ISSN 2007-9737

Similarity Driven Unsupervised Learning for Materials Science Terminology Extraction

Sapan Shah, Sarath S, Sreedhar Reddy

TRDDC, TCS Innovation Labs, Pune,
India

{sapan.hs, sarath.s8@, sreedhar.reddy}@tcs.com

Abstract. Knowledge of material properties, microstructure, underlying material composition and manufacturing process parameters that the material has undergone is of significant interest to materials scientists and engineers. A large amount of information of this nature is present in the form of unstructured sources. To access the right information for a given problem at hand, various domain specific search systems have been developed. Domain terminologies, when available, can significantly improve the quality of such systems. In this paper, we propose a novel similarity driven learning approach for automatic terminology extraction for materials science domain. It first uses various intra-domain and inter-domain unsupervised corpus level features to score and rank candidate terminologies. For inter-domain features, we use British National Corpus (BNC) as the general purpose corpus. The ranked candidate terms are then used to generate training data for learning a similarity based scoring function. The parameters of this scoring function are learnt using a Siamese neural network which uses word embeddings learnt from both the domain as well as the general purpose corpora to leverage contrasting term features. The proposed similarity based learning approach consistently outperforms other reported classification approaches on the materials dataset.

Keywords. Terminology extraction, computational terminology, domain specific search, natural language processing.

1 Introduction

A material's properties depend not only on the chemical composition of the material, but also on its internal structure. The structure in turn depends on the processes performed on the material. Knowledge of composition-process-structure-property relationships is therefore central to the success of materials engineering. A large body of knowledge of this kind is available in the form of publications, company reports, and so on, that capture results from experiments and simulations.

However, finding the right information from this large body that is relevant for a given problem is not an easy task. First, one has to sift through and select right set of documents. Then one has to scan through these documents to extract pieces of information that are relevant to the problem. Traditional search engines are not very helpful here as they are keyword centric and weak on relation processing [15, 16].

Suppose an engineer wants to know what composition of steel gives him a minimum hardness of 40RC when the annealing temperature is in the range of 500-$600°$C. A simple keyword based search for "steel and composition and hardness 40RC and annealing temperature 500-$600°$C" will not be very helpful. It will simply retrieve all the documents where the terms steel, composition, hardness, annealing, temperature, 40, 500, 600 appear somewhere in the document without necessarily being related. For instance, 50 need not be related to hardness and 500 need not be related to annealing temperature, resulting in lot of noise. What we need is an intelligent search engine that understands value relations.

To address this need, various domain specific search systems have been proposed in the literature [15, 20]. These systems are not

doi: 10.13053/CyS-23-3-3266

just keyword centric but also understand domain entities and relations to improve search accuracy. In materials science domain, [20] have developed a system that supports value constraint queries on materials entities. For example, one can use the query "steel & composition & annealing temperature:$[500, 600]°$C & hardness $\geq$ 40RC" for the case discussed above. The search engine looks for mentions of domain concepts in the text and extracts and indexes these mentions and relations between them. It also extracts values associated with the mentions and indexes them.

The generated index is then used for processing user queries. The search engine uses domain dictionaries to identify domain concepts of interest. These dictionaries are usually supplied by domain experts. However, a domain such as materials science is large and continuously growing, so expecting users to supply complete and up to date dictionaries is impractical. Tools that can automatically mine and extract domain terminologies are of great help in this context as they can serve as building blocks for constructing domain dictionaries [13].

1.1 Automatic Domain Terminology Extraction

Various approaches to terminology extraction can broadly be classified into supervised, weakly supervised and unsupervised methods. Supervised approaches cast this as a binary classification problem [8, 5, 24]. However, they need a large amount of labelled data for learning. This is hard to come by for an application domain such as materials science. Weakly supervised approaches on the other hand rely on small labelled data and a large pool of unlabelled data to learn classification models in an iterative manner. For instance, co-training based approaches [4]. However, these approaches suffer from the problem of *semantic drift* [19] wherein if non-domain terms are incorrectly added to the labelled data during earlier iterations, the later iterations are adversely affected and this downgrades the overall quality of the extracted terminologies.

Unsupervised approaches such as [23] primarily depend on various unsupervised corpus level intra-domain and inter-domain base features such as C-value, TF-IDF, domain relevance, etc. Sophisticated scoring functions are then defined using these base features that try to capture termhood and unithood [11] of various domain terminologies. Co-training based approaches are also proposed in literature where the labelled data is generated from base features in a fully unsupervised manner. This is essentially done by ranking the candidate terms using the scoring function and taking the top p terms as positive examples and bottom p as negative examples. The parameter p is critical to the performance of the classifier. With larger p, the distinction between positive and negative examples blurs, and with smaller p, we do not have enough training data.

In this paper, we propose a novel similarity driven learning approach as opposed to standard classification based approaches for unsupervised terminology extraction. In this approach, as against taking top p terms as positive examples and bottom p as negative, we take pairs of terms: we pair top p terms with each other to generate *similar* data set, and we pair top p terms with bottom p terms to generate *dissimilar* data set. Thus we have p^2 positive examples and p^2 negative examples, significantly increasing the training data size. This allows us to choose a small enough p that sharply delineates positive terms from negative terms.

We first use various corpus level statistical features to score and rank candidate terms. We use both intra-domain and inter-domain features for this purpose. For inter-domain features, we use British National Corpus (BNC) [1] as the general purpose corpus. Inter-domain features essentially measure the contrastive nature of domain specific terms. The ranked candidate terms are then used to generate training data for learning a similarity based scoring function. The parameters of the scoring function are learnt using a Siamese neural network [10] that uses word embedding representations of the candidate terms. We use two embeddings to represent a term - one learnt from the domain corpus and the other from the general corpus to leverage contrasting features present in the two corpora.

The proposed Siamese network based method has been compared with standard baselines

[1] available at http://www.natcorp.ox.ac.uk/

such as C-value, domain relevance, etc., used in terminology extraction literature. It is also compared with a co-training based unsupervised fault tolerant learning approach proposed by [23]. Our method outperforms both the baselines as well as the co-training approach. To evaluate the effectiveness of similarity driven learning, we also compare our model with a standard feed forward network having similar complexity.

The rest of the paper is organized as follows. Section 2 discusses the relevant related work. Section 3 explains pre-processing steps to identify candidate terminologies as well as the base unsupervised features used by our model. Section 4 describes the similarity driven learning approach and details the learning task for Siamese network based scoring function. Section 5 describes the evaluation dataset and discusses the experimental results. Section 6 summarizes this work and indicates future work directions..

2 Related Work

Fault Tolerant Learning (FTL) [23] is a completely unsupervised iterative learning technique for term extraction leveraging ideas from co-training [4] and transfer learning [3]. FTL trains two support vector machine classifiers separately, where predictions from one classifier are verified by the other to improve term extraction performance. Input data for one classifier is generated using the TF-IDF measure whereas the other classifier uses delimiter candidate term extraction. [22] have proposed a weakly supervised co-training based approach where they focus on learning multiple representations for terms by composing constituent words using convolutional neural network and recurrent neural network based classifiers. However, these iterative co-training approaches primarily treat term extraction as a binary classification problem. The method proposed in this paper instead learns a similarity based scoring function that captures feature similarities of domain terms as opposed to discriminating domain terms from non-terms.

A closely related work that combines word embeddings from domain specific as well as general purpose corpora is by [2]. However, the local-global vector based approach suggested by the authors only considers unigram terminologies and build a pure classification model as opposed to a similarity driven model such as the one proposed in this paper.

3 Pre-Processing

Automatic terminology extraction methods first employ various linguistic and statistical filters to identify candidate terminologies. The standard filters used in the literature include Parts of Speech (PoS) tag filter, stop words filter, frequency filter, and so on. Once the set of candidate terms are identified, scores for various unsupervised corpus level intra-domain and inter-domain features are computed. Following describes various features used by our model.

3.1 Intra-domain Features

This category of features are important in bringing out the terms that are most frequent within a particular domain. They are primarily statistical in nature. The following is a summary of the intra-domain features used in our model.

— TF-IDF [18]: It is a product of TF (frequency of term within a document) and IDF (Inverse Document Frequency - the number of documents in which a term occurs). For the purpose of term extraction, we take the average TF-IDF values across all documents in the corpus as TF-IDF feature.

— C-value [9]: This unithood feature scores candidate terms using a combination of the following criteria: assigns higher scores to more frequent terms; penalizes candidate terms if they occur as substrings of larger candidate terms; assigns higher scores to longer candidate terms.

— Term Variance (TV) [6]: It scores a candidate term by measuring its variance across all documents in the corpus. It discriminates between high frequency non-terms appearing in all documents from terms that occur frequently in a small set of documents.

3.2 Inter-domain Features

The inter-domain features used by our model include,

- Domain relevance [7]: It compares the frequency of candidate terms in domain corpus and general corpus.
- Relevance [7]: It improves domain relevance by down weighting candidate terms that occur rarely in domain corpus or occur highly frequently in general corpus.
- Weirdness [1]: This measure is similar to domain relevance but takes relative frequencies into account by considering dataset sizes.

The features described above are expected to score candidate terms such that true domain terms are assigned higher scores compared to non-terms. Accordingly, candidate terms are ranked in descending order by the score value. This ranked list is then used to measure *precision@k* which computes the number of correct domain terms identified among the top *k* candidate terms in the list.

An issue with all these statistical features is that they are very sensitive to term frequency, so they fail to identify terms that lack statistical significance. Hence they need to be augmented with learning based approaches that learn to identify other aspects of similarity to distinguish domain terms from non terms. The ranked list produced by the base features can serve as the starting point to identify a seed list of positive and negative examples to train a classifier.

3.3 Pre-trained Embeddings

For the learning phase, we represent the candidate terms using pre-trained word embeddings. Inspired by the local-global vector based approach proposed in [2], we use pre-trained 100 dimensional GloVe [17] vectors to represent statistical strength of words as they appear in general corpus (referred as *general* vectors). Whereas, word embeddings capturing domain semantic similarity are learnt using domain specific text corpus (referred as *domain* vectors). A unigram term in our system is represented by concatenating its general and domain vectors. The input representation for multiwords then corresponds to concatenation of constituent unigram terms. We currently consider multi grams with maximum size 3. The rest of the paper refers to this representation as *pre-trained* vector.

4 Proposed Similarity Driven Scoring Function

As mentioned earlier, our approach leverages feature similarity across domain terms to learn a term scoring function. Figure 1 shows the Siamese network architecture (referred as SNet) used to learn this function. The network takes a pair of terms as input and outputs a similarity score between them. Training instances for learning parameters of this network are generated in the following way.

1. The scoring functions explained in section 3 are used to generate a ranked list of candidate terms. Domain terms are expected to be ranked higher in this list.
2. Top p terms from the ranked list are denoted as *positive terms* whereas bottom p terms are denoted as *negative terms*.
3. Total $p \times (p-1)$ pairs of terms are generated using *positive terms* and assigned similarity score of 1.
4. Total $p \times p$ pairs of terms are generated by taking the cross product of the terms present in the positive and negative term sets. These pairs are assigned a score of 0.
5. The word pairs generated in step 3 and 4 constitute the training data for learning the parameters of the Siamese network in figure 1.

The data generation framework discussed above assumes that the terms closer to the top of the ranking are likely to be true domain terms and the terms closer to the bottom are likely to be non-domain terms. Hence the selection of the top p terms as positive terms and the bottom p as negative terms.

The parameters of the Siamese network are learnt using stochastic gradient descent with various choices for distance functions such as Euclidean distance and Manhattan distance. Once the network parameters are learnt, all remaining candidate terms are scored in the following way: a total of p term pairs are formed by taking the cross product of the given candidate term with all *positive terms*. These pairs are then passed through the network in figure 1 to compute their similarity scores. The average similarity score across these p pairs is then used as the score for the candidate term. Once the scores for all candidate terms are computed, they are ranked in descending order to generate a ranked list of domain terms.

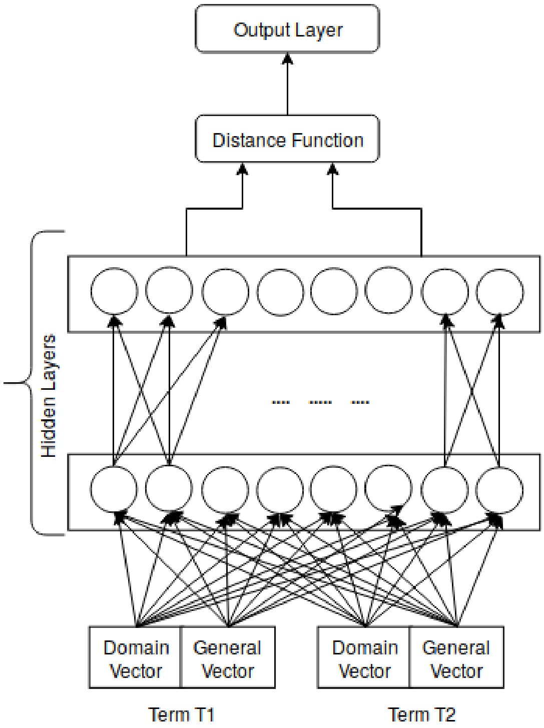

Fig. 1. Siamese Network architecture for scoring similarity between term pairs

For comparison, we also use a simple feed forward neural network (referred as FFNet) architecture. Top p terms ranked by the base features are marked as domain terms with output 1 and bottom p terms are marked as negative terms with output 0. These terms are then used to learn the network parameters. Similar to SNet, this network also takes the concatenated *pre-trained* vector of the candidate terms as input. It then uses binary cross-entropy loss to optimize network parameters. Once the parameters are learnt, all the remaining candidate terms are scored using this network and ranked in descending order to generate a ranked list of domain terms. Note that for the same p positive and p negative terms, FFNet has only $2p$ training examples, whereas SNet has $2p^2$ examples. Also the SNet approach relies on average similarity with all p positive terms.

5 Experimental Evaluation

Since materials science is the focus of our work, we use a materials science corpus for domain terminology extraction. We use British National Corpus (BNC) as the general purpose corpus.

5.1 Dataset

The text corpus used for term extraction consists of 1000 publications downloaded from ISIJ[2] International Journal. This Journal contains publications on fundamental and technological aspects of the properties, structure, characterization, processing, etc. of iron, steel and other related engineering materials. The downloaded publications are in the PDF format. We first convert these PDF files to text using Grobid [14].

Following filters are then applied on the converted text: PoS tag filter: $((\text{Adj})?(\text{Noun})+)|((\text{Adj}|\text{Noun})*(\text{Verb})?)$; stop words filter; frequency filter with minimum term frequency of 10; and shallow stemming that only converts plural forms to singular. This resulted in a total 17000 candidate terms. Few example candidate terms are: *quenching*, *quenching temperature*, *grain size*, *elongation* and *tensile strength*. The filters used for candidate term extraction have been designed by analysing few sample documents in the corpus. We use

[2] The Iron and Steel Institute of Japan - https://www.jstage.jst.go.jp/browse/isijinternational/-char/en

the pre-trained domain vectors[3] developed by [12] for materials science domain. Whereas for general vectors, we use pre-trained GloVe [17] embeddings[4].

5.2 Comparison with Previous Work

The unsupervised features described in section 3 serve as the baseline. We then compare the performance of SNet and FFNet with a simple voting algorithm and classification based algorithms such as Fault Tolerant Learning (FTL) [23] and Single Classifier (SC). The voting algorithm [25] simply uses the rankings produced by the base features. The score for a term is computed by summing the inverse of its rank in the participating base features. We use intra-domain and inter-domain features to provide different views of data for FTL approach. It starts with an initial list of s seed terms to bootstrap the classifiers. It then iteratively adds n high confidence terms to the seed list until convergence. SC is a non-co-training version of FTL that uses only a single classifier. The results were manually evaluated by three domain experts. We use *Precision@k* as the evaluation metric.

5.2.1 Experimental Setting

The SNet architecture for the materials science domain contains a single fully connected hidden layer. A distance function is then applied on the output of the hidden layer followed by a sigmoid activation. Binary cross entropy loss is then minimized using RMSProp stochastic gradient descent (SGD) [21]. The network also applies ReLU activation for the units in the hidden layer along with dropout regularization. Grid search used for hyper parameter tuning consists of: hidden layer units in {6, 8, 10, 12}; distance function in {euclidean distance, manhattan distance}; dataset size parameter p in {100, 200}; data generation features in {all-features, single best feature from each category namely C-value for intra-domain and domain relevance for inter-domain}. For data generation, the scores for multiple features are combined by taking the average of their normalized scores. The best features from the two categories are decided by considering their *precision@2000*. The architecture for FFNet also consists of a single hidden layer with similar details except for the number of positive and negative terms. This has been varied among {100, 200, 400}. For SNet, the best hyper parameter combination was found to be 8 hidden units, manhattan distance and all-features for data generation with $p = 100$; for FFNet, it was found to be 8 hidden units and 200 positive and negative terms.

Similarly we have performed grid search for hyper parameters of FTL, SC and voting algorithm. In FTL, the initial seed terms (s) are varied among {200, 400, 500, 800, 1000} and the number of terms added in each iteration (n) are varied among {20, 50, 80, 100, 150}. Different views for the classifiers are provided by using only the best intra-domain feature for one classifier and the best inter-domain feature for the other. We have also tried using combinations of best features for generating seed term list. SC also uses a similar parameter setting. For voting algorithm, we have tried the following configurations for base features: all features; top 2 features; top 2 intra-domain features; and top-2 inter-domain features.

5.3 Results and Discussion

Table 1 shows the results of our experiments. The similarity based model implemented by SNet outperforms voting algorithm, classification models such as FFNet, SC and the co-training approach of FTL. For smaller values of k such as 200, 500, the base unsupervised features have better accuracy with domain relevance giving the best results. This is to be expected as these are frequency based measures and the terms ranked closer to the top are more likely to be domain terms. However, the terms appearing later in the ranked lists for these features are not reliable. The classification and similarity based models give superior results in this case. Again this is on expected lines as these approaches learn to discern other aspects of similarity among positive and negative terms.

[3] downloaded from https://github.com/olivettigroup/materials-word-embeddings

[4] downloaded from https://nlp.stanford.edu/projects/glove/

Table 1. Evaluation of term extraction approaches using *precision@k*

Method	k=200	k=500	k=1000	k=1500	k=2000	k=3000
Unsupervised Features						
C Value	0.64	0.648	0.639	0.636	0.634	0.613
TFIDF	0.825	0.714	0.64	0.594	0.57	0.549
Term Variance Quality	0.77	0.71	0.654	0.629	0.593	0.577
Domain Relevance	**0.89**	**0.868**	0.792	0.749	0.720	0.695
Relevance	0.52	0.55	0.554	0.556	0.551	0.525
Weirdness	0.625	0.614	0.599	0.590	0.566	0.541
Proposed Methods and Previous Works						
Fault Tolerant Learning	0.820	0.830	0.804	0.769	0.725	0.725
Single Classifier	0.840	0.868	0.791	0.772	0.703	0.705
Voting Algorithm	0.885	0.866	0.791	0.748	0.720	0.694
FFNet	0.776	0.818	0.785	0.771	0.767	0.759
SNet	0.761	0.822	**0.821**	**0.815**	**0.806**	**0.764**
FFNet_dict	0.831	0.832	0.802	0.817	0.781	0.755
SNet_dict	0.856	0.858	0.825	0.815	0.797	0.765

This is evident from the table for values of $\mathrm{k} > 500$. It should also be noted that SNet consistently beats FFNet with about 3% accuracy improvement even though both models have similar network complexity (in terms of number of parameters). This can be attributed to three reasons:

— SNet's architecture is designed to explicitly learn similarity.

— Instead of relying on a single classification decision, SNet brings in ensemble effect by averaging over the similarity scores computed from the top p high confidence terms.

— For similar network complexity, due to pairing SNet has a much larger dataset available for learning parameters.

In many practical scenarios, small amount of domain terminologies are often available. For instance, in the form of domain dictionaries or lexicons. These existing domain terminologies can be exploited to improve the terminology extraction algorithms. To study the effect of such lexicon, we created a small dictionary of material properties and manufacturing processes. The terms present in this dictionary are added to the list of *positive terms* as part of dataset creation. Table 1 shows results for the classification (FFNet_dict) and similarity (SNet_dict) based models.

For smaller values of k such as 200, 500, these models perform better than FFNet and SNet. This is due to the fact that the dictionary aided models use true domain terms in addition to the terms suggested by unsupervised features. Due to this, the terms which are similar to lexicon (i.e. material properties and processes) are ranked higher. However, the number of terms representing material properties and processes is finite and not very large. Due to this, terms of various other categories (for instance, microstructural features) appear in the ranked list for higher values of k making domain lexicon less effective. This is observed in the table for values of $\mathrm{k} > 500$, where the accuracy of both SNet and FFNet approach SNet_dict and FFNet_dict respectively.

6 Conclusion and Future Work

This paper proposes a novel similarity driven learning approach for materials science terminology extraction. It uses various unsupervised features to generate training data. A similarity based scoring function is then learnt using Siamese network architecture. The proposed approach outperforms standard classification as well as co-training approaches on materials dataset. Our future work consists of generating typed dictionaries from these terminologies. We are also planning to improve term extraction further by exploiting compositional nature of multiword terms.

References

1. **Ahmad, K., Gillam, L., & Tostevin, L. (1999).** University of surrey participation in trec8: Weirdness indexing for logical document extrapolation and retrieval (wilder). *The Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland.
2. **Amjadian, E., Inkpen, D., Paribakht, T., & Faez, F. (2016).** Local-global vectors to improve unigram terminology extraction. *Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016)*, The COLING 2016 Organizing Committee, Osaka, Japan, pp. 2–11.
3. **Ando, R. K. & Zhang, T. (2005).** A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, Vol. 6, pp. 1817–1853.
4. **Blum, A. & Mitchell, T. (1998).** Combining labeled and unlabeled data with co-training. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, ACM, New York, NY, USA, pp. 92–100.
5. **Conrado, M., Pardo, T., & Rezende, S. (2013).** A machine learning approach to automatic term extraction using a rich feature set. *Proceedings of the 2013 NAACL HLT Student Research Workshop*, Association for Computational Linguistics, Atlanta, Georgia, pp. 16–23.
6. **Dhillon, I., Kogan, J., & Nicholas, C. (2004).** Chapter 4: Feature selection and document clustering.
7. **Fedorenko, D. G., Astrakhantsev, N., & Turdakov, D. (2013).** Automatic recognition of domain-specific terms: an experimental evaluation. **Vassilieva, N., Turdakov, D., & Ivanov, V.**, editors, *Proceedings of the Ninth Spring Researchers Colloquium on Databases and Information Systems, Kazan, Russia, May 31, 2013*, volume 1031 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 15–23.
8. **Foo, J. & Merkel, M. (2010).** Using machine learning to perform automatic term recognition. *LREC 2010 Workshop on Methods for automatic acquisition of Language Resources and their evaluation methods*, Valletta, Malta, pp. 49–54.
9. **Frantzi, K., Ananiadou, S., & Mima, H. (2000).** Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, Vol. 3, pp. 115–130.
10. **Hadsell, R., Chopra, S., & LeCun, Y. (2006).** Dimensionality reduction by learning an invariant mapping. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, IEEE Computer Society, Washington, DC, USA, pp. 1735–1742.
11. **Kageura, K. & Umino, B. (2001).** Methods of automatic term recognition - a review. *Terminology*, Vol. 3.
12. **Kim, E. Y., Huang, K., Tomala, A., Matthews, S., Strubell, E., Saunders, A., McCallum, A. L., & Olivetti, E. (2017).** Machine-learned and codified synthesis parameters of oxide materials. *Scientific data*.
13. **Krauthammer, M. & Nenadic, G. (2004).** Term identification in the biomedical literature. *Journal of Biomedical Informatics*, Vol. 37, No. 6, pp. 512–526.
14. **Lopez, P. (2009).** Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. *Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL'09, Springer-Verlag, Berlin, Heidelberg, pp. 473–474.
15. **Magle, T. C. (2015).** A review of quetzal: A linguistic search engine for biomedical literature. *American Laboratory*. Accessed: 2018-12-01.
16. **McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (1999).** A machine learning approach to building domain-specific search engines. *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 662–667.
17. **Pennington, J., Socher, R., & Manning, C. D. (2014).** Glove: Global vectors for word representation. *EMNLP*.
18. **Rajaraman, A. & Ullman, J. D. (2011).** *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA.

19. **Riloff, E. & Jones, R. (1999).** Learning dictionaries for information extraction by multi-level bootstrapping. *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, American Association for Artificial Intelligence, Menlo Park, CA, USA, pp. 474–479.

20. **Shah, S., Vora, D., Gautham, B. P., & Reddy, S. (2018).** A relation aware search engine for materials science. *Integrating Materials and Manufacturing Innovation*, Vol. 7, No. 1, pp. 1–11.

21. **Tieleman, T. & Hinton, G. (2014).** *RMSprop Gradient Optimization (course slides).* Uiversity of Toronto.

22. **Wang, R., Liu, W., & McDonald, C. (2016).** Featureless domain-specific term extraction with minimal labelled data. *Proceedings of the Australasian Language Technology Association Workshop 2016*, Melbourne, Australia, pp. 103–112.

23. **Yang, Y., Yu, H., Meng, Y., Lu, Y., & Xia, Y. (2010).** Fault-tolerant learning for term extraction. *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, Institute of Digital Enhancement of Cognitive Processing, Waseda University, Tohoku University, Sendai, Japan, pp. 321–330.

24. **Zhang, X., Song, Y., & Fang, A. C. (2010).** Term recognition using conditional random fields. *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering(NLPKE-2010)*, pp. 1–6.

25. **Zhang, Z., Brewster, C., & Ciravegna, F. (2008).** A comparative evaluation of term recognition algorithms. *Proceedings of The sixth international conference on Language Resources and Evaluation*, pp. 28–31.

Article received on 23/02/2019; accepted on 04/03/2019.
Corresponding author is Sapan Shah.