Generation of Search-able PDF of the Chemical Equations segmented from Document Images*

Prerana Jana
IIEST-Shibpur
India
prerana.jana@gmail.com

Anubhab Majumdar
IIEST-Shibpur
India
anubhabmajumdar93
@gmail.com

Sekhar Mandal
IIEST-Shibpur
India
sekhar@cs.iiests.ac.in

Bhabatosh Chanda
ISI-Kolkata
India
chanda@isical.ac.in

ABSTRACT

PDF format of scanned document images is not searchable. OCR tries to remedy this adversity by converting document images into editable and searchable data, but it has its own limitations in presence of equations - both mathematical and chemical. OCR system for mathematical equation is already a major research area and has provided successful result. However, chemical equation segmentation has been a less ventured road. In this paper, we present a novel method for automated generation of searchable PDF format of segmented chemical equations from scanned document images by performing chemical symbol recognition and auto-correction of OCR output. We use existing OCR system, pattern recognition technique, contextual data analysis and a standard LaTeX package to generate the chemical equation in searchable PDF format. The effectiveness of the proposed method is verified through exhaustive testing on 234 document images.

Keywords

Chemical equations, mathematical symbols, morphological operation.

1. INTRODUCTION

Text keywords are used for retrieving documents from WWW using search engines like Google. A large number of documents are being digitized today for the purpose of archival, transmission and browsing. The existing OCR systems show high accuracy in interpreting text portions, but fail to process other components like graphics, half-tones, chemical and mathematical equations properly. A few studies [19], [20], [21] are directed toward math-symbol or math equation recognition assuming that the math-zones are already marked. A number of work has been done over the past decade to detect and extract the mathematical equations present in heterogeneous document images.

Fateman et al. [4] proposed a scheme which utilised character size, font information etc. to identify the connected components. Two bags, namely *text* and *math* are defined. The *text* bag is used to keep all letters and numbers; whereas the *math* bag collects punctuation, special symbols, Roman digits, italic letters, lines and dots. Objects in the *math* bag are then grouped together according to their spatial proximity. Grouping of items in text bag is redefined next followed by review and correction to move isolated items to their proper destinations. Math component segmentation is done in [5] through physical and logical segmentation using spatial characteristics of the math zone as well as identifying some math-symbols. The document is then segmented to characters, words, lines and blocks by physical segmentation. The logical segmentation process that follows consists of two steps; first the displayed math is detected by identifying their usual center position and in the next step in-line maths is detected by identifying special symbols.

Kacem et al. [6] extracted the equations using fuzzy logic by detecting mathematical operators like '+', '-', etc. Their method was tested on a dataset consisting of 300 expressions and the success rate is about 93%. As some of the operators like '+', '-', '(' and ')' do appear in chemical equations as well, it leads to the miss-classification of chemical equations as mathematical equations reducing the success rate. A similar method has been proposed in [7] to segment the mathematical expression in printed documents. The statistical approach taken by Garain [9] on the corpus of 400 pages to differentiate normal text lines and lines containing equations/expressions is on the basis of their white spacings which are usually larger in math-equation than the normal text. However, the chemical equations in the documents bear the same property. Jin et al. [11] proposed a similar method to extract displayed formulas using Parzen classifier.

Drake and Baird [12] came up with a graphical approach; similarly Guo et al. [13] developed a Gaussian mixture model to describe spatial relationships between sub-components of

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

DocEng '16, September 12-16, 2016, Vienna, Austria
© 2016 ACM. ISBN 978-1-4503-4438-8/16/09...$15.00
DOI: http://dx.doi.org/10.1145/2960811.2960822

a math expression. Another method to check text style (regular, italic, bold) at the character level has been proposed in [10]. Garain [8] proposed a method to segment the displayed and embedded mathematical formulas from the documents using a bunch of features. The method is tested on a dataset of 200 images containing 1163 embedded and 1039 displayed expressions and the success rate is 88.3% and 97.2% respectively for embedded and displayed expressions. A method proposed by Chu and Liu [14] used features based on centroid fluctuation information on non-homogeneous regions to detect displayed and embedded formulas.

In a nutshell, in all the above methods emphasis is given only in mathematical equation. In eventual segmentation or classification, the chemical equations would automatically be included as a part of mathematical (or other) equations thereby reducing the success rate of the segmentation and effectiveness of the subsequent classification, if any.

There are some methods that are used to reconstruct chemical formula from scanned image. They have used chemical datasets. Algorri et al. [23, 24] proposed a system that reconstructs chemical molecules from scanned document. They have used connected component analysis and their own vectorisation algorithm for character recognition. Connected components that are not recognised by the OCR engine, are used to produce a graph of vectors. A rule based approach reconstructs the formula from the vector graph and the character information. ChemReader [25] starts with connected component. Alphanumerics are recognised using the GOCR open source OCR tool. Graphical components are identified using Hough transforms, corner detection and other bespoke algorithms.

Jana et al. [27] proposed a fully automated segmentation and detection technique of chemical equations present in heterogeneous document images. This paper is an extension of their work with some improvements to the original method. We propose a novel automated approach to auto-correct the extracted chemical equations and convert the same into an editable LaTeX file.

The paper is organized as follows – Detection and segmentation of chemical equations, auto-correction of the extracted chemical equations and its conversion to PDF form is presented in section 2. Section 3 presents experimental results. We conclude the paper in section 4.

2. PROPOSED WORK

The proposed method consists of three distinct steps and they are as follows: (i) Extraction of displayed chemical equations; (ii) Refining OCR output; and (iii) Converting the extracted chemical equations into search-able PDF format.

2.1 Extraction of displayed chemical equations

For this portion of the work, we have largely followed the method proposed in [27] with few improvements.

A skew free heterogeneous binary image is the input to the proposed algorithm.

Steps involved for identification of displayed chemical equations are - (i) Text line segmentation; (ii) Word blob formation; (iii) Operator identification; (iv) Displayed equation (DE) zone segmentation; and (v) Classification of extracted DE zone(s);

The details of the aforementioned steps are described in the following subsections.

2.1.1 Text line segmentation

To detect DE zones, text lines have to be segmented first from which the operators are identified to determine whether a text line is a displayed equation or not. We have taken the horizontal projection profile of the document page to segment the text line.

2.1.2 Word blob formation

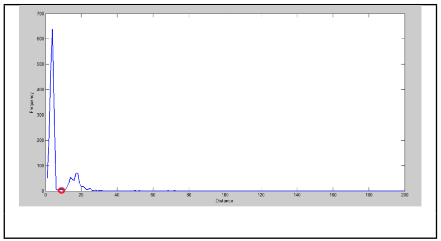


Figure 1: Distance histogram.

This is done by coalescing the characters in a word using morphological closing operation. Such character coalescing process depends on the accuracy in detecting the normal character gap and the gap between the consecutive connected components in that text line. A histogram with the distribution of distance between two consecutive characters in the document is plotted.

The distance histogram is a multi-modal histogram. The first peak corresponds to the character gaps. A distance histogram is shown in Fig. 1.

Our intention is to find out character gaps in running texts of a document page so that we can combine the consecutive characters into a single blob. Hence, we consider the upper boundary (l) of the first hump as the length of structuring element. Morphological close operation with a line structuring element of length (l) is carried out to form the word blobs.

2.1.3 Operator detection

In a mathematical or chemical equation, one or more operators are present. These operators signal us the presence of displayed equations in a document. We have considered the set of *operators* ($+$, -, $\rightarrow$, $\leftarrow$, $\leftrightarrow$, $\rightharpoonup$, $\leftharpoondown$) which are commonly used both in chemical equations as well as mathematical equations to fulfil our aim to identify displayed zones containing chemical or other equations.

After blob formation, small component like dots of i *and* j are eliminated on the basis of area. The region (R_c) corresponding to each blob is cropped using its bounding box information from the original image and the number of connected component(s) present in R_c is counted. If the number of components is more than 1, then that blob is not an operator and is removed from the blob image.

The remaining components in the blob image are operators along with some alphanumerics like a, A, (, etc. The logical AND operation is performed between the blob image and the original image. The Euler number of the operators that we have considered is 1 and based on this feature some of the alphanumerics are discarded and the resultant image is denoted by I_s

Our next task is identify operator from I_s and for this, we have used a neural network with the following feature set.

- Aspect ratio: (f_a) of each component

- Density:

$$f_d = \frac{\#pixels_o}{\#pixels_b},$$

where $\#pixels_o$ denotes the number of object pixels and $\#pixels_b$ denotes area of the bounding box.

- The horizontal and vertical projection profiles of each component is obtained.
(i)Spike for horizontal projection profile

$$f_{sh} = \frac{\#pixels_{on}}{w}$$

where $\#pixels_{on}$ denotes the number of on-pixels at the middle of the projection profile and w denotes the width of the profile.
(ii) Spike for vertical projection profile

$$f_{sv} = \frac{\#pixels_{on}}{h}$$

where $\#pixels_{on}$ denotes the number of on-pixels at the middle of the projection profile and h denotes the height of the profile.

- Ratio of

$$f_{dr} = \frac{\#pixels_{on}}{\#pixels_{off}},$$

where $\#pixels_{on}$ denotes the number of object pixels and $\#pixels_{off}$ denotes the number of background pixels along the diagonal of each component. The ratio is determined for both the right(rd) and left(ld) diagonals.

- A binary variable (f_{open}): $f_{open} \in \{1, 0\}$.
The value of f_{open} is obtained using morphological opening operation with a line like structuring element (SE) of length $\frac{w}{2}$, where w is the width of the component. If the output of the opening operation has a single component, f_{open} is set to 1, else f_{open} is 0.

- Number of end points (f_{ep}): f_{ep} is number of end points of a connected component. This is determined using thinning operation. After thinning each connected component in I_s, if a pixel has a single 8-connected neighbor, then that pixel represents an end point.

Now, $[f_a, f_d, f_{sh}, f_{sv}, f^{rd}_{dr}, f^{ld}_{dr}, f_{open}, f_{ep}]$ is the feature vector for classification of operators from I_s.

We classify all single components in I_s into following 4 classes:

1. Arrows (→, ←, ↔, ⇀, ↽)
2. Minus (-)
3. Plus (+)
4. Others (, (,), etc)

The classification is done using a two-layer feed-forward network with 100 nodes in the hidden layer and sigmoid hidden and softmax output neurons. The network is trained with scaled conjugate gradient back propagation.

We have taken a set of 7046 samples from our image dataset. This set consists of aforesaid operators and other symbol/character. Out of these 7046 images - 1000 plus, 1000 minus, 1000 arrows and 1000 other single characters are taken for the training set.

Table 1: Results of classifier for identification of *operators*

	1	2	3	4
1	319	0	0	2
2	0	120	0	0
3	0	0	2267	3
4	0	0	0	335

The remaining 3046 samples are used for testing the classifier. The accuracy of the network is depicted by the confusion matrix of the test dataset in Table 1. The reason for high number of '+' sign in the dataset is because it is the most frequently encountered *operator* as compared to the other operators or single characters.

We also need to identify the direction of the arrowhead for its correct representation. The direction of arrowhead is identified by measuring the height of the arrow elements near its two ends.

To detect '=' or '⇌' one extra step is required. For each '-' and '⇀', a rectangular window of size $l \times l/2$ is placed below the symbol to check if there is '-' and '↽' respectively within the window; if present, they are considered to form either '=' or '⇌' sign. l be the length of the symbol. The upper boundary of the window coincides with the lower boundary of the bounding box of each aforesaid symbol.

2.1.4 DE zone exraction

Initially, all the text lines consisting of at least one *operator* are considered candidate displayed equations (CDE). The *operators* are separated from CDE. The upper boundary (u_v) of the second hump of distance histogram (Fig. 1) is obtained which represents the word gaps in the text line. For each CDE zone morphological closing operation with a line structure element of length u_v is carried out. If the distance between two neighbouring components is less than u_v, it means they belong to a same word and are merged by closing operation.

For each CDE we count the number of *operators* and other corresponding components in the output of closing operation. If the number of components $\leq 2\times$ number of *operators*, then the CDE is considered displayed equation; otherwise some embedded formulae/equations may exist in the line.

2.2 Classification of extracted DE zone

The extracted DE zones can be either chemical or other (ex:- mathematical) equations. Now, each displayed equation is an input to the inbuilt OCR of MATLAB R2014a. The OCR returns each DE zone as a text string. We made a dictionary out of all the elements in the periodic table. An important observation is that an element always starts with a capital letter. Using this property, an element can be expressed by a regular expression [A-Z][a-z]*. It means an element's symbol starts with an upper case letter and may or may not have one or more lower case letters (for example H, He, Uut etc). We have designed a parser to extract the sub-string matching the regular expression mentioned above with the following grammar:

$$\begin{aligned} start &\rightarrow capital.follow \\ follow &\rightarrow small.follow| \in \\ capital &\rightarrow A|B|\ldots|X|Y|Z \\ small &\rightarrow a|b|\ldots|x|y|z \end{aligned}$$

Each of the substring returned by the parser is matched against the aforesaid dictionary and if it is a positive match then that substring is considered as a symbol of the chemical element. Let us consider, the number of substrings extracted from the OCR output by the parser is n and the number of positive matches the aforementioned dictionary is m. If m:n ratio is more than a threshold value β then this DE is considered a Chemical Equation. This threshold (β) is set to 0.7 experimentally by running the proposed algorithm on dataset containing 1390 displayed equations. The reason for the ratio not being 1 are: (i) Limitations of OCR, and (ii) Touching and broken characters.

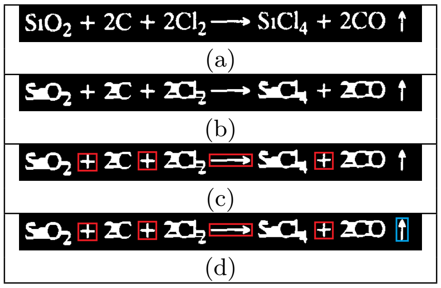


Figure 2: Up and down arrow detection (a) A chemical equation; (b) Image after blob formation ; (c) Operators are marked in red; (d) Detected arrow in blue.

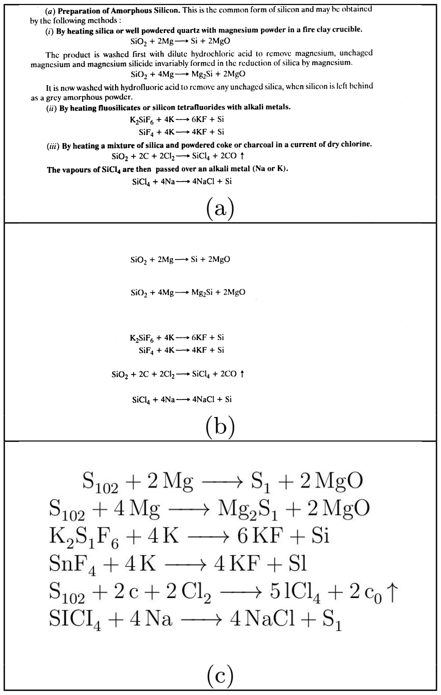
The ↑ and ↓ are frequently used in chemical equation to represent the state of compounds and are thus important to detect. For each identified chemical equation, blob formation is done as discussed in Sec. 2.1.2. After blob formation, we apply component analysis method. Then operators are marked (Fig. 2(c)) in the blob image, as they are already identified. For each component in blob image (which is not an operator), we check its immediate left component (C_l) and right component (C_r). If none of C_l or C_r is an *operator*, the the component under consideration is an up arrow or a down arrow. Fig. 2(d) shows the detected up arrow in blue. The identification of the arrowhead is done in the same way as discussed in *operator* detection.

An example of chemical equation segmentation from a sample document image is given in Fig. 3. Fig. 3(c) shows the output in PDF without any correction.

2.3 Refinement of OCR output

Due to the limitations of OCR, the output of OCR is not fully correct in the paradigm of chemical equation/formula. Out of 3406 chemical compounds in our dataset, the accuracy of OCR conversion is only 43.13%. Hence, refinement of the recognized chemical formula in the equations is an absolute necessity.

The output of closing operation is taken as input here. Each connected component i.e. character within a word blob is an input to the the OCR and the corresponding output is stored in a cell and these cells form a string (S$_{chemical}$) for each word blob. For each superscript and subscript, '^' and '_' are inserted before them respectively in S$_{chemical}$.



(*a*) **Preparation of Amorphous Silicon.** This is the common form of silicon and may be obtained by the following methods :

(*i*) **By heating silica or well powdered quartz with magnesium powder in a fire clay crucible.**

$SiO_2 + 2Mg \longrightarrow Si + 2MgO$

The product is washed first with dilute hydrochloric acid to remove magnesium, unchaged magnesium and magnesium silicide invariably formed in the reduction of silica by magnesium.

$SiO_2 + 4Mg \longrightarrow Mg_2Si + 2MgO$

It is now washed with hydrofluoric acid to remove any unchaged silica, when silicon is left behind as a grey amorphous powder.

(*ii*) **By heating fluosilicates or silicon tetrafluorides with alkali metals.**

$K_2SiF_6 + 4K \longrightarrow 6KF + Si$

$SiF_4 + 4K \longrightarrow 4KF + Si$

(*iii*) **By heating a mixture of silica and powdered coke or charcoal in a current of dry chlorine.**

$SiO_2 + 2C + 2Cl_2 \longrightarrow SiCl_4 + 2CO \uparrow$

The vapours of SiCl$_4$ are then passed over an alkali metal (Na or K).

$SiCl_4 + 4Na \longrightarrow 4NaCl + Si$

(a)

$SiO_2 + 2Mg \longrightarrow Si + 2MgO$

$SiO_2 + 4Mg \longrightarrow Mg_2Si + 2MgO$

$K_2SiF_6 + 4K \longrightarrow 6KF + Si$

$SiF_4 + 4K \longrightarrow 4KF + Si$

$SiO_2 + 2C + 2Cl_2 \longrightarrow SiCl_4 + 2CO \uparrow$

$SiCl_4 + 4Na \longrightarrow 4NaCl + Si$

(b)

$$S_{102} + 2\,Mg \longrightarrow S_1 + 2\,MgO$$
$$S_{102} + 4\,Mg \longrightarrow Mg_2S_1 + 2\,MgO$$
$$K_2S_1F_6 + 4\,K \longrightarrow 6\,KF + Si$$
$$SnF_4 + 4\,K \longrightarrow 4\,KF + Sl$$
$$S_{102} + 2\,c + 2\,Cl_2 \longrightarrow 5\,lCl_4 + 2\,c_0\uparrow$$
$$SICI_4 + 4\,Na \longrightarrow 4\,NaCl + S_1$$

(c)

Figure 3: Experimental result; (a) Input image; (b) Segmented Chemical equation; (c) OCR output in PDF format.

First, an error table is created based on the observation of OCR outputs of 280 chemical equations consisting of 1022 compounds (Table 2). Table 2 consists of two columns; first column is actual input to the OCR and the second column contains all erroneous OCR outputs. Next, this table is stored into a hash map H where each key is the erroneous OCR output and its value is the possible input set. Table 3 shows this hash map where the first column is the key and the second column contains its corresponding values. For example, if '8' is an erroneous OCR output for inputs 'g', '3' and 'a' (Table 2) then, in the hash map H (Table 3) , the key is '8' and its corresponding value is [g, 3, a] .

Each chemical compound in any equation (Fig. 4) has the following format -

Table 2: Part of the Error list

correct input	corresponding erroneous OCR outputs
g	8 3 S
O	0
3	8 'E s w
a	3. 21 8 El 8.
l	1 I
s	S
n	l1 1'1 I1 11 X1 Il
q	Cl Q
H	1-l 1-1 l-l l-1
2	7 4 Z z
I	l
u	11 U l1 ll 1l
i	1 l I

Table 3: Part of the Error Hash Map

Erroneous OCR Output	Possible Input Set
8	g 3 a
3	g
S	g s
0	O
'E	3
s	3
w	3
3.	a
21	a
El	a
8.	a
1	l i
I	l i
l1	n u
1'1	n
I1	n
11	n u
X1	n
Il	n
Cl	q

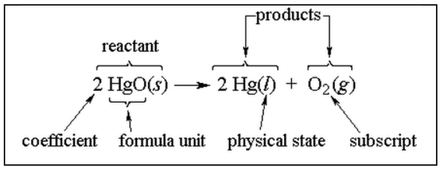


Figure 4: Different components of a chemical equation.

[Coefficient]$^{[0,1]}$[Formula Unit][State]$^{[0,1]}$.

This represents that each chemical compound in an equation may or may not start with a numeric Coefficient, must be followed by a *Formula Unit* and may or may not end with a *State* representation. Hence, refinement or auto correction of the OCR output corresponding to each word blob includes the following steps - (i) Coefficient extraction; (ii) State separation; (iii)Auto correction of the formula unit; and (iv) Auto correction of the entire equation using Context Table.

2.3.1 Coefficient Extraction

Numeric coefficient denotes the number of molecules/atoms taking part in the reaction. Coefficient extraction is done by matching its regular expression $[2-9]^{+}[0-9]^{*}$ at the beginning of S$_{chemical}$ as it has numerical values. In the regular expression, + indicates number of occurrence must be 1 or greater and * indicates the occurrence is 0 times or greater. 0 and 1 are excluded from the first digit as number of molecules or atoms cannot be 0 and if the number is 1, the numeric coefficient is not mentioned by default. Matched coefficients are stored in S$_{coeff}$.

2.3.2 State separation

The four physical states of a chemical compound - solid, gaseous, liquid, and aqueous are denoted by '(s)', '(g)', '(l)', and '(aq)', respectively. To detect the physical state of the compound, regular expression [(][A-Za-z0-9]$^{+}$[)] is used and the checking starts from the end of S$_{chemical}$. The matched substring, S is extracted from S$_{chemical}$ and Algorithm 1 is run. As mentioned earlier, OCR output for each character is stored in a cell of S. In this algorithm, S (after removing first and last character - opening and closing brackets) and H are taken as inputs and all possible *Combinations* of OCR output is produced by *GetAllCombinations* (See Fig. 5). For each cell element in S, corresponding values from hash map, H is assigned to a set, *InputSet* (See Line 3 in Algorithm 1). This set contains all possible inputs to the OCR system. Now, the key itself is added with its corresponding values in the hash table to make the *InputSet* if the length of key is 1. For example, '8' is added to the *InputSet* as its length is 1 (Fig. 5). On the contrary, in the second *InputSet*, 'Cl' is not included as its length is 2 (Fig. 5).

Each cell element of S gives one *InputSet*. Now, cartesian product of all the *InputSet* is taken to give us all possible *Combinations* and only one of the combinations is correct under proper chemical context. These *Combinations* are compared with 's', 'g', 'l' and 'aq'. If no match is found, the substring extracted from S$_{chemical}$ is concluded as a radical (Fig. 6, the compound contains a radical having the same regular expression mentioned earlier) , not a state; else, we separate the state from the compound and store it in S$_{state}$ (after adding '(' and ')' at the start and end of S).

Algorithm 1 Attempts to find out all possible combinations of the initial error corrected OCR converted texts

```
1: procedure GetAllCombinations(S,H)
2:     For each element in S, corresponding values from H is
       assigned to a set, InputSet
3:     If the InputSet is null, this indicates that the OCR output
       is not in the Hash map
4:     Since each element is OCR output of one letter, its ideal
       text length should be 1. Anything more than that indicates
       error
5:     Cartesian product of all letters in InputSet is taken and
       stored into an 2D array, Combinations
6: end procedure
```

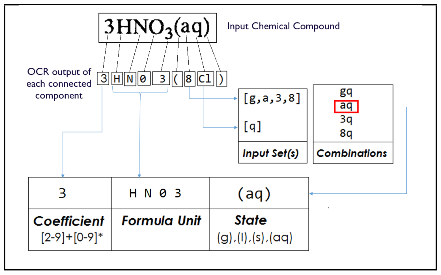


Figure 5: Extracting the formula unit, numeric coefficient and physical state from a chemical compound.

2.3.3 Refinement of the formula unit

After extracting coefficient and state, only the formula unit is left in S$_{chemical}$. The algorithm for auto correction of each formula unit is done in two steps using Algorithm 1 and Algorithm 2. First, Algorithm 1 is performed on the formula unit to get all possible combinations. Next, the output

Figure 6: Example of chemical compound having a radical in the end.

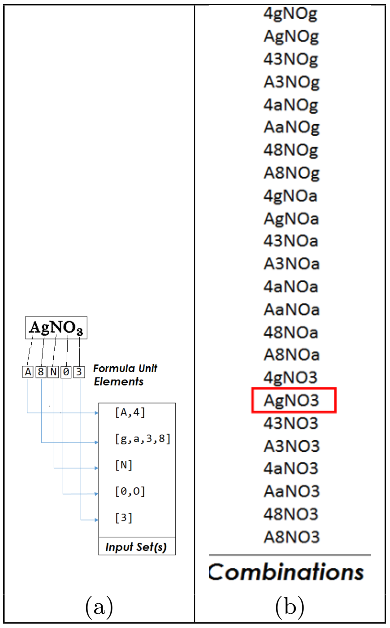


Figure 7: Example of auto correction of a formula unit.

of Algorithm 1(*Combinations*) is taken as the input of Algorithm 2. This algorithm is used to match the *Combinations* against a nearly exhaustive list of all molecules, chemical compounds, radicals and atoms namely *ChemList* collected from Wikipedia. [1].

Three cases may arise as follows–

(i) Exactly one match –
Fig. 7(b) shows the output of Algorithm 1. This is matched against *ChemList* using Algorithm 2 and algorithm finds one exact match as indicated by the red rectangle. This match is considered as the *Corrected* formula unit.

(ii) No match –
Longest common substring(s) (LCS) between *Combinations* and *ChemList* is computed and the formula unit in *Chemlist* having the longest common substring with *Combinations* is considered as *SubMatch*. There can be multiple such *SubMatch*. If there is only one, then the corresponding formula unit in *ChemList* is considered as the *Corrected* formula unit; else the *SubMatch*es having the same length as that of the *Combinations* are considered as *PossibleFormulaUnit*s. Fig. 8 (a) is a sample chemical compound. The OCR converted string is 'N 21 B l'. Algorithm 1 returns 'NaBI' and 'NaBl' as the two combinations. None of them match with any chemical compound in *ChemList*. Hence, LCS is computed between these two possible combinations and *ChemList*. Six compounds with LCS length 3 is found as shown in Fig. 8(b). Since the number of *SubMatch*es is six, the *SubMatch* having the closest length as that of *Combination* (i.e. 4) is considered as *PossibleFormulaUnit*. In this case, it is 'NaBr'.

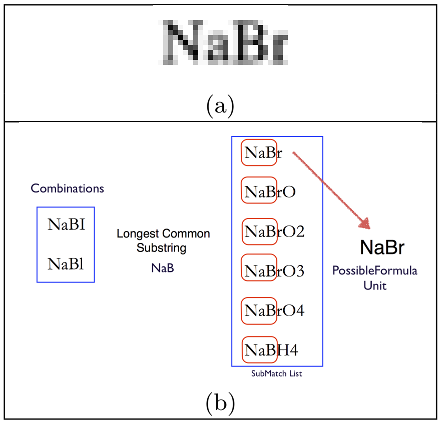


Figure 8: (a) Sample Chemical Compound; (b) LCS Match.

(iii) More than one exact match –
In Fig. 9, 'u' of 'Cu' in left hand side of the equation is 'l1' as the output of OCR. Among all possible combinations returned by Algorithm 1, 'Cu' and 'Cn' both match with *ChemList*. Hence, more than one exact match are found and both are considered as *PossibleFormulaUnit*.

The above steps are precisely mentioned in Algorithm 2. This algorithm returns *Corrected* and *PossibleFormulaUnit*s upon which context analysis is done and is discussed in the next section.

Algorithm 2 Find Match between ChemList and Combinations derived from Algorithm 1

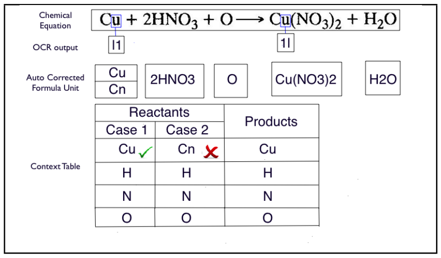
```
procedure FindMatch(ChemList,Combinations)
    Each combination is looked up against ChemList to find a match. Depending on the type of match, different steps are taken
    If it's an exact match, that combination is considered as Corrected compound
    If it's not an exact match, longest common substring of that combination and ChemList is taken. If there are multiple longest common substrings, all of them are considered for next steps
    if The longest common substring is a match with ChemList
then
        This is considered as Corrected compound
    else
        The longest common substring(s) is(are) stored as PossibleFormulaUnit(s)
    end if
end procedure
```

2.3.4 *Auto correction of the entire equation using Context Table*

Here, we have *Corrected* and *PossibleFormulaUnit*(s) and try to find out the *FinalEquation* in the context of the equation itself. If a chemical equation does not have any *PossibleFormulaUnit*, context analysis is not required. The process exits after performing Line 2 of Algorithm 3.

Algorithm 3 takes all *Corrected* and *PossibleFormulaUnit*s and returns the *FinalEquation* by

[1] http://en.wikipedia.org/wiki/Dictionary_of_chemical_formulas


Chemical Equation
$Cu + 2HNO_3 + O \longrightarrow Cu(NO_3)_2 + H_2O$
OCR output
l1
1l
Auto Corrected Formula Unit
Cu
Cn
2HNO3
O
Cu(NO3)2
H2O
Context Table
✓
✗

Reactants		Products
Case 1	Case 2	
Cu	Cn	Cu
H	H	H
N	N	N
O	O	O

Figure 9: Formation of context table.

forming the Context Table. As the universe is a closed system, all chemical equations have the same periodic elements in the left hand side, called *Reactants* as that in the right hand side, called *Products*. All the periodic elements follow the regular expression [A-Z][a-z]*. So, for each $PossibleFormulaUnit$, the set of periodic elements in the $Reactants$, P_R and in the $Products$, P_P are computed and stored in the $ContextTable$. When the set difference of P_R and P_P in the table is empty, that $PossibleFormulaUnit$ is considered as *Corrected* (Fig. 9 Case 1). In the Case 1 of P_P, the empty set condition satisfies. Hence, 'Cu' will be the *Corrected* formula unit, not 'Cn'. But if the above condition comes true for multiple possibilities, we cannot decide which of the possible formula units are actually in the original equation. This is considered an $ERROR$ case. Finally, S_{coeff} and S_{state} (if any) are added with their corresponding *Corrected* formula unit after the context analysis and this results in $FinalCompounds$ for each equation.

Now according to the stoichiometry of the chemical reaction, pre-recognised operators along with $FinalCompounds$ are concatenated together. This gives us the final auto-corrected chemical equation.

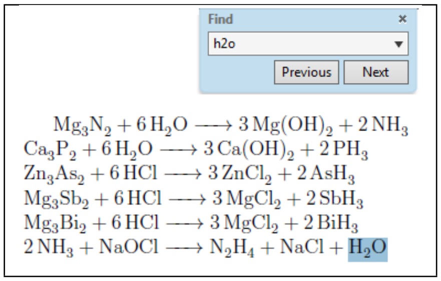
Algorithm 3 Auto-Correction of the entire equation using chemical context table

```
procedure GetFinalEqn($PossibleFormulaUnit$,$Corrected$)
    All PossibleFormulaUnits and Corrected compounds from Algorithm 2 are taken as input and all corrected compounds are directly placed in the equation
    Now for every PossibleFormulaUnits in the Reactants side, set of periodic elements are computed (Pr)
    Similarly for every PossibleFormulaUnits in the Products side, set of periodic elements are computed (Pp)
    If Pr and Pp are complete match, then that PossibleFormulaUnit is taken as Corrected and placed into the final equation
    Now finally with every corrected compound corresponding coefficient and state is added before and appended respectively
    If there are multiple Corrected compounds for one PossibleFormulaUnit then this algorithm fails and shows all possible corrected final equations
end procedure
```

2.4 Generation of chemical equation in searchable PDF format

The final auto-corrected chemical equation is then converted to LaTeX using the format specified by *mhchem* [2] package which provides commands for typesetting chemical molecular formulae and equations. This produces the searchable PDF format.

Documents like research papers or patents often have an abundance of chemical equations. To search for any particular reaction or compound would involve ploughing through the entire document manually. The proposed system can significantly and efficently reduce this effort. By providing the scanned document as input, the output would be a searchable list of all chemical equations present in the entire document. Users can search the PDF by case insensitive linear string queries as shown in Fig. 10. Each chemical element or compound present in the PDF can be searched with complete or partial search string. For example, to identify all chemical compounds containg zinc, user just have to search the generated PDF with "zn"; or to identify all chemical equations involving water, user have to search with "h2o".



$$Mg_3N_2 + 6\,H_2O \longrightarrow 3\,Mg(OH)_2 + 2\,NH_3$$
$$Ca_3P_2 + 6\,H_2O \longrightarrow 3\,Ca(OH)_2 + 2\,PH_3$$
$$Zn_3As_2 + 6\,HCl \longrightarrow 3\,ZnCl_2 + 2\,AsH_3$$
$$Mg_3Sb_2 + 6\,HCl \longrightarrow 3\,MgCl_2 + 2\,SbH_3$$
$$Mg_3Bi_2 + 6\,HCl \longrightarrow 3\,MgCl_2 + 2\,BiH_3$$
$$2\,NH_3 + NaOCl \longrightarrow N_2H_4 + NaCl + H_2O$$

Figure 10: Searchability of generated PDF

3. EXPERIMENTAL RESULT

We have implemented our algorithm in MATLAB 8.3.0.532 (R2014a) in a PC (Intel(R) Core(TM) i5-3337U CPU @ 1.80GHz running Windows 8). The proposed method has been tested on a dataset consisting of 234 document images. Out of 234 pages 50 are taken from ICDAR 2013 Mathzone segmentation datasets and other document pages are scanned from different Mathematics and Chemistry books. The summary of the experimental results is shown in Table 4. Out of 3406 chemical formula in the test dataset, 114 formula were partially corrected and 52 formula could not be corrected at all. The overall accuracy of complete refinement is 95.12%. This is measured by (#Completely Corrected formula / #Total number of formula). Due to the longest common substring match and then performing context analysis of the entire equation, there is a very small window of zero correction. Zero correction is the case when there have been no correction to the OCR output by the auto-correction algorithm. For example, OCR output of 'Mg'- I^I3 could not be corrected by our auto-correction algorithm as this erroneous conversion was not in the error hash map.

With our dataset, zero correction rate is 0.01% (It is computed as #Zero Correction Compound / #Total Compounds).These results are quite encouraging.

Consider the sample image (Fig. 3(a)) and its corresponding segmented displayed chemical equations are shown in Fig. 3(b). Fig. 3(c) shows the direct OCR output where 'i' has been wrongly identified as 'l', 'I' and '1' (for Si in all the lines of Fig. 3(c)). Similarly 'O' results in '0' (line 1,2,3). 'S' sometimes is detected as '5' (line 5). Our auto correction algorithm remedies these issues. Fig. 11 demonstrates the

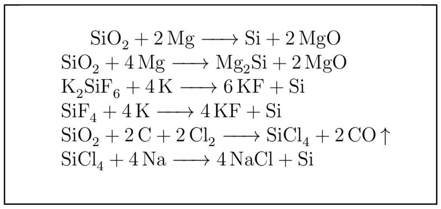
[2] ftp://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/mhchem.pdf

Table 4: Summary of Experimental Results

#Total Images	234
#Total DEs	1390
Operator recognition	99.8%
DE segmentation accuracy	98.63%
Chemical DE Classification Accuracy	98.83%
#Total Chemical Operands	3406
Complete refinement accuracy	95.12%
Zero Auto correction rate	0.01%

effect of our auto correction algorithm. This algorithm is targeted towards chemical equation with linear representation. Organic bonds cannot be detected in this system.

Some sample experimental results are shown in Fig. ??, Fig. ??, Fig. 14 and Fig. ??. More results are shown in https://sites.google.com/site/chemeqndb/home.



$$SiO_2 + 2\,Mg \longrightarrow Si + 2\,MgO$$
$$SiO_2 + 4\,Mg \longrightarrow Mg_2Si + 2\,MgO$$
$$K_2SiF_6 + 4\,K \longrightarrow 6\,KF + Si$$
$$SiF_4 + 4\,K \longrightarrow 4\,KF + Si$$
$$SiO_2 + 2\,C + 2\,Cl_2 \longrightarrow SiCl_4 + 2\,CO\uparrow$$
$$SiCl_4 + 4\,Na \longrightarrow 4\,NaCl + Si$$

Figure 11: Auto corrected output of Fig. 3(c).

Next, we try to analyse the sources of some of the errors and shortcomings of our algorithm which have negative effect on the performance figures for auto correction.

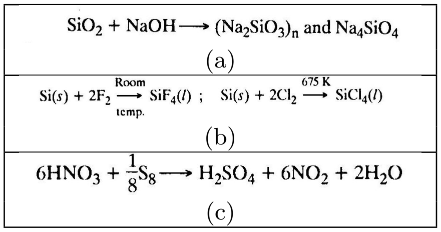
Case 1 : Chemical equations sometimes contain some texts such as '*and*', '*or*' etc between two chemical compounds (See Fig. 12(a)). Sometimes two chemical equations are conjuncted by these words in the same line. If these words are not in chemical context and OCR does not convert them correctly, our autocorrection algorithm cannot match them against *ChemList*, hence the error occurs. But OCR conversion has a high accuracy rate for such type of texts. Therefore, this is not a severe error.

Case 2 : When the chemical compound is written in formats such as $(Na_2SiO_3)_n$ (Fig. 12(a)), only Na_2SiO_3 is detected based on *ChemList* and LCS matching. This is considered as a partial autocorrection case.

Case 3 : Some equations have conditions (pressure, temperature) written over the arrows (See Fig. 12(b)). In this work, we only concentrated on chemical compounds in the equation. This does not effect the autocorrection accuracy rate as most of the time they get segmented in separate text lines; else we ignore the over arrow conditions beforehand.

Case 4 : Fractions in the numeric coefficients (Fig. 12(c)) are not dealt with in our autocorrection algorithm as they are not very common in chemical equations.

Case 5 : Here, in the equation shown in Fig. 13. both the 'g's in reactant and product side have been converted to 'S' by the OCR which results in multiple auto-corrected formula units on both side. At this point, we reach Step 17 of Algorithm 3 where context table formation cannot conclude which one is the final corrected compound. However, normally the probability of occurrence of such situation is extremely rare, so no further steps are taken to rectify this.



$$SiO_2 + NaOH \longrightarrow (Na_2SiO_3)_n \text{ and } Na_4SiO_4$$

(a)

$$Si(s) + 2F_2 \xrightarrow[\text{temp.}]{\text{Room}} SiF_4(l)\ ;\quad Si(s) + 2Cl_2 \xrightarrow{675\text{ K}} SiCl_4(l)$$

(b)

$$6HNO_3 + \frac{1}{8}S_8 \longrightarrow H_2SO_4 + 6NO_2 + 2H_2O$$

(c)

Figure 12: Sample error cases (a) presence of non-chemical words in segmented chemical equation; (b) presence of over arrow conditions; (c) fractional coefficient.

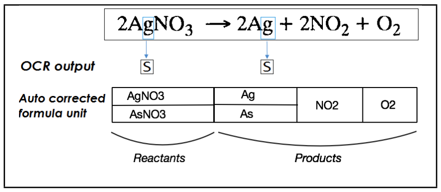


Figure 13: Error case of multiple *Corrected* compounds

4. CONCLUSIONS

We have presented an automated chemical equation segmentation and chemical context based auto correction system that is able to provide the exact searchable format of linear chemical equations in any document image. The experimental results demonstrate the efficiency of our proposed method. One of the drawbacks of our system is the time complexity as the search space in the *ChemList* is quite big and is growing over time due to discovery of new compounds. The search method can be improved and made more efficient. Since our proposed method is novel, we have not concentrated on making the system time efficient yet but more on the accuracy of the auto correction. This work leads to several research avenues. Chemical context horizon can be widened. Auto correction on non-linear or bond structure representations of chemical equations could be ventured in.

5. REFERENCES

[1] D. Blostein and A. Grabavec. *"Recognition of Mathematical Notation", Handbook of Character Recognition and document Image Analysis*, 557–582, 1997.

[2] K-F. Chan and D-Y. Yeung. *"Mathematical Expression Recognition: A Survey". IJDAR, Vol. 3, no, 1*, 3–15, 2000.

[3] U. Garain and B. B. Chaudhuri. *On OCR of Printed mathematical Expressions. "Digital Document Processing",Ed. B. B. Chaudhuri, Advances in pattern Recognition*, 235–259 , 2007.

[4] R. Fateman, T. Tokuyasu, B. Berman, and N. Mitchell. *"Optical character recognition and parsing of typeset mathematics", Visual Commun. And Image Representation, Vol 7, no 1*, 2–15, 1996.

[5] J. Y. Toumit, S. Garcia-Salicetti, and H. Emptoz. *"A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents". In Proc. of ICDAR*, 116–122, 1999.

[6] A. Kacem, A. Beliad and M. Ben Ahmed. *"Automated Extraction of printed mathematical formulas using fuzzy*

logic and propagation of context", IJDAR, vol.4 no. 2, 97–108, 2001.

[7] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida and T. Kanahori. *"INFTY - An Integrated OCR system for Mathematical Documents", Proc. of ACM Symposium on Document Engineering*, 95–104, 2003.

[8] Utpal Garain. *" Identification of Mathematical Expressions in Document Images", Proc. of ICDAR*, 1340–1344, 2009.

[9] Utpal Garain. *"Recognition of Printed Handwritten Mathematical Expressions", Ph.D Thesis, ISI, Kolkata, India*, 2005.

[10] B. B. Chaudhuri and U. Garain. *"Extraction of type atyle based meta-information from Imaged documents", IJDAR, vol. 3 no. 3*, 138–149, 2001.

[11] J. Jin, X. Han and Q. Wang. *"Mathematical formulas extraction", Proc. of ICDAR*, 1138–1141, 2003.

[12] D. M. Drake and H. S. Baird. *"Distinguishing mathematical notation from english text using computational geometry", Proc. of ICDAR*, 1270–1274, 2005.

[13] Y.-S. Guo, L. Huang and C.-P. Liu. *"A new approach for understanding of structure of printed mathematical expressions", Proc. of ICMLC*, 2633–2638, 2007.

[14] We-Te Chu and Fan liu. *"Mathematical formula detection from heterogeneous document Images", Proc. of CTAAI*, 140–146, 2013.

[15] S. P. Chowdhury, S. Mandal, A. K. Das, and B. Chanda. *"Segmentation of Text and Graphics from Document Images"*, in *Proc. of ICDAR*, 619–623, 2007.

[16] S. Mandal, S. P. Chowdhury, A. K. Das, and B. Chanda, *A simple and effective table detection system from Document Images*, IJDAR, Vol. 8(2), 172âĂŞ182, 2006.

[17] S. P. Chowdhury, S. Mandal, A. K. Das, and B. Chanda, *Segmentation of Text and Graphics from Document Images*, In Proc. of ICDAR, pp. 619âĂŞ623,2007.

[18] R. C. Gonzalez and R. Wood, *Digital Image Processing*, Addision-Wesley, 1992.

[19] D. Blostein and A. Grabavec, *Recognition of Mathematical Notation*, Handbook of Character Recognition and document Image Analysis, 577–582, 1997.

[20] K-F. Chan and D-Y. Yeung, *Mathematical Expression Recognition: A Survey*, IJDAR, Vol. 3, no: 1, 3âĂŞ15, 2000.

[21] U. Garain and B. B. Chaudhuri *An OCR of Printed mathematical Expressions, Digital Document Processing*, Ed: B. B. Chaudhuri, Advances in pattern Recognition, 235âĂŞ259 , 2007.

[22] A. Fujiyoshi, M. Suzuki, S. Uchid, *Grammatical Verification for Mathematical Formula Recognition Based on Context-Free Tree Grammar*, Mathematics in Computer Science, 279–298, 2010.

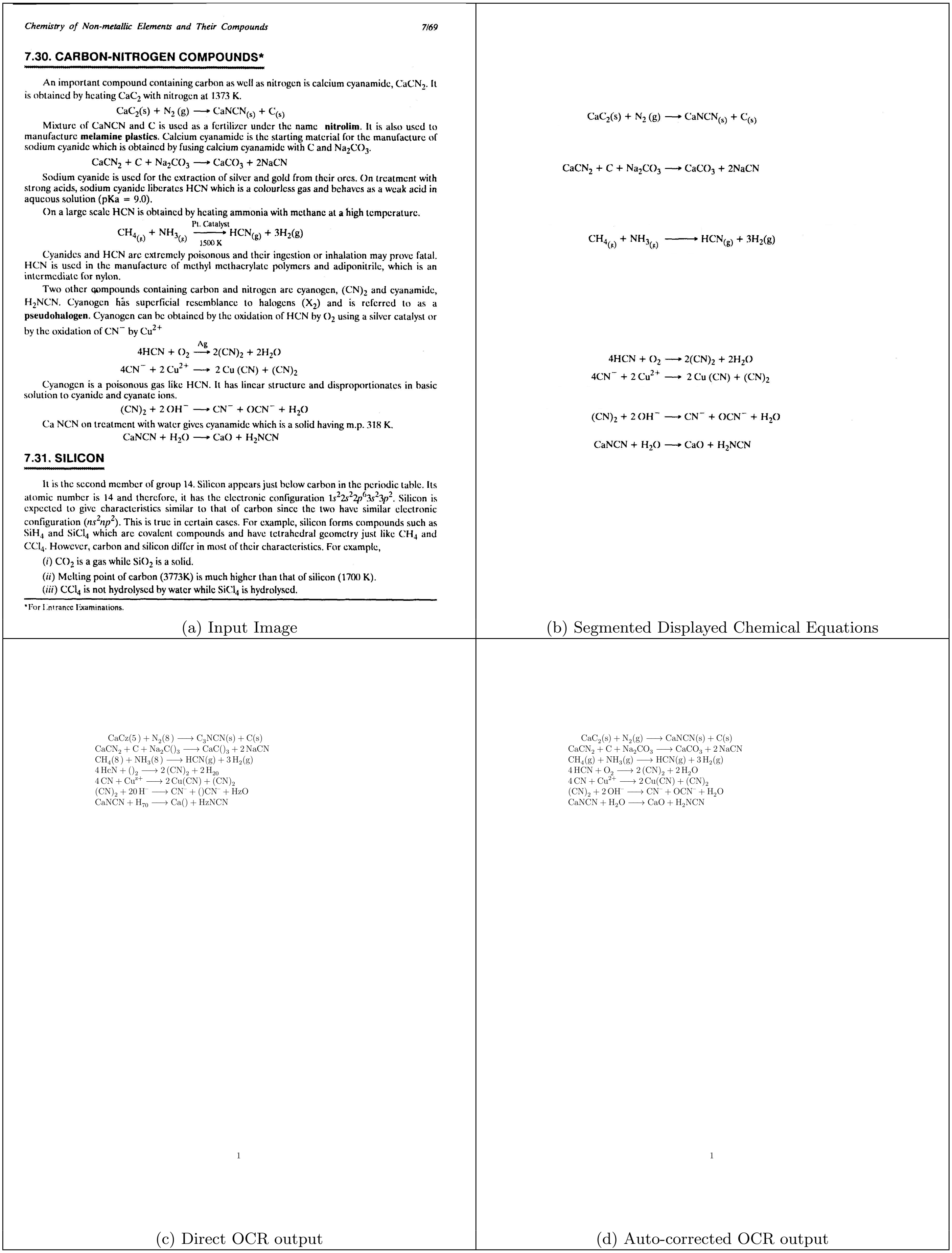
[23] M. E. Algorri, M. Zimmermann, C. M. Friedrich, S. Akle, and M. Hofmann-Apitius, *Reconstruction of chemical molecules from images*, In Proc. 29th Annual International IEEE Conference on Engineering in Medicine and Biology Society, 4609–4612, 2007.

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

[25] J. Park, G. R. Rosania, K. A. Shedden, M. Nguyen, N. Lyu, and K. Saitou, *Automated extraction of chemical structure information from digital raster images*, Chemistry Central journal, vol. 3(1), 2009.

[26] A. K. Jain, J, Mao, K. M. Mohiuddin *Artificial Neural Network: A Tutorial*, Computer (Volume:29 , Issue: 3), Mar 1996.

[27] P. Jana and A. Majumdar *"Automated Segmentation and Classification of Chemical and other Equations from Document Images", 8th International Conference on Advances in Pattern Recognition*, 127-129, 2015.



Chemistry of Non-metallic Elements and Their Compounds *7/69*

7.30. CARBON-NITROGEN COMPOUNDS*

An important compound containing carbon as well as nitrogen is calcium cyanamide, $CaCN_2$. It is obtained by heating CaC_2 with nitrogen at 1373 K.

$$CaC_2(s) + N_2\,(g) \longrightarrow CaNCN_{(s)} + C_{(s)}$$

Mixture of CaNCN and C is used as a fertilizer under the name **nitrolim**. It is also used to manufacture **melamine plastics**. Calcium cyanamide is the starting material for the manufacture of sodium cyanide which is obtained by fusing calcium cyanamide with C and Na_2CO_3.

$$CaCN_2 + C + Na_2CO_3 \longrightarrow CaCO_3 + 2NaCN$$

Sodium cyanide is used for the extraction of silver and gold from their ores. On treatment with strong acids, sodium cyanide liberates HCN which is a colourless gas and behaves as a weak acid in aqueous solution (pKa = 9.0).

On a large scale HCN is obtained by heating ammonia with methane at a high temperature.

$$CH_{4_{(g)}} + NH_{3_{(g)}} \xrightarrow[1500\text{ K}]{\text{Pt. Catalyst}} HCN_{(g)} + 3H_2(g)$$

Cyanides and HCN are extremely poisonous and their ingestion or inhalation may prove fatal. HCN is used in the manufacture of methyl methacrylate polymers and adiponitrile, which is an intermediate for nylon.

Two other compounds containing carbon and nitrogen are cyanogen, $(CN)_2$ and cyanamide, H_2NCN. Cyanogen has superficial resemblance to halogens (X_2) and is referred to as a **pseudohalogen**. Cyanogen can be obtained by the oxidation of HCN by O_2 using a silver catalyst or by the oxidation of CN^- by Cu^{2+}

$$4HCN + O_2 \xrightarrow{Ag} 2(CN)_2 + 2H_2O$$

$$4CN^- + 2\,Cu^{2+} \longrightarrow 2\,Cu\,(CN) + (CN)_2$$

Cyanogen is a poisonous gas like HCN. It has linear structure and disproportionates in basic solution to cyanide and cyanate ions.

$$(CN)_2 + 2\,OH^- \longrightarrow CN^- + OCN^- + H_2O$$

Ca NCN on treatment with water gives cyanamide which is a solid having m.p. 318 K.

$$CaNCN + H_2O \longrightarrow CaO + H_2NCN$$

7.31. SILICON

It is the second member of group 14. Silicon appears just below carbon in the periodic table. Its atomic number is 14 and therefore, it has the electronic configuration $1s^22s^22p^63s^23p^2$. Silicon is expected to give characteristics similar to that of carbon since the two have similar electronic configuration (ns^2np^2). This is true in certain cases. For example, silicon forms compounds such as SiH_4 and $SiCl_4$ which are covalent compounds and have tetrahedral geometry just like CH_4 and CCl_4. However, carbon and silicon differ in most of their characteristics. For example,

(*i*) CO_2 is a gas while SiO_2 is a solid.

(*ii*) Melting point of carbon (3773K) is much higher than that of silicon (1700 K).

(*iii*) CCl_4 is not hydrolysed by water while $SiCl_4$ is hydrolysed.

*For Entrance Examinations.

(a) Input Image

$$CaC_2(s) + N_2\,(g) \longrightarrow CaNCN_{(s)} + C_{(s)}$$

$$CaCN_2 + C + Na_2CO_3 \longrightarrow CaCO_3 + 2NaCN$$

$$CH_{4_{(g)}} + NH_{3_{(g)}} \longrightarrow HCN_{(g)} + 3H_2(g)$$

$$4HCN + O_2 \longrightarrow 2(CN)_2 + 2H_2O$$

$$4CN^- + 2\,Cu^{2+} \longrightarrow 2\,Cu\,(CN) + (CN)_2$$

$$(CN)_2 + 2\,OH^- \longrightarrow CN^- + OCN^- + H_2O$$

$$CaNCN + H_2O \longrightarrow CaO + H_2NCN$$

(b) Segmented Displayed Chemical Equations

$$\mathrm{CaCz(5\,) + N_2(8\,) \longrightarrow C_3NCN(s) + C(s)}$$

$$\mathrm{CaCN_2 + C + Na_2C()_3 \longrightarrow CaC()_3 + 2\,NaCN}$$

$$\mathrm{CH_4(8\,) + NH_3(8\,) \longrightarrow HCN(g) + 3\,H_2(g)}$$

$$\mathrm{4\,HcN + ()_2 \longrightarrow 2\,(CN)_2 + 2\,H_{20}}$$

$$\mathrm{4\,CN + Cu^{z+} \longrightarrow 2\,Cu(CN) + (CN)_2}$$

$$\mathrm{(CN)_2 + 20\,H^- \longrightarrow CN^- + ()CN^- + HzO}$$

$$\mathrm{CaNCN + H_{70} \longrightarrow Ca() + HzNCN}$$

1

(c) Direct OCR output

$$\mathrm{CaC_2(s) + N_2(g) \longrightarrow CaNCN(s) + C(s)}$$

$$\mathrm{CaCN_2 + C + Na_2CO_3 \longrightarrow CaCO_3 + 2\,NaCN}$$

$$\mathrm{CH_4(g) + NH_3(g) \longrightarrow HCN(g) + 3\,H_2(g)}$$

$$\mathrm{4\,HCN + O_2 \longrightarrow 2\,(CN)_2 + 2\,H_2O}$$

$$\mathrm{4\,CN + Cu^{2+} \longrightarrow 2\,Cu(CN) + (CN)_2}$$

$$\mathrm{(CN)_2 + 2\,OH^- \longrightarrow CN^- + OCN^- + H_2O}$$

$$\mathrm{CaNCN + H_2O \longrightarrow CaO + H_2NCN}$$

1

(d) Auto-corrected OCR output

Figure 14: Experimental result