Recognising Chemical Formulas from Molecule Depictions

Noureddin Sadawi

School of Computer Science, University of Birmingham, UK,
N.M.Sadawi@cs.bham.ac.uk
http://www.cs.bham.ac.uk/~nms

Abstract

In this paper we present a system for analysing and understanding structural molecular diagrams of the type commonly found in documents from the chemical and life science disciplines. Our approach can automatically reconstruct the basic formula representation corresponding to such a diagram and, further, reconstruct its full SMILES (Simplified Molecular Input Line Entry specification) specification [17] — a format that requires semantic understanding of the diagram.

1 Introduction

The majority of molecular drawings found in recent publications in the fields of chemistry and the life sciences are generated using computer programs. However, the publication process renders them as images, leaving them inaccessible to software tools for molecular modelling and manipulation, or even to visually impaired readers. Therefore, a freely available and high quality system for analysing such diagrams and extracting a non-image based description of the molecules depicted would be of significant value. This paper describes our current progress on building such a tool.

One of the first projects in this area was Kekulé [7, 8]. Developed as a commercial tool, Kekulé worked from scanned images, applied vectorisation, bespoke dashed line and wedge recognition, a neural network for character recognition (after significant preparation of the image involving deskewing, scaling, under-sampling and contrast and density adjustments), ligature recognition and a graph compilation algorithm on remaining vector data to produce an internal data structure from the diagram. This internal data structure could then be transformed into a number of standard chemical formats including SMILES. The tool was able to recognise most common elements of chemical structural diagrams and provided interactive molecular diagram correcting and editing facilities as well. Kekulé is no longer available commercially and was never released non-commercially.

CLiDE (Chemical Literature Data Extraction) [5, 16] is another commercial product. Its approach starts with connected component extraction, then uses contour analysis in its vectorisation algorithm to find curved and straight polygon approximations to connected components in the diagram. Again a neural network is used for character recognition and considerable attention was given to handling difficult cases of damaged and noisy images. As with Kekulé, most common elements of chemical structural diagrams are recognised.

IBM Almaden developed a system [3] that uses many-sided bounding polygons of a particular constrained form to approximate a hull for each connected component. Based on distances between the different hulls, the connected components are grouped into diagrams, segmenting them from the rest of the document. The resulting groups are vectorised, and features of these vectorisations are used to classify components in the diagram. Components identified as characters are recognised in a feature based approach on the pixel version of the character image. Accuracy only for simple planar diagrams is claimed.

The chemOCR project [2, 1] also uses a connected component approach and their own vectorisation algorithm, but with a support vector machine for character recognition. Connected components that are not recognised as characters are used to produce a graph of vectors. A rule based approach reconstructs the molecule description from the vector graph and the character information.

ChemReader [12] starts with connected component extraction. Character components are classified and recognised using the GOCR open source OCR tool [15]. Graphical components are recognised using Hough transforms, corner detection and other bespoke algorithms.

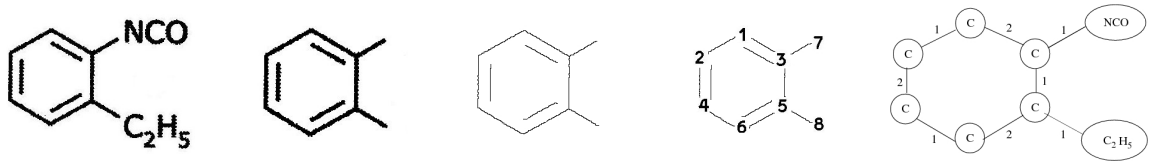

Figure 1: Recognition steps for the molecule C_9H_9NO.

OSRA [6] is, as far as we know, the only free, open source tool in this field that is currently available. It heavily uses other open source tools for image manipulation, vectorisation and OCR.

There have been a number of tools for recognition of handwritten chemical diagrams [14, 11].

2 Molecule Recognition Procedure

Our molecule recognition takes a single molecule depiction either from a scanned image or a digitally generated one, and returns a graph structure representing that molecule. This data structure can be used for further post-processing. The recognition is carried out by a combination of standard techniques and bespoke algorithms. The following is a schematic overview of the single steps in our procedure. Figure 1 presents intermediate steps of our recognition procedure for an example molecule image.

Algorithm 1 (Recognition Procedure)
INPUT:

An image of a chemical molecule

OUTPUT:

A graph $G = (V, E)$ representing the molecule structure

METHOD:

1. *Image binarisation*
2. *Connected Component Labelling*
3. *Character Extraction and Recognition*
4. *Image Thinning*
5. *Line Endpoint Extraction*
6. *Creation of Graph Representation*

Steps 1 and 4 are done using standard algorithms — image binarisation is performed using Otsu's method [10] and image thinning using Hilditch's algorithm [9] — and we will therefore not explain those in great detail here. The remaining steps of the procedure are given in more detail in the following subsections.

2.1 Segmentation

We perform the segmentation of the single components of the image by applying connected component labelling. In our particular approach we traverse the image from left to right, top to bottom searching for unlabelled black pixel. Once such a pixel has been found we label the entire connected component containing that pixel using a grass-fire algorithm [13] that recursively searches outwards for neighbouring unlabelled black pixels. Once the entire component has been labelled the systematic traversal continues.

Figure 2: Zoning of a character image

Figure 3: Types of bonds

2.2 OCR and Removal of Letters

In order to recognise characters in the input image, we use a set of numerical features that are extracted from the segmented symbols. These features are extracted via superimposing a 3 by 3 grid on each character candidate as displayed in Fig. 2, hence zoning the sub-image [4], and then calculating the percentage of feature pixels in each zone. To improve performance, we create a vector of 10 features containing the percentages of the 9 resulting values and a width to width plus height ratio.

These features are used in a classification process to assign a character label to each symbol. However, this step is followed by a context dependent verification step in some cases. For instance, in case of letters that may have similar shapes to those of a single straight-line (lower case L for instance), we classify them according to their position. That is, if they are located close to another letter and the letter is on the left side of this candidate l, they will be classified as a character. This is best represented in the case of the chemical symbol of Chlorine, Cl. However, problems can occur in case of single letters in particular fonts (e.g., a capital I) that resemble single lines; we discuss this in more detail in Sec. 4. This step ends with erasing the identified characters and grouping of characters that have been identified as consecutive symbols in the input image.

2.3 3D Bond Extraction

When drawing molecular structures, showing accurate 3D arrangements can be essential. To achieve this, different symbols are used. As Fig. 3 illustrates, solid triangles are used to show bonds coming out of the plane, while several short parallel lines are depict bonds going into the plane, and normal lines are used to show planar bonds.

To extract precise geometric information about bonds in the shape of solid triangles, we use a disc of a certain radius that can fit into the base of an average solid triangle. When a triangle base is found, another disc is drawn around the border of the first disc in 8 directions (see Fig. 4). Here, we use numerical features to accurately determine both the location and orientation of any existing bonds of such type. We then save the location and orientation of the triangle, and when the image is later thinned and line end points are extracted, we find the nearest line to this disc and label it accordingly.

Also, to detect 3D bonds represented as dashed lines, we scan the image for short lines whose centres are located on the same line and that are relatively close to each other (see Fig. 5). The figure also shows a case when the dashed lines are vertical. For the OCR stage, these lines might look like candidate lower case l, but due to the strict rules that we use, the adjacent symbol from the left hand side is not a C so these lines will not be classified as ls. At the end of this step, precise geometric information of any existing 3D bonds is saved for later use.

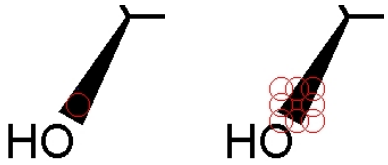

Figure 4: Detecting 3D bonds (solid triangle case)

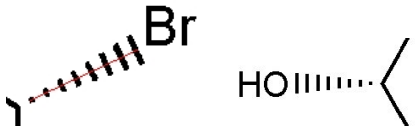

Figure 5: Detecting 3D bonds (dashed lines case)

2.4 End Point Extraction

After removing the characters in the character recognition step and after handling any existing 3D bonds, remaining parts of the image is thinned to a single pixel width using Hilditch's thinning algorithm [9]. Since the remaining components of the input image are the lines and shapes which form the actual drawing, we traverse the image horizontally, vertically and diagonally to find line junctions (locations of bonds meeting points). We measure the distance between consecutive feature points and use a certain threshold to determine whether those points belong to connected lines. Whenever a candidate junction if found, those lines are disconnected as shown in Fig. 1.

After detecting junctions, we scan the image horizontally and look for any feature points. Whenever a feature point is found, we visit all connected feature points in a breadth-first search manner until all connected points are visited. Since the lines are of one-pixel width, this guarantees us the precision of our calculated lines start and end points. This process is repeated recursively until all lines are processed and an initial list of line coordinates is created.

There are several special cases that need attention. Fig. 6 illustrates three different molecules with bonds represented as a mixture of parallel lines of varying length. In each of these cases the middle line actually represents several bonds between at least three Carbon atoms. For example, the right-most of the three molecules contains three Carbon atoms, two connected with a triple bond and two connected with a single bond. We handle these cases by finding parallel lines that are close to each other and of varying length. We determine the number of separate bonds involved by drawing auxiliary lines perpendicular to the line joining the endpoints of the shorter lines, thus cutting the longer line at the point of two joining bonds.

2.5 Graph construction

After accurate information on all bonds and atoms has been gathered, the final step of the procedure is to combine the information on the molecule extracted during the recognition process in a data structure that can

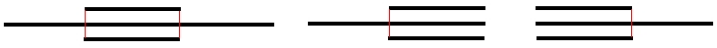

Figure 6: Case of short bonds on sides of longer bond

be used for further processing. We thereby construct an undirected graph $G = (V, E)$, where V is a set of labelled vertices and E is a set of labelled edges.

Each vertex $v \in V$ is labelled with the atom or molecular sub-formula it represents. In our example in Fig. 1, we introduce a vertex for each of the carbon atoms as well as one for each of the sub-formulas NCO and C_2H_5. Note that we explicitly represent each carbon atom that is normally omitted in the molecule depiction with a label C in the graph.

The edges in the graph represent the bonds between the single components of the molecule. Thereby each $e \in E$ is labelled with an integer from 1 to 5, representing one of the the following five types of bonds: 1: a single bond, 2: a double bond, 3: a triple bond, 4: a protruding 3D bond (i.e., a solid triangle), and 5: a receding 3D bond (i.e., a dashed triangle).

3 Extracting Meaningful Chemical Representations

After the initial recognition procedure the resulting graph data structure can be used to further post-processing. In particular, we use it a basis for drivers that transform the gathered information into representations that are meaningful from a chemical point of view. A first, simple driver simply extracts the chemical formula of the molecule recognised. A second, more elaborate driver extracts the SMILES representation (Simplified Molecular Input Line Entry specification) [17], a format that actually represents the full chemical structure of the molecule and can be used to reconstruct the graphical form of the molecule.

3.1 Molecule Formula Generation

Extracting the molecule formula amounts to an enumeration of the single atoms occurring in the graph structure. In addition we might need to add hydrogen atoms for single unsaturated Carbon atoms in the molecule. The algorithm proceeds as follows:

Algorithm 2 (Molecule Formula Generation)
INPUT:

Molecule graph $G = (V, E)$

OUTPUT:

The molecule formula

METHOD:

1. *Initialise empty counter for the molecule formula*
2. *For each $v \in V$ do*
   (a) *If v has label C then*
      i. *Add one C to the molecule formula*
      ii. *Compute the valence w of v by adding for each $e \in E$ attached to v*
         - *if e has label $l = 1, 2,$ or 3, then $w := w + l$*
         - *if e has label $l = 4,$ or 5, then $w := w + 1$*
      iii. *Add $4 - w$ H atoms to the molecule formula*
   (b) *else add the label of v to the molecule formula.*

For example, in the molecule of Fig. 1, add the vertices NCO and C_2H_5 directly to the formula. For the 6 vertices we add one additional Hydrogen atom H for each of the four vertices that have valence 3 and none for the two that have valence 4. This results in the final molecule formula of C_9H_9NO.

3.2 SMILES Output

SMILES notation is a string representation of atoms, bonds and their connectivity. In SMILES, all atoms are represented by their normal symbols except Hydrogen which can either be omitted or displayed. To represent different types of bonds, SMILES puts nothing between atoms connected via single bonds, a = between atoms connected via double bond and a # to represent a triple bond. Also, stereo type bonds (3D bonds) are represented by a @@. Additionally, SMILES uses numbers to indicate ring closures and parentheses to imply branches. SMILES notation is not necessarily unique. For example, `C1=CC=C([NCO])C([C2H5])=C1` and `[NCO]C1(=CC=CC=C1([C2H5]))` are both valid representations of our molecule in Fig. 1, where the 1 indicates the circular bond between two Carbon atoms and entries in square brackets are molecular substructures.

While there exists a complex algorithm to generate unique SMILES notation [18], which relies on a canonical reordering of the atoms in the graph, we have chosen to implement a simpler generation algorithm that produces correct, but not necessarily unique SMILES in all cases. The basic idea of the algorithm is to generate the output string by first choosing a convenient starting vertex, traverse the graph in a depth first fashion, following edges with the largest label first, adding edges and atoms to the notation until all edges have been traversed and all vertices visited.

The detailed algorithm works as follows:

Algorithm 3 (Generating SMILES Representation)
INPUT:
Molecule graph $G = (V, E)$
OUTPUT:
SMILES notation for molecule
METHOD:

1. *Initialise empty backtrack stack S*
2. *Initialise output expression O*
3. *Choose starting vertex $v \in V$, preferably vertex with only one bond*
4. *For current $v \in V$ mark v visited and add label of v to output O*
5. *If v has single edge e not marked traversed*
   (a) *Mark e traversed*
   (b) *If e leads to an already visited vertex v' then*
      i. *Make circular bond in O*
      ii. *If we are in an open substructure, close substructure and add closing round bracket to O*
      iii. *If S is empty then end*
      iv. *Else pop from S and go to step 5*
   (c) *Else add bond to O and go to step 4 with v' as new current vertex*
6. *Else if v has several edges not marked traversed*
   (a) *Choose edge e with greatest label*
   (b) *Open new substructure and add open round bracket to O*
   (c) *Add bond to O*
   (d) *Push remaining untraversed edges onto S*
   (e) *Goto step 5a*
7. *Else if v has no edges not marked traversed then goto step 5(b)ii.*

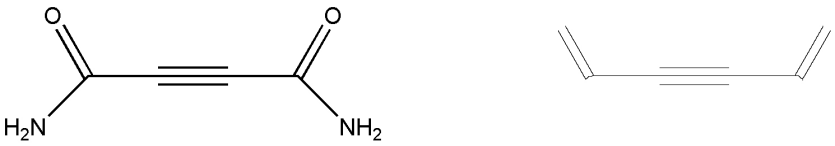

Figure 7: Problems with Hilditch's thinning algorithm

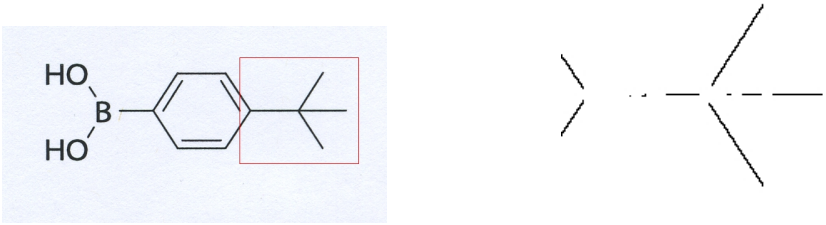

Figure 8: Disconnecting at undesired location after thinning.

4 Results and Discussion

We have tested our procedure using 60 different images created with the chemical software package ChemDraw and 50 scanned images from several different sources. The 60 images created with ChemDraw are essentially perfect images, containing no noise, and our procedure managed to recognise 59 of the images correctly. In once case the recognition failed due to the fact that Hilditch's thinning algorithm does not work properly on all patterns, this leads the system to end up with inaccurate results. As shown in Fig. 7, when diagonal lines are 2-pixels wide, the algorithm considers all pixels of this pattern to be boundary points and hence completely deletes them, leading to the dropping of two bonds.

Out of the 50 scanned images only 23 were recognised correctly. From the remaining 27 images the majority, 24, could not be recognised correctly due to failures during the end point extraction. These can be primarily attributed to insufficient noise removal which resulted in some lines shapes being jagged after the thinning algorithm and thus were falsely recognised as candidate junction and therefore lines were disconnected at undesired locations. Fig. 8 shows one of these cases for which the system outputs a wrong molecule name.

Another two images where wrongly recognised as Iodine atoms were classified as vertical single bonds. Fig. 10 illustrates this problem which can be attributed to some of the source material involving fonts that have capital I characters indistinguishable from vertical lines.

A final failure case, was again due to a diagonal line being completely erased after thinning. This case is shown in Fig. 9.

5 Conclusion

A number of chemical molecule structural diagram recognition systems have been built and discussed in the literature. We are only aware of one that is open source and free.

We have presented a prototype tool for the recognition of chemical structural molecular diagrams that we will release freely when it has reached a more mature level of robustness and functionality. Although our prototype is at an early stage of development, it already yields promising results. We are currently working to improve its robustness, especially in the area of noisy scanned images, and add further chemical diagram elements to its repertoire.

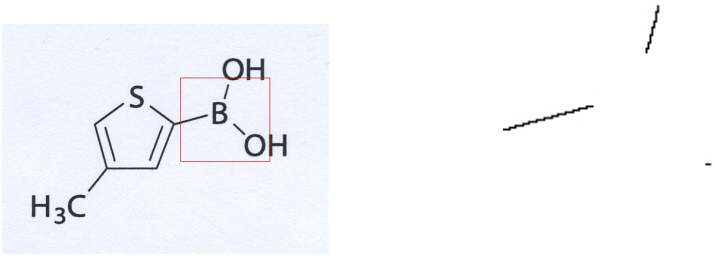

Figure 9: Hilditch's thinning algorithm erases a diagonal line.

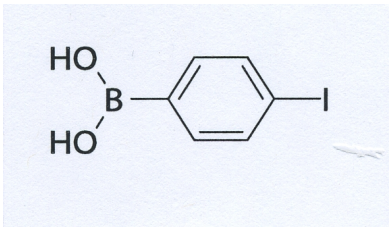

Figure 10: Iodine atom looks like and is classified as vertical single bond.

Our current implementation still has some shortcomings in handling scanned images. These issues arise mostly from our still inadequate vectorisation and noise removal implementations. Furthermore, the depiction of 3D bonds can vary quite significantly from image source to source and our handling thereof is not yet fully satisfactory.

References

[1] M. E. Algorri, M. Zimmermann, C. M. Friedrich, S. Akle, and M. Hofmann-Apitius. Reconstruction of chemical molecules from images. In *29th Annual International IEEE Conference on Engineering in Medicine and Biology Society, EMBS 2007*, pages 4609–4612, 2007.

[2] M. E. Algorri, M. Zimmermann, and M. Hofmann-Apitius. Automatic recognition of chemical images. In *Eighth Mexican International Conference on Current Trends in Computer Science, ENC 2007*, pages 41–46, 2007.

[3] S. Boyer. Optical recognition of chemical graphics. pages 627–631, 1993.

[4] A. Jain D. Trier and T. Taxt. Feature extraction methods for character recognition-a survey. *Pattern Recognition*, 29(4):641–662, April 1996.

[5] P. Ibison et al. Chemical literature data extraction: The clide project. *Journal of Chemical Information and Computer Sciences*, 33(3):338–344, 1993.

[6] Igor V. Filippov and Marc C. Nicklaus. Optical structure recognition software to recover chemical information: Osra, an open source solution. *Journal of Chemical Information and Modeling*, 49(3):740–743, 2009.

[7] J. R. McDaniel and J. R. Balmuth. Kekulé: Ocr-optical chemical (structure) recognition. *Journal of Chemical Information and Computer Sciences*, 32(4):373–378, 1992.

[8] Joe R. McDaniel and Jason R. Balmuth. Automatic interpretation of chemical structure diagrams. In *Selected Papers from the First International Workshop on Graphics Recognition, Methods and Applications*, pages 148–158, London, UK, 1996. Springer-Verlag.

[9] N. J. Naccache and R. Shinghal. An investigation into the skeletonization approach of Hilditch. *Pattern Recognition*, 17(3):279–284, 1986.

[10] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9:62–66, January 1979.

[11] Tom Y. Ouyang and Randall Davis. Recognition of hand drawn chemical diagrams. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 846–851. AAAI Press, 2007.

[12] Jungkap Park, Gus R. Rosania, Kerby A. Shedden, Mandee Nguyen, Naesung Lyu, and Kazuhiro Saitou. Automated extraction of chemical structure information from digital raster images. *Chemistry Central journal*, 3(1), February 2009.

[13] I. Pitas. *Digital Image Processing Algorithms*. Prentice Hall, 1993.

[14] J. Ramel, G. Boissier, and H. Emptoz. Automatic reading of handwritten chemical formulas from a structural representation of the image. *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, page 83, 1999.

[15] Joerg Schulenburg. GOCR: Open source character recognition, 2008. `http://jocr.sourceforge.net/`.

[16] Aniko T. Valko and A. Peter Johnson. CLiDE Pro: The latest generation of CLiDE, a tool for optical chemical structure recognition. *Journal of Chemical Information and Modeling*, 49(4):780–789, 2009.

[17] D. Weininger. Smiles, a chemical language and information system. *Journal of Chemical Information and Computer Sciences*, 1:31–36, 1988.

[18] D. Weininger, A. Weininger, and J. L. Weininger. Smiles .2. algorithm for generation of unique smiles notatio n. *Journal Of Chemical Information And Computer Sciences*, 29(2):97–101, 1989.