuses on parsing images and image-related text through datasets, capturing the representation needed to relate the visual information with the corresponding text, and utilizing data programing and statistical learning to build models for KBC. Specifically, we implemented the text/image modules in Fonduer to encode a list of visual information into feature representations, e.g. the position, context, and the contents in the image, as well as the relationship to the text in the document. We presented our work in the application of constructing entity relationship between organic compounds with chemical synthesis diagrams/schemes from literature publications. Compared with golden data, the knowledge base extraction results in a precision of 0.382, recall of 0.834, and F1 of 0.524. We aim to build a general platform to extract text-image relation for multimodal dataset and enable users to incorporate domain expertise as supervising intuitive knowledge for a range of visual information extraction tasks.**

I. INTRODUCTION

Knowledge base construction (KBC) is the process that incorporates information extraction and information integration together, populating a knowledge base, i.e., a relational database together with inference rules, with information extracted from documents and structured sources. [1] There have been an increasing interests from academia to industry to construct knowledge base under various frameworks, e.g. DeepDive [2] from Stanford, NELL [3] from CMU, YAGO [4,5] from MPI, DeepQA [6] from IBM and EntityCube [7] from Microsoft. Traditionally, these KBC frameworks focus on unstructured text information extraction, with little information from visual representations existing in the dataset. Recently, Fonduer [8] was introduced as a machine-learning-based KBC system for richly-formatted data, and in particular deals with the challenges of extracting tabular information in table-embedded documents. However, an effective and complete extraction and representation of images (including diagrams) from documents is absent in Fonduer. Furthermore, there is no mechanism in Fonduer to couple the texts from the document, image captions as well as contents inside the images. To further expand the generality of the knowledge-base-construction platform for richly-formatted dataset, we developed on top of Fonduer to extend the ability to understand image, as well as the relationship of image with its context and the related text entity in the richly-formatted dataset.

To illustrate our work for visual information extraction, we choose the application of building a knowledge base with the relation between organic chemical compound entity and the diagram entity illustrating the synthesis approach for the chemical compound in image format from the dataset of published literature papers in scientific journals. The motivation for building a knowledge base of chemical synthesis diagrams originates from the high demand of an efficient approach to search for the synthesis approach of new or rare organic compound. It can be very time consuming and cost a lot human labor to manually go through literature papers and find the corresponding synthesis figure for a new or rare compound of interest. The constructed knowledge base can be of great value for (1) it collects the recent progress in synthesis of new organic compounds in academia; (2) users can search for the synthesis route of a organic compounds it he/she encounters an unfamiliar compound; (3) it offers the opportunity for chemistry theorist to study/conclude the general mechanism for organic synthesis by going through this knowledge base; (4) the constructed dataset can be used for training machine-learning models to predict the synthesis route of novel organic compounds. [9, 10, 11] Furthermore, the knowledge base generation can be pipelined together with training of predictive models for designing chemical synthesis with the available chemical structure recognition tool for synthesis diagrams [12]. Thus, an adaptive machine-learning model can be trained directly from enormous research publications in organic synthesis field, which offers great potential to boost the performance of current deep learning models [11]. However, traditional KBC platforms are unable to construct richly-formatted dataset like this publication literatures dataset. Thus, we extend the functionality of Fonduer to handle visual information extraction, especially for image and text relationship to construct the knowledge base of organic compound synthesis.

The image explaining the synthesis approach for a chemical compound can be easily represented in structural diagrams with organic chemical structures, arrows, and text for reaction conditions. The organic chemical structures can be in various format, constructing from straight/dashed lines, polygons, rings, etc. This standard presentation is widely used in fields including biology, chemistry and medicine. However, due to the abundance of organic chemical compound species and complicated combinations

*This work was not supported by any organization

[1]W. Li is a graduate student from Department of Computer Sciences, Materials Science and Engineering, University of Wisconsin-Madison, WI, 53706, USA wli284@wisc.edu

[2]Z. Song is a graduate student from Department of Computer Sciences, Materials Science and Engineering, University of Wisconsin-Madison, WI, 53706, USA zsong39@wisc.edu

[3]X. He is a graduate student from Department of Computer Sciences, Information Science, University of Wisconsin-Madison, WI, 53706, USA xhe75@wisc.edu

(a) **Approach to the synthesis of the C1–C11 and C14–C18 portion of Leucascandrolide A†‡**

T. J. Hunter,§[a] J. Zheng§[b] and G. A. O'Doherty*[b]

An asymmetric synthesis of the C1 to C11 and C14 to C18 fragments of the macrocyclic portion of the antibiotic Leucascandrolide A was achieved in 21 total steps from an achiral dienoate. The key 4-hydroxy-2,5-pyran portion of the natural product was established by oxy-Michael cyclization of a 5,7,9,11-tetraol intermediate, which in turn was established by an iterative asymmetric-hydration of dienoates. Alternative strategies for establishing the polyol stereochemistry were explored.

(b)

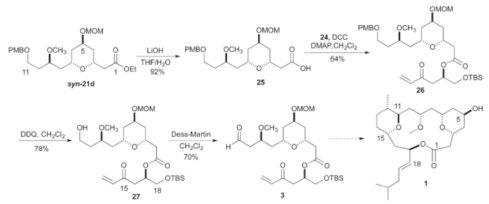

Fig. 1. (a) A sample literature paper title and abstract. (b) A sample image from literature paper in (a) representing a synthesis route for an organic compound.

among functional groups, it can be challenging to relate the diagrams with its dedicated chemical nomenclature. It is even more difficult to find out how a specific organic compound is synthesized from a pool of diagrams. While one may manually read through the literature and match the organic product with the desired reaction scheme, this procedure can be time-consuming, error-prone and tedious. Moreover, domain expert with an adequate amount of background knowledge from organic chemistry is a necessity for the completion of this work. Therefore, it will be a great relief if the process is automated based on a knowledge base that is dedicated to the relation of organic product with reaction scheme. Our goal in this project is to implement the image visual information extraction functionality, so that we can easily construct richly-formatted knowledge base like this, and many other potential applications in various areas.

SUMMARY OF CONTRIBUTIONS AND OUTLINE

We demonstrated our work by building knowledge base for organic compound and the synthesis diagrams that produce the compound. Our contribution is the addition of image information representation to the existing framework Fonduer, in which users can construct knowledge base for text-image, image-image relation schema. We extended the Fonduer to support parse images from html and pdf files. We built the image filters and feature generator for multimodal features. We provided image support for data programming and classification model. In section 3, we describe our complete workflow for the construction of knowledge base, as well as the training and testing of the machine-learning-based model, by data programing in Snorkel framework [13]. In section 4, we show our experimental setup. In section 5, we present the knowledge base extraction results compared with golden labels.

II. RELATED WORKS

We demonstrate previous related work in below sections.

A. Knowledge Base Construction

Knowledge base construction (KBC) is the process that incorporates information extraction and information integration together, populating a knowledge base, i.e., a relational database together with inference rules, with information extracted from documents and structured sources. [1] KBC has been intensively studied by people in the passed few years. Most of the existing approaches for information extraction or knowledge base construction were either classic rule-based approaches [12,14] or machine-learning-based systems. [1,2,4,5,8] Among these, many have been studied the information extraction and integration from textual and unstructured data or structured data, but not from visual data.

B. Fonduer Framework

Fonduer is a machine-learning-based framework for knowledge base construction from richly formatted data. [8] Before Fonduer, a well-known challenge in KBC was to deal with multiple modalities of data, i.e. data can have various format, including plain text, tables, structural data and visual data. Many existing systems only utilized the textual features, thus failed to build high-quality knowledge bases from richly formatted data like tables and figures. Fonduer aimed to confront this challenge by utilizing structural and visual cues for data extraction and matching and in particular, it achieved high performance when dealing with tabular information extraction from table-embedded documents. However, visual data such as images and figures were out of Fonduer's scope.

C. Snorkel Framework and Data Programming

Snorkel is a system that use data programming to rapidly creating, modeling, and managing training data. [16] Data programming is a paradigm that lies in the heart of Snorkel. In Snorkels data programming model, the users start by creating a list of labeling functions (LFs), which are just a set of rules that can label the data automatically. However, the labels might be noisy, depend on the quality of the labeling functions. To resolve the noise, Snorkel builds generative models from the LFs using deep-learning, to learning the effectiveness of each LF, and thus can assign weights to LFs based on their quality. Snorkel and data programing free the developers from manually labeling data, which is the most time consuming part and the bottleneck of the current supervised machine learning systems.

III. KBC WORKFLOW

Our work is built upon the framework of Fonduer. We combined information extraction and information integration for organic compound and synthesis diagrams extraction and pairing, exploiting the related expertise and leveraging the existing frameworks including spaCy [16] for natural language processing (NLP), Snorkel for statistical learning and data programing, SQLAlchemy for data storage and query, Optical character recognition (OCR) [17] for content recognition in images. We also incorporated image information extraction such as histogram of gradient (HOG) [18], local binary pattern (LBP) [19], binarized image [20],

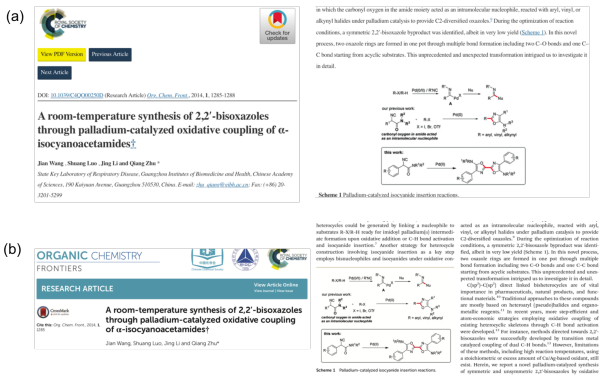

Fig. 2. A sample document in html format (a) and pdf format (b).

and convolutional neural network (CNN) [21] in computer vision for more effective feature generation. We present our techniques in detail in the following sections.

A. Dataset

The dataset consists of literature paper published on the publication group website in both pdf format and the html format. We collected the literature papers that report the novel synthesis route for new compounds in organic chemistry. Fig. 2 shows a sample document we collected in both html format and pdf format. As a starting point, we collected 24 literature paper on organic synthesis area for code development. Later in the project, we expanded our dataset to 102 literature paper to include more relations and more variety in candidates.

All the literatures we used are from the journal Organic Chemistry Frontiers (Royal Society of Chemistry). We chose this journal because it has a rich set of most up-to-date noval organic compound synthesis schemes, and both PDF and HTML are available and well formatted. The papers have generally 3-4 pages and 4-7 images, with the longest one having 21 pages and 32 images.

B. Parse the documents

For initial parsing of the dataset file, we read the html text as element tree in python and parse the node in hierarchy order to obtain the sentences and figures raw information. We also use unix command pdftotext to obtain the text information in the pdf file, and unix command pdftohtml to obtain the image in the pdf file. We then use positional information and context information for the linkage of entities (sentences, images) obtained from the html file and pdf file. The workflow of the parser is shown in Figure 3.

For the purpose of our particular application, we will create a binary relation knowledge base, namely an organic compound product - synthesis diagram relation. We thus need to parse the documents to extract the said two entities. Fonduer has already incorporated basic HTML parser with existing APIs, namely BeautifulSoup and LXML, then pass the extracted texts to the NLP tool - spaCy for fine-grained lingual processing. The parsed sentence was analyzed by spacy NLP modules and return tokenized words. Currently,

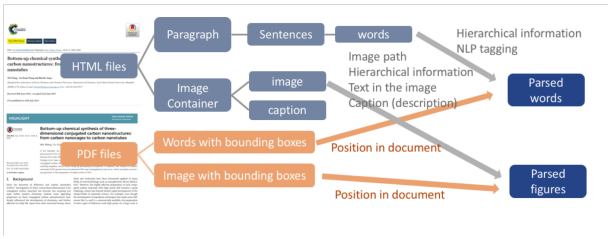

Fig. 3. A schematic workflow of parser on HTML files and PDF files to obtain parsed words and parsed figures.

Fonduer only allows the default setup of the spaCy for lingual processing, and this includes a set of default delimiters like space, dash, parenthesis and bracket. However, using default delimiters will also mistakenly separate valid organic compound nomenclature, like (E)-2,6-heptadienoyl. Therefore, we also enable the customizable delimiter set during lingual parsing. This parser finally yields an array of Phrase class, which will be pipelined to the downstream candidate extractor engine. During our inspections and experiments with this parser, we made several improvements for the robustness of our applications.

Because organic compounds are more richly formatted than the plain text (subscription, superscription, punctuation, etc.), the associated HTML will have multiple nested tags to represent such format. Therefore, we need to allow users to customize the flatten method during HTML parsing, such that the compound is not mistakenly separated across two Phrase objects.

As for the image extraction, we added a new class `DetailedFigure`, which incorporates an array of useful information for downstream processes. For instance, because each image from standard scientific literature will have its caption, we maintained a field called description to store the associated caption. Note that the caption is also parsed from the HTML (with necessary customized tag flattening), and this field is very critical for correctly matching the desired organic compound.

We also have fields to store the coordinates of the images within a page extracted from the dataset. Note that since the HTML actually does not give us numerical values of the page coordinates, we have to resort to its associated PDF to extract the position and size information. Specifically, we used the Linux program, pdftohtml, to convert PDF files into HTML, XML and PNG images, where we can obtain the bounding box of each image and line of words. We linked the image bounding box with the figure name in pdf by examining the bounding boxes of the caption text and the image, if the text bounding box is near the image bounding box and the text contains the words like figure 1 or schema 1 within a certain range, we then captured an image-caption pair (see the code in Figure 4). Then we can connect the image in pdf with the image in HTML by the figure name. We have completed parsing and extracting arrays of key information from HTMLs, PDFs as well as the diagrams from 102 literatures, selecting 7523 candidates with customized matchers

```
def match(box1, box2, page):
    return
    abs(box1['top'] + box1['height'] - box2['top']) < page.get('height')/10
    and
    abs(box1['left'] - box2['left']) < page.get('width')/5
```

Fig. 4. A sample code showing how to link a bounding box of image with the caption text.

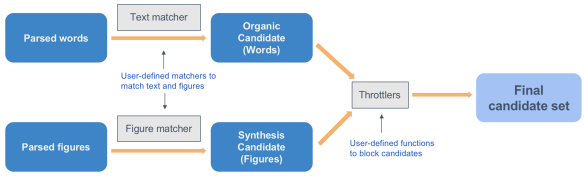

Fig. 5. Workflow of matchers and throttlers to extract the organic/figure entity and candidate set.

and throttlers, and creating 35233 features and 11 labeling functions for the on-ward probabilistic inference and weakly supervised training.

C. Entity and Candidate Generation

We utilized the domain knowledge in organic compounds to construct the organic compound matcher. The domain knowledge is reflected in the regular expression, user-defined matcher function, and organic compound dictionary to fit common prefix, suffix, as well as naming conventions in organic chemistry. For the images, we examined the captions under the images, as well as the words inside the images to select the synthesis diagrams that contains useful synthesis approach information. The candidate pairs between selected organic compound and synthesis diagrams were then filter by user-defined filter function that screen out obviously unmatched pairs of organic compound and synthesis diagram, e.g. a pair of compound A and a image show the synthesis of compound B. In building the parser and extractor, we allow the flexibility to user to develop their own matcher and filter functions for text entities and images. The information extracted from the image can be effectively utilized in the construction of matcher and filter function per user need. The workflow of matcher and candidate filter is demonstrated in Figure 5.

We mainly used regular expression for organic compound/text matcher. We applied our domain knowledge to write up a set of regular expressions summarized in Figure 6.

Prefix	`\(?((mono|bi|di|tri|tetra|hex|hept|oct|iso|a?cycl|poly).*)?(meth|carb|benz|fluoro|chloro|bromo|iodo|hydro(xy)?|amino|alk).+`
Suffix	`.+(ane|yl|adiene|atriene|yne|anol|anediol|anetriol|anone|acid|amine|xide|dine|(or?mone)|thiol|cin)\)?`
Dash	`(\w+\-|\(?)([a-z|\d]\'?\-)\w*`
Comma & Dash	`(\w+\-|\(?)([a-z|\d]\'?,[a-z|\d]\'?\-)\w*`
Inorganic & Abbreviation	`([A-Z][a-z]?\d*\+?){2,}`

Fig. 6. Sample regular expressions for organic compound extraction.

```
def candidate_filter(c):
    (organic, figure) = c
    to_remove = ['synthesis', 'syntheses', 'product', 'reaction', 'the', 'of', 'for', 'and']
    for key in to_remove:
        if key in organic.text.split():
            return False
    if same_file(organic, figure):
        if mentionsFig(organic, figure) or mentionsOrg(figure, organic):
            return True
    return False
```

Fig. 7. A sample throttler function for filtering the candidate pairs.

(a)

```
(2-di-tert-butylphosphino-2′-(N,N-dimethylamino)biphenyl)
(3) and cornolactone
(3) carbonyl
(4-nitrobenzenesulfonyl)
(5′ R )-5′,8-cyclopurine
(5′R)-5′,8-cyclo-2′-deoxypurine
(5′R)-5′,8-cyclopurine
(5′R)-cdG phosphoramidite
(5′S)- and (5′R)-5′,8-cyclo-2′-deoxypurine
(5′S)- and (5′R)-5′,8-cyclopurine
(5′S)-cdA
(5′S)-cdG
(CuOTf)2·PHH and ethyl
(DIC) and 4-(dimethylamino)pyridine
(E)-1,2-bis(silyl)ethenes with acid
(E)-3-methyl-3-en-1-ynes
```

(b)

name	caption
Scheme 1	The intramolecular Mannich reaction.
Scheme 2	Formal synthesis of (−)-FR901483 (1).
Scheme 3	Total synthesis of (−)-hippodamine (7).
Scheme 4	Intramolecular Mannich cyclization for the synthe
Scheme 5	Total synthesis of dihydrolycoflexine (25) and 5-epi

(c)

organic	fig_name	document
aryl	Scheme 14	Metal-mediated C–O bond forming reaction
(−)-platensimycin	Scheme 2	Formal synthesis of (−)-platensimycin -
K2CO3	Scheme 4	Photoredox-catalyzed cascade annulation
lactone	Scheme 4	Recent synthetic studies towards natural
the investigated [2]rotaxanes	Scheme 1	Synthesis and characterization of a doub
AcOH	Scheme 4	Organocatalytic asymmetric synthesis of
diaryl	Fig. 2	Metal-mediated C–O bond forming reaction
IV	Scheme 7	Recent advances in the intramolecular Ma
K2CO3	Scheme 4	Photoredox-catalyzed cascade annulation
diaryl	Scheme 10	Metal-mediated C–O bond forming reaction

Fig. 8. (a) A subset of extracted organics (b) A subset of extracted figures names and captions (c) A subset of candidate organic figure pair.

For images, we created black/white lists to filter out those obviously unqualified candidates. The image candidates were stored in a new class called `OmniDetailedFigure` for consistency. Furthermore, we incorporated lambda functions as matchers by inspecting if the organic compound is mentioned in the caption of the image and vise versa, to find matches between organic compounds and images. After adopting the lambda functions, nearly 93% of the candidate pairs were filtered by our matchers. Figure 7 shows a sample throttler function we use to block false candidates.

After the extractions, our candidate is in the form of the data structure as (`ImplicitSpan, DetailedImage`), where `ImplicitSpan` and `DetailedImage` represent the potential product of chemical synthesis and the image for the synthesis respectively. Figure 8 shows the extracted organic compound entity, synthesis figure entity and the candidate organic-figure pair. After parsing and extracting candidates, we filled the `ImplicitSpan` table in the database, where each row stores the text information of the candidate organic compound. The result is actually not perfect as one may discover that the parenthesis and brackets are sometimes mismatched because its tricky to distinguish whether it belongs to the text or the organic compound itself. We also filled the `DetailedImage` table in the database with its name, caption, OCR extracted text, and position coordinates.

D. Feature Annotations

For each extracted candidate pair from the previous section, we annotate it with a set of abundant features to make sure that the text-image candidate pair conveys enough information for determining a matched relation or not. We manually constructed a set of position-related and context related features for the organic compound (text phrase) and the

```
def _generate_approximate_feats(span):
    phrase_num = span.sentence.phrase_num
    doc = span.sentence.document
    string = span.get_span()
    freq75, freq90, freq100 = 0, 0, 0
    for i in range(len(doc.phrases)):
        if fuzz.partial_ratio(string, doc.phrases[i].text) >= 75:
            freq75 += 1
        if fuzz.partial_ratio(string, doc.phrases[i].text) >= 90:
            freq90 += 1
        if fuzz.partial_ratio(string, doc.phrases[i].text) == 100:
            freq100 += 1
    yield "APPEARED_[{}]_TIMES_FUZZY_75".format(freq75//10 if freq75//10 < 4 else 'MANY')
    yield "APPEARED_[{}]_TIMES_FUZZY_90".format(freq90//10 if freq90//10 < 4 else 'MANY')
    yield "APPEARED_[{}]_TIMES_FUZZY_100".format(freq100//10 if freq100//10 < 4 else 'MANY')
```

Fig. 9. A sample feature generator for organic compound entity which represents the appearance frequency in the document.

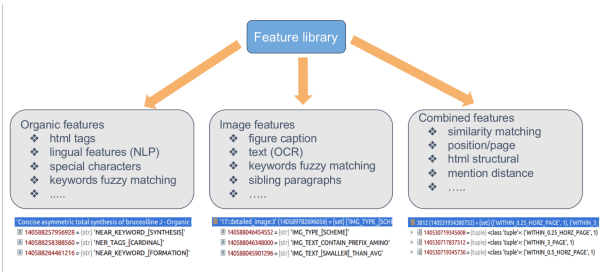

Fig. 10. A subset of feature library comprised of three major types of features: organic features, image features and combined features.

synthesis diagram (image) that encode the spatial relationship within the document. For the text phrases, we also encode the NLP tags of the word and previous/subsequent words into the feature set. For the images, we encode the spatial correlation with the paired text phrases, common information between the caption of the image and the sentence of the text phrase.

For text representation, similar to the original text feature annotation in Fonduer, we also have created features spanning from the lingual tags from the parsing phase including dependency labels, dependency parents, ner_tags, pos_tags, etc. We further encode into the features the frequencies of the dedicated organic compound found in the phrase/sentence, in the entire document and in the caption. We also have the HTML tag information added into the features.

For image representation, we created features based on the candidate image description/caption, the OCR text from the image, and the information from previous paragraph of the image, where the candidate were mentioned. Specifically, we use fuzzy matching to obtain the matching score with the organic compound and have several levels of threshold as the feature keys (see code snippet example below in Figure 9). Figure 10 shows a sample feature generator for organic features. For the relation/combined representation, we deploy the visual distance, page-alignment as well as the HTML tags to mine the parent-child, ancestor-descendant and sibling relations of a particular image with the surrounding paragraph and other images. Figure 10 shows a sample subset of features we annotated for each candidate pair.

In order to encode more visual information of images in the feature set for more general application, we also added common feature extraction methods in computer vision field in our work, such as local binary pattern (LBP), histogram of gradient (HOG), binarized image, and convolutional neural

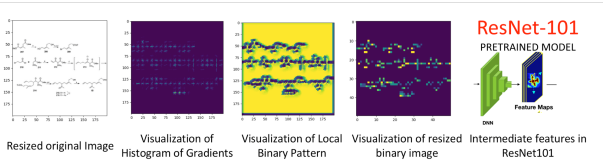

Fig. 11. A visualization of Histogram of Gradients, local binary pattern, binarized image and CNN feature (Pretrained ResNet101) on a sample synthesis image

network (CNN) features. Figure 11 visualizes the image features on a sample synthesis image. LBP is a texture descriptor made popular by the work of Ojala et al. in their 2002 paper [18]. LBPs compute a local representation of texture. This local representation is constructed by comparing each pixel with its surrounding neighborhood of pixels. HOG is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. [19] Binary image was obtained by using Otsus method [20] that pixels with intensity value lower than the threshold will be zero and higher than threshold will be one. We use ResNet-101 pretrained model [21] on ImageNet to obtain the intermediate features of the synthesis images. The high level abstract features of the CNN structure can provide highly-folded signals that could potentially be used for discriminative model training.

E. Labeling Functions

As we are utilizing Snorkel as the framework to train a machine-learning-based model for knowledge construction, we applied user-defined labeling functions, which express various heuristics, patterns, strategies to label the derived candidate pairs. The hand-labeled training data is expensive to obtain and inhibitive for very large dataset. We use domain knowledge in organic chemistry, as well as common context information to determine what kind of organic compound is more likely to be the product of the reaction, not the reactant, what kind of images is more likely to be a synthesis diagram for the organic compound. The labeling functions can be not perfect, or even of quite low-quality. However, with data programming, we can train the machine learning models to learn which features are more important in determining matched pair of organic compound and synthesis diagram.

As of now, we have 11 labeling functions including regular expressions, caption matching, OCR text matching, page matching, etc, as shown in Figure 12. Notably, we found that there is a significant amount of captions that start with Synthesis of, which can be treated as a strong signal that the following lemma would be the desired organic compound. We thus add this pattern into the labeling function set (see the code snippet in Figure 12). In addition, long captions composed of multiple sentences typically are structured as the first sentence being the summary and the rest being the detailed descriptions or the reaction conditions. Therefore, if the organic compound is not present in the very first sentence of the caption, it is highly likely that this compound does not

```
org_fig_lfs = [
    LF_fig_name_match,
    LF_text_desc_match,
    LF_ocr_text_match,
    LF_text_length_match,
    LF_match_whitelist,
    LF_match_blacklist,
    LF_match_page,
    LF_pos_near,
    LF_check_redundant_word_in_organic,
    LF_keyword_of,
    LF_first_period,
]
```

```
def LF_keyword_of(c):
    organic, figure, = c.get_contexts()
    words = figure.description.split(' ')
    words_lower = figure.description.lower().split(' ')
    if organic.text in words and 'synthesis' in words_lower:
        org_idx = words.index(organic.text)
        syn_idx = words_lower.index('synthesis')
        return 1 if syn_idx + 2 == org_idx else -1
    return 0
```

Fig. 12. A collection of labeling functions for data programming and a sample labeling function in detail.

match this figure. Our current labeling functions might not be very accurate. However, we can learn the effectiveness of our labeling functions by taking advantage of the Snorkels data programming model, and further improve the labeling functions in the future debugging phase.

F. Training and Testing

We first train a model of the labeling functions to estimate the accuracies of the labeling functions. We use Snorkel here to automatically learn a generative model over the provided labeling functions, which acts as various weak supervision sources. The generative model estimates their accuracies and correlations. This step doesnt require any ground truth data, but learns from the agreements and disagreements of the labeling functions. The output of generative model is a set of probabilistic labels as a single, noise-aware training label set. We use the training marginals and the feature extracted in section 3.3 to train a wide variety of state-of-the-art machine learning models, such as sparse logistic regression, popular deep learning models. The testing set contains the original 24 papers, while the training set contains the 78 newly added papers. We only labeled the 24 papers in the testing set manually as the golden labels.

IV. EXPERIMENTAL SETUP

We used macOS/Linux to deploy the Snorkel/Fonduer framework. The machine sued for generating the train/test results in this report was a Macbook Pro Laptop produced in mid 2015 with a processor of 2.5 GHz Intel Core i7, memory of 16GB and storage of SSD.

V. RESULTS

As mentioned in section III.E, we created 11 labeling functions and use those labeling functions to train a generative model. The output of the generative model were probabilistic labels, which we can use in next stage to train a discriminative model and perform classification tasks on the candidate set. An initial evaluation to the labeling functions is shown in Figure 13.

Next, we applied the featurizer to our candidates and extracted multimodality features from text and images. We also extracted additional image features, such as HOG and LBP features. We concatenated those features together and generated a final feature matrix that contains 55907 features for each training and testing examples.

Logistic regression was then used for discriminative model training. In this step, we compared the result of whether add

	TP	FP	FN	TN	Empirical Acc.	Learned Acc.
LF_text_length_match	0	0	14	1121	0.987665	0.995822
LF_check_redundant_word_in_organic	0	0	89	15	0.144231	0.988582
LF_pos_near	286	901	0	0	0.240944	0.990305
LF_keyword_of	173	150	83	625	0.774006	0.989395
LF_match_blacklist	0	0	0	37	1.000000	0.988628
LF_match_whitelist	444	1087	0	0	0.290007	0.989849
LF_text_desc_partial_match	0	0	0	0	NaN	0.988584
LF_first_period	758	1550	0	0	0.328423	0.991336
LF_dict_match	0	0	0	0	NaN	0.988613
LF_match_page	0	0	449	1975	0.814769	0.994374
LF_ocr_text_match	493	942	297	2079	0.674888	0.999723
LF_text_desc_full_match	519	1310	0	0	0.283762	0.990929
LF_page_not_match	0	0	0	0	NaN	0.988494

Fig. 13. Labeling function empirical accuracy.

```
plot_tp_entity(TP[0])
```

```
./Synthesis of trans -disubstituted-2,3-dihydrobenzofurans by a formal [4
hides and sulfonium salts - Organic Chemistry Frontiers (RSC Publishing) I
2.gif
```

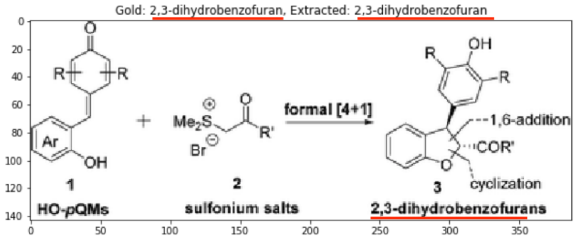

Fig. 14. Example of true prediction.

additional image features to our model. As shown in Table 1, for models without image features, we got a precision and recall of 0.392 and 0.809. If we add image features to our model, we got a precision and recall of 0.382 and 0.834.

TABLE I
PREDICTION ACCURACY

	precision	recall	F_1
With Image Feature	0.392	0.809	0.528
Without Image Feature	0.382	0.834	0.524

One of the possible explanations to this result is that, the raw features we extracted from the images were pre-trained and then concatenated to the feature matrix. The pre-trained image features may contains a lot of noise, which cannot cooperate well with the multimodality features. To better combine those two different sets of features together, it is not enough to simply concatenate them. One possible way to improve the result was to integrate the training of ResNet101 into Fonduers training process, and re-learn the weights of image net in each iteration. This possible improvement will be discussed in later section. Below we showed several visualizations of the result. Figure 14 shows an example of the true prediction.

Furthermore, using the predictions we can identify the organic compound and synthesis figure in the original documents, as shown in Figure 15.

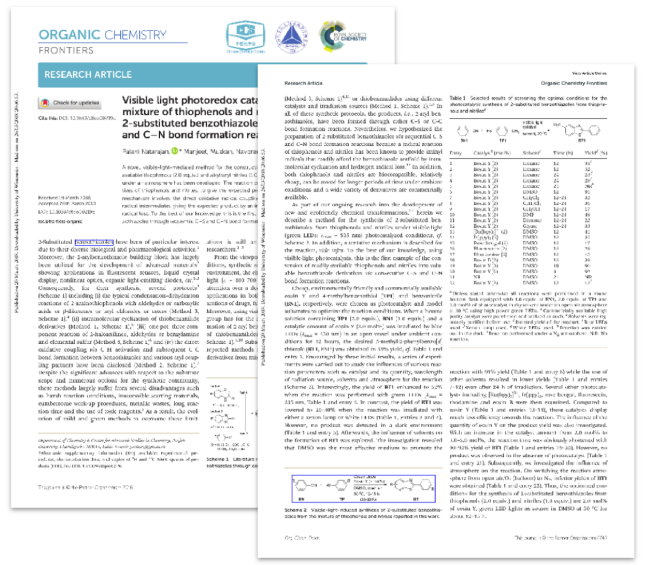

Fig. 15. Example of extracted organic and figure in original paper(circled out with blue boxes).

VI. CHALLENGES AND FUTURE WORK

We identified several challenges that affect the performance of our applications. First, in our context, extracting relations from papers is non-trivial. The organic structures are so complicated that even as persons with domain knowledge, we still find it difficult to identify the organic compound and associate it with correct synthesis images.

Second, when reading an organic chemical paper, there is too much information coming from the text and images. However, most of them are distracting and noisy signals, and only a small proportion of them are related to the synthesis process, which can be utilized for the classification tasks. In our current implementation, we tried to extract as much information that we think would be useful for the classification as we can. However, to further debug and improve the performance, we need more time to evaluate our feature library and labeling functions.

Third, as mentioned before, the feature matrix that we used to train the discriminative model was a half sparse and half dense hybrid. Also, the dense features generated from the images are low level encoding of images, while the sparse features are human supervised high level encoding of lingual or spatial relationship. Thus, directly concatenate the features and learn the weights for the features together may affect the performance of our discriminative model. One direction for our future work would be integrating the extraction of image features into Fonduer's training process and train the deep learning model ResNet101 together with the sparse features, such that the weights of image network will be learned and updated in each training iteration. We believe that update the weights of ResNet101 while training the discriminative model will capture more underlying semantics from images and thus improve the performance.

VII. CONCLUSIONS

In summary, we extended Fonduer's KBC pipeline and made it possible to build knowledge bases from text and images. Specifically, we extended the parser and data model and added compatibility for image candidate. With the matchers and labeling functions, we can generate candidate pairs that capture the relationship between organics and synthesis figures. Lastly, we extended Fonduer's feature library and added additional image features, such as HOG, LBP and CNN to train the discriminative model. Compared with golden data, the knowledge base extraction results in a precision of 0.382, recall of 0.834, and F_1 of 0.524. The generated knowledge base of organic synthesis images can be of value for chemistry research and artificial intelligence in auto-design synthesis route. We hope this work can bring some insights into building a general platform to extract text-image relation for multimodal dataset and enable users to incorporate domain expertise as supervising intuitive knowledge for a range of visual information extraction tasks.

APPENDIX

Source Code repository for this work is available at `https://github.com/leewaymay/839_fonduer`.

ACKNOWLEDGMENT

We hereby acknowledge the helpful advice from Theodoros Rekatsinas and Sen Wu.

REFERENCES

[1] R, Christopher, et al. "Feature engineering for knowledge base construction." arXiv preprint arXiv:1407.6439 (2014).
[2] Niu, Feng, et al. "DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference." VLDS 12 (2012): 25-28.
[3] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka Jr, and T. Mitchell. Toward an architecture for never-ending language learning. In AAAI, 2010.
[4] G. Kasneci, M. Ramanath, F. Suchanek, and G. Weikum. The YAGO-NAGA approach to knowledge discovery. SIGMOD Record, 2008.
[5] N. Nakashole, M. Theobald, and G. Weikum. Scalable knowledge harvesting with high precision and high recall. In WSDM, 2011.
[6] D. Ferrucci et al. Building Watson: An overview of the DeepQA project. AI Magazine, 2010.
[7] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J. Wen. Statsnowball: A statistical approach to extracting entity relationships. In WWW, 2009.
[8] Wu, Sen, et al. "Fonduer: Knowledge Base Construction from Richly Formatted Data." arXiv preprint arXiv:1703.05028 (2017).
[9] Simm, Gregor N., and Markus Reiher. "Context-Driven Exploration of Complex Chemical Reaction Networks." Journal of chemical theory and computation 13.12 (2017): 6108-6119.
[10] Fooshee, David, et al. "Deep learning for chemical reaction prediction." Molecular Systems Design & Engineering (2018).
[11] Segler, Marwin HS, Mike Preuss, and Mark P. Waller. "Planning chemical syntheses with deep neural networks and symbolic AI." Nature 555.7698 (2018): 604.
[12] OSRA: Optical Structure Recognition Application. https://cactus.nci.nih.gov/osra/ (Accessed in 2018).
[13] Ratner, Alexander J., et al. "Snorkel: Fast training set generation for information extraction." Proceedings of the 2017 ACM International Conference on Management of Data. ACM, 2017.
[14] Poria, Soujanya, et al. "A rule-based approach to aspect extraction from product reviews." Proceedings of the second workshop on natural language processing for social media (SocialNLP). 2014.
[15] Atzmueller, Martin, Peter Kluegl, and Frank Puppe. "Rule-Based Information Extraction for Structured Data Acquisition using TextMarker." LWA. 2008.

[16] SpaCy: Industrial-strength Natural Language Processing in Python, https://spacy.io/ (Accessed in 2018).
[17] Mori, Shunji, Hirobumi Nishida, and Hiromitsu Yamada. Optical character recognition. John Wiley & Sons, Inc., 1999.
[18] Ojala, Timo, Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.
[19] Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." IEEE Transactions on pattern analysis and machine intelligence 24.7 (2002): 971-987.
[20] Yousefi, Jamileh. "Image Binarization using Otsu Thresholding Algorithm." (2011).
[21] He, Kaiming, et al. "Deep residual learning for image recognition. CoRR abs/1512.03385 (2015)." (2015).