

# A STUDY INTO MATH DOCUMENT CLASSIFICATION USING DEEP LEARNING

Fatimah Alshamari<sup>1,2</sup> and Abdou Youssef<sup>1</sup>

<sup>1</sup>Department of Computer Science,  
The George Washington University, Washington D.C, USA  
<sup>2</sup>Department of Computer Science, Taibah University, Medina, KSA

## ABSTRACT

*Document classification is a fundamental task for many applications, including document annotation, document understanding, and knowledge discovery. This is especially true in STEM fields where the growth rate of scientific publications is exponential, and where the need for document processing and understanding is essential to technological advancement. Classifying a new publication into a specific domain based on the content of the document is an expensive process in terms of cost and time. Therefore, there is a high demand for a reliable document classification system. In this paper, we focus on classification of mathematics documents, which consist of English text and mathematics formulas and symbols. The paper addresses two key questions. The first question is whether math-document classification performance is impacted by math expressions and symbols, either alone or in conjunction with the text contents of documents. Our investigations show that Text-Only embedding produces better classification results. The second question we address is the optimization of a deep learning (DL) model, the LSTM combined with one dimension CNN, for math document classification. We examine the model with several input representations, key design parameters and decision choices, and choices of the best input representation for math documents classification.*

## KEYWORDS

*Math, document, classification, deep learning, LSTM*

## 1. INTRODUCTION

Recently, online libraries have been increasing exponentially in the number of publications [1]. These documents are required to be manually annotated by an expert in order to be indexed in online libraries. The annotation, if done manually, is an expensive process in terms of cost and time. For this reason, some libraries ask the authors to help with this process by adding a subject code before submitting the document [1]. However, this does not appear to solve the problem adequately, because the final subject for indexing a document should be reviewed by an expert. Accordingly, many studies have been conducted in automated text classification to tackle the problem. Most of these studies aim to find a reliable classification approach that could replace the expert role [2].

Thus, different classification approaches have been investigated in text classification across several areas. These approaches could include traditional machine learning algorithms or deep learning classification models. However, mathematics documents differ from documents in other areas because the former contain mathematical symbols, formulas, and expressions. Inasmuch as they are indispensable for a reader to understand the mathematics content, using these mathematical expressions as features in any classifiers model could improve the classification

Natarajan Meghanathan et al. (Eds): CSE, BMLI, EDU, AI & FL, CCNET - 2020

pp. 13-22, 2020. CS & IT - CSCP 2020

DOI: 10.5121/csit.2020.101702

process. In document classification, features are crucial as they can improve the accuracy and efficiency of the classifier or hurt them. In this study, we investigated the impact of math features either alone or in conjunction with the text contents of documents on the math document classification performance. Therefore, we conducted comparative experiments by examining a number of models, several key design parameters, and decision choices to choose the best deep learning math documents classifier design.

This paper is organized as follows. Section 2 presents an overview of related work. Section 3 discusses the dataset we used and its different set-ups. Section 4 illustrates our approach. In Section 5, we describe our experiments, and discuss our results. Our conclusions and future work are provided in Section 6.

## 2. RELATED WORK

In the field of document classification, many studies have been conducted in different areas aiming to find a reliable model [2], which include classical machine learning and deep neural networks models.

Harish et al presented a general technical review that studies the text classification and conducted a performance comparison of a set of known machine learning classification models [3]. Their study concludes that under different criteria, such as algorithm parameters and time complexity, each model behaves differently. There is no optimum combination between text representation techniques with machine learning models that could produce the best result.

In a recent study, Scharpf, et al investigated the impact of selecting and combining different representation techniques on the classification accuracy and cluster purity on scientific documents, sections and abstracts [4]. They applied different encodings techniques such as tfidf, and doc2vec on text and math content separately. Their study showed that text encoding outperforms standalone math encoding, and combining text and math encodings does not improve the classification performance. They observed a low correlation between text and math similarity, which suggests that text and math should better be treated as separate features.

Dalal and Zaveri [2] presented some issues with text classification that could affect the performance of any model. The pre-processing and feature selection techniques should be applied carefully because they have a critical impact on performance.

Our work is aligned with [5], [6], and [1], where they focused on mathematics document classification. Barthel et al presented a large-scale experiment by applying simple pre-processing. Although they used a large dataset, about three million documents, they limited the experiments to metadata only. Despite the fact they presented good results, it would not be possible to classify papers that have short or no abstract using their model. In addition, if a formula is too simple, during the pre-processing stage, they added it to the only-text features; this enrichment of the only text features could have some impact on performance that was not investigated in that paper. In our paper, we are curious to comprise a large set of formula features beyond what is revealed in the metadata only. Therefore, we adopt the full-text dataset including the metadata.

In a recent paper [6], Suzuki and Fujii investigated math document classification, but they used MATHML that produces the math features in a tree structure, whereas we use LLaMaPUn [7], and our math features are linear rather than tree-structured.

In terms of deep neural networks [8], [9] and [10] studied different deep neural network models on the text classification task. Semberecki and Maciejewski applied long short-term memory (LSTM) model in documents classification to study different representations approaches [8]. Their evaluation showed that the vector representation approach outperformed a standard bag-of-word approach based on the LSTM model in the document classification task. In [9] a recurrent Convolutional Neural Network (CNN) model was proposed for text classification. They applied a recurrent structure to capture the contextual information in order to construct the representation of the text. In addition they applied max-pooling to capture the main tokens in the text. Their experiments were conducted on four different datasets, and showed that their proposed model outperformed CNN and Recursive Neural Networks (RNN). A hybrid architecture consisting of an LSTM model and CNN model was proposed for different text classification tasks in [10]. Their experiments showed that combining these two models in one architecture outperforms both CNN and LSTM alone.

### 3. DATASET

#### 3.1. Dataset Collection

In order to evaluate our models, we obtained one of the HTML5 dataset subsets [11]. It is a large scale dataset recently published by the arXivLive project [12]. It consists of over a million scientific documents from the arXiv.org [13], converted from the original LaTeX format to HTML format using the LaTeXML conversion tool [14], then grouped based on the conversion success level into three subsets. Since not all the math documents converted fully (without errors and warnings) using LaTeXML, we selected only the documents that had no conversion issues. This resulted in a smaller subset, but large enough, which we used for this work. This subset consists of 705,095 multi-label documents assigned to one or more subjects from seven major areas: Physics, Computer Science, Mathematics, Quantitative Biology, Quantitative Finance, Statistics, Electrical Engineering and Systems Science, and Economics. For the Mathematics area, ArXiv uses a set of 32 subjects to label math documents inclusively. The first subject that is assigned to a document reflects the principal contribution of that document, and all other subjects, if any, are considered a secondary principal contribution.

Our focus in this study is document classification in the Mathematics areas; therefore, from the used subset we retrieved a collection of about 171,000 full-text English-language documents with math subject as a first label. The number of documents in each subject varies from 372 to 20522. The popularity of math paper submissions varies from subject to subject, causing a bias between the numbers of documents in the different subject areas. Therefore, we will consider weighted accuracy in future work. In addition, the size or length of the documents varies; some documents are short papers while others are dissertations with a significantly large number of pages. For this reason, the input length is one of the parameters that we considered in the training process. Table 1 provides detailed information regarding the subject distribution in the dataset.

Table 1. Dataset Statistics based on the number of documents in each subject.

Subject	#of Document	Subject	#of Document	Subject	#of Document
mathGT	3075	mathST	3155	mathCO	14116
mathGR	4902	mathMP	20627	mathCT	327
mathKT	416	mathAG	10577	mathDS	7895
mathMG	1335	mathAT	1697	mathGM	819
mathNT	7698	mathAP	18145	mathHO	480
mathNA	4869	mathCA	6947	mathIT	6413
mathOA	2076	mathAC	4370	mathLO	2317
mathOC	4321	mathCV	4819	mathPR	11523
mathQA	3040	mathDG	9724	mathRT	1817
mathRA	2703	mathFA	8470	mathSG	542
mathSP	786	mathGN	1164		
<b>Total # of Documents</b>		171165			

### 3.2. Dataset Preparation

One of the advantages of the HTML format is preserving all the properties of formulas and mathematical expressions. However, NLP tools require text data; therefore, an additional step of converting the HTML format has been done. In order to get the raw text, we applied the LLaMaPUn [7] toolkit. The raw text is segmented into three different sets: (1) Text and Math, (2) Text, and (3) Math. The Text-and-Math set is the original output of the LLaMaPUn, Figure 1 shows the input for this set. While the Text set consists of a clean text without any math formulas or expressions, as shown in Figure 2, the Math set consists of only math formulas and expressions, as shown in Figure 3.

2. SKEW MONOIDAL CATEGORIES

A *skew monoidal category* [25] is a category  $\mathcal{C}$  together with a distinguished object  $I$ , a functor  $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$  and three natural transformations

$$\lambda_A : I \otimes A \rightarrow A \quad \rho_A : A \rightarrow A \otimes I \quad \alpha_{A,B,C} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$$

Figure 1. The input for Text and Math set

Skew Monoidal Categories A *skew monoidal category* is a category together with a distinguished object a functor and three natural transformations

Figure 2. The input for Text set

$$\mathcal{C} \mid \otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$$

$$\lambda_A : I \otimes A \rightarrow A \quad \rho_A : A \rightarrow A \otimes I \quad \alpha_{A,B,C} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$$

Figure 3. The input for Math set

### 3.3. Dataset Pre-processing

Data pre-processing is an essential step for any text-based task. For each set resulting from the segmentation step, we applied different pre-processing. For the Text set, we applied the following processing: text normalization and text removal. Specifically, in the text normalization step, we replace all punctuation with a keyword PUN, and all numbers with keyword NUM. In the text removal step, we used the Natural Language Toolkit (NLTK) [15], to omit stop-words since they carry less significance in classification [2]. Also, we discarded all non-English words and words with length more than 25. In addition, as math documents contain theories' names and other named entities, which are case-sensitive, we preserved case-sensitivity. For the Math set, math formulas and expressions are case-sensitive, hence, we used LLaMaPUn to preserve case-sensitivity, font styles, weights and faces [16]. Finally, a combination of the set-specific (i.e. Text set and Math set) pre-processing steps has been applied on the Text-and-Math set.

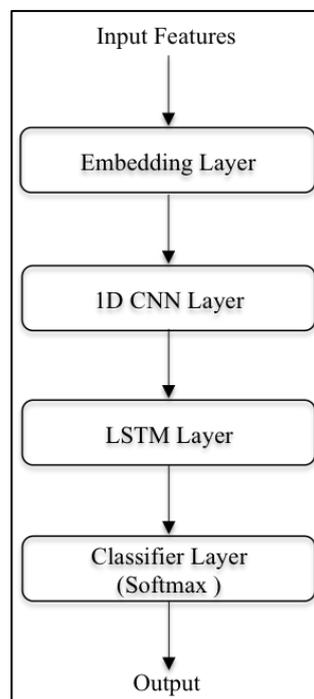


Figure 4. The Approach for Classifying Mathematics Document

## 4. THE DEEP LEARNING APPROACH

Our approach consists of (1) examining a deep learning model architecture with several key design parameters; (2) evaluating the model on document classification, and (3) investigating the best feature set based on the performance. Recently, deep learning models have shown significant improvements in different NLP tasks such as Question Answering [17], Information Extraction [18], and Text Classification [8], [9] and [10]. Different architectures including RNN [19] and its variants LSTM [20], BiLSTM [21], CNN [22], and Attention [23] have been widely studied for different Neural Network Language Processing (NLP) tasks.

While there is no rule that defines which model is the right one for a specific task, the task objective and the input characteristics play an important role in narrowing the set of models candidates. For example, the RNN has two limitations which are exploding gradient and vanishing gradient [24]. The vanishing gradient could accrue with a long sequence input, which

is the case in document classification. Fortunately, the LSTM model tackles the gradient problems and is able to capture long-term dependencies in any length sequences. In addition, combining CNN with LSTM in one architecture outperforms LSTM model in text classification [10].

Therefore, our focus in this paper is to study LSTM with one CNN layer on the math documents classification with different setups. Our model, shown in Figure 4, consists of four layers: an embedding layer, a 1D CNN layer, an LSTM layer, and a softmax layer. The model starts with feeding the document as raw input features into the embedding layer, which passes its embeddings to the CNN layer; the CNN layer derives important features and passes them to the LSTM layer [24]. Finally, the output of the LSTM layer is fed to the softmax layer that determines the class of the input document.

As we are dealing with math documents which contain text with math formulas and expressions, we define three different sets of input representations: Text-Only and Math-Only, and Text-Math as detailed in section 2.2.

## 5. EXPERIMENTAL RESULTS AND ANALYSIS

### 5.1. Experiments

Our model is built using the deep learning platform Keras [25] with Tensorflow [26] as the backend framework. In order to train and evaluate the classifier, we divided the dataset randomly into 80% training and 20% testing split. For all setups we report and compare the accuracy on the test set. The categorical-crossentropy is the loss function used, and for optimizing all models Adam is used [27]. All Out-Of-Vocabulary (OOV) words are randomly initialized.

The main parameters we focused on are the documents length and the word embeddings. We examined different input length (1000, 1500, 2000), where an input length of  $N$  means that from each document, only the first  $N$  tokens are taken while the remaining tokens are ignored. This is commonly done in deep learning for memory and speed considerations. In the embedding setups, we evaluated our model based on the word embeddings types using two main methods: Randomly Initialized embeddings, and GloVe embeddings. For the GloVe embeddings we have experimented with the type of the data used to train the embeddings, and the size of embeddings dimensions. Specifically, we used pre-trained embeddings from [11] which are trained on global data in different subject areas including math; we refer to the pre-trained GloVe embedding as GloVe-PR. GloVe-PR came in one dimension size 300 and two flavors based on its training data: Text only excluding math symbols and expressions, and text-and-math which includes both the text and the math symbols and expressions.

In addition to the pre-trained embeddings, we trained our own GloVe on our dataset, which includes math subjects only. We follow similar setups and parameters used in [11] with additional dimension sizes in (100, 200, 300); we refer to this mode of embedding as GloVe-M. For all pre-trained embedding experiments, GloVe-PR and GloVe-M we evaluate both tuning the word embeddings and freezing the word embeddings layer. For the baseline, we trained the LSTM model using randomly initialized embeddings.

### 5.2. Results

We report in Tables 2, 3 and 4 the performance of the models based on the different setups. Table 2 illustrates the accuracy performance corresponding to embedding dimension of 100 in all

models and all different input lengths. In this table Text-Only outperforms both the Text-Math and Math-Only, where the highest results across all models are given by GloVe-M+Tuned model. We also observe that increasing the input length provides some improvement, though the gains are very modest. The results of 200 embedding dimensions are shown in Table 3. Overall, we observed similar trends to the ones we observed in Table 2, except for the input length impact on the results. In particular, adding more input data resulted in sustainable improvement in the performance.

Table 2. Classification Experimental Results of the 100 Embedding Dimension. The numbers represent the classification accuracy on the test set.

Input length	Model	Text-Math	Text-Only	Math-Only
1000	Baseline	57.4	64.08	37.7
	Random-Embedding	56.78	63.68	39.42
	GloVe-M	52.27	60.77	22.1
	GloVe-M+Tuned	57.98	<b>65.1</b>	35.43
1500	Baseline	59	65.94	38.94
	Random-Embedding	58.07	65.91	41.53
	GloVe-M	54.33	62.68	12.74
	GloVe-M+Tuned	58.5	<b>67.08</b>	37.04
2000	Baseline	59.43	65.86	39.85
	Random-Embedding	59.2	66.33	42.59
	GloVe-M	54.37	63.73	23.84
	GloVe-M+Tuned	60.19	<b>67.87</b>	38.18

Table 3. Classification Experimental Results of the 200 Embedding Dimension. The numbers represent the classification accuracy on the test set.

Input length	Model	Text-Math	Text-Only	Math-Only
1000	Baseline	58.07	64.55	37.38
	Random-Embedding	56.8	64.4	38.68
	GloVe-M	52.78	71.74	10.45
	GloVe-M+Tuned	58.41	<b>65.38</b>	34.76
1500	Baseline	58.87	65.87	38.93
	Random-Embedding	58.14	65.9	41.34
	GloVe-M	55.31	63.49	26.65
	GloVe-M+Tuned	59.37	<b>66.41</b>	36.89
2000	Baseline	59.82	66.2	46.71
	Random-Embedding	59	67.21	43.65
	GloVe-M	56.7	<b>68.07</b>	20.48
	GloVe-M+Tuned	60.33	68.04	38.56

Table 4 shows the results using 300 dimension embedding with different input lengths. In general, the Text-Only yields the best performance compared to both the Math-Only and Text-and-Math across all setups. Specifically, in the GloVe-PR+Tuned model with the 2000 input length, Text-Only outperforms the Math-Only and Text-Math with 24.13% and 7.7% gain respectively. For embedding types, both GloVe-PR+Tuned and GloVe-M+Tuned produced the best results across different input length.

Table 4. Classification Experimental Results of the 300 Embedding Dimension. The numbers represent the classification accuracy on the test set.

Input length	Model	Text-Math	Text-Only	Math-Only
1000	Baseline	58.04	64.6	36.71
	Random-Embedding	57.15	63.5	40.04
	GloVe-PR	52.8	62.36	39
	GloVePR+Tunned	58.48	<b>65.52</b>	40.22
	GloVe-M	54.03	62.18	26.52
	GloVe-M+Tunned	58.29	65.1	31.11
1500	Baseline	59.31	65.94	38.61
	Random-Embedding	58.48	66.08	41.58
	GloVe-PR	55.81	63.67	40.38
	GloVePR+Tunned	59.56	67.06	41.67
	GloVe-M	55.84	63.66	28.79
	GloVe-M+Tunned	59.73	<b>67.08</b>	31.17
2000	Baseline	60.12	65.86	39.91
	Random-Embedding	55.95	66.24	42.96
	GloVe-PR	56.94	65.37	42.62
	GloVePR+Tunned	60.54	<b>68.24</b>	44.11
	GloVe-M	56.32	64.39	14.42
	GloVe-M+Tunned	60.24	67.87	37.63

Interestingly, the result shows that the classifiers performance degraded when including the math formulas and expressions. The best Text-Only model performed significantly better than the other models that contain math information. Thus, considerable improvement can be achieved when omitting all the math information from the text.

We believe the reason that the Text-Only model performs better is twofold. First, unlike text tokens, the math symbols in equations and expressions are abstract and often generic, thus lacking any differentiation power. Second, when we include math tokens and fix the number of tokens taken to represent a document, the number of text tokens in the representation in the Text-and-Math model is fewer than in the Text-Only model, thus ending up with fewer differentiating features in the Text-and-Math model than in the Text-Only model, resulting naturally in lesser accuracy.

With respect to the input length and embeddings size, all models seem to benefit from increasing both factors. However, this improvement is slight with increasing the embeddings size as shown in Table 5.

Table 5. Best Performance in Text Only among different Dimensions and Input-Length.

Model	Dimension	Input-Length	Accuracy
GloVePR+Tunned	300	2000	<b>68.24</b>
GloVe-M	200	2000	<b>68.07</b>
GloVe-M+Tunned	100	2000	<b>67.87</b>

## 6. CONCLUSIONS

Mathematics document classification is an important problem to address due to the growing demand for automatic models to handle such task. Document classification in general and the Mathematics document classification in particular are still an open challenge given the large set of subjects associated with a document.

Thus, in this study, we carried out comparative experiments using well-known deep learning models, LSTM with one dimension CNN. One main goal of this study is to investigate the impact of inclusion/exclusion of the math features on the classification output. The other main goal was to identify optimal models, optimal feature representations, and optimal design choices.

The results showed that overall the model performance improved with excluding all math information from the document, using the Text-only representations. Furthermore, the result of testing the impact of input length and embeddings size showed that increasing both factors impacted the performance positively. Specifically, larger improvement was achieved with increasing the input length and smaller improvement achieved with increasing the embedding size.

Of all the models and variations in parameters and design choices considered, the best choice was determined to be: 300D embedding with input length of 2000, and with post-trained (i.e. tuned) GloVe embedding that was pretrained on global data.

This study can be improved in several ways. For instance, in our future work, we are planning to use the full text from a document for further experiments and testing. We also plan to investigate other embedding representations such as Word2Vec/Fasttext and contextualized embeddings (i.e. ELMO and BERT), and test other deep neural architectures.

## REFERENCES

- [1] Řehůřek, Radim, & PetrSojka, (2008) “Automated classification and categorization of mathematical knowledge”, International Conference on Intelligent Computer Mathematics, pp543-557.
- [2] Dalal, Mita K., & Mukesh A. Zaveri, (2011) “Automatic text classification: a technical review”, International Journal of Computer Applications, Vol. 10, No. 5, pp37-40
- [3] Harish, Bhat S., Devanur S. Guru, & ShantharamuManjunath,(2010) “Representation and classification of text documents: A brief review”, IJCA, Special Issue on RTIPPR (2), pp110-119.
- [4] P. Scharpf, M. Schubotz, A. Youssef, F. Hamborg, N. Meuschke, andB.Gipp,(2020) “Classification and Clustering of arXiv Documents, Sections, and Abstracts, Comparing Encodings of Natural and Mathematical Language”, Proceedings of the JCDL Conference 2020, doi: 10.1145/3383583.3398529.
- [5] Barthel, Simon, SaschaTönnies, & Wolf-TiloBalke, (2013) “Large-Scale Experiments for Mathematical Document Classification”, International Conference on Asian Digital Libraries, pp83-92.
- [6] Suzuki, Tokinori, & Atsushi Fujii, (2017) “Mathematical document categorization with structure of mathematical expressions”, ACM/IEEE Joint Conference on Digital Libraries, pp1-10.
- [7] DeyanGinev& Jan Frederik Schaefer, (2019) “LLaMaPUn: common language and mathematics processing algorithms”, <https://github.com/dginev/llamapun/>
- [8] Semberecki, Piotr & HenrykMaciejewski, (2017) “Deep learning methods for subject text classification of articles”, Federated Conference on Computer Science and Information Systems, pp1357-360.
- [9] Lai, Siwei, et al., (2015) “Recurrent convolutional neural networks for text classification”, Twenty-ninth AAAI conference on artificial intelligence
- [10] Zhou, Chunting, et al., (2015) “A C-LSTM neural network for text classification”, arXiv:1511.08630
- [11] DeyanGinev, (2018) “an HTML5 conversion of arXiv.org”, <https://sigmathling.kwarc.info/resources/arxmliv-dataset-082018/>
- [12] Stamerjohanns, Heinrich, Michael Kohlhase, DeyanGinev, Catalin David, & Bruce Miller, (2010) “Transforming large collections of scientific publications to XML”, Mathematics in Computer Science, Vol. 3, No. 3, pp299-307
- [13] Arxiv E-print archive, <http://arxiv.org>
- [14] Miller, Bruce, (2010) “LaTeXML: ALatex to xml converter”,<http://dlmf.nist.gov/LaTeXML/>
- [15] Loper, Edward, & Steven Bird,(2002) “NLTK: the natural language toolkit”,arXiv preprint cs/0205028

- [16] Ginev, Deyan, & Bruce R. Miller, (2019) "Scientific Statement Classification over arXiv.org.", arXiv:1908.10993
- [17] Yu, Lei, et al., (2014) "Deep learning for answer sentence selection.", arXiv:1412.1632
- [18] Narasimhan, Karthik, Adam Yala, & Regina Barzilay, (2016) "Improving information extraction by acquiring external evidence with reinforcement learning", arXiv:1603.07954
- [19] Rumelhart, David E., Geoffrey E. Hinton, & Ronald J. Williams, (1986) "Learning representations by back-propagating errors", nature Vol. 323, No. 6088, pp533-536
- [20] Hochreiter, Sepp, & Jürgen Schmidhuber, (1997) "Long short-term memory", Neural computation, Vol. 9, No. 8, pp1735-1780
- [21] Schuster, Mike, & Kuldip K. Paliwal, (1997) "Bidirectional recurrent neural networks", IEEE Transactions on Signal Processing, Vol. 45, No. 11, pp2673-2681
- [22] Fukushima, Kunihiko, & Sei Miyake, (1982) "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition", In Competition and cooperation in neural nets, Berlin, Heidelberg, pp267-285.
- [23] Wang, Yequan, et al., (2016) "Attention-based LSTM for aspect-level sentiment classification", Empirical methods in natural language processing, pp606-615.
- [24] Hallac, Ibrahim R., Betül Ay, & Galip Aydın, (2018) "Experiments on Fine Tuning Deep Learning Models With News Data For Tweet Classification", IEEE International Conference on Artificial Intelligence and Data Processing, pp1-5.
- [25] François Chollet et al., (2015) "Keras: The python deep learning library", <https://keras.io>
- [26] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S, (2016) "Tensorflow: Large-scale machine learning on heterogeneous distributed systems", arXiv:1603.04467
- [27] Kingma, Diederik P., and Jimmy Ba., (2014) "Adam: A method for stochastic optimization", arXiv:1412.6980

## AUTHORS

**Fatimah Alshamari** is a Ph.D research student. She received B.C.Sc (Bachelor of Computer Science) degree in 2007, and M.C.Sc (Master of Computer Science) degree in 2014. She is now Assistant Lecturer of Taibah University. Her research interests include Natural Language Processing and Mathematics Language Processing.

**Abdou Youssef** has 30 years of research and teaching experience in the field of computer science. He is currently a tenured Professor at The George Washington University, Washington, D.C, which he joined as Assistant Professor in Fall of 1987. His current research interests are applied data science, math search and math language processing, audio-visual data processing, pattern recognition, theory and algorithms. He has published over 125 papers in those areas, and co-edited the book Interconnection Networks for High-Performance Parallel Computers, published by IEEE Computer Society Press in 1994. His research has been funded by NSF, NSA, and NIST. Currently, he is developing novel techniques for part-of-math tagging, math semantics extraction and question answering, and big-data applications such as fraud detection in the retail business, next-generation recommendation systems, and more.