MolRec at CLEF 2012 — Overview and Analysis of Results

Noureddin M. Sadawi, Alan P. Sexton and Volker Sorge

School of Computer Science, University of Birmingham
Email: N.M.Sadawi|A.P.Sexton|V.Sorge@cs.bham.ac.uk
URL: www.cs.bham.ac.uk/~nms|aps|vxs

Abstract. We present the results and analysis of our chemical structure recognition system, MolRec, in the CLEF 2012 chemical structure recognition task. MolRec analyses a diagram image, extracts vectorised components from the image and applies a rule based system to construct an internal representation of the chemical structure. This internal representation can then be exported to MOL or SMILE format.
The task assigned in CLEF was to analyse two sets of chemical diagram images clipped from patent documents. The first set is of 965 diagram images, the results of which could be evaluated automatically using OpenBabel. The second set is a more challenging collection of 95 images which include elements not supported by OpenBabel and which therefore have to be evaluated manually. On the first set, MolRec achieved recognition rates of between 94.91% and 96.18% over 4 runs with slightly different parameters. On the more exacting second set, MolRec's recognition rate was between 46.32% and 58.95%. Overall the results testified to high performance on a large sample of quite complex diagrams but also to the challenges posed by the more difficult images that appear in real patent documents.

1 Introduction

We present the recognition results of our MolRec system on the CLEF 2012 corpus of chemical molecule diagrams. MolRec is a rule based system, in that after an initial preprocessing phase, the primary recognition task is performed by a rule engine, in which largely disjoint rules are repeatedly applied to an initial set of geometric primitives, thereby rewriting the set into a graph representation of the given molecule diagram. This final graph structure then serves as a basis from which other efficient electronic representation formats, such as MOL files, can be generated.

A previous implementation of the system has already performed well in the TREC 2011 competition [7]. For CLEF 2012 we have used an improved system with a fully overhauled implementation of the rewrite engine that not only leads to better recognition performance but is also computationally much more efficient.

In this note we will first give a short overview of the system, detailing its overall structure and briefly summarising the rewriting rules that perform the

primary recognition (Sec. 2). For a more detailed overview we refer the reader to [5]. We will then present the results of our system on the CLEF 2012 recognition task (Sec. 3) and follow it up by a more detailed analysis and discussion of images that were not successfully recognised (Sec. 4) but which motivate future improvements.

2 Overview of MolRec

MolRec employs a rule-based approach for the recognition of chemical structure diagrams. It consists of two modules, a vectoriser and a rule-engine. The vectoriser preprocesses an input image, analyses the chemical structure diagram it represents and generates a set of geometric primitives. These primitives are then picked up by the rule engine which rewrites them into a graph structure representation of the recognised molecule. In a post-processing step this graph structure can then be translated into a variety of output formats such as MOL files [6] or SMILES [1,8].

2.1 Vectorisation

The vectorisation works essentially in three steps:

1. Image binarisation
2. Optical character recognition (OCR)
3. Separation of bond elements

Binarisation For the first step of image binarisation we use Otsu's method [4]. This is followed by labelling of connected components.

OCR In a second step optical character recognition is performed by extracting a set of structural features from connected components and applying a nearest neighbour classification based on a Euclidean metric.

All connected components recognised as characters are removed from the image. Some contextual information is used to disambiguate difficult cases. For example, the lower case, sans serif letter "l" is often visually indistinguishable from short line segments in a molecule diagram, but in all examples we have come across, it does not appear except beside other letters (usually after a capital "C", to denote a Chlorine atom).

The result of this step is a skeleton molecule with all detected characters removed.

Separation of Bond Elements At this point we produce a new copy of the (character free) diagram and apply a thinning algorithm to connected components to thin them to a single pixel width. Using the thinned lines as a guide, we walk the corresponding paths in the original image to determine the average

line width by finding the largest disk that fits wholly with the stroke width of the line. At the same time, we build a polyline representation of the thinned lines. At every junction where three or more polylines meet, we split them into separate polylines. Closed polylines are also identified.

Because of scanning, discretisation and thinning artifacts, these polylines are not, as we would like, smooth idealised representations of the lines in the original diagram. Therefore we clean them up by applying the Douglas-Peucker line simplification algorithm [2], where we set the simplification threshold to between 1 and 2 average line widths as found above. This is sufficient to smooth out the polylines, removing almost all artifacts, without losing the significant corners in the lines in the diagram. Basing the threshold on the average line width allows the algorithm to adapt to the different line styles that appear in molecule diagrams in practice.

In addition to detecting and separating polylines we also detect circles as well as lines with arrows heads and solid triangles. The latter two are then annotated with their respective direction.

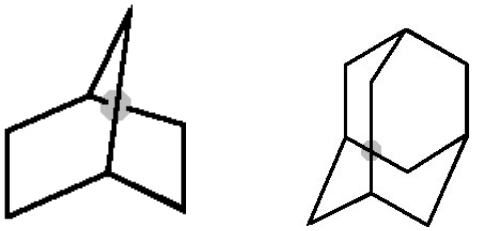
2.2 Rule Engine

The rule engine essentially works with the geometric primitives resulting from the vectorisation. In particular it uses character groups from the OCR step, as well as line segments, circles, solid triangles and arrows from the bond separation. The goal of the rule engine is to rewrite the input set of primitives into a graph structure that represents the molecule in terms of the atoms (or superatoms) and different types of bonds between them.

Rule are defined in terms of preconditions and consequences. A rule is applicable if there exist geometric objects that satisfies its preconditions. The consequence results in the removal of existing geometric objects and the addition of elements to the graph as well as possibly the addition of new geometric objects. In general, preconditions of different rules are mutually exclusive, and thus the order of rule application is irrelevant. Rules work with a number of parameters, both fuzzy and strict, that set certain thresholds, for instance the minimal bond length, under which decisions will be made. These parameters allow for the customisation of MolRec and its adaptation to particular requirements of datasets. In this section we will only briefly summarise the main rules and present an example. For more details we refer the reader to [5].

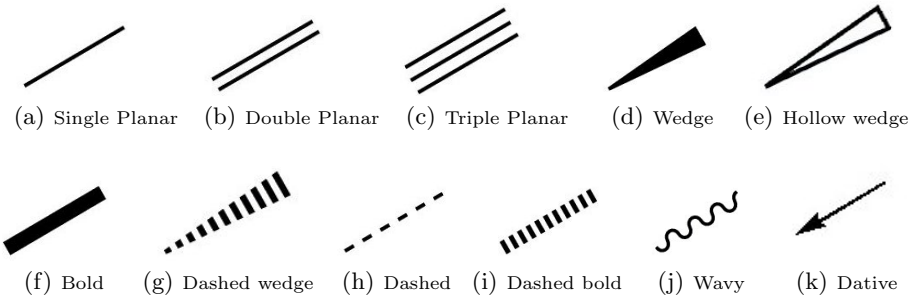
The rule engine consists of 18 rules altogether. Two of these rules have to be applied before all other rules. These two rules deal with the recognition of bridge bonds, 3-dimensional structures representing multiple different connection paths between different parts of the molecule. These are typically presented in a $2^1/_2$-dimensional perspective drawing form such as in Fig. 1.

The other 16 rules can be applied in arbitrary order. They deal with the recognition of a number of other bonds that MolRec can handle and that are presented in Fig. 2. All these bonds consist of one or several geometric objects, which a rule can select using its preconditions and rewrite into a corresponding graph entry for the recognised bond type.



(a) Open-bridge (b) Closed-bridge

Fig. 1. Closed and open bridge bonds (circled)



(a) Single Planar (b) Double Planar (c) Triple Planar (d) Wedge (e) Hollow wedge

(f) Bold (g) Dashed wedge (h) Dashed (i) Dashed bold (j) Wavy (k) Dative

Fig. 2. Bond types recognised by MolRec.

However, there are also single geometric objects that possibly represent more than one bond. An example are so called implicit nodes presented in Fig. 3. Here carbon atoms are understood to be at the grey circled areas separating the bonds. These cases are dealt with by rules that pick double or triple bonds, respectively, while also producing new geometric objects by effectively cutting the bonds at the implicit nodes. These new objects can then be further processed by other rules.

Not all decisions on bond types can be made by inspecting locally a number of geometric objects only. Consequently some rules mark some of their results as possibly ambiguous. These ambiguities have to be resolved taking context information into account. This is done after all geometric objects have been rewritten within the context of the resulting graph. Furthermore, at this stage MolRec also adds the character groups to the graph, which can be used as further aid for disambiguation. For example, disambiguation of lower case “l”, capital case “I”, the digit “1” and a vertical single bond is carried out at this stage. In addition character groups identifying more than one atom are identified as superatoms. Their structure is looked up in a dictionary and the character group is replaced with the molecule subgraph corresponding to that superatom.

Fig. 3. Implicit Nodes (circled)

An Example Rule - Wavy Bond Wavy bonds (Figure 2(j)) are commonly used in chemical structure diagrams. As the name suggests, they have a *wavy* form although a less commonly used *saw-tooth* form can be encountered in the literature.

A vectorisation process will most likely turn a wavy bond into a connected sequence of short line segments arranged in a saw-tooth, or a zig-zag, pattern. As illustrated in Figure 4, a straight line can pass through the centre points of these line segments. A pattern of this form can be identified using the following conditions.

1. $L = \{l_1, \ldots, l_n\}$, where $n \geq 3$, is a set of line segments,
2. $\forall l \in L : \mathit{length}(l) \in dl$, where dl is the *dash length*: a parameter of the system representing a range of acceptable values for the length of an individual dash in a dashed line.
3. All elements of L are connected.
4. The centre points of the elements of L are approximately collinear.[1]
5. Two elements of L, called the end elements, dash-neighbour[2] precisely one other element of L. All other elements of L, called internal elements, dash-neighbour precisely two other elements of L.
6. Two end points that are not connected must be the pair of end points that are furthest apart.

Consequence A wavy bond between the furthest two endpoints. The new wavy bond has unknown direction.

To understand this rule, note that condition 1 simply selects possible line segments, such as those from Figure 4, for consideration. Condition 2 ensures that each segment is of an appropriate length. That the endpoints of the line segments are connected is guaranteed by condition 3. The *approximate collinearity* of their centre points in condition 4 ensures that this sequence of line segments, although having a zig-zag form at the micro-structure level, is straight at a macro-structure level. Finally, the last two conditions ensures that the segments form a single sequence as would be obtained from a wavy bond, and not, for example, from any

[1] Approximate collinearity is a precisely defined relationship under which a set of points can be considered to be collinear within the constraints of the limitations of the construction, printing and scanning technologies used for the image.

[2] Dash neighbouring is a precisely defined relationship which specify the conditions under which two line segments can be considered to be consecutive dashes in a dashed line.

Fig. 4. Dashes in a Wavy Bond

kind of star structure with multiple end points. The consequence merely identifies the structure as a wavy bond between the end points and leaves undecided the directionality of the bond.

3 Analysis of MolRec's Performance

We were given a set of 961 test images by CLEF12's organisers. This collection was split into two sets. The first set was of 865 images selected for automatic evaluation by comparison of generated MOL files with the ground truth MOL files using the OpenBabel toolkit[3]. However, there are chemical diagrams whose valid MOL files are beyond OpenBabel's ability to compare (typically because they contain some form of Markush structure) and a second set of 95 such diagrams was selected for manual, visual evaluation. This second set was intentionally included to provide a greater challenge to the participating diagram recognition systems.

We ran MolRec four times on these sets where we slightly adjusted its internal parameters and MolRec achieved the results illustrated in Table 1 and Table 2. The two tables show the number of correct and incorrect recognitions for the four runs on the manual and automatic evaluation sets respectively. Notice that because most of the diagrams mis-recognised in some runs were also mis-recognised in other runs, there were a total of 52 different diagrams mis-recognised in the manual evaluation set and a total of 46 different diagrams mis-recognised in the automatic evaluation set. Some of these diagrams failed for multiple reasons, so we were able to identify the reasons illustrated in Table 3.

Run	# Recognitions	# Mis-Recognitions	Accuracy
1	44	51	46.32%
2	56	39	58.95%
3	44	51	46.32%
4	54	41	56.84%

Table 1. Four Runs on the Manual Evaluation Set (95 images)

Run	# Recognitions	# Mis-Recognitions	Accuracy
1	832	33	96.18%
2	821	44	94.91%
3	821	44	94.91%
4	832	33	96.18%

Table 2. Four Runs on the Automatic Evaluation Set (865 images)

Reason	Manual Set #Images	Automatic Set #Images
Character Grouping	26	0
Touching Characters	8	1
Vectorisation of Four-way Junction	6	7
Missed Solid Wedge	0	6
Missed Dashed Wedge	0	6
OCR Errors	5	11
Missed wavy bond	2	1
Missed charge sign	1	2
Incorrect Stereocentre	0	1
Atom too close to line endpoint	0	1
Line end too close to closed node	0	1

Table 3. Reasons for Mis-Recognition of Molecules.

4 Evaluation and Analysis of Results

We now briefly discuss some of the problems that have lead to MolRec mis-recognising molecule diagrams in the test set.

4.1 Character Grouping

An error in the implementation of our character group formation algorithm lead to the digit "1" being repeated when it appears within a atom group, so, for example, MolRec recognised R_{21} incorrectly as R_{211}. A separate problem was the difficulty in correctly separating different atom groups which are closely spaced, as shown in Figure 5.

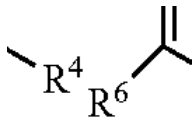


Fig. 5. Atoms too Close

4.2 Touching Characters

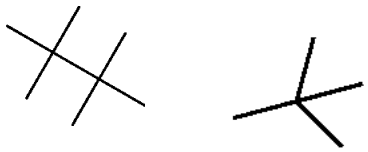
We currently do not handle touching characters and therefore they will likely cause mis-recognition. Figure 6 shows several examples.

(a) (b) (c) (d)

Fig. 6. Example of Touching Characters

4.3 Vectorisation of Four-way Junctions

Vectorising junctions where four lines meet was another reason for recognition failure. Two examples are given in Figure 7. MolRec misses such junctions.



(a) (b)

Fig. 7. Example of Four-way Junctions Missed by MolRec

4.4 OCR Errors

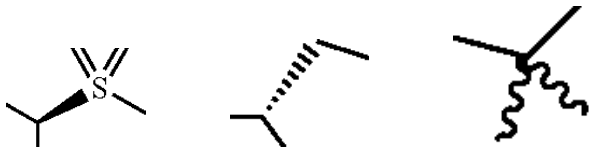
There were some cases of OCR errors. These included a "G" interpreted as an "O", and the "alkyl" atom (Figure 8(a)) being mis-recognised. Also, as illustrated in Figure 8(b), there was several cases where an "I" was interpreted as a vertical single bond.

4.5 Missed Solid Wedge, Dashed Wedge and Wavy Bonds

As shown in Figure 9, MolRec incorrectly recognised a number of solid wedge, dashed wedge and wavy bonds.

(a) (b)

Fig. 8. OCR Errors

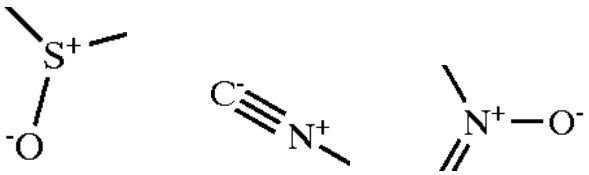


(a) (b) (c)

Fig. 9. Examples of Missed Solid Wedge, Dashed Wedge and Wavy Bonds

4.6 Missed Charge sign

The plus and minus signs, or "+" and "-", are often used to indicate the existence of positive and negative charges respectively. As shown in Figure 10, they are usually placed to the top right of an atom. While correctly recognising the positive charge sign, MolRec missed three negative charge signs including one that was placed at the top left of an atom name.

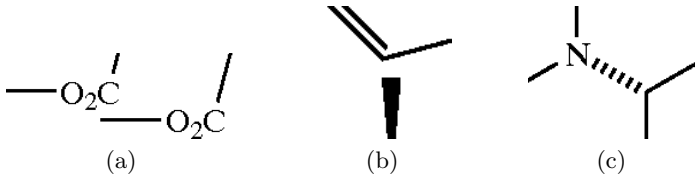


(a) (b) (c)

Fig. 10. Missed Charge Signs

4.7 Other Mis-recognition Reasons

These include an atom that was too close to a bond's endpoint and which was therefore erroneously considered connected (Figure 11(a)), a solid wedge bond that was too close to a closed node so they were considered connected (Figure 11(b)) and a dashed bold bond whose stereocentre was incorrectly determined (Figure 11(c)).



(a) (b) (c)

Fig. 11. Other Mis-recognition Reasons

4.8 Errors in the Dataset

While analysing our results we came across a small number of cases where the ground truth was incorrect and our recognition was correct. As shown in Table 4, there were 11 images in total. Such ground truth dataset errors are very difficult to avoid in such a complex task.

US20040254236A1_p0037_x0488_y1434_c00146
US20060122222A1_p0015_x0638_y2146_c00043
US20060122222A1_p0015_x0638_y2624_c00044
US20060122222A1_p0025_x0363_y1056_c00070
US20060122222A1_p0027_x0379_y1064_c00078
US20060122222A1_p0040_x1313_y0721_c00105
US20060154945A1_p0017_x0404_y2072_c00059
US20070155803A1_p0020_x1323_y2114_c00037
US20070155803A1_p0029_x1358_y1732_c00073
US20070179154A1_p0046_x1376_y0890_c00066
US20070270434A1_p0015_x1376_y2386_c00029

Table 4. Images with Incorrect MOL files

5 Conclusions

Despite scoring high recognition rates throughout four runs, the presented experiments demonstrate that there is still plenty of room to improve MolRec. We believe many of the mis-recognition problems can be solved with some relatively simple enhancements of our system, e.g. the error in character grouping or the vectorisation of four-way junctions. Tackling the notoriously difficult touching character segmentation problem is one aspect where we plan to explore viable solutions. Another area we plan to investigate is the recognition of more general Markush structures. Additionally, robust charge sign spotting, accurate identification of solid wedge bonds and precise identification of dashed wedge bonds are also areas we need to address further. However, we are pleased with the performance of MolRec and hope to participate in similar events in the future so

that we can contribute to progress in the state of the art of chemical structure diagrams.

References

1. Daylight Chemical Information Systems, Inc. SMILES — a simplified chemical language, 2008. http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html.
2. David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica*, 10(2):112–122, 1973.
3. Open Babel: The open source chemistry toolbox. http://openbabel.org/.
4. N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9:62–66, January 1979.
5. Noureddin M. Sadawi, Alan P. Sexton, and Volker Sorge. Chemical structure recognition: A rule based approach. In Christian Viard-Gaudin and Richard Zanibbi, editors, *19th Document Recognition and Retrieval Conference (DRR 2012)*. SPIE, January 2012.
6. Symyx. CTfile formats, 2010. http://www.symyx.com/downloads/public/ctfile/ctfile.jsp.
7. Text REtrieval Conference (TREC). Image2structure task, 2011. http://trec.nist.gov/pubs/trec20/t20.proceedings.html.
8. D. Weininger. SMILES, a chemical language and information system. *J Chem Inform Comput Sci*, 1:31–36, 1988.