Graph Based Automatic Domain Term Extraction

Hema Ala
LTRC, IIIT-Hyderabad, India
hema.ala@research.iiit.ac.in

Dipti Misra Sharma
LTRC, IIIT-Hyderabad, India
dipti@iiit.ac.in

Abstract

We present a Graph Based Approach to automatically extract domain specific terms from technical domains like Biochemistry, Communication, Computer Science and Law. Our approach is similar to TextRank with an extra post-processing step to reduce the noise. We performed our experiments on the mentioned domains provided by ICON TermTraction - 2020 shared task. Presented precision, recall and f1-score for all experiments. Further, it is observed that our method gives promising results without much noise in domain terms.

1 Introduction

Domain Term, is a word or group of words, carrying a special, possibly complex, conceptual meaning, within a specific domain or subject field or community. Because of their low ambiguity and high specificity, these words are also particularly useful to conceptualize a knowledge subject. For each domain, there is an essential need to identify the domain-specific terms as they play a vital role in many *Natural Language Processing Applications* such as Neural Machine Translation(NMT) (Dinu et al., 2019), Information Retrieval (Chien, 1999), Information Extraction (Yangarber et al., 2000), Text Classification (Liu et al., 2005), etc. The task of automatically extracting domain specific terms from a given text of a certain academic or technical domain, is known as *Automatic Technical Domain Term Extraction*. This is a predominant task in NLP. Extracted terms can be useful in more complex tasks such as NMT (Dinu et al., 2019), Ontology Construction (Kietz et al., 2000; Wu and Hsu, 2002), Domain Identification, Semantic Search, Question-Answering, Word Sense Induction, etc. Several research works have been carried out to extract domain-specific terms. Most of them are either rule based (Collard et al., 2018) or dictionary based (Kim and Cavedon, 2011). Also, there are few term extraction techniques which uses machine learning algorithms (Fedorenko et al., 2014), thereby demanding a huge labelled corpus. But, the existence of domain term annotated corpus is very rare in case of many domains. Also, the availability of such huge labelled corpus is almost nil for low resource languages. Therefore, our domain term extraction approach is motivated more by unsupervised than supervised strategies. Hence, we used a Graph Based Approach which extract not only unigrams but also collocations. ***Collocations*** are expressions of multiple words which commonly co-occur in a given context than its individual word parts. These are the phrases that express stronger sentiment which can be easily captured with bigram, trigram and so on. Hence our approach is not restricted to just unigram extraction, it also considers multi-word domain terms [1]. To demonstrate the performance of our approaches, we used data provided by ICON TermTraction - 2020 shared task. The discussion and analysis on the performance of the approaches are mentioned in section 4. In this paper we performed our experiments on four domains in English. We are still in the process of exploring the possible unsupervised approaches to extract domain terms in a flexible and intuitive manner. Further, it can be applicable to all domains irrespective of any language.

2 Background & Motivation

There have been a lot of studies regarding the automatic domain term extraction. But very less work carried on unsupervised approaches that too on technical domains like, computer science, chemistry, etc. Automatic domain term extraction is a categorization or classification task where terms

[1] Covalent Bond, Amino Acid, Hydrophobic Hydrogen Bond, Artificial Intelligence, Support Vector Machines, Natural Language Processing, etc are few examples of bigram and trigram collocations

Proceedings of the 17th International Conference on Natural Language Processing: TermTraction 2020 Shared Task, pages 1–4
Patna, India, December 18 - 21, 2020. ©2020 NLP Association of India (NLPAI)

are categorized into a set of predefined domains (Velardi et al., 2001; Xu et al., 2002). Further this task is used in many NLP applications such as domain ontology construction and NMT with Domain Terminology by injecting custom terminology into neural machine translation at run time (Dinu et al., 2019). In order to effectively make use of domain terms in various applications, an ultimate approach which is fast, flexible and reliable is highly required. In spite of many contributions on automatic domain term extraction, very limited study is done so far using unsupervised approaches. Most of the explorations are done using supervised methods such as focusing on various features like contextual, domain concepts and topics to measure the semantic similarity of terms to assign domain concepts to domain-specific terms (Kim and Cavedon, 2011). Similarly, another experimental evaluation is done by comparing the performance of two existing approaches for Automatic Domain Term Recognition: *Machine Learning Method and Voting Algorithm*(Fedorenko et al., 2014). But, the major well known drawback with these supervised algorithms is that, they demand huge labelled training data. Therefore, an unsupervised algorithm is more preferable. Most of such unsupervised approaches extract domain-specific terms using frequency count (VRL, 2009). The basic underlying idea is that, in a particular domain, domain-specific terms occur with markedly higher frequency than they do in other domains, similar to term frequency patterns captured by TF-IDF. Apart from these methods, another experimental approach for domain term extraction is executed using Deep Learning where possible *term spans* within a fixed length in the sentence, is considered to predict a domain term. Deep Learning technique is proven to yield high recall and a comparable precision on term extraction task (Gao and Yuan, 2019). However, for training such Deep Learning models, an enormous training data is mandatory. Conversely, availability of this sort of corpus for diverse multilingual domain is very scarce. Our goal is to formulate a flexible and reliable approach which successfully extracts domain terms irrespective of the domain and language of a document. Accordingly, we present experiments which extract domain terms in a given document disregarding of any domain without having a dependency on labelled corpus. Our approach , *TextRank* is an inspiration from PageRank algorithm (Brin and Page, 1998). (Mihalcea and Tarau, 2004) Introduced TextRank a graph-based ranking model for text processing, and showed how this model can be successfully used in natural language applications. In particular keyword and sentence extraction. We reimplemented TextRank from Mihalcea and Tarau (2004) for extracting domain-specific terms from technical domains like, computer science , chemistry, etc by handling noise generated in the outputs. *TextRank* is merely a graph based approach where *words* are considered as nodes and the *relation* between them as edges. Based on syntactic filters, such as Parts of Speech (POS) Tags, words are selected as nodes and relation between the words is based on word co-occurrences . A window size (N) is assumed for word co-occurrences. For all words that fall in a particular window, an edge is allocated, resulting into a graph of nodes and edges. An undirected and unweighted graph is considered in our approach. This is further discussed in detail in Section 3.

3 Approach

A graph-based ranking algorithm is a way of deciding on the importance of a vertex within a graph by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information. Applying a similar line of thinking to lexical or semantic graphs extracted from natural language texts, results in a graph-based ranking model that can be applied to a variety of natural language processing applications, where knowledge drawn from an entire text is used in making local ranking/selection decisions. We implemented the TextRank algorithm described in (Mihalcea and Tarau, 2004). Mihalcea and Tarau (2004) described usage of TextRank for keyword extraction and sentence extraction but we adopted that technique for automatic domain term extraction by doing few modifications in syntactic filters, and adding a post processing step for noise removal using top 1000 common words in English from Wikipedia. we used Noun, Proper Nouns, Adjectives as syntactic filters, window size $(N = 4)$is used in all experiments and calculated precision , recall and F1 score.

TextRank is completely unsupervised, and unlike other supervised systems, it relies completely on information drawn from the text itself, which makes it easily portable to other domains, and languages. Intuitively, TextRank works well because it does not only rely on the local context of a text unit (vertex), but rather it takes into account information recursively drawn from the entire text (graph). Through the graphs it builds on texts, TextRank identifies connections between various entities in a text, and implements the concept of recommendation. A text unit recommends other related text units, and the strength of the recommendation is recursively computed based on the importance of the units making the recommendation.

The brief explanation of each step in Text Rank algorithms is given as follows, firstly the text is tokenized, and annotated with part of speech tags, for this task we used Spacy (Honnibal and Montani, 2017). To evade the excessive growth of graph size by including all possible combinations of sequences consisting of more than one lexical unit(word), we consider only single words as nodes to build the graph, with multi-word domain terms being eventually reconstructed in the post-processing step. Following, all words that pass the syntactic filter are added to the graph, and an edge added between those words that co-occur within a window of N words. After the graph construction (undirected , unweighted graph), the value of each vertex is set to 1. Next, the ranking algorithm will run on the graph for several iterations until it converges usually for 20-30 iterations, at a threshold of 0.0001(Mihalcea and Tarau, 2004). Once a final score is achieved for each vertex (for each word)in the graph, vertices are sorted in reversed order of their score then the top K words in the ranking are retained for post-processing. In our experiments we take top $K = n/3$ where n is total number of unique words in the text. In post processing step along with constructing n-grams we reduce the noise using top 1000 English words from Wikipedia. To construct n-grams from unigrams we get, first annotate the text with technical domain terms we get then retrieve the terms which occur side by side in the text.

4 Experiments & Results

We evaluate our approach on data provided by ICON TermTraction - 2020 shared task for four domains, Biochemistry, communication , Computer Science and Law. Each domain contains files with text related to that domain. In each domain we have minimum 10 files and maximum 16 files. We did experiments on individual files for the respective domain. As our approach comes under unsupervised learning, there is no requirement of training data. Results of each file in specific domain showed in table 1 , 2 , 3 for Biochemistry, Communication and Computer Science respectively. For Law do-

File	Precision	Recall	F1
1	0.15	0.45	0.22
2	0.07	0.21	0.10
3	0.17	0.35	0.23
4	0.16	0.48	0.24
5	0.07	0.67	0.13
6	0.36	0.62	0.45
7	0.22	0.57	0.32
8	0.17	0.63	0.26
9	0.22	0.63	0.33
10	0.24	0.54	0.33

Table 1: Scores of individual files in BioChemistry

File	Precision	Recall	F1
1	0.08	0.54	0.14
2	0.06	0.5	0.11
3	0.06	0.31	0.10
4	0.16	0.5	0.24
5	0.12	0.77	0.20
6	0.09	0.68	0.16
7	0.05	0.69	0.09
8	0.25	0.37	0.05
9	0.06	0.56	0.11
10	0.08	0.55	0.15

Table 2: Scores of individual files in Communication

File	Precision	Recall	F1
1	0.17	0.52	0.26
2	0.18	0.61	0.27
3	0.12	0.48	0.19
4	0.09	0.55	0.17
5	0.14	0.63	0.23
6	0.06	0.68	0.12
7	0.12	0.57	0.20
8	0.16	0.46	0.23

Table 3: Scores of individual files in Computer Science

Domain	Precision	Recall	F1
BioChemistry	0.18	0.52	0.26
Communication	0.08	0.54	0.14
Computer Science	0.13	0.56	0.20
Law	0.05	0.5	0.10

Table 4: Average precision , recall and f1-scores

main also the behaviour is same as above three domains. From the results of all domains, we can observe that Recall is very high compared to Precision, from this we can infer our algorithm is not producing much noise. In table 4 we have averaged precision , recall and f1-score for each domain. overall we got promising results for technical domain term extraction for all given domains.

5 Conclusion & Future Work

In this paper we showed a graph based approach for automatic technical domain term extraction for four technical domains(BioChemistry, Computer Science, Communication,Law). Our approach showed high recall in all cases for all domains, from this we can conclude that our model has the power to extract domain-specific terms without much noise. Our approach doesn't depend on any language dependant resources except POS tagger, hence we can adopt this method for any language. We plan to extend our approach to possible Indian languages like Telugu, Hindi etc. And we would like to improve this approach with different word relationships(edge relations like we did using co-occurrence of words in given window). One approach for that is like using similarity of words using word2vec etc.

References

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine.

L-F Chien. 1999. Pat-tree-based adaptive keyphrase extraction for intelligent chinese information retrieval. *Information processing & management*, 35(4):501–521.

Jacob Collard, TN Bhat, Eswaran Subrahmanian, Ram D Sriram, John T Elliot, Ursula R Kattner, Carelyn E Campbell, and Ira Monarch. 2018. Generating domain terminologies using root-and rule-based terms 1. *Washington Academy of Sciences. Journal of the Washington Academy of Sciences*, 104(4):31–78.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. *arXiv preprint arXiv:1906.01105*.

Denis Fedorenko, N Astrakhantsev, and D Turdakov. 2014. Automatic recognition of domain-specific terms: an experimental evaluation. *Proceedings of the Institute for System Programming*, 26(4):55–72.

Yuze Gao and Yu Yuan. 2019. Feature-less end-to-end nested term extraction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 607–616. Springer.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jörg-Uwe Kietz, Raphael Volz, and Alexander Maedche. 2000. Extracting a domain-specific ontology from a corporate intranet. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Su Nam Kim and Lawrence Cavedon. 2011. Classifying domain-specific terms using a dictionary. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 57–65.

Tao Liu, Xiao-long Wang, Y Guan, Zhi-Ming Xu, et al. 2005. Domain-specific term extraction and its application in text classification. In *8th Joint Conference on Information Sciences*, pages 1481–1484.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Paola Velardi, Michele Missikoff, and Roberto Basili. 2001. Identification of relevant terms to support the construction of domain ontologies. In *Proceedings of the ACL 2001 Workshop on Human Language Technology and Knowledge Management*.

NICTA VRL. 2009. An unsupervised approach to domain-specific term extraction. In *Australasian Language Technology Association Workshop 2009*, page 94.

Shih-Hung Wu and Wen-Lian Hsu. 2002. Soat: a semi-automatic domain ontology acquisition tool from chinese corpus. In *COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*.

Feiyu Xu, Daniela Kurz, Jakub Piskorski, and Sven Schmeier. 2002. A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. In *LREC*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.