Topic recommendation using Doc2Vec

Petros Karvelis
Laboratory of Knowledge and Intelligent Computing
Department of Computer Engineering,
Technological Educational Institute of Epirus
Arta, Greece
pkarvelis@teiep.gr

Dimitris Gavrilis
Department of Electrical Engineering and Computer Technology,
University of Patras,
Patras, Greece
gavrilis@upatras.gr

George Georgoulas
Control Engineering Group
Department of Computer Science, Electrical and Space Engineering
Luleâ University of Technology,
Luleâ, Sweden
geogeo@ltu.se

Chrysostomos Stylios
Laboratory of Knowledge and Intelligent Computing
Department of Computer Engineering,
Technological Educational Institute of Epirus
Arta, Greece
stylios@teiep.gr

***Abstract*— The ever-increasing number of electronic content stored in digital libraries requires a significant amount of effort in cataloguing and has led to self-deposit solutions where the authors submit and publish their own digital records. Even in self-deposit, going through the abstract and assigning subject terms or keywords is a time consuming and expensive process, yet crucial for the metadata quality of the record that affects retrieval. Therefore, an automatic, or even a semi-automatic process that can recommend topics for a new entry is of huge practical value. A system that can address that has to rely basically on two components, one component for efficiently representing the relevant information of the new document and one component for recommending an appropriate set of topics based on the representation of the previous stage. In this work, different candidate solutions for both components are investigated and compared. For the first stage both distributed Document to Vector (doc2vec) and conventional Bag of Words (BoW) components are employed, while for the latter two different transformation approaches from the field of multi-label classification are compared. For the comparison, a collection of Ph.D. abstracts (~19000 documents) from the MIT Libraries Dspace repository is used suggesting that different combinations can provide high quality solutions.**

Keywords*— *Recommender system, multilabel classification, word2vec, doc2vec, bag of words

I. Introduction

Topic recommendation is an important step in all digital libraries especially in academic environments or libraries with substantial amount of content. Topics are used in information retrieval and are also an important part of the navigation within the digital libraries. An accurate set of subject terms contributes significantly to the quality of the metadata record.

Furthermore, taking into account the increasing number of publications and also the self-deposit workflows that have been developed as parts of most repositories (such as DSpace, EPrints [1]), it is often the case that authors themselves suggest keywords and subject terms for their own publications. Author contributed keywords are usually accurate but when authors are asked to pick subjects from a thesaurus, lack of training in combination with ambiguous terms found in large thesauri, often lead to poor quality records. A potential solution for more standardized subject assignment is the use of automatic recommendation engines relying on text processing and machine learning tools

Text processing is in the front of many research and commercial applications. Among the various new technologies developed, distributed word representations are in the spotlight of late. However distributed word representations are not so new. In fact, some early proposals date back in the mid-80s and early 90s [2]. Nevertheless, this topic became of interest later starting with the introduction of neural net-work language models trying to solve the language modeling task of predicting a probability distribution over the “next” word, given some preceding words [3].

These works manifested very high word-prediction, but also made it clear that more computationally efficient models are needed. Based on these very promising results, this work examines the use of distributed and distributional representations of texts/abstracts for the purpose of recommending an appropriate set of topics/subjects for them. The topic recommendation is treated as a multilabel classification problem, which allows more than one topics/subjects to be assigned to the same abstract.

The rest of the paper is structured as follows Section 2 describes the overall Topic Recommendation scheme and each one of the involved components, Section 3 presents the Results and finally Section 4 concludes the paper with some final remarks, recommendations and suggestions for future work.

II. 2 Topic Recommendation

The topic recommendation consists of a number of stages. Fig. 1 provides a flowchart of the overall framework of the Topic Recommendation System. At first the document is “harvested” and it is parsed to extract the text that corresponds to the abstract of the thesis. Then standard preprocessing steps before the application of the feature extraction stage. Two feature extraction approaches are tested in this work: the doc2vec approach, which is considered state of the art for most text mining applications and the more conventional Bag of Words (BoW) representation which, has been the standard approach before the introduction of the distributed word representations. The final stage uses the extracted feature and a multi-label classifier to assign a number of labels/topics to the original text. In the following sections each of these stages are described in more detail.

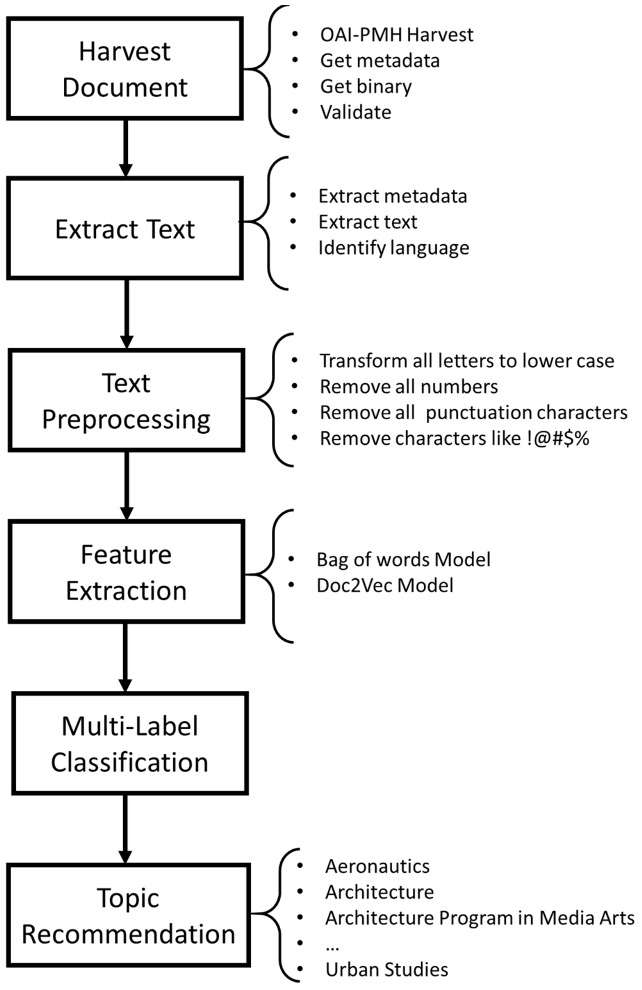


Fig. 1. The proposed Topic Recommendation System flowchart. The original document is harvested and the appropriate text is extracted. Then text preprocessing takes place for text normalization followed by feature extraction. Finally, the topic assignment is performed by the multilabel classifier.

A. Document Harvesting and Text Extraction

In order to acquire the data for the experiment presented in this paper, OAI-PMH provider of the MIT's DSpace repository was used. A selected harvesting based on all the Theses related Sets of the repository was executed using the repolytics (http://www.repolytics.com/) engine. The OAI-PMH repository contained various representations of the data. The METS and OAI_DC were chosen and retrieved but ultimately the OAI_DC was used as it contained all the necessary information. The subject terms array was mapped from the dc:subject element whereas the dc:description element was used to provide the abstract. Multiple dc:description elements were used for each record and all these were concatenated into one.

B. Text Preprocessing

The goal behind text preprocessing is to separate the text into individual words and also get rid of irrelevant formatting options. This step is very important in text mining applications and is usually called text normalization. It includes a number of steps like transforming all words to lower case, removal of stop words and stemming [4]. In our case, we used the following preprocessing steps:

a. Transform all words to lower case: using this step we transform all words to a uniform representation.

b. Remove all numbers from text.

c. Remove all punctuations: this step is crucial since words like “mathematics,” and “mathematics” after that step will be considered the same.

d. Remove special characters like “@#$%^&*(){}:”<>?/.,’;][“

Finally, all the extracted texts are stored into a single file where each line represents a single document.

C. Feature Extraction

A major challenge of the text classification problem is the representation of the documents having arbitrary lengths using vectors of the same dimension so that it can be used by conventional classification schemes. In the proposed system, two methods were incorporated and tested; one that used to be the standard for text processing (BoW) and one that seems to yield even more promising results (doc2vec).

Feature using BoW

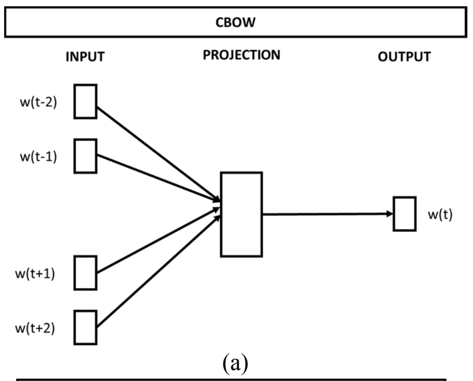
First the simplest (BoW) representation, where each document is represented by a vector of the words counts that appear in the document is considered. Thus, in order to represent a document into a vector, a vocabulary of terms was produced. A vocabulary of 1000 words was created for the selected dataset. This is a common parameter setting [5].

Feature using doc2vec

Word transformations to vectors is a key step for many Natural Language Processing (NLP) algorithms nowadays. However, representing a meaning with distinct symbols fails to homogenize the same meaning of words like “salt” and “pepper”. Early attempts tried to solve this problem by clustering words based on their meaning ending, representing words as high dimensional sparse vectors [6].

Most recently, a new idea was proposed inspired by the neural network language model, representing words as dense vectors, which are learned from large corpus [7-9]. These representations are often called word embeddings and the model proposed is known as Word to Vector (word2vec). These embeddings are easy to work with, since the vectors can be manipulated by a huge number of algorithms like dimensionality reduction, clustering, classification, similarity searching and many more [10]. Another advantage of the vector representation of words learned by word2vec models is the fact that they are able to carry semantic meanings being useful in various NLP tasks.

Two models have been presented in order to produce such dense embeddings of words: The Continuous Bag of Words (CBOW) [7] and the Skip-Gram [8]. Each of the two models train a network to predict neighboring words. Suppose that a sequence of words $[w_1, w_2, ..., w_{T-1}, w_T]$ is provided. The CBOW model, first randomly initializes the vector of each word and then using a single layer Neural Network whose outcome is the vector of the predicted word w_t, optimizes the original guesses. One can easily understand that the size of the Neural Network controls the size of the word vector. On the other hand, the Skip-gram model uses the word w_t, in order to predict the context words as one can see in the following Fig. 2.



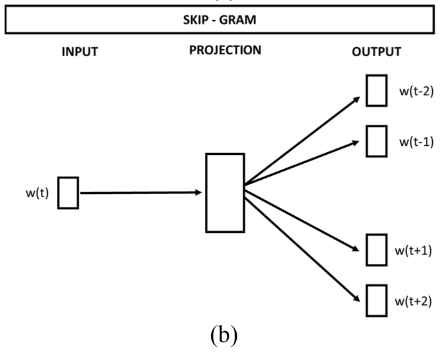


Fig. 2. The two topologies for the word2vec models (a) CBOW and (b) Skip-gram.

The prediction task can be decomposed into the following formulas. The objective of the word2vec model is to maximize the average log probability [13]:

$$\frac{1}{T}\sum_{t=l}^{T-l}\log p\left(w_t \mid w_{t-l}, ..., w_{t+l}\right) \tag{1}$$

The word w_t can be easily be predicted using a multiclass classifier such as the SoftMax:

$$p\left(w_t \mid w_{t-l}, ..., w_{t+l}\right) = \frac{e^{y_{w_t}}}{\sum_i e^{y_{w_i}}} \tag{2}$$

where each of the terms y_{w_i} is the unnormalized log-probability for each output word w_i and is computed by:

$$y = a + bf(w_{t-l}, ..., w_{t+l}; V) \tag{3}$$

where a, b are the SoftMax parameters and f is constructed by a concatenation or average of words vectors extracted from the vocabulary matrix V. The hierarchical SoftMax can be used in order to speed up the training process [8], [11], [12].

A year after the proposed word2vec model the paragraph2vec and finally the doc2vec models were presented by the same research team [13]. The paragraph2vec model is similar to word2vec model where a whole paragraph can be represented by a single vector. Paragrpah2vec uses the same principle in a way that it treats each sentence as one vector by concatenating the paragraph vector with the word vectors. An important advantage of paragraph vectors is that they are learned from unlabeled data and thus can work well for tasks that do not have enough labeled data. First, they inherit an important property of the word vectors: the semantics of the words. In order to demonstrate this effect a search for the nearest vectors to the following vectors-words “astronomy”, “power”, “pixel” and “cancer”, was performed and the results are presented in the following Table 1.

Table 1. Examples of Nearest neighbor words.

Word	1st Nearest Word	2nd Nearest Word	3rd Nearest Word
“astronomy”	observatory	telescope	astrophysical
“power”	efficiency	energy	mw
“pixel”	image	cmos	detector
“cancer”	tumor	breast	cells

D. *Multi-Label Classification*

Multi-label classification originated from the text classification field [14] and the need to be able to assign more than one labels to a text (e.g. in our case a thesis that can be related both to mathematics and mechanical engineering) and since then, has found many applications in various fields ranging from image annotation to fault diagnosis [15].

Unlike conventional classification problems, in a multi-label setting each instance/object is associated with a set of labels $Y \subseteq \Lambda$ [16]-[18]. In turn multi-label problems can be considered as part of the even larger multi-output classification family.

Multi-label classification problems can be tackled using: a) problem transformation and b) algorithm adaptation. The first one involves the transformation of the multi-label classification problem into one or more conventional classification problems by transforming the original data set into more than one data sets, with each instance of the new data set having a single output. On the other hand, the algorithm adaptation problems directly attack the posed problem by adapting existing algorithms to output simultaneously more than one classes/labels. In this work, only algorithms of the first variety are tested and more specifically, the Binary Relevance (BR) method and the Classifier Chains (CCs) method, which is an extension of the BR method.

Binary Relevance

BR is the simples and most intuitive approach to multilabel classification. It is a one vs all approach that involves the training of q binary classifiers, where q is the number of unique classes/labels. Each classifier is trained to recognize whether an instance belongs to a specific class. During the recall phase, all the labels that are assigned by the independent classifiers are aggregated to form the set of labels of that specific instance. BR is fast and it is scalable. However, it fails to take into consideration potential correlations between the labels classes.

Classifier Chains

CCs is an extension of the BR method. It also uses binary classifiers, but this time in a cascade manner; each time the output of a binary classifier takes as input the outputs of the classifiers previously encountered in the chain. This way the classifier is made aware of potential correlations between the labels.

Evaluation measures

As in every learning problem, multilabel classification requires a set of measures in order to assess and compare the performance of the different proposed methods. However, the existence of a set of relevant outputs instead of a single one, requires the definition of extensions of methods conventionally involved in classification problems. These measures are primarily divided into two groups: example based and label based. In the rest of the section we present some of the more often encountered ones in the multilabel literature.

Accuracy

This is the ratio of the size of the union and intersection of the predicted and actual label sets (represented by the logical AND ($\wedge$) and OR ($\vee$) operations in bit-vector notation, respectively), taken for each example, and averaged over the number of examples (example based measure).

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i \wedge \hat{y}_i|}{|y_i \vee \hat{y}_i|} \quad (4)$$

Hamming loss

Hamming loss it also an example based measure:

$$\text{Hamming-Loss} = \frac{1}{Mq}\sum_{i=1}^{M}|\hat{y}_i \Delta y_i| \quad (5)$$

where M is the number of examples, q is the number of labels and Δ stands for the symmetric difference of two sets. It evaluates the fraction of misclassified instance-label pairs, i.e. a relevant label is missed or an irrelevant is predicted [19]

In the case of label based measures, each class label is treated separately and the quantities of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) involving that label are calculated. Using those four values micro and macro averages values for almost all binary classification measures can be calculated.

a) Macro-averaging

$$B_{macro} = \frac{1}{q}\sum_{j=1}^{q} B\left(TP_j, FP_j, TN_j, FN_j\right) \quad (6)$$

b) Micro-averaging

$$B_{macro} = \frac{1}{q}\sum_{j=1}^{q} B\left(TP_j, FP_j, TN_j, FN_j\right)\ , \quad (7)$$

where B is a binary measure of accuracy. In this work, the F_1 measure was selected, because it combines both precision and recall [18]:

$$F_1 = 2\frac{precision \times recall}{precision + recall}\ , \quad (8)$$

III. Results and Discussion

In order to evaluate the different methods proposed in this paper, both for text representation and multilabel classification, all the sets from the MIT's DSpace repositories where harvested using OAI-PMH. For each one of these (approximately 19,000) records the OAI DC and METS metadata formats where harvested. For each record, the dc:description and dc:subject elements were of particular interest as the former contained the abstract and the latter, the set of subject terms associated with the record. 47 topics were collected for the 19,000 records, which are presented in Table 2.

For each one of the four combinations (BoW-BR, BoW-CCs, doc2vec-BR and doc2vec-CCs) the original dataset was divided in a training and a testing set, using the fixed parameter set described in section 2 and logistic regression classifier as the basic classifier. The results are summarized in Table 2 and Table 3. As it can be seen the distributed representation consistently outperforms the BoW representation. Moreover, the CCs has a profound effect on the performance of the system. This was expected since in the specific application the labels/topics are usually correlated and BR is known for its inability to capture such correlations.

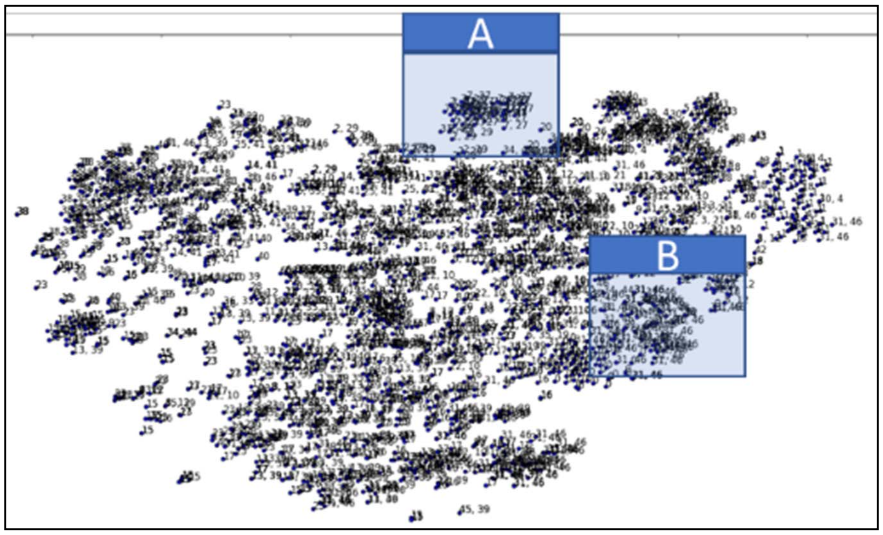
Table 1. Comparison of the BoW-BR vs. doc2vec-BR

	Accuracy	B_{macro}	Hamming-Loss	B_{micro}
BoW-BR	0.097	**0.567**	0.182	0.175
doc2vec-BR	**0.219**	0.538	**0.062**	**0.353**

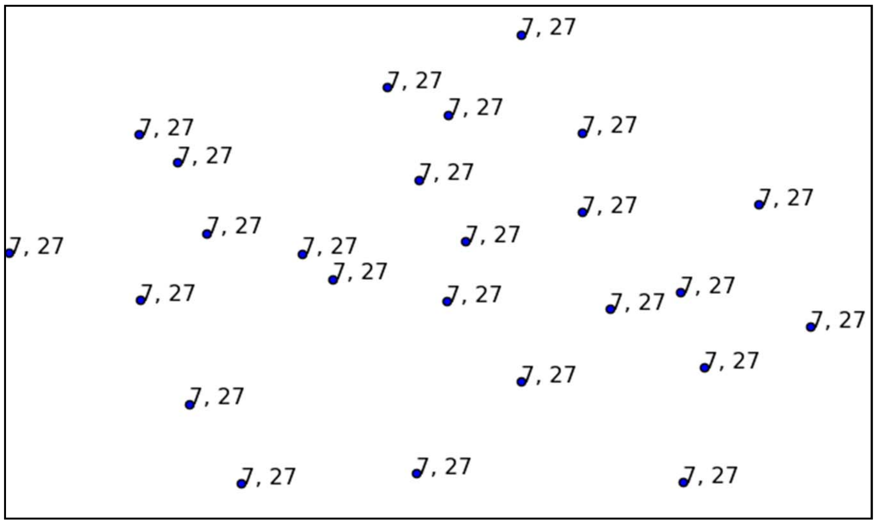
Table 2. Comparison of the BoW-CC vs. doc2vec-CC

	Accuracy	B_{macro}	Hamming-Loss	B_{micro}
BoW-CC	0.555	0.556	0.026	0.571
doc2vec-CC	**0.606**	**0.610**	**0.022**	**0.640**

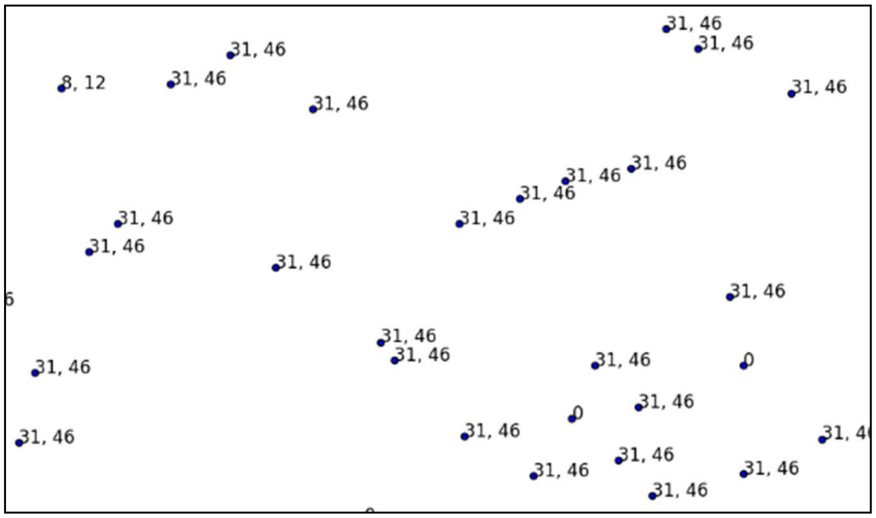
Furthermore, an insight of the effectiveness of the learned vectors can be given by projecting them down to two dimensions using t-Distributed Stochastic Neighbor Embedding (t-SNE) [20]. t-SNE is particularly well suited for the visualization of high-dimensional datasets such as word embeddings. As it can be seen in Fig. 3 it becomes apparent a number of clusters are formed in the two dimensions and almost all the documents of each cluster share the same topics.



(a)



(b)



(c)

Fig. 3. Visualization of the document vectors using t-SNE. (a) The document vectors projected into two dimensions where each document is represented by a dot and the coded topics are on the top of each document-point (b) Area A: A cluster of documents for topics 7 (linguistics,) and 27 (philosophy) and (c) Area B: a cluster of documents for topics 31 (electrical engineering) and 46 (computer science).

IV. Conclusions

This work presented an automatic approach for topic recommendation of PhD thesis documents based solely on their abstracts. Two different text representations were tested in combination with a multilabel framework. Our initial results suggest that doc2vec representations offer better representation information compared to conventional BoW yielding quite high performance when combined with a multilabel classification scheme that takes into consideration the correlation between the assigned topics.

Given that the CCs does not scale so well with increased number of labels in future work more scalable approaches will be tested, along with other text representation options.

References

[1] R. Tansley, M. Bass, D. Stuve, M. Branschofsky, D. Chudnov, G. McClellan and M. Smith, "The DSpace institutional digital repository system: current functionality," *In Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries (JCDL '03)*, IEEE Computer Society, Washington, DC, pp. 87-97, USA (2003).

[2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 96, pp. 391-407, 1990.

[3] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," *In Proceedings of the international workshop on artificial intelligence and statistics, Society for Artificial Intelligence and Statistics*, pp. 246–252, 2005.

[4] G. Miner, J. Elder, T. Hill, R. Nisbet, D. Delen and A. Fast, "Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications," *Academic Press*, 2012.

[5] D. Bamman and N. Smith, "Contextualized Sarcasm Detection on Twitter," *In Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM 2015)*, Oxford, UK, 2015.

[6] P. Brown, P. deSouza, R. Mercer, V. Pietra, J. Lai, "Class-based n-gram models of natural," *Computational Linguistics*, vol. 18, no. 4, 1992.

[7] T. Mikolov, K. Chen, G. Corrado, and Dean, "Efficient estimation of word representations in vector space," *In ICLR*, 2013.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *In NIPS*, pp. 3111–3119, 2013.

[9] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," *In NAACL HLT*, pp. 746–751, 2013.

[10] C. Aggarwal, "Recommender Systems", *Springer International Publishing*, Switzerland, 2016.

[11] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," *In Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pp. 246–252, 2005.

[12] A. Mnih and G. Hinton, "A scalable hierarchical distributed language model," *In Advances in Neural Information Processing Systems*, pp. 1081–1088, 2008.

[13] Q. Le, T. Mikolov, "Distributed Representations of Sentences and Documents," *Proceedings of The 31st International Conference on Machine Learning*, pp. 1188–1196, 2014.

[14] R. Schapire, Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2, pp.135-168, 2000.

[15] G. Georgoulas, V. Climente, J. Antonino-Daviu, I. Tsoumas, C. Stylios, A. Arkkio and G. Nikolakopoulos, "The use of a Multi-label Classification Framework for the Detection of Broken Bars and Mixed Eccentricity Faults based on the Start-up Transient," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 625-634, 2016.

[16] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data. In Data mining and knowledge discovery handbook", *Springer*, pp. 667-685, 2009.

[17] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int J Data Warehousing and Mining*, pp. 1–13, 2007.

[18] J. Read, "Scalable multi-label classification," Ph.D. thesis, University of Waikato, 2010.

[19] M. Zhang, and Z. Zhou, "A review on multi-label learning algorithms," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 8, pp. 1819-1837, 2014.

[20] L. Maaten and G. Hinton, "Visualizing High-Dimensional Data Using t-SNE," *Journal of Machine Learning Research*, pp. 2579-2605, 2008.