Mixed Heuristic Search for Sketch Prediction on Chemical Structure Drawing

Bo Kang,[*] Hao Hu,[†] Joseph J. LaViola Jr.[‡]
University of Central Florida, Department of EECS, Orlando, FL, USA

Abstract

Sketching is a natural way to input chemical structures that can be used to query information from a large chemical structure database. Based on a user's incomplete sketch of a chemical structure, sketch prediction becomes a challenging problem not only due to arbitrary drawings orders among users but also similarities among chemical structure layouts. In this paper, we present a graph-based approach to handle the sketch prediction problem. We use *multisets* as the data representation of hand-drawn chemical structures and create an undirected graph to handle data in all *multisets*. This approach transforms the sketch prediction problem into a search problem to find a *hamiltonian path* in the corresponding sub-graph with polynomial time complexity. We introduce mixed heuristics to guide the search procedure. Through an initial experiment on a hand-drawn chemical structure dataset, we demonstrate that in comparison with a baseline method, the proposed approach improves the prediction accuracy and efficiently predicts chemical structures from only partially sketched drawings.

CR Categories: I.7.5 [Document management and text processing]: Document capture—Graphics recognition and interpretation I.2.8 [Artificial intelligence]: Search methodologies—Search with partial observations;

Keywords: Chemical Structure Sketch Prediction, Graph Search, Heuristics, Hamiltonian Path

1 Introduction

Sketch-based interfaces present a pencil-and-paper-like approach to entering organic chemical structures on a computer which supports chemical structure prediction queries based on the partial sketch. For instance, users can sketch partial chemical structures instead of the full chemical diagram to explore similar chemical structures within a large chemical structure database, such as SciFinder. In addition, partial sketch prediction can guide users to draw their intended structures, especially when the user forgets how to draw the exact solution. However, sketch prediction is a difficult problem not only due to users' arbitrary drawing orders but also due to the similarity between chemical structure layouts. In this paper, we propose a chemical structure sketch prediction framework to tackle such problems.

1.1 Sketched Chemical Structure Variation

We asked users to enter different chemical structures using a digitizing Tablet PC in order to get a better sense of the types of variation that exists when users sketch chemical structures. A hand-drawn chemical structure is composed of chemical bonds and chemical symbols.

Figure 1 shows several hand-drawn chemical structure samples. In Figure 1(A), both the top and bottom hand-drawn samples represent *bromobenzene*. Yet their visual representations are different due to the position of single bonds and double bonds to construct the *benzene* ring. Besides, each chemical bond or chemical symbol is labeled as a number to indicate the drawing order. For instance, the *benzene* ring in the top sample is drawn in a clockwise manner started from the single bond which is labeled as 1. By contrast, the *benzene* ring in the bottom sample is drawn in a counterclockwise manner started from the double bond which is labeled as 1. Participants also draw a *benzene* in one stroke to represent the ring and then update three single bonds to double bonds. Thus, the same chemical structure can be drawn with an arbitrary drawing order.

In Figure 1(B), both the top and bottom hand-drawn samples represent *phenol*. Due to the symmetrical property of chemical structures, users might change their drawing orientations from the top sample to the bottom sample. Thus, the drawing orientation of a chemical structure can vary. In Figure 1(C), both top and bottom hand-drawn samples represent *1-bromopentan-2-ol*. Compared to the top sketch, a user explicitly draws several *carbon* chemical labels between every two chemical bonds. Thus, drawing conventions can vary with chemical structures. In Figure 1(D), the top and bottom hand-drawn samples represent two isomers. They have similar visual appearance except the position of the chemical label on structures. Therefore, chemical structure similarity could mean molecule isomers with different visual chemical structure layouts. These examples illustrate the variety of ways that a chemical structure can be written, making sketch prediction a challenging problem with a large search space.

1.2 Related Work

Sketch prediction is a challenging problem which depends on sketch recognition [Gennari et al. 2005; Hammond and Davis 2004; LaViola Jr. and Zeleznik 2007; Ouyang and Davis 2007; Peterson et al. 2010; Sezgin and Davis 2005] since it needs to interpret the incomplete user sketch and discriminate between object classes. In general, [Tirkaz et al. 2012] proposes a method by learning visual appearances of partial drawings through semi-supervised clustering, followed by a supervised classification step that determines object classes. [Costagliola et al. 2014] uses an Attributed Relational Graph to represent symbol and exploits a novel spatial descriptor to represent relations between two stroke primitives in order to make the symbol matching. [Mas et al. 2007] presents a syntactic approach to on-line recognition of sketched symbols. The symbols are defined by an adjacency grammar whose rules are generated automatically given the small set of seven symbols. The system can recognize partial sketches in arbitrary drawing order, using the grammar to check the validity of its hypotheses.

Regarding chemical structure sketch recognition, [Ouyang and Davis 2011] provided a state-of-the-art sketch recognition framework which combines multiple levels of visual features using a jointly trained conditional random field. [Sadawi et al. 2012] illustrates a rule-based system to recognize chemical structures.

[*] e-mail:bkang@cs.ucf.edu

[†] e-mail:hao_hu@knights.ucf.edu

[‡] e-mail:jjl@eecs.ucf.edu

SBIM 2014, August 08 – 10, 2014, Vancouver, British Columbia, Canada.
Copyright © ACM 978-1-4503-3018-3/14/08 $15.00

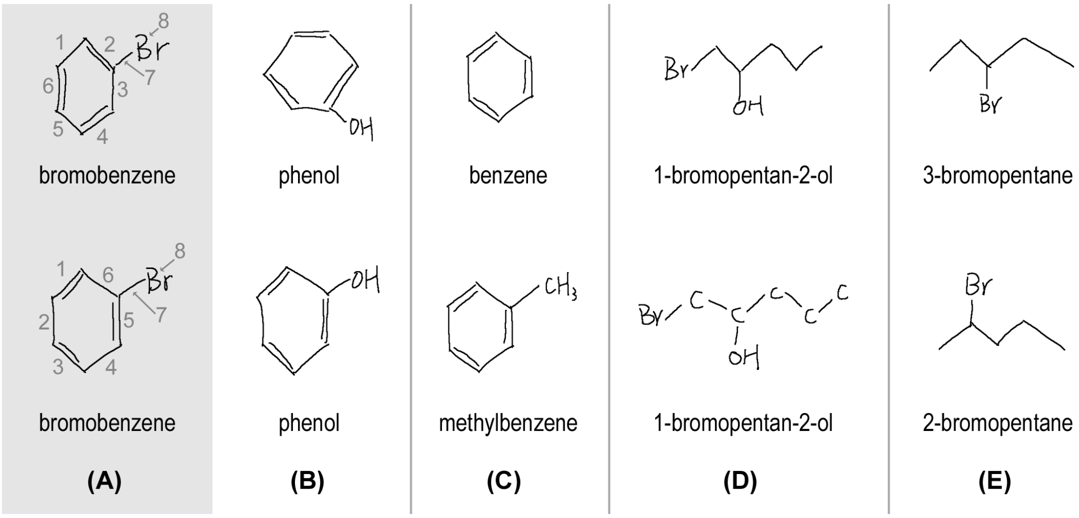

Figure 1: *Chemical structure hand-drawn samples. In Figure 1(A), the same chemical structure bromobenzene can be drawn in different drawing orders between the top sample and the bottom sample. Each chemical label or chemical symbol is marked with a number to represent the drawing order. In Figure 1(B), phenol can be drawn in different chemical structure layouts. In Figure 1(C), methylbenzene contains benzene in its visual layout. In Figure 1(D), 1-bromopentan-2-ol can be drawn by explicitly writing multiple carbon labels in the bottom sample. In Figure 1(E), 3-bromopentane and 2-bromopentane have similar visual layouts.*

With respect of structure prediction work, [Shatabda et al. 2013] proposed mixed heuristics to guide protein structure prediction. More generally, [Doppa et al. 2013] introduced a framework for structured prediction. Given a structured input, their framework uses a search procedure guided by a learned heuristic to uncover high quality candidate outputs and use another cost function to select a final prediction among the output.

In contrast to this prior work, the key contribution of our work is to treat the chemical structure sketch prediction as a structural prediction problem [Daumé et al. 2009]. After sketch recognition, hand-drawn chemical structures can be interpreted as a sequence of encoded symbols. Then we use *multisets* to represent such an encoded sequence. A *multiset* is a set which may contain multiple occurrences of the same encoded symbol. Through such structured input, we introduce a graph-based approach to store all *multisets* and use mixed heuristics to guide the search to find a *Hamiltonian path* in one sub-graph from the graph.

As the sketch prediction work depends on sketch recognition, we briefly discuss the recognition method which provides both reasonable correct recognition results and efficient recognition performance. After that, we explain our sketch prediction framework by explaining the graph-based approach and propose several heuristics to efficiently do the local search to meet our goal.

2 Sketch Recognition

Our sketch recognition system initially parses the ink strokes into a number of lines or arc primitives. We cluster these primitives to generate chemical symbols (chemical labels or chemical bonds). By using a domain specific encoding policy, we further change recognized chemical symbols to the corresponding encoded symbols, so that encoded symbols have more stroke information than chemical symbols. According to the drawing order, the generated encoded symbols form an encoded sequence [Sezgin and Davis 2005]. These encoded sequences are used as input to the sketch prediction process. Currently, our system can recognize two types of chemical symbols: chemical bonds (single bond, double bond, triple bond, hash bond and dash bond) and chemical labels (combination of letters).

2.1 Geometric Primitives Extraction

Each drawing stroke is a combination of atomic geometric primitives that can include line primitives and arc primitives. Breaking up a stroke in this fashion lets us distinguish cases where one stroke might contain many chemical single bonds. The segmentation procedure relies on geometric straw features [Wolin et al. 2008; Xiong and LaViola 2009] to find stroke corners on a stroke and decompose a stroke into a number of line primitives and arc primitives.

2.2 Symbol Interpretation

After extracting geometric primitives for each stroke, we cluster line primitives and arc primitives as symbol candidates and label them as chemical symbols. By interpreting the domain knowledge and geometric features of chemical structures, we assume that only line primitives can construct chemical bonds; both line primitives and arc primitives can form chemical labels. Based on it, we apply a clustering method [Ouyang and Davis 2007], which uses a time sliding window with a fixed length to group contiguous strokes. If these strokes contain an arc primitive, we send grouped strokes to a template-based symbol recognizer [Vatavu et al. 2012] to label them. As for the chemical bonds, before clustering, we use domain knowledge and geometric features to form different types of chemical bonds. After clustering, we perform domain knowledge verification to check if recognized chemical symbols satisfy chemical rules.

2.3 Encoding Policy

We encode recognized chemical symbols as shown in Table 1. Regarding chemical bonds, the slope feature of bonds further restricts a bond as a specific encoded symbol. For instance, a single bond with a positive slope is interpreted as 2; the double bond with both lines horizontal are interpreted as 4. Each chemical label directly

Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code
LineHorizontal	0	ParallelHorizontal	4	None	-1	Br	12
LineVertical	1	ParallelVertical	5	C	8	OH	14
LinePositiveSlope	2	ParallelPositiveSlope	6	N	9	NO_2	16
LineNegativeSlope	3	ParallelNegativeSlope	7	H	11	....	

Table 1: *Encoding examples in the codebook.*

maps to a corresponding encoded symbol. Incorrect recognition results lead to the error (-1) encoded symbol.

3 Sketch Prediction

The purpose of sketch prediction is to analyze the partial drawings interpreted as the encoded sequence through the sketch recognition process and match the most similar chemical structure from training encoded sequences. Figure 2 shows an example predicting four chemical structures based on the user's partial drawing.

We propose a graph-based method by first building a graph G from training encoded sequences which depends on mixed heuristics. Then using additional heuristics, we search in G for the substances that are good fit for the partial drawing.

Before illustrating the details on graph construction and partial drawing prediction, we first analyze the similarity between encoded sequences and then introduce the *multiset* as data representation which is used as the constraint heuristic in the partial drawing matching.

3.1 Encoded Sequences Analysis

Corresponding to Figure 1, there will be four different cases between encoded sequences. Assume there are two encoded sequences S_1, S_2:

- ***Case 1*** S_1, S_2 have the same label and encoded symbols collection with different orders (Figure 1(A)).
- ***Case 2*** S_1, S_2 have different labels and one encoded symbols collection is the subset of another one (Figure 1(C)).
- ***Case 3*** S_1, S_2 have the same label but different encoded symbol collections and orders (Figure 1(B, D)).
- ***Case 4*** S_1, S_2 have different labels, both have the same encoded symbol collections yet with different orders (Figure 1(E)).

From the above cases, we notice that encoded symbol collection and sequence order are two critical properties to disambiguate one encoded sequence from others. In order to use these two features to classify a partial drawing as a chemical structure, we further represent the encoded sequence as a *multiset* to declare the prediction boundary among training encoded sequences. A *multiset* is a set which may contain multiple occurrences of the same element. Using *multisets* allows the same encoded symbol to appear more than once in the set. Besides, it automatically incorporates other sequence orders which construct the same collection of encoded symbols. However, since several *multisets* may represent the same chemical structure because of diverse chemical structure layouts (Case 3 above), we set up the ground truth among these *multisets* to help us disambiguate.

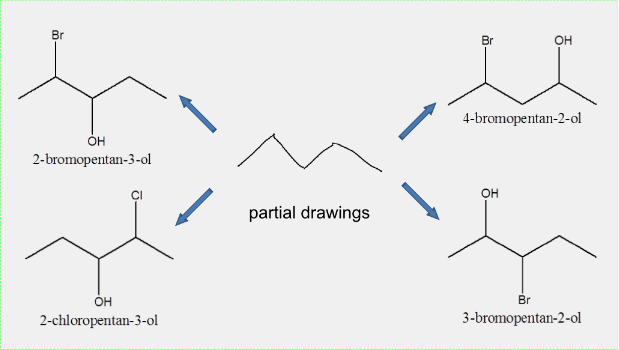

Figure 2: *Sketch prediction based on the user's partial drawing.*

3.2 Graph Representation

We represent all encoded sequences with their corresponding *multisets* as an undirected edge-weighted graph:

$$G = (V, E, \mathcal{W}) \tag{1}$$

where V is the vertex set and each vertex $v \in V$ maps to an encoded symbol which can exist in more than one encoded sequences. E is the edge set in which each edge $e \in E$ indicates the connection between every two contiguous encoded symbols in an encoded sequence. The weight $\mathcal{W} = (w_{ij}, w_{ji})$ represents the frequency at which an edge between vertices i and j appears in all training encoded sequences. As the order between two encoded symbols can vary, there are two edge weights w_{ij} and w_{ji} on the same edge.

As each vertex $v_i \in V$ on G can map to the same encoded symbol in different encoded sequences, we define the vertex constraint C_{v_i} on the vertex v_i, which is the set of *multiset* corresponding to the above encoded sequences.

3.3 Graph Construction

Algorithm 1 demonstrates the graph construction process. For each training encoded sequence S, we build the sequence onto the graph G using several steps.

Algorithm 1 Build encoded sequences onto a graph.

```
1: Input: encoded sequences $\mathbb{S}$
2: Initialize an empty graph $G$.
3: for each encoded sequence $S$ with length $N$ in $\mathbb{S}$ do
4:     $M \leftarrow$ Corresponding multiset of $S$.
5:     $G_{Sub} \leftarrow$ Construct_Vertices_On_Graph($S$, $G$).
6:     for $i$ = 1 to N do
7:         $V_{current} \leftarrow G_{Sub_{i-1}}$, $V_{next} \leftarrow G_{Sub_i}$.
8:         Add $M$ to $C_{V_{current}}$, $C_{V_{next}}$.
9:         Add or Update an edge from $V_{current}$ to $V_{next}$.
10:    end for
11:    Add or Update an edge from $G_{Sub_{N-1}}$ to $G_{Sub_0}$.
                                   ▷ $S$ forms a Hamiltonian Cycle on $G_{Sub}$
12: end for
13: Output: Graph $G$.
```

First, we compute the current encoded sequence S's corresponding *multiset* M (line 4 of the algorithm). In line 5 of the algorithm, we search S on G to find a subgraph G_{Sub} of G. For every encoded symbol $s_i \in S$, our goal is to match one vertex v_i on the graph to indicate this symbol. We utilize the two heuristics below to guide the search to find such vertex.

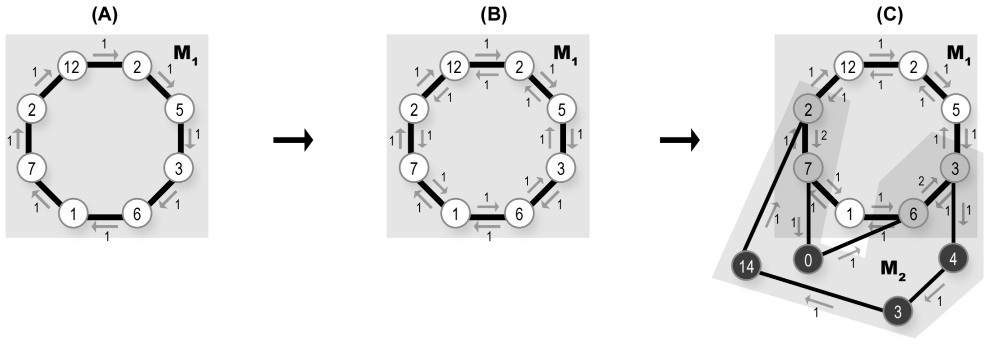

Figure 3: *Iterative graph construction by building three encoded sequences. (A) is generated after building bromobenzene's encoded sequence (2,5,3,6,1,7,2,12). (B) is generated after building bromobenzene's encoded sequence (2,7,1,6,3,5,2,12). (C) is generated after building pheno's encoded sequence (2,7,0,6,3,4,3,14). The arrow represents the drawing order between two encoded symbols and the number on the arrow shows the weight of each edge. The color of node indicates which multiset it belongs to. For instance, the node with number 7 in (C) is included by both multisets M_1 and M_2 while the node with number 12 in (C) is contained only by M_1.*

Heuristic-1 For the encoded symbol s_i, we can find out all the matching vertices which corresponds to s_i. If we have chosen the vertex v_{i-1} on the graph which is corresponding to s_{i-1}, we check if an edge on the graph G exists between the vertex v_{i-1} and each matching vertex. If only one edge exists, we can select that edge's endpoint v_i to represent s_i. If more than one edge exists, we further apply *Heuristic-2* to select v_i. However, if no edge exists, we pick v_i from vertex candidates which holds the maximum number of vertex constraints.

Heuristic-2 Each edge in the graph G has a weight to represent the connection frequency between two encoded symbols in all training encoded sequences. If there are multiple vertices which have an edge to connect the vertex v_{i-1}, we will select the vertex with the maximal weight on $e_{i-1,i}$ as v_i.

For these two heuristics, applying *Heuristic-1* alone will average the weight of edges which increase the ambiguity to select a vertex for the encoded symbol. While applying *Heuristic-2* alone may create exceeding vertices that represent the same encoded symbol, which also make it difficult to select a vertex. Therefore, we combine these two heuristics to maximize the utilization of existing vertices on the graph and at the same time to guarantee to choose the correct vertex based on edge weights.

For every encoded symbol $s_i \in S$, we search for a set of vertex candidates in which each vertex has the same encoded symbol as s_i. If no vertex candidate matches s_i, we create a vertex to map on this encoded symbol. If more than one vertex candidate match the current encoded symbol, we further apply *Heuristic-1* and *Heuristic-2* till only one vertex candidate is left to represent the encoded symbol s_i. If no vertex on the graph satisfies these heuristics, we create a vertex to map on this encoded symbol.

The intuition behind these two heuristics is that as there might be similar encoded sequences with the same prefix, *Heuristic-1* can guarantee to match the current sequence S onto the graph with the longest matching encoded sequence. The edge weight property records users' drawing sequence behavior. So, *Heuristic-2* can project S onto the graph with the most frequent encoded sequence.

After retrieving the subgraph G_{Sub} from G to represent the current encoded sequence, we iterate each vertex G_{Sub_i} in G_{Sub} to update its vertex constraint by adding the current encoded sequence's corresponding *multiset* M (line 8 of the algorithm). In addition, between every two contiguous vertices in G_{Sub}, if no edge exists, we create an edge; otherwise, we update the edges corresponding weight (line 9 of the algorithm).

In line 11 of the algorithm, after every encoded symbol in the sequence has found the vertex on G, between the last vertex and the first vertex, if no edge exists, we create an edge; otherwise, we update the edges corresponding weight. By doing this every training encoded sequence S forms a *hamiltonian cycle* in a subgraph of G.

Figure 3 builds three encoded sequences onto the graph. Based on the encoding policy in Table 1, both sequence (A) and (B) reflect the top hand-drawn sample in Figure 1(A). The benzene ring in sequence (A) is drawn in the counterclockwise manner whereas the same ring in sequence (B) is drawn in the clockwise manner. Both sequence (A) and (B) are represented by the same *multiset* M_1. Sequence (C) corresponds to the top hand-drawn sample in Figure 1(B). The sequence is represented by the *multiset* M_2.

3.4 Partial Encoded Sequence Prediction

Algorithm 2 illustrates the partial sketch prediction process. It returns a predicted ground truth chemical structure representing a input partial sequence.

Algorithm 2 Sequence prediction on the graph

```
1:  Input: partial encoded sequence $S'$ and graph $G(V, E, \mathcal{W})$.
2:  $P \leftarrow$ Search_Partial_Path_On_Graph($G$,$S'$).
3:  $C \leftarrow$ Get_Search_Constraints($P$).
4:  while $\exists$AmbiguityInSearchConstraint($C$) do
5:      Search_Hamiltonian_Path($P, C$).
6:  end while
7:  $M \leftarrow$ the multiset with the smallest size in $C$.
8:  if $\exists$AmbiguityInMultiset($M$) then
9:      Post Processing.
10: end if
11: Output: Ground truth chemical structure which $M$ represents.
```

In line 2 of the algorithm, we search the partial encoded sequence S' on the graph G to find all corresponding vertices to form a path P. During the search, after deciding the first vertex v_0 for the encoded symbol s'_0, we define the global search constraint C as the vertex constraint C_{v_0}. After selecting every v_i toward the encoded symbol s'_i, we update C as:

$$C = C \cap C_{v_i} \quad (2)$$

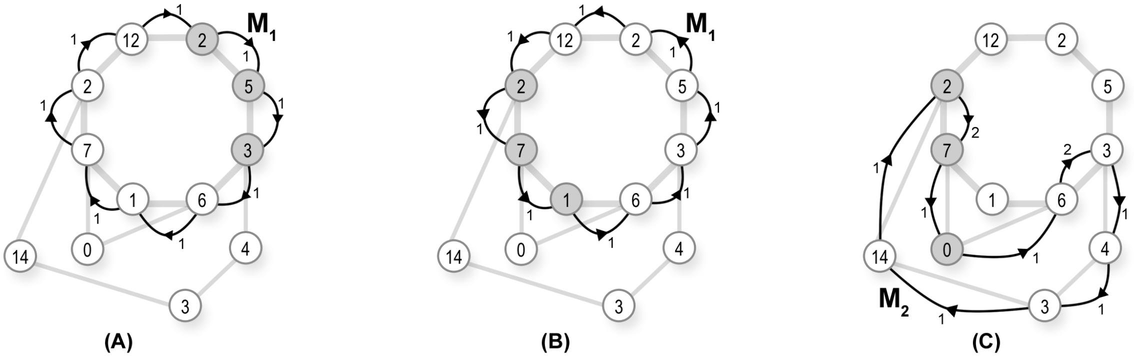

Figure 4: *Predictive graph search for three different partial sequences. (A) shows the search result for the partial sequence (2,5,3). (B) shows the search result for the partial sequence (2,7,1). (C) shows the search result for the partial sequence (2,7,0).*

For every encoded symbol $s'_i \in S'$, in order to decide the mapping vertex on G, similar to the graph construction search, we initially search for a set of vertex candidates in which each vertex has the same encoded symbol as s'_i. If no vertex candidate matches any encoded symbol s'_i, the algorithm will terminate without any prediction. If more than one vertex candidate match the current encoded symbol, we first use the same *Heuristic-1* in the constructive graph search to reduce the vertex candidate. After that, we apply the below *Heuristic-3* to further eliminate vertex candidates.

Heuristic-3 When searching for the vertex v_i toward the encoded symbol s_i, we already get the updated C from the last $i-1$ matched vertices. For every vertex candidate's v_i, we check whether $C_{v_i} \cap C =\not\emptyset$. If the intersection is not empty, we mark this vertex candidate as consistent with regard to C. If the intersection is empty, it means there will be no predicted *multiset* in C after selecting the vertex candidate v_i. In other words, this v_i should be eliminated from vertex candidates. In the specific case, many vertex candidates maintain the consistency with C. As the number of *multisets* differ in the intersection set between C and every vertex candidate, we select the vertex candidate, which contains the largest number of *multisest* in the intersection set.

Though applying *Heuristic-3*, it is possible that global search constraint C still includes more than one *multiset*. We further utilize the edge weight property in *Heuristic-2* in the constructive graph search to eliminate vertex candidates. However if we still cannot make a decision to select a vertex after applying the above heuristics, we will randomly picks one vertex from vertex candidates.

After deciding all vertices for S', we retrieve the updated global search constraint C (line 3 of the algorithm). From line 4 till line 6 of the algorithm, based on the existing path P, we continue to search for a *Hamiltonian path* which attempts to visit all vertices in one of the subgraphs corresponding to one *multiset* in C. During the search (line 5 of the algorithm), similar to the search in line 2 of the algorithm, we use *heuristic-3* and *heuristic-2* continuously to complete the *Hamiltonian path* search procedure. However if we still cannot a make decision to select a vertex after applying the heuristics, we will randomly pick one vertex from vertex candidates. When a new vertex is added into the path, we update the global search constraint C. Such a search procedure will be terminated when the global search constraint C only contains one *multiset* (line 4 of the algorithm).

Regarding *case 2* in Section 3.1, the global search constraint C might contain both the *multiset* M_a and *multiset* M_b ($M_a \subset M_b$). Under such conditions, we apply *Heuristic-4*.

Heuristic-4 Each *multiset* assigns a counter variable and initializes its value as the size of itself. When updating the global search constraint C, we also update the counter in each *multiset* of C by subtracting 1 from it. When the counter reaches 0 for the *multiset* M with the smallest size, it indicates that the *Hamiltonian path* of subgraph in G is found which corresponds to M.

We add *Heuristic-4* into the search termination in line 4 of the algorithm, which checks if the counter in any one *multiset* of C becomes zero. If such a *multiset* exists, we will select it and neglect other *multisets* which are both in the global search constraint C.

After the graph search procedure from line 2 to line 6 of the algorithm, if the global search constraint C only leaves one multiset or the search procedure finds a *Hamiltonian path* in the multiset with the smallest size, we can successfully get a *multiset* M from the global search constraint C (line 7 of the algorithm). Otherwise, the search procedure fails to make the prediction.

3.4.1 Post Processing

Due to *case 3* and *case 4* in Section 3.1, a *multiset* can contain multiple labels, in which some labels represent ground truth hand-drawn chemical structures, whereas the others do not. From line 7 to line 9 of the Algorithm, we do post processing to disambiguate and select the ground truth chemical structure from the *multiset* M.

Regarding *case 3*, as different *multisets* can reflect the same chemical structure, we set up the ground truth *multiset* among them. We build a relation between the ground truth hand-drawn chemical structure and other various hand-drawn samples which can group all different *multisets* corresponding to the same ground truth. As a result, we eliminate such ambiguity through this pre-constructed relationship.

Regarding *case 4*, though two encoded sequences' *multisets* are identical, we additionally analyze recognized symbols' spatial information in order to make the distinction. During sketch recognition, we capture chemical bonds' 2D relationship on the drawing canvas. Then we generate another spatial encoded symbol sequence by sorting all the recognized chemical bonds from the left to the right and from the top to the bottom of the canvas. During sketch prediction, we match the partial spatial encoded information with training chemical structures. If the partial spatial encoded sequence is the prefix of any complete spatial sequence, the corresponding ground truth of that complete sequence will be the prediction candidate for the partial encoded sequence.

Based on the graph from Figure 3, Figure 4 shows three partial encoded sequence prediction examples. All of them correctly predict the ground truth chemical structures. Compared to partial se-

CR	$0-19\%$		$20\%-39\%$		$40\%-59\%$		$60\%-79\%$		$80\%-99\%$		100%	
User	Graph	Baseline	Graph	Baseline	Graph	Baseline	Graph	Baseline	Graph	Baseline	Graph	Baseline
1	0.58	0.36	0.67	0.53	0.78	0.70	0.82	0.72	0.86	0.85	1.0	1.0
2	0.64	0.50	0.71	0.56	0.75	0.57	0.79	0.70	0.81	0.78	1.0	1.0
3	0.71	0.49	0.79	0.52	0.80	0.67	0.82	0.73	0.84	0.84	1.0	1.0
4	0.67	0.35	0.75	0.47	0.82	0.71	0.84	0.75	0.90	0.85	1.0	1.0
5	0.65	0.52	0.73	0.59	0.84	0.64	0.87	0.78	0.89	0.81	1.0	1.0
6	0.57	0.44	0.58	0.50	0.76	0.56	0.80	0.61	0.88	0.79	1.0	1.0
Average	0.64	0.43	0.70	0.53	0.79	0.64	0.82	0.71	0.86	0.82	1.0	1.0

Table 2: *User-dependent sketch prediction correctness result across six different completion rates (CR).*

CR	$0-19\%$		$20\%-39\%$		$40\%-59\%$		$60\%-79\%$		$80\%-99\%$		100%	
User	Graph	Baseline	Graph	Baseline	Graph	Baseline	Graph	Baseline	Graph	Baseline	Graph	Baseline
1	0.43	0.26	0.55	0.35	0.57	0.46	0.57	0.45	0.62	0.45	0.71	0.52
2	0.32	0.20	0.41	0.27	0.47	0.34	0.49	0.35	0.56	0.38	0.69	0.47
3	0.41	0.19	0.55	0.24	0.57	0.41	0.56	0.48	0.60	0.48	0.65	0.51
4	0.52	0.24	0.59	0.30	0.61	0.36	0.67	0.46	0.72	0.47	0.80	0.58
5	0.41	0.34	0.56	0.39	0.64	0.52	0.64	0.57	0.73	0.58	0.79	0.62
6	0.34	0.31	0.50	0.32	0.61	0.42	0.64	0.47	0.64	0.53	0.69	0.59
Average	0.40	0.26	0.53	0.31	0.58	0.42	0.59	0.46	0.64	0.48	0.72	0.55

Table 3: *User-independent sketch prediction correctness result across six different completion rates (CR).*

quences (B) and (C), when parsing the partial sequence (2,7), the algorithm cannot determine the *Hamiltonian path* between vertex 0 and vertex 1 which both connect to the vertex 7. Under this condition, the algorithm will randomly pick a vertex and follow one *Hamiltonian path* to search.

4 Evaluation

We ran an experiment to explore the effectiveness of our approach. In comparison with a baseline method, we evaluated the performance of our graph-based approach in two ways: prediction accuracy and execution time. In order to guarantee correct sketch prediction across all four cases in the previous section, we only show the performance result below by using all heuristics. We implemented this graph-based algorithm in C# and integrated it with our existing sketch recognition system. We used the HP Compaq tc4400, 12.1 inch tablet PC running Windows 7 to conduct the experiment.

4.1 Experiment and Data Set

As we did not find any sketched datasets which tackle the combinatorial drawing orders in the chemistry domain, we designed 4 sets of 2D molecular structures to capture users' drawing orders from common organic molecules, with each set having 5 visually similar structures. In total, there are 20 chemical structure class categories for the evaluation. All these chemical structures cover users' arbitrary drawing orders. Some sets also cover the similarity problem among chemical structure layouts. We recruited six participants with an organic chemistry background to draw these chemical structures. Participants were asked to draw each chemical structure 10 times consecutively. During data collection, the sketch recognizer gives good recognition results. However, if the recognizer makes an error, we let the user re-draw the same structure. During that process, the ground truth chemical structure is shown on the drawing canvas for a short period of time. Users were asked to draw the same ground truth structure the first time. After that, they draw the same structure using their own drawing styles another 9 times. In total, we collected 1200 encoded sequence instances of training data. The encoded symbols among these sequences fall in a range between $0-21$. The length of encoded sequences is between $5-33$. The number of *multisets* is within the range of $47-107$. Based on this training data set, we obtained the testing data by extracting the ordered partial sequences from each complete encoded sequence in the training data set.

4.1.1 Baseline Method

Since our method is designed to tackle the chemical structure sketch prediction problem with diverse drawing orders and similar chemical structure layouts, we need to choose a baseline method which concentrates on the same aspects of the problem. Based on the encoding policy for chemical structures, we simply implemented a HMM based sketch prediction method [Sezgin and Davis 2005]. The schematic representation of the HMM topology is the *left-to-right* model [Rabiner 1990]. Such a model can represent the temporal structure of encoded sequences. We trained each chemical structure using one HMM model which can handle variable length encoded sequence data. Regarding one chemical structure's corresponding model, we set the number of hidden states as the maximal length of all training encoded sequences. We achieve the sketch prediction goal by finding the *Viterbi path* in the model.

4.2 Sketch Prediction Accuracy Analysis

We conducted both user-dependent and user-independent tests to assess sketch prediction accuracy between the graph-based approach and the baseline approach. In the user-dependent test, we use each participant's own complete dataset to train models of both methods and use its own extracted test dataset to validate models. In the user-independent test, we conduct a 6-fold cross-validation on these 6 participants' datasets. For both metrics, we ran an average of 1683 tests each among 6 different users.

For both the graph-based approach and the baseline approach, to measure sketch prediction accuracy, we compute prediction correctness under the different sequence completion rates. A correct prediction means that the prediction result is the same as the ground truth chemical structure. Therefore, the prediction correctness is the ratio of the number of correct predictions among the total predictions. The completion rate is calculated by dividing the length of the partial encoded sequence by the length of its corresponding complete encoded sequence in the training data set. In some cases, the decision cannot be clearly made due to insufficient partial information. We consider these cases as the correct prediction as long as the predicted multiset contains the entire partial sequence.

In the user-dependent evaluation, Table 2 shows the sketch prediction correctness for six users. For each user, we report the prediction correctness between two methods in six different completion rate ranges. For instance, for partial sequences from user 2, in which the completion rate of each of them is between 20% and 39%, the prediction correctness to use the graph-based method is 0.71, which means that 71% of partial encoded sequences, are correctly matched to the ground truth chemical structure. The last row of the table shows the average prediction correctness for all users in different completion rate ranges. For both methods, the result from Table 2 implies that when the completion rate goes up, the prediction correctness increases. Besides, when the completion rate achieves 100%, there is no wrong prediction for partial sequences. Indeed, under this circumstance, the sketch prediction problem becomes the sketch recognition problem. Compared with the baseline method, the graph-based method shows prediction correctness gains in different completion rate ranges.

In the user-independent evaluation, Table 3 gives the sketch prediction result. In comparison with the user-dependent test, the prediction correctness in the user-independent test goes down for both methods. This can be explained by users' diverse drawing styles, which might generate different visual layout for the same chemical structure. In other words, by using other users data to train models, if the current user draws the same chemical structure using non ground-truth layouts, it becomes easier to fail to predict the ground-truth structure. From the last row of the table, the average prediction correctness of the graph-based method outperforms the baseline method which spans all completion rate ranges.

4.3 Sketch Prediction Execution Time Analysis

Theoretically, finding a Hamiltonian path in a graph is a *NP-complete* problem [Garey and Johnson 1979]. We reduce the graph exhaustive search complexity to polynomial time after applying mixed heuristics. The running time of the heuristic search algorithm depends on three factors: the length of the partial sequence, the size of the search constraint and the number of vertices in the graph. We run the performance test based on our dataset. Our experimental machine has a 2.30 GHz Intel Core i5 with 4 GByte main memory. For all six users' tests across completion rates, the average time for sketch prediction takes roughly *0.14s*. So it meets the real-time requirement of providing a sketch prediction result in our sketch-based system.

5 Discussion and Future Work

Our evaluation results demonstrate promising prediction accuracy with real-time execution. However, when both the size of dataset and length of an encoded sequence increases, the prediction accuracy might decrease and real-time sketch prediction cannot be guaranteed. Therefore, we need to further explore ways to represent hand-drawn chemical structures more efficiently. One possible method is to apply more advanced encoding policies to reduce the encoding length. On the other hand, with the longer encoded sequence, the performance of applying Heuristic-3 will be significantly downgraded since we need to compare and find the proper *multiset* in every prediction step from more *multisets*. Therefore, we will try to find more sophisticated search algorithms to boost the search performance.

Besides the above algorithmic improvement, we could further improve the algorithm by making it stroker order invariant. Although the same chemical structure can be sketched using different stroke orders and these different ordering pose an issue with our algorithm, it will have no impact on the visual sketched structure. Thus, it is possible to pre-process the sketched structure such that the encoding is order invariant, which might reduce the computational complexity of the search algorithm and improve the prediction accuracy.

Currently, our graph-based sketch prediction method relies on the domain specific dataset and encoding policy, which does not fit for other domains. Thus it is important to generalize our method to handle different datasets and encoding policies.

Regarding the sketch recognition, our sketch prediction method depends on the correct recognition toward chemical symbols and bonds. In other words, we do not consider the recognition error effects in the sketch prediction. In the future, we would like to investigate how to improve our graph approach to handle sketch recognition errors.

6 Conclusion

In this paper, we formalize the chemical structure sketch prediction problem as a structure prediction problem. We use *multisets* to represent hand-drawn chemical structures, and propose a graph-based method to handle all *multisets*. Through such structured input, we transform sketch prediction to a graph search problem by using mixed heuristics to guide the search to find a *Hamiltonian path* in one sub-graph. We evaluated the graph-based approach through our hand-drawn chemical structure data set. The results shows that our prediction method outperforms a baseline method both in user-dependent and user-independent tests.

Acknowledgements

This work is supported in part by NSF CAREER award IIS-0845921 and NSF awards IIS-0856045 and CCF-1012056. We would like to thank Jennifer Reed and Simon Tang for discussions on the similarity among chemical structures and the design of test cases for our evaluation. Also, we would like to thank the members of ISUE lab for their support and the anonymous reviewers for their useful comments and feedback.

References

COSTAGLIOLA, G., DE ROSA, M., AND FUCCELLA, V. 2014. Technical section: Recognition and autocompletion of partially drawn symbols by using polar histograms as spatial relation descriptors. *Comput. Graph. 39* (Apr.), 101–116.

DAUMÉ, III, H., LANGFORD, J., AND MARCU, D. 2009. Search-based structured prediction. *Mach. Learn. 75*, 3 (June), 297–325.

DOPPA, J. R., FERN, A., AND TADEPALLI, P. 2013. Hc-search: Learning heuristics and cost functions for structured prediction. In *AAAI*, AAAI Press, M. desJardins and M. L. Littman, Eds.

GAREY, M. R., AND JOHNSON, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

GENNARI, L., KARA, L. B., STAHOVICH, T. F., AND SHIMADA, K. 2005. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Comput. Graph. 29*, 4 (Aug.), 547–562.

HAMMOND, T., AND DAVIS, R. 2004. Automatically transforming symbolic shape descriptions for use in sketch recognition. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI Press, AAAI'04, 450–456.

LaViola Jr., J. J., and Zeleznik, R. C. 2007. A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 11 (Nov.), 1917–1926.

Mas, J., Sanchez, G., Llados, J., and Lamiroy, B. 2007. An incremental on-line parsing algorithm for recognizing sketching diagrams. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 01*, IEEE Computer Society, Washington, DC, USA, ICDAR '07, 452–456.

Ouyang, T. Y., and Davis, R. 2007. Recognition of hand drawn chemical diagrams. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, AAAI Press, AAAI'07, 846–851.

Ouyang, T. Y., and Davis, R. 2011. Chemink: a natural realtime recognition system for chemical drawings. In *Proceedings of the 16th international conference on Intelligent user interfaces*, ACM, New York, NY, USA, IUI '11, 267–276.

Peterson, E. J., Stahovich, T. F., Doi, E., and Alvarado, C. 2010. Grouping strokes into shapes in hand-drawn diagrams. In *AAAI*, AAAI Press, M. Fox and D. Poole, Eds.

Rabiner, L. R. 1990. Readings in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ch. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, 267–296.

Sadawi, N. M., Sexton, A. P., and Sorge, V. 2012. Chemical structure recognition: a rule-based approach. In *DRR*, SPIE, C. Viard-Gaudin and R. Zanibbi, Eds., vol. 8297 of *SPIE Proceedings*.

Sezgin, T. M., and Davis, R. 2005. Hmm-based efficient sketch recognition. In *Proceedings of the 10th international conference on Intelligent user interfaces*, ACM, New York, NY, USA, IUI '05, 281–283.

Shatabda, S., Newton, M. A. H., and Sattar, A. 2013. Mixed heuristic local search for protein structure prediction. In *AAAI*, AAAI Press, M. desJardins and M. L. Littman, Eds.

Tirkaz, C., Yanikoglu, B., and Metin Sezgin, T. 2012. Sketched symbol recognition with auto-completion. *Pattern Recogn. 45*, 11 (Nov.), 3926–3937.

Vatavu, R.-D., Anthony, L., and Wobbrock, J. O. 2012. Gestures as point clouds: A $p recognizer for user interface prototypes. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ACM, New York, NY, USA, ICMI '12, 273–280.

Wolin, A., Eoff, B., and Hammond, T. 2008. Shortstraw: a simple and effective corner finder for polylines. In *Proceedings of the Fifth Eurographics conference on Sketch-Based Interfaces and Modeling*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SBIM'08, 33–40.

Xiong, Y., and LaViola, Jr., J. J. 2009. Revisiting shortstraw: improving corner finding in sketch-based interfaces. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, ACM, New York, NY, USA, SBIM '09, 101–108.