Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 6
June 2024

Reformatting Algorithm of Scientific Articles

Sayel M. Fayyad[1*], Hesham Abusaimeh[2], Ruba S. Khasawneh[3] and Rola A. Al-Lataifeh[4]

[1]Faculty of Engineering Technology, Department of Mechanical Engineering, Al-Balqa Applied University-Amman – Jordan.

[2]Computer Science Department-Middle East University-Amman –Jordan, [3,4]Baisc Science Department, Irbid University College- Al-Balqa Applied University-Amman – Jordan

* Corresponding author

Abstract : This study offers a suggested method to enhance the reformatting of papers using the guidelines and templates provided by scientific publications. The suggested methodology is semi-automatic, which reduces time and effort to a significant degree. The manual reformatting method employed by writers and editors requires a great deal of time and work. Every author attempts to submit an article online through the website of a scientific journal; this submission involves a series of steps and requires particular information at each stage as well as in the pre- or post-processing stages. The article must follow the journal's author instructions precisely and in a specific format. Any article conversion into a journal format is a laborious, time-consuming procedure that may not be done accurately. The editors and reviewers of scientific journals have the same issue with reformatting submitted papers. They find it to be a laborious procedure that requires follow-up and tracking to ensure the article is formatted correctly. Thus, the suggested method will address every one of these issues and provide a quick, simple, and very accurate way to reformat papers according to journal format. Here, VB.NET interfaces and utilities are utilized to create an algorithm for reformatting works submitted to journals for publication, complete with stringent formatting guidelines for writers. By decreasing the amount of time required to reformat the articles by 66% and raising the percentage of compatibility between the article and the journal format or template to acceptable rates, which reach a range of 95% to 99%, the suggested system produced positive results when compared to manual or traditional methods.

Keywords : Templates, text, VB.NET, algorithm, reformat, articles, scientific journals, and compatibility.

1. Introduction

Templates are pre-made text formatting tools that allow you to change a given text's style to a necessary or specialized format. When an author wants to change the style of their essay or article to meet publication criteria, it can be a laborious process. The process of reformatting an article before submitting it to a journal can be challenging and time-consuming for many authors. Based on observations and interviews with numerous authors, it takes, on average, 10 to 15 hours to complete this process. The primary goal of this thesis is to provide a ready-made reformatting system that will allow writers, editors, and authors to easily convert their own articles from any format to the format needed by a particular publication.

All of Microsoft Office's apps allow for some text formatting, but it must be changed manually by the user. For instance, the font size should be changed to 12 and the text should be bolded.

These changes take a lot of time, require laborious labor, are tedious, and cause delays in task processing. Such issues will be addressed in this work, and the user's task in the newly proposed system will be to define the settings of the new paper format in the user interface, upload his article in the required format, or select the required format for the headings, titles, figures, tables, citation of references, and abstract. The article will then be formatted according to these required formats. The suggested algorithm will handle the necessary modifications, and appropriate programs such as VB.net will be used to assess and categorize the articles that are received by the journals by calculating the compatibility percentage. Following this, the articles will be sent to be modified (in accordance with the suggested algorithm) based on the format that is closest to the journal's requirements.

Review of Related Literature

A Convolutional Recurrent Generative model was presented by Bhunia et al. (2017) to address the word level font transfer problem. Any printed text

picture may have its font style changed from its original font to the one that was needed by this network system. The whole word pictures were used to train the network end-to-end. As a result, it gets rid of the essential character segmentation pre-processing stages. A conditional setting that aids in learning a one-to-many mapping function was added to the model. This model architecture's ability to be utilized in a generator that effectively handles word pictures of any width was one of its benefits. The first architecture that can handle photos with different widths was the one that was suggested. The novel approach was contrasted with a few of the most advanced picture translation techniques. The system's capacity to learn the typeface distributions demonstrated its performance on the same dataset.

Heraje and Shetty (2017) the goal of this study was to identify and extract written characters from a given input picture. It included identifying and locating the character in a picture. For the prepared English text, optical character recognition is carried out. When a system is first taught for every letter and number in the English language as well as the intended output, the machine learning approach is applied. Lastly, a plot showing the system's accuracy is created based on the output that was gathered.

Taqdir and Uttarhanica (2016) spoke about utilizing handwritten English characters as data, processing the characters to train a neural network algorithm, and then identifying the pattern to convert the characters to a better adaptation. The HCR technology converts images into a format that can be edited. This method alters images in the form of documents by editing, modifying, and storing dates for an extended amount of time. Pre-processing, segmentation, feature extraction, classification, and recognition are some of the steps in this technique. The first documentation system of a high-quality digitization method applied to an early printed book was supplied by Reul et al. (2016). A detailed explanation of the approach, which includes preprocessing, layout analysis, and text recognition, was provided using the 1488 Nuremberg printing of "Der HeiligenLeben." The amount of time needed for each step was noted. Excellent results were obtained from the character recognition operation at both the word (92.19%) and character (97.57%) levels. This study provides realistic estimations for the amount of human labor required to extract entire text from incunabula.

Sakila and Vijayarani (2015) applied the template matching approach to scanned images of documents including both numeric characters and upper- and lowercase letters. The Performance Index approach was applied to compare the template picture with the input image, and the results were compared using the cross correlation and normalized cross correlation methods. In this work, several comparisons were made, including single-character comparisons from words, sentences, and paragraphs as well as multi-character (word) comparisons from words, sentences, and paragraphs.

Bhatia et al. (2013) they believed that a wide variety of handwriting styles from various people should be handled. Because various people's handwriting has varying character shapes, form distortion is a prevalent phenomenon in practical image capture methods and settings. This work's character recognition procedure is broken down into two stages. During the initial stage, Otsu's approach is used to produce a threshold value, which is used to transform the picture into binary form. The median filter was used to eliminate the noise in the next stage. Subsequently, a feature extraction process was carried out, employing the Fourier descriptor approach with the Fourier transform. This yielded a correlation between the template created using training data and the test data. The Back Propagation technique is used to design and train a multilayer feed forward neural network. Testing is done to ensure that the pattern matches test data after training. Analyses and results are provided for several neural network convergence objectives.

A programming framework for processing structured and semi-structured text by example, called STEPS, was introduced by Yessenov et al. (2013). By using examples, STEPS users design and modify hierarchical structures. In a user research conducted among fourteen computer scientists, STEPS outperforms traditional programming in a between-subjects manner. At least for post-docs and graduate students studying computer science, it was discovered that STEPS is quicker than other options. Additionally, it was discovered that STEPS may be enhanced with regard to how it presents the set of potential operations and inferred operations,

both of which have been the subject of earlier research.

A web-based address system application was described by Bencze et al. (2006) that used LATEX template documents to create personalized PDF pages. Users can use a web browser to access the application's functionality, which were housed on a server. Only the server side needed to have the functional LATEX system installed. The address book can be managed by the registered user. Users have the ability to upload LATEX templates and use those templates to create numerous PDF documents. Every chosen recipient had a template tailored to them, with the relevant address book data element replaced. An invitation card or a letter that has to be addressed to several recipients might serve as examples of applications. Additionally, users may utilize a basic web-based document editor and built-in templates to generate simple documents (such letters).

Materials and methods

It is imperative that a new method or system be used to reformat the papers that are submitted to journals. A unique program created by VB.NET application for this purpose will be used to classify the submitted articles under the proposed system. This classification will include figuring out the compatibility percentage; if this percentage is low, medium, or less than 95%, the text needs to be restructured in accordance with the journal guidelines or requirements. If not, it is accepted (but not in its entirety by the journal).

Tools used in simulation

Many tools used in simulation of this study, includes the following:

Oracle Virtual Machine- Virtualbox

Cross-platform virtualization software like Oracle VM Virtual Box gives users the ability to expand their current computer's capability to run various operating systems simultaneously. Oracle VM VirtualBox, which was created for IT professionals and developers, is perfect for testing, developing, showcasing, and deploying solutions across various platforms on a single machine. It runs on a variety of operating systems, including Windows, Mac OS X, Linux, and Oracle Solaris systems. Numerous cutting-edge features of Oracle VM VirtualBox, such notable speed gains, a more potent virtualization system, and a greater range of supported guest operating system platforms, can provide real business advantages.

Python and spyder3

Python is a high-level programming language that is incredibly user-friendly and is swiftly setting the benchmark for helpful tools in scientific computing. Its numerous features include being open source, fully standardized for usage on several operating systems (Windows, Mac OS, and Linux), incredibly adaptable, and simple to use and understand.

Ubuntu

Ubuntu is an open-source and free software-based operating system (OS). The user of Ubuntu may create documents, spreadsheets, and more in addition to reading email and browsing the web. Ubuntu offers users the strength and adaptability they need for work, learning, and personal use. Ubuntu is virus-free, simple to install, and ideal for servers, desktop computers, and laptops.

PyPDF2

A PDF library written entirely in Python called PyPDF2 can divide, combine, crop, and alter the pages of PDF files. Passwords, custom data, and viewing options can also be added to PDF files. It can combine full files together and extract text and metadata from PDFs.

The Proposed System

The present study employs a VB.NET software application interface to facilitate the uploading of manuscripts by users, who can be either writers or editors. The system is designed to verify the article and categorize it based on its closest match to the necessary journal format. Once the compatibility percentage has been verified, the paper may be forwarded to be reformatted according the journal's guidelines.

Simulation Results and Discussion

The results are broken down into three phases. The first step is a code that can transform any PDF file into a collection of words or characters that can be altered and transformed into any other kind of data in a different format.

Using the following code, any PDF file written in any format can be converted to a data that can be edited and processed to be converted to the necessary format after being cleaned up and made

ready to be entered into the last phase of the proposed system, which is to be formatted according to the author's instructions as required by the Journal. Table 1 displays this code.

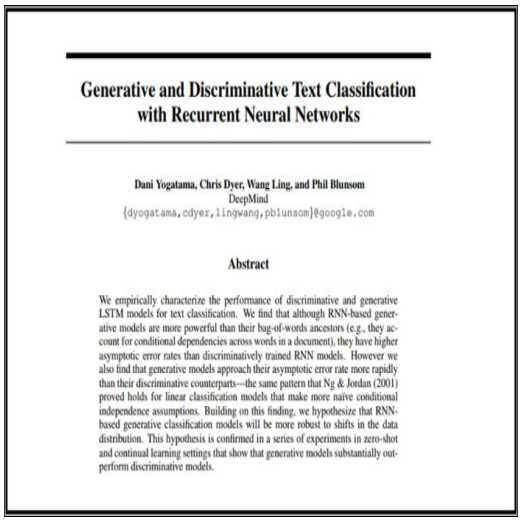
Table 1 The initial code used to convert any PDF file written by any format to a data

```
print('\n')
print('total number of pages are', numOfPages)
#print(extracted.encode('utf-8'))
"
content_file = open('converted_PDF_Text.txt','r').read()
tokenized_file = open('tokenizedFile.text','w')
for tokens in word_tokenize(content_file):
tokenized_file.write(tokens)
tokenized_file.write('\n')
  #print(tokens)
"""
```

```
"""
@author: text_classification
"""
import PyPDF2
fromnltk.tokenize import sent_tokenize, word_tokenize
getFile = input('enter the absolute path of PDF file: ')

text_file = open('converted_PDF_Text.txt','w')
pdfFileObj = open(getFile,'rb')  #'rb' for read binary mode
pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
numOfPages = pdfReader.numPages
for page in range(numOfPages):
pageObj = pdfReader.getPage(page)
 #'9' is the page number
extracted = pageObj.extractText()
print(extracted)
text_file.write('#')
text_file.write('\n')
text_file.write(extracted)
```

The Original PDF-file is shown in Figure 1 below



Generative and Discriminative Text Classification with Recurrent Neural Networks

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom
DeepMind
{dyogatama,cdyer,lingwang,pblunsom}@google.com

Abstract

We empirically characterize the performance of discriminative and generative LSTM models for text classification. We find that although RNN-based generative models are more powerful than their bag-of-words ancestors (e.g., they account for conditional dependencies across words in a document), they have higher asymptotic error rates than discriminatively trained RNN models. However we also find that generative models approach their asymptotic error rate more rapidly than their discriminative counterparts—the same pattern that Ng & Jordan (2001) proved holds for linear classification models that make more naïve conditional independence assumptions. Building on this finding, we hypothesize that RNN-based generative classification models will be more robust to shifts in the data distribution. This hypothesis is confirmed in a series of experiments in zero-shot and continual learning settings that show that generative models substantially outperform discriminative models.

Figure 1. PDF file of the original segment of the paper

Figure 2 shows the Converted text after running the code.

```
1 #
2 GenerativeandDiscriminativeText
3 withRecurrentNeuralNetworks
4 DaniYogatama,ChrisDyer,WangLing,andPhilBlunsom
5 DeepMind
6 f
7 dyogatama,cdyer,lingwang,pblunsom
8 g
9 @google.com
10 Abstract
11 Weempiricallycharacterizetheperformanceofdiscriminativeandgenerative
12 LSTMmodelsfortextWethatalthoughRNN-basedgener-
13 ativemodelsaremorepowerfulthantheirbag-of-wordsancestors(e.g.,theyac-
14 countforconditionaldependenciesacrosswordsinadocument),theyhavehigher
15 asymptoticerrorratesthandiscriminativelytrainedRNNmodels.Howeverwe
16 alsothatgenerativemodelsapproachtheirasymptoticerrorratemorerapidly
17 thantheirdiscriminativecounterpartsSthesamepatternthatNg&Jordan(2001)
18 provedholdsforlinearmodelsthatmakemorena
19 ¨
20 econditional
21 independenceassumptions.BuildingonthiswehypothesizethatRNN-
22 basedgenerativemodelswillbemorerobusttoshiftsinthedata
23 distribution.Thishypothesisisinaseriesofexperimentsinzero-shot
24 andcontinuallearningsettingsthatshowthatgenerativemodelssubstantiallyout-
25 performdiscriminativemodels.
26 1Introduction
27 Neuralnetworkmodelsusedinnaturallanguageprocessingapplicationsareusuallytraineddiscrim-
28 inatively.Thisstrategysucceedsformanyapplicationswhentrainingdataisabundantandthedata
29 distributionisstable.Unfortunately,neuralnetworksrequirealotoftrainingdata,andtheytendto
30 generalizepoorlywhenthedatadistributionshifts(e.g.,newlabels,newdomains,newtasks).In
31 thispaper,weexploreusinggenerativemodelstoobtainimprovementsinsamplecomplexityand
32 abilitytoadapttoshiftingdatadistributions.
33 Whileneuralnetworksaretraditionallyusedasdiscriminativemodels(Ney,1995;Rubinstein&
34 Hastie,1997),theirxibilitymakesthemwellsuitedtoestimatingclasspriorsandclass-conditional
35 observationlikelihoods.WefocusonasimpleNLPtaskStextdiscriminative
36 andgenerativevariantmodelsbasedonacommonneuralnetworkarchitecture(
37 x
38 2).Thesemodels
39 useanLSTM(Hochreiter&Schmidhuber,1997)toprocessdocumentsassequencesofwords.In
40 thegenerativemodel,documentsaregeneratedwordbyword,conditionedonalearnedclassembed-
41 ding;inthediscriminativemodeltheLSTMfireadsfthedocumentandusesitshiddenrepresentation
```

Figure 2 after running code 1

Code for cleaning data

This code will automatically remove any unnecessary words or numbers from texts so that they may be categorized into one of three text classifications (good, terrible, or medium) based on composition percentage. The cleaning code is displayed in Table 2.

Table 2. The cleaning code

```
# -*- coding: utf-8 -*-
#import nltk
import pandas as pd
df = pd.read_csv('text_data.csv')
df.head()
#with open('text_data.csv') as td:
#    for rows in td:
#        print(rows)

X = df.iloc[:,0] #all enteries of col_0
y = df.iloc[:,1] #all enteries of col_1

############### Tfidfvectorinzation is "term frequency–inverse document frequency"
from sklearn.feature_extraction.text import TfidfVectorizer
#tf = TfidfVectorizer(tokenizer=lambda doc: doc,lowercase=False, analyzer='word', min_df = 0, stop_words = 'english')
tf = TfidfVectorizer()

#X is a list having my **Text** which im reading from a CSV
tfidf_matrix =  tf.fit_transform(X)
feature_names = tf.get_feature_names()
print(tfidf_matrix.todense())
# generate the training and testing variables from the csv and pass it to the model
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(tfidf_matrix, y, test_size=0.33)
#X_train, X_test, y_train, y_test = train_test_split(tfidf_matrix, y, test_size=0.33, random_state=42)

#intiating the value neighbors to 1 (maximum no. of neighbors)
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=1)
neigh.fit(X_train, y_train) #fitting the training set to KNN classifier
y_preds = neigh.predict(X_test)

# computing the accuracy score of the KNN classifier with given dataset
from sklearn.metrics import accuracy_score
from sklearn import metrics
# try K=1 through K=25 and record testing accuracy
#this will replicate in the given graph
k_range = range(1, 300)

# We can create Python dictionary using [] or dict(). So that a well-defined
#traning and testing set will be creating and stored for the graph
scores = [ ]
# We use a loop through the range 1 to 30 (could be changes according to the user discreation)
# We append the scores in the dictionary
for k in k_range:
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
scores.append(metrics.accuracy_score(y_test, y_pred)) #append the scores dictionary
print('scores are', scores)
# import Matplotlib (scientific plotting library) used for polting the graph
import matplotlib.pyplot as plt
# plot the relationship between K and testing accuracy
# plt.plot(x_axis, y_axis)
plt.plot(k_range, scores)
plt.xlabel('Value of K for KNN')
plt.ylabel('Testing Accuracy')
```

to the specified format, is displayed in Figure 3.

The outcome of executing the last code, which depends on KNN, or K-nearest neighbor, to determine whether or not the input text is closest

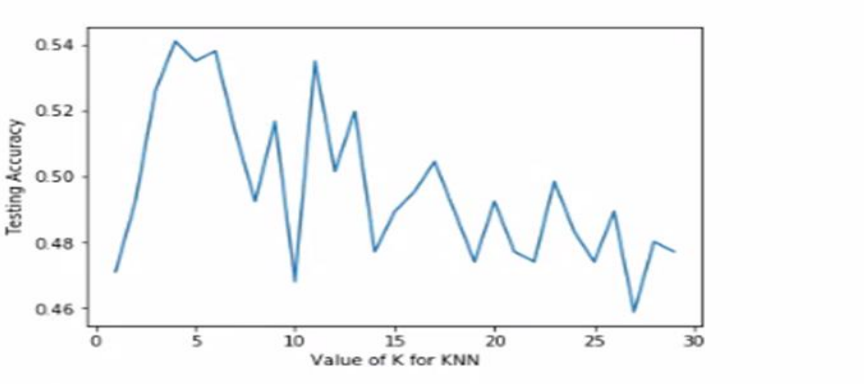


Figure 3 Text testing accuracy depends on KNN

VB.NET Simulation interface

The VB.NET interface for creating the simulation and calculating the comp.% of the papers is displayed in Figure 4.

Figure 5 VB.NET interface

Using the left-hand boxes labeled "Format type" in the simulation interface, the author or editor may indicate the format requirements for the journal template. They can then input text to be formatted based on the kind of document (article title, author name, abstract, introduction, etc.). The author can then pick the kind and choose the type after that. At this point, the cumulative percentage of compatibility will be automatically determined once the comp.% for each section of the document has been calculated. These procedures are shown in Figure 5.

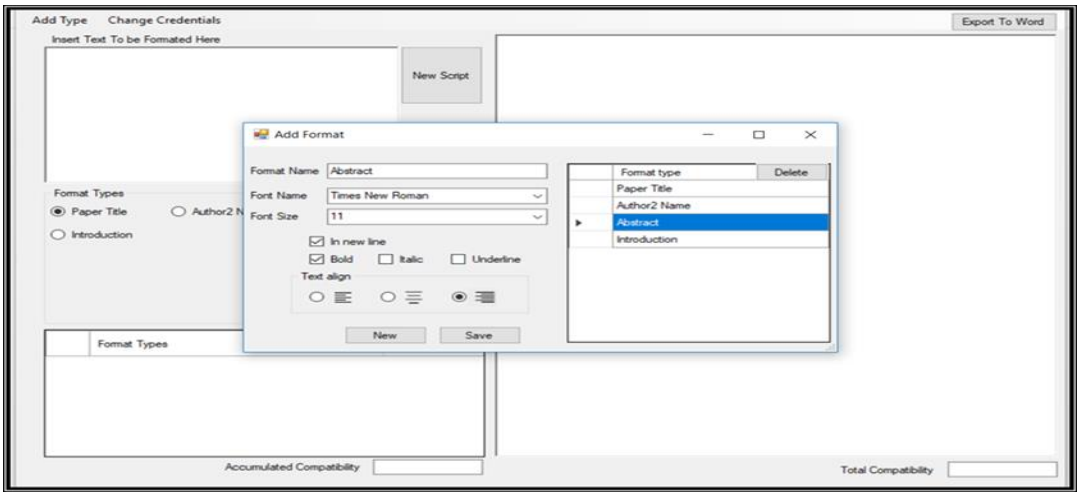


Figure 5 Specifying the format for each section of the document

A text uploaded to the system and its compatibility percentage are displayed in Figure 6.

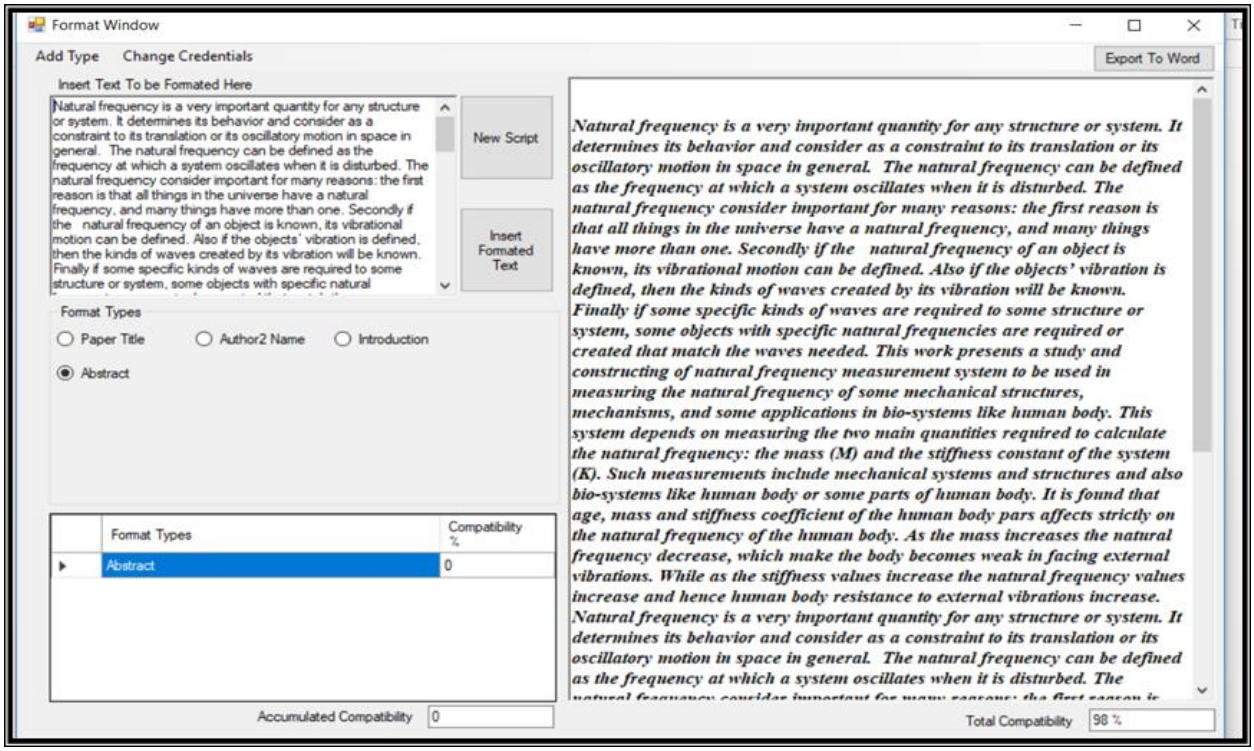


Figure 6. Inserting text using comp.% before and after, as well as formatting the test

Additionally, if necessary, the simulation may convert the formatted text output into a Word document. The word document is seen in Figure 7.

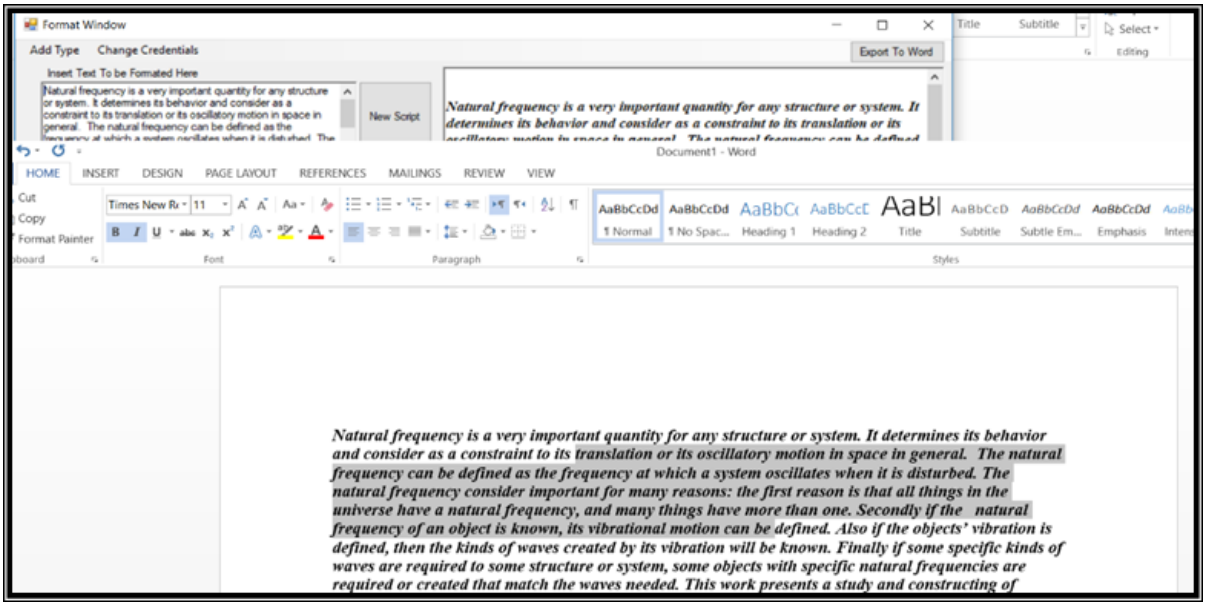


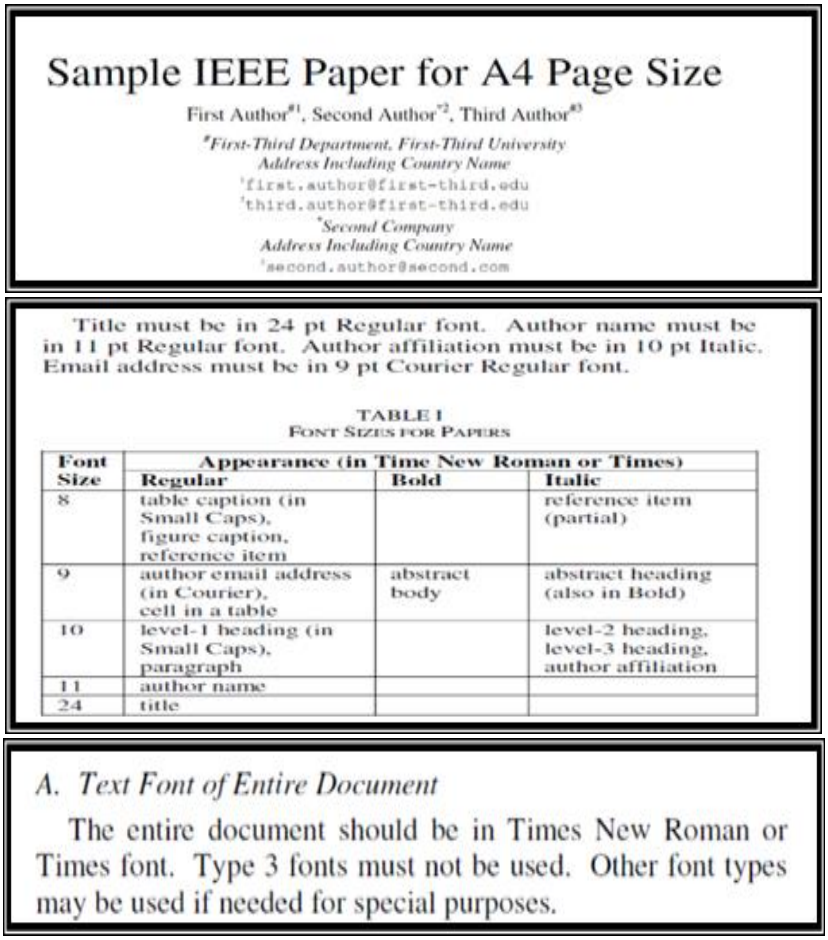
Figure 7 Export the results as a word page file

The final findings show that the compatibility percentage after conversion is between 95% and 98%.

Results of VB.NET simulation

-IEEE format

The format needed by IEEE journals is displayed in Figure 8, which includes the attributes of every section of the manuscript, including the title, the format of the authors' names, the abstract, the body of the study, and so on.



Sample IEEE Paper for A4 Page Size

First Author[#1], Second Author[*2], Third Author[#3]

[#]*First-Third Department, First-Third University*
Address Including Country Name
[1]first.author@first-third.edu
[3]third.author@first-third.edu
[*]*Second Company*
Address Including Country Name
[2]second.author@second.com

Title must be in 24 pt Regular font. Author name must be in 11 pt Regular font. Author affiliation must be in 10 pt Italic. Email address must be in 9 pt Courier Regular font.

TABLE I
FONT SIZES FOR PAPERS

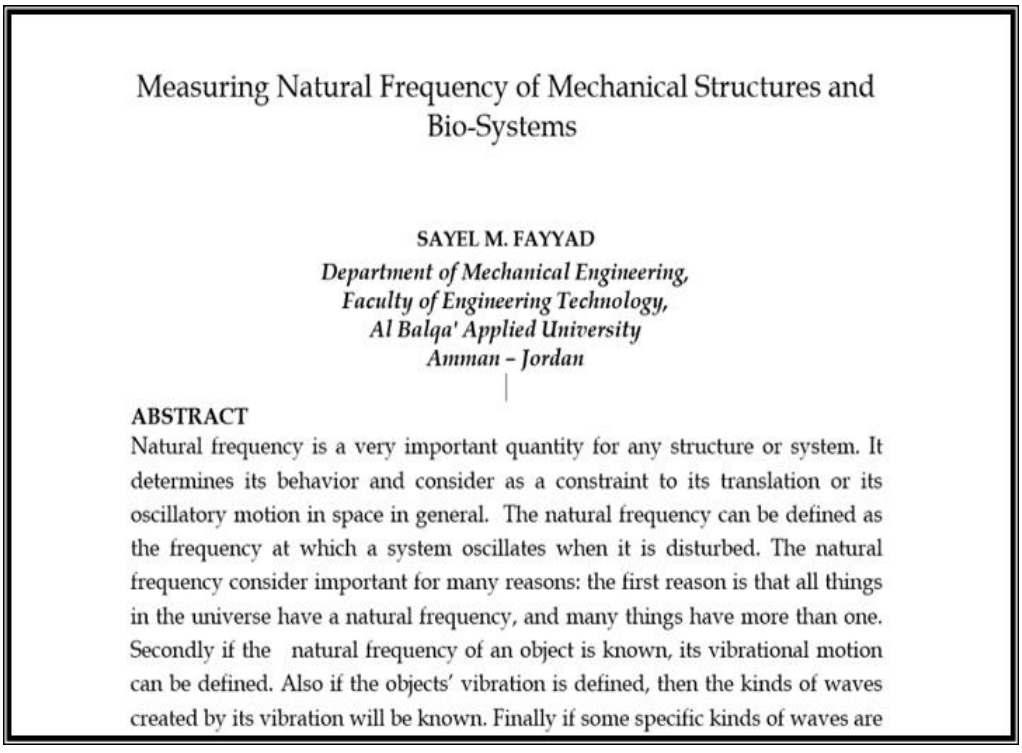
Font Size	Appearance (in Time New Roman or Times)		
	Regular	Bold	Italic
8	table caption (in Small Caps), figure caption, reference item		reference item (partial)
9	author email address (in Courier), cell in a table	abstract body	abstract heading (also in Bold)
10	level-1 heading (in Small Caps), paragraph		level-2 heading, level-3 heading, author affiliation
11	author name		
24	title		

A. Text Font of Entire Document

The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

Figure 8: (a) IEEE journal necessary format, (b) typefaces of text used in the article, and (c) font of complete documents

The original manuscript is displayed in Figure 9 prior to reformatting.



Measuring Natural Frequency of Mechanical Structures and Bio-Systems

SAYEL M. FAYYAD
Department of Mechanical Engineering,
Faculty of Engineering Technology,
Al Balqa' Applied University
Amman – Jordan

ABSTRACT

Natural frequency is a very important quantity for any structure or system. It determines its behavior and consider as a constraint to its translation or its oscillatory motion in space in general. The natural frequency can be defined as the frequency at which a system oscillates when it is disturbed. The natural frequency consider important for many reasons: the first reason is that all things in the universe have a natural frequency, and many things have more than one. Secondly if the natural frequency of an object is known, its vibrational motion can be defined. Also if the objects' vibration is defined, then the kinds of waves created by its vibration will be known. Finally if some specific kinds of waves are

Figure 9 sample of the paper before reformatting

While the document is depicted in Figure 10 throughout the reformatting process, the comparative percentages before and after are 9.44% and 95%, respectively.

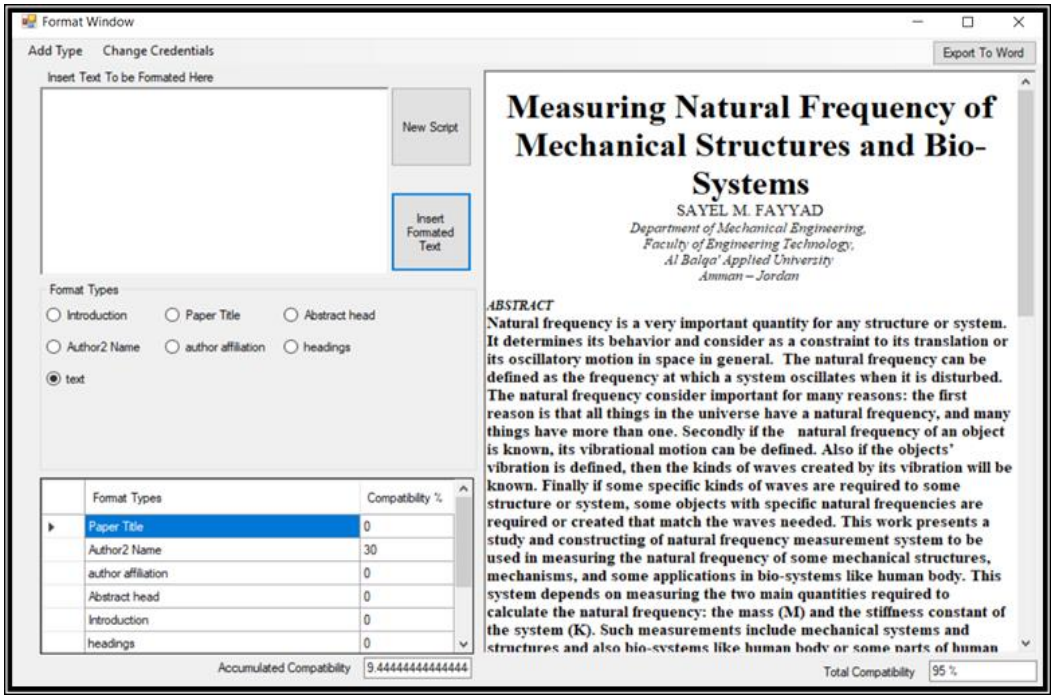


Figure 10 reformatting of the paper and comp.%

Figure 11 shows the final output paper in word file.

Measuring Natural Frequency of Mechanical Structures and Bio-Systems

SAYEL M. FAYYAD
Department of Mechanical Engineering,
Faculty of Engineering Technology,
Al Balqa' Applied University
Amman – Jordan

ABSTRACT

Natural frequency is a very important quantity for any structure or system. It determines its behavior and consider as a constraint to its translation or its oscillatory motion in space in general. The natural frequency can be defined as the frequency at which a system oscillates when it is disturbed. The natural frequency consider important for many reasons: the first reason is that all things in the universe have a natural frequency, and many things have more than one. Secondly if the natural frequency of an object is known, its vibrational motion can be defined. Also if the objects' vibration is defined, then the kinds of waves created by its vibration will be known. Finally if some specific kinds of waves are required to some structure or system, some objects with specific natural frequencies are required or created that match the waves needed. This work presents a study and constructing of natural frequency measurement system to be used in measuring the natural frequency of some mechanical structures, mechanisms, and some applications in bio-systems like human body. This system depends on measuring the two main quantities required to calculate the natural frequency: the mass (M) and the stiffness constant of the system (K). Such measurements include mechanical systems and structures and also bio-systems like human body or some parts of human body. It is found that age, mass and stiffness coefficient of the human body

Figure 11 Final paper in the required format

-Elsevier formatting

Procedia journal format is displayed as an Elsevier journal template in Figure 12.

Figure 12 Some Elsevier journal format sample

Figure 13 displays the outcome of the suggested VB.NET system for paper reformatting along with the percentage change between before and after the Elsevier journal.

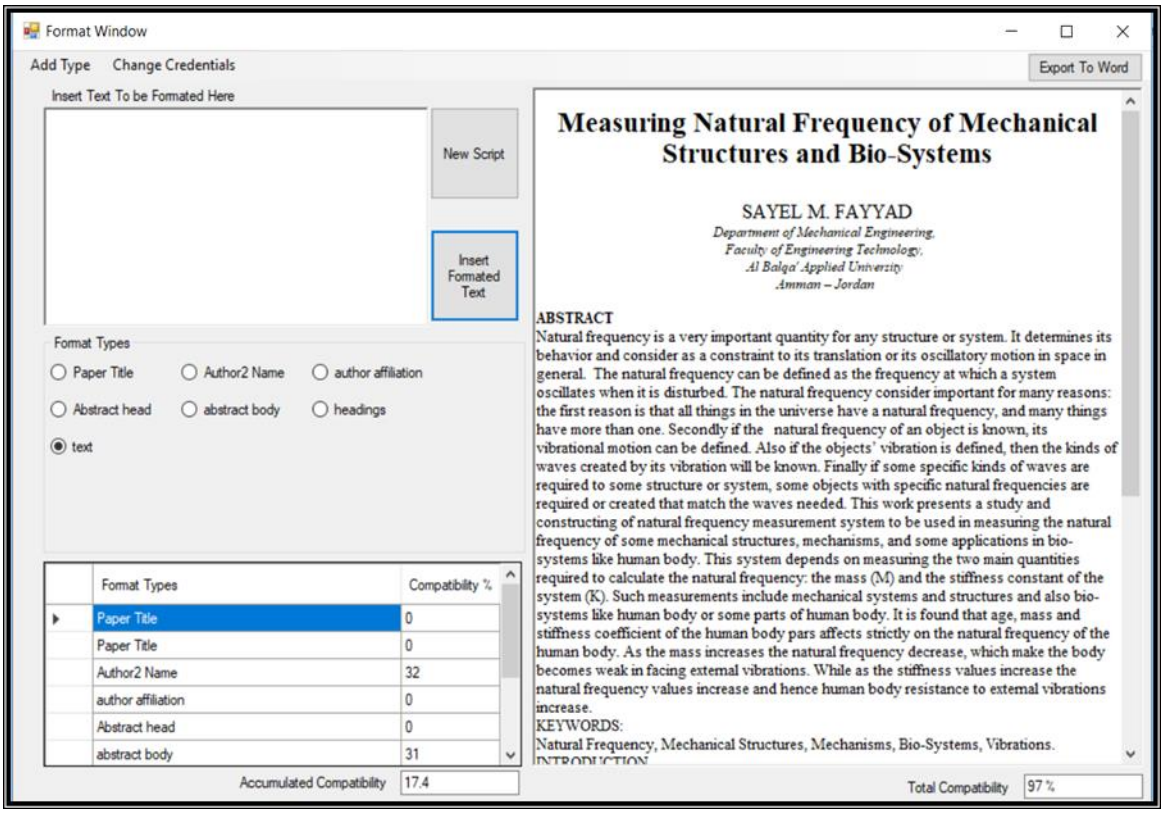


Figure 13 Reformatting article findings using the suggested approach on Elsevier journal

Figure 14 displays the final output paper from the second test, which had a comp.% of 17.4 before reformatting and 97% after the suggested system's processing.

Measuring Natural Frequency of Mechanical Structures and Bio-Systems

SAYEL M. FAYYAD
Department of Mechanical Engineering,
Faculty of Engineering Technology,
Al Balqa' Applied University
Amman – Jordan

ABSTRACT

Natural frequency is a very important quantity for any structure or system. It determines its behavior and consider as a constraint to its translation or its oscillatory motion in space in general. The natural frequency can be defined as the frequency at which a system oscillates when it is disturbed. The natural frequency consider important for many reasons: the first reason is that all things in the universe have a natural frequency, and many things have more than one. Secondly if the natural frequency of an object is known, its vibrational motion can be defined. Also if the objects' vibration is defined, then the kinds of waves created by its vibration will be known. Finally if some specific kinds of waves are required to some structure or system, some objects with specific natural frequencies are required or created that match the waves needed. This work presents a study and constructing of natural frequency measurement system to be used in measuring the natural frequency of some mechanical structures, mechanisms, and some applications in bio-systems like human body. This system depends on measuring the two main quantities required to calculate the natural frequency: the mass (M) and the stiffness constant of the system (K). Such measurements include mechanical systems and structures and also bio-systems like human body or some parts of human body.

KEYWORDS:
Natural Frequency, Mechanical Structures, Mechanisms, Bio-Systems, Vibrations.

INTRODUCTION

Figure 14. The final output paper for the second sample

Results Discussion

The proposed algorithms have the ability to reformat any article with a bad or medium compatibility percentage with the required format of any journal. The compatibility percentage of the final text is in the range of 95-98%. The two important things that emerged are the compatibility percentage before and after reformatting and the time required to reformat any paper using the proposed system. The comp.% is a tool used to determine whether to go to the next phase in article submission or to reformat the manuscript by either the author or the journal editor. The typical way of reformatting the work in journal format takes longer than the suggested model. Figure 15 illustrates a comparison of these timeframes. It decreases time by one-third on average, as measured by reformatting six articles using both the manual technique and the suggested algorithm.

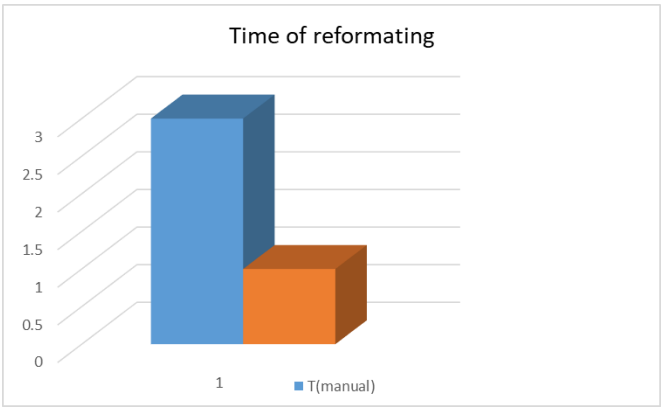


Figure 15: Comparison of reformatting time (in hours) between the existing and suggested systems

Figure 16 compares the final percentages of compatibility between the suggested model and the manual methods employed in paper reformatting. It is noticeable that the time, effort, and cost of reformatting any manuscript using the suggested model are lowered, while the compatibility between the papers and journal templates using the proposed model is enhanced when compared to existing approaches, but with less time and effort.

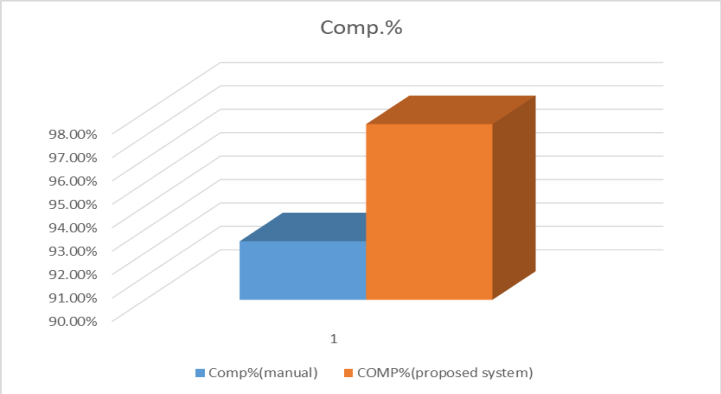


Figure 17 The compatibility comparison between the manual technique and the suggested system.

5. Conclusion

This endeavor is the first step toward creating a smart model and online platform to reformat any given paper (submitted to any scientific journal for review and subsequent publication) into the required format. The study's goal is to create an automatic reformatting mechanism for such submitted papers to journals, streamlining the process and saving both effort and time. Unfortunately, a complete system is not constructed owing to time constraints, restricted utilities, and a lack of understanding of some important sophisticated programs and high level programming languages; yet, this thesis marks the first or fundamental step toward a full version of such system.

In the first level, the system turns a supplied file (PDF or text) into a data collection that can be processed and changed. The second stage seeks to clean up the data so that it may be used in future neural network or artificial intelligence systems. The third stage employs the VB.NET programming language to assess the compliance of the provided text with the journal format and transform it into the proper format.

The recommended technique performs effectively when validating the composition percentage before and after reformatting, as well as reformatting the text to the suitable journal format.

The recommended approach reduces the amount of labor and time necessary to reformat any text while still generating satisfactory results. The reformatted texts have excellent compatibility rates, up to 99%. The time necessary to reformat any paper is reduced by 66 percent.

References

[1] Alomari, Z., El Halimi, O., Sivaprasad, K., and Pandit, C. (2015). Comparative Studies of Six Programming Languages. At: https://www.researchgate.net/publication/274572185.

[2] Bencze, I. Fark P., Hatala, L., and Jeszenszky, P. (2006). Server side PDF generation based on LATEX templates. TUGboat, Volume 27, No. 1—Proceedings of EuroTEX2006.

[3] Bhunia A., Bhunia A. K., Banerjee P., Konwer, A., Bhowmick, A., Roy P., and Umapada, Pal U. (2017). Word Level Font-to-Font Image Translation using Convolutional Recurrent Generative Adversarial Networks. E-mail of corresponding author: 2partharoy@gmail.com.

[4] Hu J., Fang L., Cao Y. Zeng H., Li H. Yang Q., and Chen Z. (2008). Enhancing Text Clustering by Leveraging Wikipedia Semantics. SIGIR'08, July 20–24, 2008, Singapore.

[5] Kawade, G. and Ballal, S. (2012). A Role of Virtual Machine in Operating System. Software Engineering. Volume 3, Issue 1, 2012, pp.-21-24.

[6] Kriesel, D. (2005). A brief introduction to neural networks.dkriesel.com. http://www.dkriesel.com/en/science/neural_networks.

[7] Kumar G. and Bhatia P. (2013). Neural Network based Approach for Recognition of Text Images. International Journal of Computer Applications (0975 – 8887) Volume 62– No.14.

[8] Reul C., Dittrich M. and Martin Gruner M. (2016), Case Study of a highly automated Layout Analysis and OCR of an incunabulum: 'Der HeiligenLeben' (1488). Available at https://go.uniwue.de/itf954ocr2 http://www.kallimachos.de

[9] Shetty R., and Heraje N. (2017). Recognition of Formatted Text using Machine Learning Technique. American Journal of Intelligent Systems, 7(3): 64-67.

[10] Stallings W. (2005). Operating Systems: Internals and Design Principles, Fifth Edition. Prentice Hall, 2005.

[11] Taqdir and Uttarhanica (2016) Recognition of Handwritten Characters Using Neural Networks. International Journal of Innovative Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2016.

[12] Vijayarani S., and Sakila A. (2015). Template Matching Technique for Searching Words in Document Images. International Journal on Cybernetics & Informatics (IJCI) Vol. 4, No. 6.

[13] Widjaja, M., and Hansun S. (2015). Implementation of Porter's Modified Stemming Algorithm in an Indonesian Word Error Detection Plug in Application. International Journal of Technology (2015) 2: 139-150.

[14] Yessenov K. et al. (2013). A Colorful Approach Text Processing by example. UIST'13, October 8–11, 2013, St. Andrews, UK.