JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

Markov Logic Networks for Optical Chemical Structure Recognition

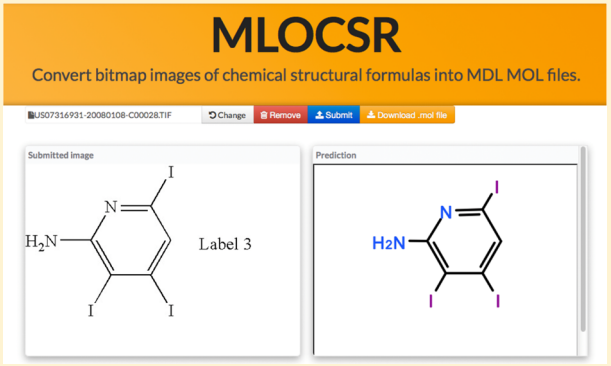
Paolo Frasconi,*,† Francesco Gabbrielli,*,† Marco Lippi,*,‡ and Simone Marinai*,†

†Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Firenze, Via di Santa Marta, 3, 50139 Firenze, Italy

‡Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università degli Studi di Siena, Via Banchi di Sotto, 55, 53100 Siena, Italy

Supporting Information

ABSTRACT: Optical chemical structure recognition is the problem of converting a bitmap image containing a chemical structure formula into a standard structured representation of the molecule. We introduce a novel approach to this problem based on the pipelined integration of pattern recognition techniques with probabilistic knowledge representation and reasoning. Basic entities and relations (such as textual elements, points, lines, etc.) are first extracted by a low-level processing module. A probabilistic reasoning engine based on Markov logic, embodying chemical and graphical knowledge, is subsequently used to refine these pieces of information. An annotated connection table of atoms and bonds is finally assembled and converted into a standard chemical exchange format. We report a successful evaluation on two large image data sets, showing that the method compares favorably with the current state-of-the-art, especially on degraded low-resolution images. The system is available as a web server at http://mlocsr.dinfo.unifi.it.



INTRODUCTION

In spite of the availability of a wide array of data exchange formats, the vast majority of patent files and scientific papers in chemistry and related disciplines (pharmaceutics, medicine, biology, etc.) are still communicating information about small molecules via structural diagrams, which are drawn with specialized software and later embedded into electronic documents in the form of bitmap images. These reports are typically available online, but the information about compounds is not machine-readable and lacks a structured representation to enable effective indexing and searching.[1,2]

The problem of extracting a structured representation from bitmap images of chemical formulas emerged in the early 1990s when a number of software systems were developed by various research groups.[3−6] CLiDE[6] developed into a commercial product, while the IBM system described in ref 5 was granted a US patent.[7] More recent systems include refs 8−11. In particular, OSRA[10] is an open-source project. A new version of CLiDE has been recently developed.[12] AsteriX and OSRA can also automatically analyze entire scientific articles in pdf format and detect pictures representing chemical diagrams.

Although these systems have several distinct characteristics in their implementation details, they follow a common design strategy where some modules are designed to extract low-level information from images and one or more higher level modules are in charge of the interpretation of the extracted low-level elements and their assembly into an annotated connection table.

As for the low-level information extraction process, the mentioned systems mainly differ in the algorithms which are employed for the identification of connected components, lines, characters, and other distinctive traits of chemical diagrams such as solid/dashed wedges. OSRA[10] performs a binarization of the image and then applies a vectorization algorithm, by employing the Potrace library [http://potrace.sourceforge.net/] and several heuristic rules in order to retrieve nodes, bonds, circles, and solid/dashed wedges, based on geometrical properties empirically estimated. A peculiarity of OSRA is that the test image is processed at three different scales, and finally an empirical confidence estimation function is employed in order to determine the best output among the three candidates. CLiDE[6,12] extracts approximation polygons from the connected components founds in the vectorized image and then identifies atoms and bonds starting from the end-points of such polygon, treating as special cases more complex structures such as dashed bonds. Even in CLiDE, the reconstruction of the molecule is based on empirical hard-coded geometrical rules. ChemReader[9] first extracts connected components, which are classified as either text or graphics: text is interpreted by an optical character recognition (OCR) tool, while graphical components are processed by a group of operators, such as generalized Hough transform and corner detection. A simple graph reconstruction algorithm finally assembles the found atoms and edges into a molecule.

Received: April 8, 2014

   dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

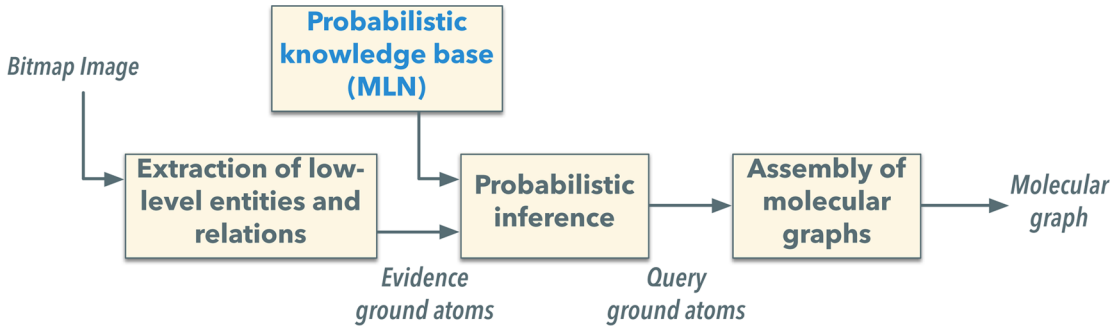


Figure 1. Scheme of the formula recognition system.

Chemical background knowledge is mostly employed in the higher level modules of these systems and, in particular, during the graph reconstruction process. For example chemical knowledge is used to fix valency errors (as in Kekulé, OSRA, and ChemReader), to interpret ambiguous situations regarding crossing bonds, or to handle OCR mistakes (as in CLiDE and OSRA).

One difficulty in the low-level processing phase is the recognition of text characters. This is because most OCR packages are designed to take advantage of dictionary lookup and improve their accuracy by excluding clearly malformed strings. In chemical diagrams, however, strings are often short and subscripts are very frequent. To address this difficulty, specialized OCR based on artificial neural network were specifically designed in Kekulé and CLiDE. Another difficulty concerns image resolution. Several parameters of the low-level pipeline need to be adjusted to take into account the expected size of the graphical primitives and the expected amount of corruption due to spatial sampling. Moreover, in the case of images found in scientific papers or patents, the resolution expressed in dpi (dots per inch, which may be available in the image file metadata) does not necessarily match the actual amount of aliasing, often because molecules can be drawn at different relative scales when pasted together with other graphical diagrams within the same image.

In this paper, we introduce MLOCSR (for Markov logic optical chemical structure recognizer), a method which also follows a pipelined design strategy based on low-level and high-level processing phases but is distinguished by several novel algorithmic ideas. The low-level module is designed to be resolution independent thus removing the need for an explicit definition of the image resolution. This is achieved by estimating the character size and the thickness of bonds and then linking the parameters used for tuning the low-level algorithms to these values. The character size is estimated with a tight interplay between the OCR engine and several image processing algorithms, designed to take into account specific features of chemical structural diagrams.

The higher-level module is based on a Markov logic network[13,14] (MLN), used as a probabilistic first-order logic inference engine. Chemical and graphical knowledge is represented in our system as a collection of *weighted* first-order logic formulas. Each chemical structure formula is associated with one logical world. The output from the low-level pipeline (for a given molecular diagram) is directly mapped into a set of logical constants (i.e., object identifiers for graphical elements such as points, lines, character boxes, etc.) and a set of ground facts describing relations which hold true on the low-level objects. Chemical primitives (atoms, bonds, and their attributes) are finally obtained as the result of probabilistic inference, namely by computing the most probable world given the ground atoms extracted at the lower-level. Performance evaluation on several public domain data sets show that our system achieves state-of-the-art recognition accuracy, both at the whole molecule level and at the individual constituents (i.e., atoms and bonds) level.

METHODS

System Overview. Our approach for reconstructing the structure of a chemical compound is sketched in Figure 1 and consists of three main modules: a low-level extractor of graphical primitives (represented as logical entities and relations), a probabilistic logical reasoning engine based on Markov logic, and a final stage for assembling the output molecular graph.

The low-level subsystem takes as input a bitmap image and extracts graphical primitives represented as a set of logical ground (i.e., variable free) atoms. Objects of interest are three distinct types of graphical points:

C-points: detected as intersections between lines; they typically represent carbon atoms.

D-points: detected as end points of lines in double and triple bonds.

T-points: detected as end points of lines terminating in a text box.

Examples of these three point types can be seen in Figure 2b. Several binary and ternary relations are also extracted. Binary relations are mostly associated with lines and include:

Carbon to Carbon Lines. LineBetweenCpoints(c1, c2) is true if c1 and c2 are C-points and a line connecting them was detected (for example the lines highlighted in red in Figure 2c).

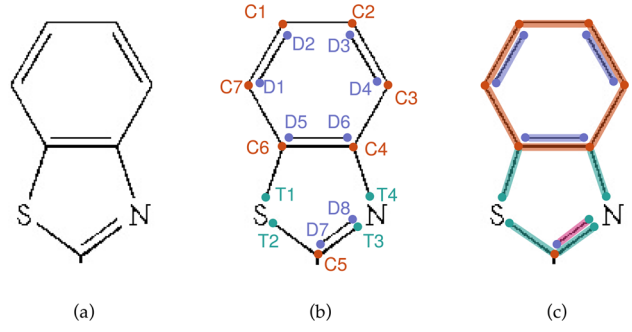


Figure 2. (a) Fragment of a bitmap image after binarization and smoothing. (b) Extracted points. C1, ..., C7 are C-points, D1, ..., D8 are D-points, and T1, ..., T4 are T-points. (c) Extracted relations.

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

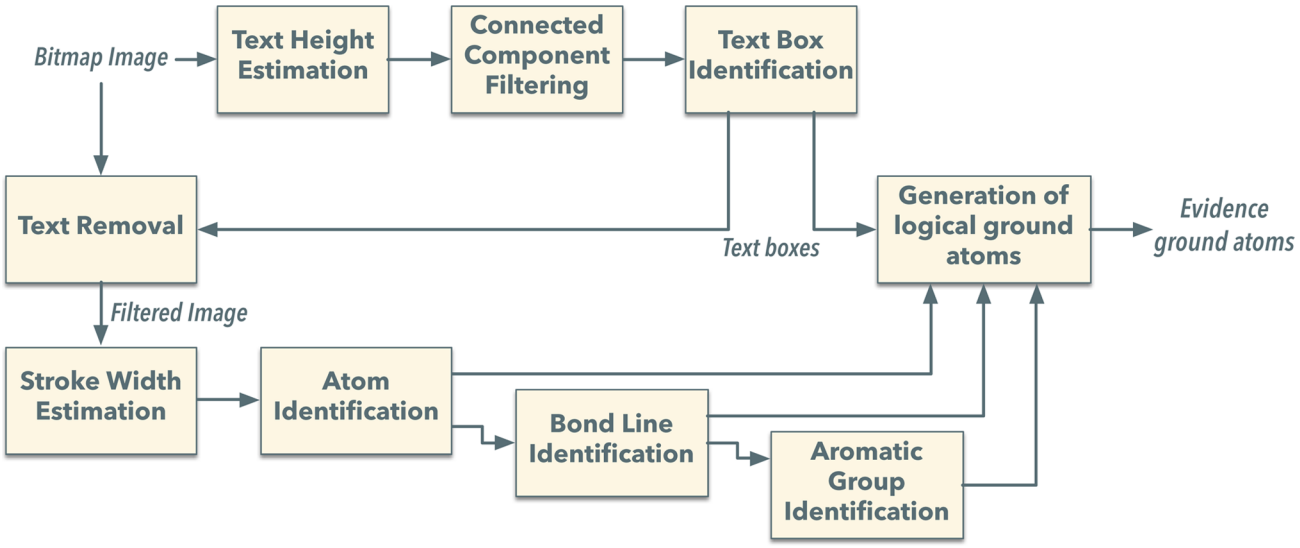


Figure 3. Scheme of low level processing.

Carbon to Text Box Lines. LineBetweenCpointAndTpoint-(c, t) is true if c is a C-point, t a T-point, and a line connecting them was detected (for example the lines highlighted in cyan in Figure 2c).

Double Lines. LineBetweDpoints(d1, d1) is true if d1 and d2 are D-points and a line connecting them was detected (for example the lines highlighted in violet in Figure 2c).

Double to Text Box Lines. LineBetweenDpointAndTpoint-(d, t) is true if d is a D-point, t a T-point, and a line connecting them was detected (for example the line highlighted in magenta in Figure 2c).

Ternary relations are used to represent collinearity of points, for example CollinearCCC(c1, c2, c3) is true if the three C-points c1, c2, and c3 are detected to be collinear. Twenty-six evidence predicates are defined in our system (a complete list is available in the Markov logic network file, see the Supporting Information).

The information extracted at this level is however noisy. Spurious end-points or lines may be detected for example when the formula structure contains wedge or weaved lines, or when textual annotations (which are not part of the formula) are present and too close to the formula itself. In order to cleanup information while taking advantage of geometric and chemical knowledge, we designed a probabilistic knowledge base using a Markov logic network (MLN). The probabilistic reasoning procedure takes the knowledge base and the information extracted at the low level for a particular molecule, and produces the most likely explanation as groundings of new logical predicates. These groundings are then fed into the final stage which is in charge of assembling the output molecular structure in the form of an annotated graph which can finally be saved as an MDL Molfile.

Extraction of Low Level Entities and Relations. Gray level images are binarized with global thresholding, that is appropriate when dealing with computer-generated structural diagrams[15] such as those appearing in the chemical literature. Contours are smoothed by a morphological closing operator performed with suitable structuring elements. Connected components (CCs) are then found on the binarized image. Each connected component c is described by its bounding box, having width $w(c)$ and height $h(c)$, and by the number of black pixels in the component $(n(c))$.

At this point, the image vectorization module identifies and localizes two kinds of low-level entities: textual symbols (such as chemical element symbols or superatoms [Superatoms are strings representing chemical formulas, such as SO_2, COOH, or even simply elements, such as N.]) and graphical items (such as lines and circles). Here we take advantage of a tight interplay between an OCR engine and several image processing algorithms which have been especially designed to take into account some specific features of chemical structural diagrams.

The main steps are summarized in Figure 3. The Text Processing section describes the upper part of Figure 3 providing details on text height estimation, connected components filtering, text box identification, and removal. The lower part of Figure 3 is described in the Image Vectorization section that describes the stroke width estimation and the identification of atoms, bond lines, and aromatic groups.

Text Processing. Most graphics recognition methods identify textual objects by looking for CCs whose dimensions fall within a given interval. Acceptable components are identified by taking into account thresholds that are either fixed a-priori or computed from statistics of components in the input image.[16] In the case of chemical drawings, the number of characters in each image is however too limited to produce reliable frequency estimates. Therefore, it seems appropriate to rely also on the recognition of CCs by means of an OCR engine [We use the open source tesseract: https://code.google.com/p/tesseract-ocr/.] to identify the text in the image.

Text Height Estimation. The set O of connected components whose width-to-height ratio is compatible with potential characters $(\alpha_1 < w(c)/h(c) < \alpha_2)$ is submitted to the OCR engine for the purpose of estimating the text height T.

1. If nitrogen (N) or hydrogen (H) atoms are recognized, the average height of the corresponding CCs is computed and retained as first estimate of T. These two atom names occur frequently in structural formulas and are hardly confused with other graphical items. Moreover, they never appear as subscripts or superscripts.
2. If neither N nor H atoms are found, then T is computed on a subset of the CCs analyzed by the OCR engine. In

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

particular, we ignore CCs that are unlikely to correspond to characters because they are too thin ($n(c)/a(c) < \beta_1$), too dark ($n(c)/a(c) > \beta_2$), too small ($a(c) < \gamma$ or $h(c) < \delta$), or very large ($h(c) > H/3$ or $w(c) > W/3$). Here $a(c) = w(c)\cdot h(c)$ is the area of c and H, W are the height and width of the input image, respectively. The absolute thresholds used to filter the CCs are not too critical since the purpose here is to identify a few CCs that most likely correspond to characters and can be used to compute T without requiring to find all the characters in the input image.

Subsequently, the current T value is refined taking into account the CCs recognized as oxygen (O) with dimensions compatible with the current value of T. This check is required to exclude CCs corresponding to aromatic rings from the T computation, without missing actual oxygen atoms. If no textual characters are recognized, T is not computed and the text processing step is terminated.

Filtering Connected Components. In this step, we refine the set of CCs to be submitted to the OCR engine, in this case for the purpose of character recognition. First, we remove from O a component c if

$$\frac{|T - h(c)|}{\max\{T, h(c)\}} > \varepsilon \quad (1)$$

Here the main assumption is that all characters (except subscripts and superscripts, which are not removed by eq 1) have the same font size, so that the estimated T can be used for all CCs.

Second, we add to O components that likely correspond to thin characters such as uppercase I, lowercase l, and 1. As addressed also by other systems (e.g.[16,17]) disambiguation of these thin characters with respect to dashes or single bonds is essential. To this purpose, for every CC c previously discarded because of a low width-to-height ratio, let $S(c)$ denote the square region with side $h(c)$ centered on c (see red squares in Figure 4). We consider the following three cases:

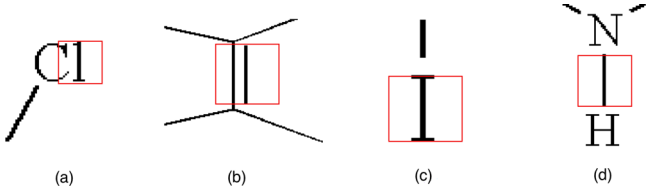


Figure 4. Disambiguation of short bonds and thin characters.

- If $S(c)$ and a CC $c' \in O$ overlap, then c is added to O. The rationale is that a CC close to a text box is likely to contain a character, as shown in Figure 4a.
- Else, if $S(c)$ overlaps only with a CC $c' \notin O$, then we count the number of pixels $\overline{n}(c')$ that belong to c' and fall within $S(c)$. If $\overline{n}(c') < 0.5n(c)$, then c is added to O. For example the right vertical line in Figure 4b would not be considered as a text candidate.
- The last case is when $S(c)$ does not overlap with any other CC. This is handled as a special case: c is added to O but is subsequently accepted as a character only if recognized as I (corresponding to a Iodine atom), as in Figure 4c. In the example shown in Figure 4d, the line inside the red square would not be accepted as a character (unless the OCR engine misrecognizes the line as an I).

At this point the OCR engine is invoked separately on each element of O. Results are interpreted in the following step.

Text Box Identification and Removal. Identification of character strings is a classical problem in document image analysis and has been addressed by several authors. Chemical drawings, however, are rather special in this respect because text strings are typically very short. For example, methods based on the Hough transform of connected components,[16,18] which assume relatively long sequences of characters of the same height, are not applicable. Our approach is more closely related to methods that iteratively merge together characters and update the thresholds when building text strings (see, e.g., ref 19). In particular, iterative grouping of characters into strings that allows us to easily identify subscripts and superscripts. As a major difference with respect to Su and Cai,[19] grouping in our case relies on geometric rules whose thresholds are adapted as a function of the T value, allowing us to achieve resolution independence. The same grouping strategy is followed for both horizontally and vertically aligned text. Following a common approach in graphics recognition systems (see, e.g., ref 20), we finally remove all the CCs in O recognized as text and feed the filtered image to the subsequent graphical modules that extract graphical entities.

Image Vectorization. The vectorization step follows the identification and removal of text components and aims at extracting the low-level graphical entities, such as straight segments, from the bit-map image. The most important substep is the identification of lines (that in chemical diagrams correspond to bonds) from the image. According to ref 20 three main approaches can be considered for finding the lines from the image:

1. Methods based on parametric model fitting use a line model to detect the lines. The most used technique is the Hough transform that is a global transformation of the image that allows to find straight lines also in the presence of noise (such as broken lines). Even if the Hough transform has been used to vectorize chemical diagrams[9] it does not guarantee that close segments with different slope would not be mixed together[20] and can provide inaccurate locations of extrema points.
2. The most widely used approach in the vectorization of line segments is based on the extraction of the line skeleton and its subsequent processing. Despite their wide adoption, skeleton-based methods can generate wrong junction points, in particular when dealing with noisy images or low resolution ones where segments can have a thickness of a few pixels.[20]
3. The last approach is based on the extraction of the contours of the graphical objects in the image and the subsequent matching of opposite contours.[20,21] These methods position the junction points more accurately also in the presence of low resolution images, but are sometimes too much dependent on thresholds and parameters that should be hand-tuned.

The technique described in this paper adopts a contour-based processing where the parameters depend on the estimated stroke width (S), similarly to the text processing that is based on the estimated text height T.

Stroke Width Estimation. S is estimated by first identifying the strokes of the bonds with three steps.

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

- Edges of graphical objects are extracted with the Canny algorithm and potential segments are identified by applying the Hough transform on the edge image.
- Potential segments are verified by counting the black pixels in the segment. Segments with less than 75% black pixels are discarded as false positives (Figure 1 left).
- Each segment is inspected at regularly spaced positions and the number of contiguous black pixels in lines orthogonal to the segment is used to compute S (Figure 5 right).

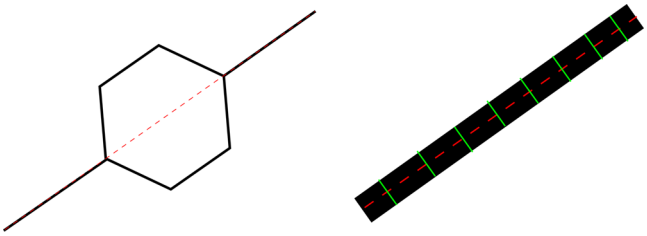


Figure 5. (left) Checking potential segments. (right) Computing S.

It is important to remark that the search made with the Hough transform can be inaccurate. However, false positive segments are identified and discarded in the second step while false negative ones are not critical since in this step we do not aim at identifying all the segments. On the other hand the inaccurate identification of the slope and extrema of the segments has little influence on the S computation performed in the third step.

Atom Identification. In this step, both C-points and simple D-points are identified. Each contour, with at least $2{\cdot}S$ pixels, is approximated with a polygon using the Douglas-Peucker algorithm.[22] This algorithm iteratively adds new polygonalization points until all the contour points have a distance to the approximating polygon lower than a predefined *precision* value. In the proposed system the precision is fixed considering S according to the following equation:

$$\text{precision} = \sqrt{2}\max(2, S) \tag{2}$$

In structural diagrams the contours can correspond to simple bonds (straight segments) or to more complex items that comprise more bonds.

Simple bonds are identified by analyzing the polygonalization and checking all the triangles that can be defined considering the polygonalization points. The contour corresponds to one single bond if there is no triangle with all the sides longer than S. The farthest approximation points in these contours are identified and labeled as end points that could be C-points or B-points.

Contours not recognized as simple bonds are analyzed to look for C-points that implicitly represent Carbon atoms. For each polygonalization point three cases can occur:

- The point can be merged with one C-point. In this case the C-point position is updated as the average of the two points.
- If there is a text box closer than $\eta{\cdot}T$ to the point, the text is added to the list of superatoms and the point is labeled as T-point.
- Otherwise the point is added to the list of C-point.

In some cases multiple instances of the same atom are found in a given image. Disambiguation will be later performed by the Markov logic network.

Bond Line Identification. Each pair of end points belonging to the same contour that are farther than $\sigma{\cdot}S$ pixels is checked to verify whether there is one segment linking the two points. To this purpose the density of black pixels in the rectangle that connects the points and with width equal to S is computed. If this density is higher than a threshold, the two points are connected with a bond line. At the end of this step collinear touching segments are merged together.

To represent the three-dimensional arrangement of atoms three special bonds are used, in addition to solid lines that describe planar bonds.

Wedges. represent bonds that point out of the planar compound toward the observer or in the opposite direction. [What we call here wedges are sometimes in the literature also indicated as black wedges. Conversely, what we call hash bonds are sometimes named dashed-wedged or simply dashed. We shorten the nomenclature in order to improve the readability of the paper.] One wedge can be found if there are three C-points belonging to the same contour that are linked together and such that the area comprised in the triangle defined by these points is mostly black. the cases with two C-points and a single T-point, or two T-points and a single C-point are considered as well.

Dashed Lines. represent bonds that point in the opposite direction with respect to wedges. Dashed lines can be found by looking for small connected components having only two end points. These segments are then grouped together on the basis of their distance.

Wavy Lines . are another type of three-dimensional bond and are identified with a suitable analysis of polygonalization points of a given contour. The details are omitted in this paper, but the technique is similar to the approach described in ref 17.

Aromatic Group Identification. Aromatic groups are identified by looking for circles inside polygons. The circles are identified by looking for connected components with a square bounding box where all the carbon points have roughly the same distance from the square center. A new object that corresponds to the center of the aromatic ring is then created within the Markov logic domain.

Parameters and Thresholds Estimation. The parameters and thresholds which are used by the low-level modules are listed in Table 1, together with the value chosen within our system. A cross-validation estimation of such a large number of parameters is practically unfeasible, and therefore we relied on tuning their values using a small subset of chemical images from different data sets.

Table 1. Parameters and Thresholds Employed by the Low-Level Modules

name	value	description
α_1, α_2	0.4, 1.75	lower/upper bound for character width-to-height ratio
β_1, β_2	0.1, 0.8	lower/upper bound for character pixels-to-area ratio
γ	17	minimum area for textual connected component
δ	8	minimum height for textual connected component
ε	0.33	relative height of a component for OCR submission
η	0.75	threshold for connecting text boxes to points
σ	6	multiplicative coefficient for S for identifying bonds

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Using Markov Logic to Refine Entities and Relations. In order to reconstruct the molecular graphs, the graphical entities extracted in the image processing stages (in particular lines, points and text boxes) need to be mapped into the chemical entities of interest (in particular atoms and bonds). It is rather natural to describe this mapping by means of a set of logical rules and constraints. However, the inherent presence of noise makes it difficult to use classical logical inference techniques: if noises creates contradictions, inferred consequences would be meaningless. For this reason we employ Markov logic the process of reconstruction of the molecular graph necessitates to disambiguate atoms and bonds. The information collected by the image processing stages can be easily coded into a set of logic predicates, while atoms and bonds have to follow rules and constraints, which can be described in a natural way through probabilistic logic formulas: Markov logic therefore represents the ideal framework for this task.

A Markov logic network (MLN; see refs 13 and 14 for details) defines a probability distribution over logical worlds [A logical *world*, is a mapping from symbols to objects, functions, and relations. It assigns a truth value to every ground (i.e., variable free) atom that can be constructed from the available predicates and constant symbols.] in first-order logic. In general, a (nonprobabilistic) first-order logic knowledge base can be seen as a set of *hard* constraints over possible worlds: if a world violates even only one formula, then it is impossible. In Markov logic, violations are allowed: a world violating a formula will be *less probable*, but not impossible. Every formula F_i in the knowledge base is associated with a real-valued weight w_i expressing the strength of the constraint. The probability distribution defined by an MLN is specified by the following log-linear model

$$P(x) = \frac{1}{Z}\exp\left(\sum_{i=1}^{n} w_i n_i(x)\right) \tag{3}$$

where x denotes a world, $n_i(x)$ is the number of groundings of formula F_i which are true in x, and the *partition function* Z acts as a normalization factor ensuring that P is a valid distribution. The higher w_i, the more a world which violates F_i is unlikely. MAP (maximum-a-posteriori) inference is then the problem of determining the maximizer of eq 3, a #-P complete problem for which however several approximate algorithms exist.

In many applications of MLNs it is convenient to split predicates into *evidence* predicates (which are assumed to be known at inference time) and *query* predicates (corresponding to predictions). MAP inference in this case becomes the problem of finding the most probable truth assignment to query groundings, y^*, given the available evidence x, i.e. computing

$$y^* = \arg\max_y P(y|x) = \arg\max_y \frac{1}{Z_x}\exp\left(\sum_{i\in\mathcal{F}_y} w_i n_i(x, y)\right)$$

where $\mathcal{F}_y$ are the formulas which contain query predicates.

In MLOCSR, we employ a function-free fragment of Markov logic. The truth value of evidence predicates is determined by the low-level pipeline described above. Evidence predicates describe the following:

- Presence of C-points, D-points, T-points, and lines connecting them;
- Presence of circles representing aromatic bonds;
- Distance-based geometrical properties, in particular to describe whether the distance between two extracted C-, D-, or T-points is below a certain threshold;
- Collinearity of three extracted C-, D-, or T-points;
- Chemical information describing the text recognized by the OCR (e.g., if a string is recognized as OH, then it is a hydroxyl group).

Query predicates (whose truth value is inferred by the MLN) are used for the following:

- Atom resolution, i.e. deciding whether two geometric points are associated with the same atom;
- Bond type identification (e.g., single, double, stereo).

In the following, we provide some details on the knowledge base used in MLOCSR. We break down the presentation into rules related to the geometrical properties of the low-level extracted entities and rules embedding chemical knowledge. The complete knowledge base (which has over 100 formulas) is available in the Supporting Information.

Geometric Rules. These rules are mainly used for atom resolution (i.e., recognizing C-, D-, or T-points associated with the same atom) and bond dype identification. For example, if two points extracted by the low-level processing stage are very close each other, then they likely correspond to the same atom. This rule can be translated into logic as

$$\text{VeryCloseCpoints}(c1, c2) \Rightarrow \text{SameCarbon}(c1, c2) \tag{4}$$

In the above formula, c1 and c2 represent logical variables (which stand for objects). In this sections, for the sake of simplicity, we omit the quantifier ∀ meaning that all variables are universally quantified. By adding a positive but not large weight to the formula, we allow worlds which violate the rule but those worlds which satisfy the rule will receive a higher probability.

Another set of geometric constraints was designed to reconstruct chemical bonds. For example, a double bond between two carbon atoms is likely to exist if we identified two C-points and two D-points arranged like C1, C7, D1, and D2 in Figure 2b. In logic this is written as

$$\begin{aligned}&\text{VeryCloseDpointAndCpoint}(d1, c1)\\ &\quad \wedge \text{VeryCloseDpointAndCpoint}(d2, c2)\\ &\quad \wedge \text{LineBetweenDpoints}(d1, d2)\\ &\quad \wedge \text{LineBetweenCpoints}(c1, c2)\\ &\quad \Rightarrow \text{DoubleBondBetweenCpoints}(c1, c2)\end{aligned} \tag{5}$$

Similar rules were designed to predict double bonds between C-points and T-points.

Within this group of rules which are used to correctly infer the chemical bonds within the molecule, a crucial role is played by the analysis of collinear points. For example, if three C-points c1, c2, and c3, which are not very close in the molecular graph, are (almost) collinear in such order, then c1 and c3 cannot be bonded:

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

$$\begin{aligned}&\text{CollinearCCC}(c1, c2, c3)\\&\quad \wedge \neg\text{VeryCloseCarbons}(c1, c2)\\&\quad \wedge \neg\text{VeryCloseCarbons}(c2, c3)\\&\quad \wedge \text{CarbonLine}(c1, c2)\\&\quad \wedge \text{CarbonLine}(c2, c3)\\&\quad \Rightarrow \neg\text{AreCarbonsBonded}(c1, c3)\end{aligned} \tag{6}$$

This rule helps correct those errors introduced in the low-level processing, where two consecutive lines are interpreted as a single line, in which case both an atom and a bond would be missed. Figure 6 illustrates this example for C-points c1, c2, and c3. An extension of eq 6 (with four variables) is used to handle the case of broken lines.

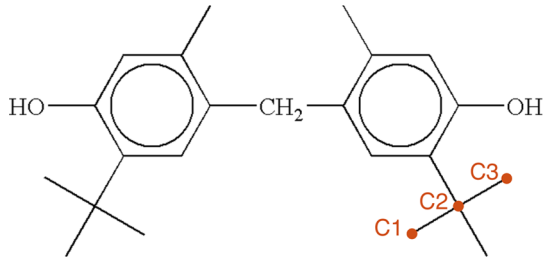


Figure 6. Example of a chemical drawing containing collinear carbon points: in this case, C1 and C3 should not be predicted as bonded.

Detection of wedge and hash bonds requires ad-hoc rules and relies on evidence predicates which indicate the presence of black-triangles (wedge bond) or dashed-lines. For example, the following rule is used to encourage the prediction of a wedge bond such as the one depicted in Figure 7, which connects a C-point (c1) and two T-points (t2 and t3):

$$\begin{aligned}&\text{BlackTriangleCTT}(c1, t2, t3)\\&\quad \wedge \text{VeryCloseTpoints}(t2, t3)\\&\quad \Rightarrow \text{WedgeBondBetweenCpointAndTpoint}(c1, t2)\end{aligned} \tag{7}$$

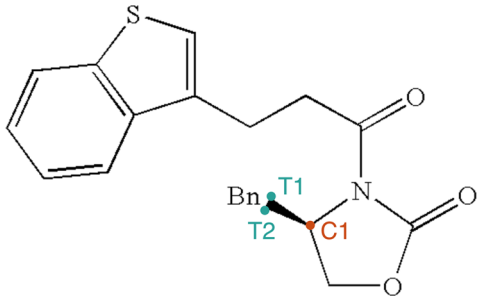


Figure 7. Example of a chemical drawing containing a wedge bond, which is predicted if a black-triangle pattern between three points is observed.

Chemical Rules. The powerful formalism of first-order logic allows to include in the knowledge base of the model also some chemical properties. An example in this sense is given by valence rules, which regulate the number of chemical bonds which an atom of a certain element can engage. For example, the hydroxyl group (OH), when connected to a carbon point, typically forms a single bond, whereas an oxygen atom is typically involved in a double bond. Therefore, rules like the following one can be easily conceived:

$$\begin{aligned}&\text{IsHydroxyl}(t) \wedge \text{LineBetweenCpointAndTpoint}(c, t)\\&\quad \Rightarrow \neg\text{DoubleBondBetweenCpointAndTpoint}(c, t)\end{aligned} \tag{8}$$

where predicate IsHydroxyl(t) indicates that T-point t contains a hydroxyl group. In some cases, the chemical knowledge is used in combination with geometric predicates, in order to disambiguate more complex situations:

$$\begin{aligned}&\text{IsOxygen}(t1) \wedge \text{LineBetweenCpointAndTpoint}(c1, t1)\\&\quad \wedge \text{VeryCloseCpoints}(c1, c2)\\&\quad \wedge \text{VeryCloseTpoints}(t1, t2)\\&\quad \Rightarrow \text{DoubleBondBetweenCpointAndTpoint}(c2, t1)\end{aligned} \tag{9}$$

where predicate IsOxygen(t) indicates whether T-point t contains an oxygen atom. Figure 8 shows an example where the formula in eq 9 is applicable.

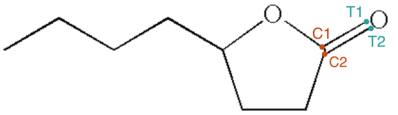


Figure 8. Example of a chemical drawing containing a text box with a single oxygen atom: in this case a double bond with a carbon atom should receive high probability.

Assembling Molecular Graphs. Given the inferred truth values of the query predicates, it is finally necessary to assemble the atom and bond entities in order to create the molecular graph, and produce a machine-readable file.

Small molecules may be formally represented as annotated underacted graphs, where nodes are atoms and edges represent chemical bonds. A large variety of data formats exist for exchanging the structural information on compounds. The most immediate approach is perhaps to represent the graph directly via a "connection table" (a list of atoms and a list of annotated bonds) which may be stored as an ASCII file, as it happens with several widespread formats such as the MDL Molfile[23] or the Chemical markup language[24] (CML). A second approach is to serialize the graph into a string. SMILES[25] for example use depth-first search and a number of simple conventions to obtain a string which is easily parsable by humans. InChI[26] is an alternative serial representation which is based on the McCay algorithm[27] to obtain a canonical ordering of the atoms (to address the graph isomorphism issue) and is guaranteed to create a unique identifier for each compound. Public domain tools exist that convert among existing formats[28] while preserving the essential information about a compound.

Our system produces as output an MDL Molfile, and therefore it is necessary to produce the connection table (atoms and bonds) describing the molecule, starting from the output of Markov Logic inference.

In order to determine the atoms in the molecule, the transitive closure of the SameCarbon predicate is simply applied, and then a carbon atom is inserted into the molecular graph for each representative in such closure set. As for text-boxes elements, the text recognized by the OCR will be used

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

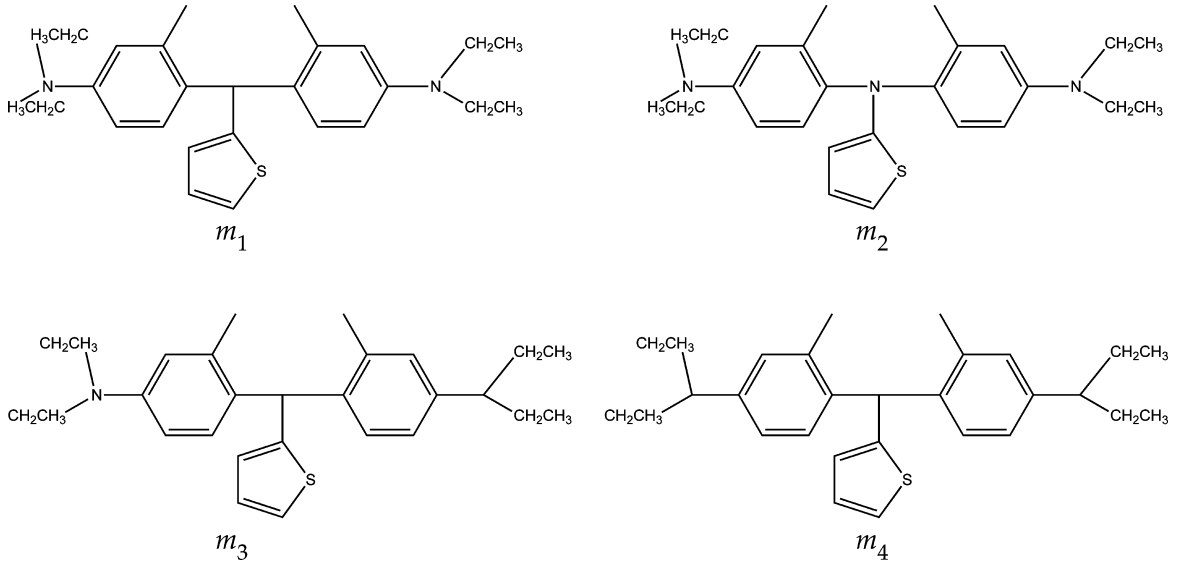


Figure 9. Tanimoto similarity does not necessarily reflect recognition accuracy. In this example m_1 is the ground truth and m_2, m_3, and m_4 are possible predictions. When using path fingerprints, $\tau(m_1, m_2) = 0.42$ although the difference between the two structures is just one atom (the central nitrogen). The large similarity drop is due to the fact that the error is in a node touched by several paths. Interestingly, m_3 and m_1 also differ by a single atom but $\tau(m_1, m_3) = 0.88$. The fourth molecule m_4 has two errors, but $\tau(m_1, m_4) = 0.61$ is still better than $\tau(m_1, m_2)$.

for the atom label, after checking for typical spelling errors (e.g., CooH is corrected into COOH), as it happens also for OSRA. It is worth remarking that, at this point, two possible behaviors can be adopted within the recognition process: (1) the string returned by the OCR module is simply inserted "as is" into the MOL file as a *superatom*, with no further manipulation; (2) the recognized string is compared against a list of superatoms, that is a list of known molecular groups (such as CH_2, COOH, MeO, and so on) for which the expansion in the SMILE format is known: if the string corresponds to one of such superatoms, all the atoms and bonds produced by the expansion are automatically inserted into the MOL file. These two recognition modes will be subject of deep analysis in the experimental section.

Concerning the introduction of bonds within the molecular graph, a bond is created between two atoms if at least one bond between two points representing such atoms is predicted by the inference process. The cardinality of the predicted bond is chosen according to a priority between bond types: first aromatic bonds are checked, followed by triple, double, and finally by single ones. If a single bond is predicted, stereochemistry is then also checked, by observing the truth values of the corresponding WedgeBond and HashBond predicates: in such cases, the wedge/hash bit is accordingly set within the produced MOL file.

EXPERIMENTAL RESULTS

Data Sets. We tested our system on some benchmark data sets, and we compared against OSRA[10] (version 1.4.0). A first data set can be downloaded directly from OSRA Web site [http://cactus.nci.nih.gov/osra/uspto-validation-updated.zip] and it consists of 5719 images produced by the US Patent Office Complex Work Units (each image has the corresponding ground truth file in MOL format). Such data set is very redundant, as it contains groups of almost identical images/molecules: the performance of the tested predictors could therefore be ill-conditioned by this data distribution. For this reason, we constructed a second data set, by clustering images and taking one representative for each cluster [The spectral clustering algorithm was used to this aim.]: following this approach, we obtained a nonredundant subset of 937 molecules (the list is provided as Supporting Information). The third data set employed in our experiments is ChemInfty,[29] a publicly available [http://www.iapr-tc11.org/mediawiki/index.php/Chem-Infty_Dataset:_A_ground-truthed_dataset_of_Chemical_Structure_Images] collection of 869 molecules extracted from Japanese patent applications published in 2008.

Performance Analysis Procedures. Recognition quality may be evaluated by comparing reconstructed structures against the corresponding ground truth. Two approaches have been used in the literature for this purpose: InChI match and Tanimoto similarity. The InChI string is an identifier of a chemical compound, i.e. two strings are identical if and only if the two structures are chemically indistinguishable. However, a higher number of perfectly recognized structures does not necessarily imply a better effectiveness of a chemical OCR system: if a compound is not perfectly recognized, it may contain just one error or several errors. To obtain a less crude measure of performance, some authors have proposed the use of the Tanimoto similarity,[9,10] which we briefly explain here. The *fingerprint* of a molecule is a bit vector of length obtained by hashing certain features or substructures of the molecule (e.g., paths obtained by depth-first traversal of the molecular graph).[30] The Tanimoto similarity between two molecules (m_1 and m_2) is then defined as the Jaccard index between the two sets described by the two fingerprints ($\mathrm{fp}(m_1)$, $\mathrm{fp}(m_1)$), i.e.

$$\tau(m_1, m_2) = \frac{|\mathrm{fp}(m_1) \cap \mathrm{fp}(m_2)|}{|\mathrm{fp}(m_1) \cup \mathrm{fp}(m_2)|}$$

Neglecting the fact that features may collide due to hashing, the Tanimoto similarity is large when two molecules share many common features. However, when using paths as features, similarity may significantly depend on the position of the errors in the chemical graph, and not on the number of errors, as illustrated in Figure 9. As conceded in ref 10, using other kinds of fingerprints does not provide better accuracy measures. We

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

therefore propose a third approach to measure recognition accuracy, inspired from information retrieval. First, we match vertices in the predicted structure to atoms in the ground truth. To compute the match we form a bipartite graph whose vertices are true and predicted atoms or superatoms and whose edges connect predicted and true vertices and weighted by the Euclidean distance between the 2D coordinates. We then compute a minimum-weight bipartite matching[31] to match every ground truth vertex to a prediction vertex. This approach is only feasible if the system outputs a connection table where superatoms are left unexpanded: in fact, if superatoms are expanded, the 2D atom coordinates needs to be computed by a layout algorithm which does not necessarily adhere to the layout in the ground truth. Occasionally, we found cases in the available data sets where the 2D layout in the ground truth is not perfectly congruent with the layout in the original image. To address this problem, we transformed the 2D coordinates according to the homography found by the RANSAC algorithm.[32] The matching between the true molecular graphs (V, E) and the prediction (V', E') is a function $m: V \rightarrow V' \cup \{v\}$ where conventionally $m(v) = \nu$ if v is not matched. The number of correctly recognized (super)atoms is defined as

$$\mathrm{TP}^a = \sum_{v \in V} 1\{\lambda(v) = \lambda(m(v))\} \tag{10}$$

where $1\{e\}$ is the indicator of a Boolean expression e and $\lambda(v)$ is the label of vertex v, with $\lambda(\nu)$ = nil. For our purposes, the label $l(v)$ is an an element symbol (such as N, O, Cu, etc.) if v is an atom, and a string of element symbols or abbreviations, possibly with subscripts/superscripts and/or parentheses (e.g., N–$(CH_2CH_3)_2$, MeO_2SO, Ph_2P, OBoc, etc.), if v is a superatom. Precision, recall, and F_1 measures may be then defined as usual:

$$P^a = \frac{\mathrm{TP}^a}{|V'|}$$

$$R^a = \frac{\mathrm{TP}^a}{|V|}$$

$$F_1^a = \frac{2P^aR^a}{P^a + R^a} \tag{11}$$

Similar quantities may be defined for the edges (chemical bonds) using the following definition of "true positive" bonds:

$$\mathrm{TP}^b = \sum_{\{u,v\} \in E} 1\{\lambda(u) = \lambda(m(u))\} \cdot 1\{\lambda(v) = \lambda(m(v))\} \cdot 1\{\mu(\{u, v\}) = \mu(\{m(u), m(v)\})\} \tag{12}$$

where $\mu(\{u, v\})$ is the label of edge $\{u, v\}$. Note that in some circumstances F_1^a or F_1^b may be smaller than 1 when applied to pairs of molecules that have the same InChI, as the matching between the two molecular graph might not be perfect (see Figure 10). The precision, recall, and F_1 measures should be seen as complementary to the traditional performance measures based on InChI and Tanimoto. The latter put emphasis on chemical faithfulness of the predicted structure while the former put emphasis on the graphical faithfulness and quantify more precisely the human effort that would be required to manually correct wrong predictions. In order to measure precision, recall and F_1, we manually created ad-hoc MDL ground truth files where superatom are not expanded and reflect the image contents (see the Supporting Information).

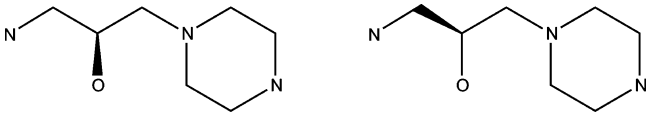


Figure 10. Graphically distinct but chemically equivalent structures.

RESULTS AND DISCUSSION

We performed experiments in two distinct settings, depending on whether the two algorithms (both OSRA and MLOCSR) were allowed to expand superatoms or not. In the first case, the InChI measure could be used to measure the performance of the predictors, while in the second case no InChI generation is possible, and therefore, the measurements on the recognition of atoms and bonds presented in the previous section were employed. When disabling the expansion of superatoms, the ground truth MOL files had to be accordingly rearranged: for this reason, this setting was employed only on the cluster representatives of the USPTO data set and on the ChemInfty data sets, being the adaptation of the whole USPTO data set too time-consuming.

An MLN consisting of 128 first-order logic rules was employed in our experiments, using the Alchemy software package developed at the University of Washington [http://alchemy.cs.washington.edu]. MaxWalkSAT[33] with 3 different tries and 1 000 000 steps for each try was used as the MAP inference algorithm.

OSRA by default processes the input image at three different resolutions, choosing the best output by employing a quite complex empirical confidence function[10] which counts the number of atoms, rings, fragments, and other objects in the predicted molecule. Since OSRA expands the superatoms by default, in order not to treat unfairly the system, in our experimental setting which does not perform superatom expansion we forced OSRA to use the resolution chosen as the best one according to the standard expanding version.

Results measuring the accuracy in the geometric reconstruction of the molecular diagrams are shown in Table 2, within the setting in which superatoms are not expanded. Table 3 reports instead results in the expanded setting, where the quality of the predictions can be analyzed with chemical performance measurements.

Table 2. Results Measuring the Geometric Quality of the Reconstruction[a]

	method	F_1 atoms	F_1 bonds	perfect F_1
USPTO cluster	MLOCSR	99.1	98.8	84.1
	OSRA	97.5	97.8	77.6
ChemInfty	MLOCSR	94.2	94.2	54.0
	OSRA	85.3	88.4	45.2

[a]In this setting, superatoms are not expanded. Perfect F_1 indicates the percentage of molecules where the predictor achieve a value of F_1 equal to 100 for both atoms and bonds.

In the first setting, it is clear that MLOCSR is more accurate than OSRA in correctly identifying the atoms and bonds in the molecule, both on the clustered version of the USPO data set, and on the ChemInfty data set, perfectly recognizing 6.5% and 8.8% more of the molecules exactly reconstructed by OSRA in the two cases, respectively. Note that, for most molecules, predictions are affected by a small number of errors (one or two atoms or edges). This is the case for both MLOCSR and

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Table 3. Results Measuring the Quality of Reconstruction from a Chemical Point of View[a]

	method	InChI basic	InChI full	Tanimoto
USPTO	MLOCSR	86.1	79.4	0.948
	OSRA	85.2	81.4	0.940
USPTO cluster	MLOCSR	79.1	71.9	0.929
	OSRA	77.5	74.0	0.917
ChemInfty	MLOCSR	35.1	35.0	0.776
	OSRA	36.9	36.0	0.740

[a]In this setting superatoms are expanded. InChI basic does not consider the stereochemistry level, which is included in the InChI (full) measurement.

OSRA predictions. Since the total number of atoms ranges from a few dozens to one hundred or more and even a single error implies $F_1 < 100$ for that molecule, the perfect F_1 measure, which counts the number of molecules having $F_1 = 100$ for both atoms and bonds, is much lower than the F_1 measure on atoms or bonds.

In the second setting, we considered the percentage of molecules with perfect InChI and the Tanimoto index as the performance measurement. We counted both the number of molecules for which the InChI was correct except for the stereochemistry level (InChI basic) and the number of molecules having the full InChI correctly predicted (InChI full). Table 3 shows that the performance of MLOCSR is superior to that of OSRA for InChI basic (except for ChemInfty) and for the Tanimoto index. On InChI full, on the other hand, OSRA has a slightly advantage over MLOCSR, which indicates that our system has still margins of improvement in the recognition of stereo bonds.

We also conceived an additional experiment in order to assess the performance of the predictors when the quality of the input images degrades: we resampled the images in the USPTO clustered data set at three different levels, which we will call high degradation (HD), medium degradation (MD) and low degradation (LD). [Using ImageMagick software, such three levels correspond to image resampling using three different parameters r, specifically r = 210 (HD), 240 (MD), and 270 (LD).] Results in Table 4 show the performance of MLOCSR and OSRA in this setting, showing the advantage of our approach even in this scenario. Here we report only geometric measurements in the not-expanded setting: the task being extremely challenging, measuring the percentage of correct InChI would produce too low results for both systems.

Table 4. Results with Images Having Degrading Quality, Obtained by Resampling the Original USPTO Clustered Data Set[a]

	method	F_1 atoms	F_1 bonds	perfect F_1
LD	MLOCSR	96.0	97.0	45.3
	OSRA	89.5	93.4	14.1
MD	MLOCSR	88.3	90.8	13.8
	OSRA	82.2	89.5	2.2
HD	MLOCSR	80.3	83.8	2.3
	OSRA	76.0	81.5	1.4

[a]We employ the setting in which superatoms are not expanded. Perfect F_1 indicates the percentage of molecules where the predictor achieve a value of F_1 equal to 100 for both atoms and bonds.

4. CONCLUSIONS

We have introduced a new method for optical recognition of chemical diagrams. The main novelty in our approach is the use of Markov logic to incorporate chemical and graphical knowledge in the form of soft first order logic formulas involving low-level graphical primitives. Our system achieves state-of-the-art recognition accuracy and compares favorably to existing approaches. The advantages are more evident in the more difficult data sets containing lower quality images, showing the benefit of a sound probabilistic reasoning engine over hand-coded deterministic heuristics.

In this paper we have also suggested alternative approaches for measuring recognition performance which can complement the widespread use of correct InChI and Tanimoto index. In particular, the use of the F_1 measure for atoms, bonds and full molecules can provide a better estimation of the recognition quality as it is directly related to the amount of work that would be necessary to manually correct mistakes.

One obvious direction for further improving this research is the use of learning algorithms to fine-tune Markov logic weights or even to learn new formulas from data by using structure learning algorithms such as those described in ref 34−36. The application of supervised learning techniques in this context is however not straightforward: the available background knowledge simply consists of molecular connection tables with no information about the graphical elements which are extracted at the lower level. This means that the truth state of query predicates for Markov logic are not directly observed but need to be reconstructed (possibly in a semiautomatic fashion) starting from graphical primitives and the available MDL files.

While our system is able to handle a vast set of formulas, there remains several directions for further improvement, addressing specific cases which are not currently handled. These include some graphical ambiguities due to touching and broken characters, or characters touching lines; Markush features such as substituent replacement in R-groups, link nodes, or repeating units; recognition of chemical tables or reactions.

Our contributions shows that Markov logic is a viable approach to reason about the information extracted by the low-level image processing modules. Such an approach is not necessarily specific to the low-level modules we have developed and, in principle, an MLN could be used in other related systems such as OSRA.[10] An actual implementation in this direction, however, is not straightforward due to the tighter integration between the low-level feature extractors and higher-level chemical knowledge in OSRA.

■ ASSOCIATED CONTENT

Supporting Information

ZIP containing the ground truth files (MOL format with unexpanded superatoms) for the ChemInfty data set. ZIP containing the ground truth files (MOL format with unexpanded superatoms) for the cluster representatives of the USPTO data set. TXT containing the list of cluster representatives of the USPTO data set. TXT containing the Markov logic network in Alchemy format. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Authors

*E-mail: paolo.frasconi@unifi.it (P.F.).

dx.doi.org/10.1021/ci5002197 | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

*E-mail: francescogabbrielli@gmail.com (F.G.).
*E-mail: lippi@diism.unisi.it (M.L.).
*E-mail: simone.marinai@unifi.it (S.M.).

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We dedicate this paper to Giovanni Soda, a friend and colleague who participated to this research with fruitful discussions before he passed away on July 2, 2014. We would like to thank Claudio Celletti for a preliminary implementation of some modules of this system. This research is partially supported by Italian Ministry of Education, University, and Research (PRIN project 2009LNP494).

■ REFERENCES

(1) Kind, T.; Scholz, M.; Fiehn, O. How large is the metabolome? A critical analysis of data exchange practices in chemistry. *PLoS One* **2009**, *4*, e5440.
(2) Gaulton, A.; Overington, J. P. Role of open chemical data in aiding drug discovery and design. *Future Med. Chem.* **2010**, *2*, 903−7.
(3) Contreras, M. L.; Allendes, C.; Alvarez, L. T.; Rozas, R. Computational perception and recognition of digitized molecular structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 302−307.
(4) McDaniel, J.; Balmuth, J. Kekule: OCR-optical chemical (structure) recognition. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 373−378.
(5) Casey, R.; Boyer, S.; Healey, P.; Miller, A.; Oudot, B.; Zilles, K. Optical recognition of chemical graphics. *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR)*; Tsukuba Science City, Oct 20−22, 1993; pp 627−631.
(6) Ibison, P.; Jacquot, M.; Kam, F.; Neville, A. G.; Simpson, R. W.; Tonnelier, C.; Venczel, T.; Johnson, A. P. Chemical literature data extraction: The CLiDE Project. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 338−344.
(7) Boyer, S.; Casey, R. G.; et al. Apparatus and method for optical recognition of chemical graphics. U.S. Patent no. 5,157,736, 1992.
(8) Algorri, M.-E.; Zimmermann, M.; Friedrich, C.; Akle, S.; Hofmann-Apitius, M. Reconstruction of Chemical Molecules from Images. *Proceedings of the 29th Annual International Conference on Engineering in Medicine and Biology Society (EMBS'07)*; Lyon, Aug 22−26, 2007; pp 4609 −4612.
(9) Park, J.; Rosania, G. R.; Shedden, K. A.; Nguyen, M.; Lyu, N.; Saitou, K. Automated extraction of chemical structure information from digital raster images. *Chem. Cent J.* **2009**, *3*, 4.
(10) Filippov, I. V.; Nicklaus, M. C. Optical structure recognition software to recover chemical information: OSRA, an open source solution. *J. Chem. Inf Model* **2009**, *49*, 740−3.
(11) Lounnas, V.; Vriend, G. AsteriX: A Web Server To Automatically Extract Ligand Coordinates from Figures in PDF Articles. *J. Chem. Inf. Model.* **2012**, *52*, 568−576.
(12) Valko, A. T.; Johnson, A. P. CLiDE Pro: The Latest Generation of CLiDE, a Tool for Optical Chemical Structure Recognition. *J. Chem. Inf. Model* **2009**, *49*, 780−787.
(13) Richardson, M.; Domingos, P. Markov logic networks. *Machine Learning* **2006**, *62*, 107−136.
(14) Domingos, P.; Kok, S.; Lowd, D.; Poon, H.; Richardson, M.; Singla, P. In *Probabilistic Inductive Logic Programming*; De Raedt, L., Frasconi, P., Kersting, K., Muggleton, S., Eds.; Springer: New York, 2008; pp 92−117.
(15) Lu, S.; Su, B.; Tan, C. L. Document image binarization using background estimation and stroke edges. *IJDAR* **2010**, *13*, 303−314.
(16) Tombre, K.; Tabbone, S.; Plissier, L.; Lamiroy, B.; Dosch, P. In *Document Analysis Systems V*; Lopresti, D., Hu, J., Kashi, R., Eds.; Lecture Notes in Computer Science; Springer: Berlin Heidelberg, 2002; Vol. *2423*; pp 200−211.
(17) Sadawi, N. M.; Sexton, A. P.; Sorge, V. MolRec at CLEF 2012−Overview and Analysis of Results. *CLEF (Online Working Notes/Labs/Workshop)*, Rome, Italy, September 17−20, 2012.
(18) Fletcher, L. A.; Kasturi, R. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Machine Intell.* **1988**, *10*, 910−918.
(19) Su, F.; Cai, S. A Character Extraction and Recognition Method for Line Drawings. Image and Signal Processing. *2nd International Congress on Image and Signal Processing*, Tianjin, Oct 17−19, 2009; pp 1−5.
(20) Hilaire, X.; Tombre, K. Robust and Accurate Vectorization of Line Drawings. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 890−904.
(21) Tombre, K.; Ah-Soon, C.; Dosch, P.; Masini, G.; Tabbone, S. In *Graphics Recognition Recent Advances*; Chhabra, A., Dori, D., Eds.; Lecture Notes in Computer Science; Springer: Berlin Heidelberg, 2000; Vol. *1941*, pp 3−18.
(22) Douglas, D.; Peucker, T. Algorithms for the reduction of the number of points required for represent a digitized line or its caricature. *Can. Cartogr.* **1973**, *10*, 112−122.
(23) Accelrys Software Inc. CTfile Formats. http://download.accelrys.com/freeware/ctfile-formats/ctfile-formats.zip (last accessed August 4, 2014).
(24) Murray-Rust, P.; Rzepa, H. Chemical markup, XML, and the Worldwide Web. 1. Basic principles. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 928−942.
(25) Weininger, D.; Weininger, A.; Weininger, J. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97−101.
(26) McNaught, A. The IUPAC international chemical identifier. *Chem. Int.* **2006**, *28* (6), 12.
(27) McKay, B. D. Practical graph isomorphism. *Congressus Numerantium* **1981**, *30*, 45−87.
(28) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J. Cheminf.* **2011**, *3*, 33.
(29) Nakagawa, K.; Fujiyoshi, A.; Suzuki, M. Ground-truthed dataset of chemical structure images in Japanese published patent applications. *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, Boston, MA, June 9−11, 2010; pp 455−462.
(30) Flower, D. R. On the properties of bit string-based measures of chemical similarity. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 378−386.
(31) Bondy, J. A.; Murty, U. S. R. *Graph theory*; Springer: New York, 2008; Vol. *244*.
(32) Hartley, R.; Zisserman, A. *Multiple view geometry in computer vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2003.
(33) Kautz, H.; Selman, B. Pushing the envelope: planning, propositional logic, and stochastic search. *Proceedings of the thirteenth national conference on Artificial intelligence*, 1996; Vol. *2*, pp 1194−1201.
(34) Kok, S.; Domingos, P. Learning the structure of Markov logic networks. *Proceedings of the 22nd international conference on Machine learning*, Bonn, Germany, Aug 7−11, 2005; pp 441−448.
(35) Huynh, T. N.; Mooney, R. J. Discriminative structure and parameter learning for Markov logic networks. *Proceedings of the 25th international conference on Machine learning*, Helsinki, Finland, July 5−9, 2008; 416−423.
(36) Jaeger, M.; Lippi, M.; Passerini, A.; Frasconi, P. Type Extension Trees for feature construction and learning in relational domains. *Artif. Intell.* **2013**, *204*, 30−55.