Automatic Technical Domain Identification

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

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

Abstract

In this paper we present two Machine Learning algorithms namely Stochastic Gradient Descent and Multi Layer Perceptron to Identify the technical domain of given text as such text provides information about the specific domain. We performed our experiments on Coarse-grained technical domains like Computer Science, Physics, Law, etc for English, Bengali, Gujarati, Hindi, Malayalam, Marathi, Tamil, and Telugu languages, and on fine-grained sub domains for Computer Science like Operating System, Computer Network, Database etc for only English language. Using TFIDF as a feature extraction method we show how both the machine learning models perform on the mentioned languages.

1 Introduction

We can frame Automatic Domain Identification of given text as a text classification problem where one needs to assign predefined categories to given texts. Text classification is a classic topic for Natural Language Processing (NLP), the range of text classification research goes from designing the best features to choosing the best possible machine learning classifiers. Therefore we use Term Frequency & Inverse Document Frequency (TFIDF) to represent our text in terms of vectors, so that the machine learning algorithms will find the relationships between them and classifies the given text. Many machine learning algorithms showed the best performances on text classification , but very limited number studies have explored technical domains like computer science, chemistry, management, etc that too on Indian languages (Kaur and Saini, 2015). There are numerous applications of text classification in Natural Language Processing Tasks like Machine Translation,etc. For these tasks technical domain identification would be the first process. It determines the domain for a given input text, subsequently Machine Translation can choose its resources as per the identified domain. The task can also be viewed at the coarse-grained or fine-grained level based on the requirement. We did our experiments on data provided by ICON TechDOfication-2020 shared task, for English, Bengali, Gujarati, Hindi , Malayalam , Marathi, Tamil and Telugu languages for coarse grained domain classification . For fin-grained classification we have Computer Science domain in English.

2 Related Work

We treat Automatic Technical Domain Identification as a text classification task where we assign predefined categories like chem for chemistry, cs for computer science for the given text. Text classification is a fundamental task in NLP applications and it is a crucial technology in many applications, such as web search, ads matching, and sentiment analysis. Many researchers found variety of algorithms to solve the text classification problem.

The algorithms will vary based on the langauge of text and domain of the text as well. McCallum et al. (1998) compared the theory and practice of two different first-order probabilistic classifiers, both of which make the naive Bayes assumption. The multinomial model is found to be almost uniformly better than the multi-variate Bernoulli model. Joachims (1999) introduced Transductive Support Vector Machine for text classification. While general Support Vector Machines (SVMs) try to produce a general decision function for a learning task, Transductive Support Vector Machines take a particular test set into account and try to minimize misclassification of just those particular samples. Nigam et al. (1999) used maximum entropy for text classification by computing the conditional distribution of the class given the text, and compared accuracy to naive Bayes and

Proceedings of the 17th International Conference on Natural Language Processing: TechDOfication 2020 Shared Task, pages 27–30
Patna, India, December 18 - 21, 2020.

showed that maximum entropy is sometimes significantly better, but also sometimes worse. Lodhi et al. (2002) proposed a novel approach for categorizing text documents based on the use of a special kernel called string subsequence kernel. Machine learning for text classification is the foundation of document categorization, news filtering, document routing, and personalization.

In text domains, effective feature selection is crucial to make the learning task efficient and more accurate, based on this point Forman (2003) presented an extensive comparative study of twelve feature selection metrics like Document Frequency, etc for the high-dimensional domain of text classification, focusing on support vector machines and 2-class problems, typically with high class skew. In social media such as Twitter, Facebook the users may become overwhelmed by the raw data. One solution to this problem is the classification of short text, In Sriram et al. (2010) they did the same, they proposed an approach to use a small set of domain-specific features extracted from the author's profile and text to classify the text to a predefined set of generic classes such as News, Events, Opinions, Deals, and Private Messages. Apart from machine learning algorithms there are some deep learning techniques as well for text classification. In contrast to traditional methods,Lai et al. (2015) introduced a recurrent convolutional neural network for text classification without human designed features. In their model, they apply a recurrent structure to capture contextual information as far as possible when learning word representations, which may introduce considerably less noise compared to traditional window-based neural networks.

Conneau et al. (2016) presented a new architecture (VD-CNN) for text processing which operates directly at the character level and uses only small convolutions and pooling operations. Joulin et al. (2016) used fasttext for word features and then averaged to get a sentence representation for text classification. Yao et al. (2019)proposed a novel approach for text clasiification termed as Graph Convolutional Networks termed as Text-GCN, it can capture global co-occurence information and uses limited labelled texts/documents well. Though there exists a lot of work on text classification, very few works are done for technical domains and on Indian languages like ours. Therefore we present our approach on the provided Indian Languages along with technical domains.

3 Approach

We evaluate our two models namely, Stochastic Gradient Decent and Multi Layer Perceptron on technical domains(Chemistry,Communication Technology, Computer Science, Law , Math and Physics,Bio-Chemistry, Management) for coarse grained technical domain classification for all above mentioned languages(though the number of domains may differ from language to language). For fine grained technical domain classification we have only Computer science in which sub-domains include AI, Algorithm , Computer Architecture, Computer Networks , Database Management system , Programming and Software Engineering for English. We used TFIDF for all experiments.

3.1 Term Frequency & Inverse Document Frequency (TF-IDF)

We use TF-IDF as our feature extraction method in our experiments, The most basic form of weighted word feature extraction is Term frequency (Salton and Buckley, 1988) TF, where each word is mapped to a number corresponding to the number of occurrences of that word in the whole corpora. Methods that extend the results of TF generally use word frequency as a boolean or logarithmically scaled weighting.

$$W(d,t) = TF(d,t) * log(\frac{N}{df(t)}) \qquad (1)$$

(Jones, 1972) proposed Inverse Document Frequency (IDF) as a method to be used along with term frequency in order to lessen the effect of implicitly common words in the corpus. IDF assigns a higher weight to words with either high or low frequency term in the document. This combination of TF and IDF is well known as Term Frequency-Inverse document frequency (TF-IDF). The mathematical representation of the weight of a term in a document by TF-IDF is given in Equation 1. Here N is the number of documents and $df(t)$ is the number of documents containing the term t in the corpus. The first term in equation 1 improves the recall while the second term improves the precision.

3.2 Stochastic Gradient Decent (SGD)

We used SGD classifier from scikit-learn (Pedregosa et al., 2011). SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification

and natural language processing. Though SGD is an optimizer it's alone can be used as a classifier for text classification using different loss functions. The class SGDClassifier implements a plain stochastic gradient descent learning routine which supports different loss functions and penalties for classification. We performed our experiments with hinge loss which is equivalent to linear Support Vector Machine (SVM).

3.3 Multi Layer Perceptron

For Multi Layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLPClassifier relies on an underlying Neural Network to perform the task of text classification. MLPClassifier trains iteratively, at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters. It can also have a regularization term added to the loss function that shrinks model parameters to prevent overfitting. In our experiments we used MLPClassifier from (Pedregosa et al., 2011).

4 Experiments & Results

We evaluate two machine learning algorithms on Data provided by ICON TechDOfication-2020 shared task. The data statistics in terms of number of sentences for all languages is mentioned in table 1. We are provided with various technical domains like physics chemistry etc by ICON TechDOfication 2020 shared task for mentioned languages, however the domains in each language are different. We have Physics(phy), Maths(math), Chemistry(che), Law(law) and Computer Science(cse) n English. Similarly Bengali and Gujarathi have BioChemistry(bioche), cse, Communication Technology(com-tech), Management(mgmt) and phy. Hindi and Telugu have bioche,cse , phy, mgmt, com-tech and other, where Hindi has extra math domain. Malayalam has cse, bioche, com-tech domains, and for Marathi we have bioche, com-tech, phy and cse. In fine-grained domain identification like identifying sub-domain of Computer Science, we have AI (ai),Algorithm (algo),Computer Architecture (ca), Computer Networks (cn), Database Management system (dbms),Programming (pro) and Software Engineering (se) subdomains.

As mentioned in section 3.1 we used TFIDF for all experiments in this paper. For SGD classifier we used hinge loss, and we took alpha as 0.00001, it is a constant that multiplies the regularization term. The higher the value, the stronger the regularization. Maximum number of iterations taken for this algorithm is 15. In MLPClassifier we used relu activation function, solver as sgd which used to find the gradients and optimize the loss function. We adopted the same alpha as SGD Classifier. As MLP is neural network based classifier, there is a need to give hidden layer sizes, we used [100,90] for two hidden layers apart from input and output layer.

Lang.	Train	Dev	Test
English	23962	4850	2500
Bengali	58500	5843	1923
Gujarati	36009	5724	2683
Hindi	148445	14338	4212
Malayalam	40669	3390	1515
Marathi	41997	3780	1789
Tamil	72483	6190	2071
Telugu	68865	5920	2612
English(CS)	13580	1360	1930

Table 1: Data Statistics (no. of sentences)
English(CS) is fine-grained classification task for Computer Science Domain in English

Lang.	Acc.	P	R	F1
English	0.76	0.76	0.76	0.76
Bengali	0.66	0.71	0.68	0.66
Gujarati	0.58	0.57	0.58	0.57
Hindi	0.43	0.44	0.41	0.40
Malayalam	0.44	0.47	0.4	0.37
Marathi	0.48	0.5	0.47	0.43
Tamil	0.44	0.43	0.45	0.36
Telugu	0.55	0.6	0.56	0.57
English(CS)	0.70	0.70	0.70	0.70

Table 2: Classification Report for SGD Classfier
English(CS) is fine-grained classification task for Computer Science Domain in English
Acc:Accuracy P:Precision R:Recall F1:F1-score

We present Accuracy, Precision, Recall and F1-score for all the tasks(for all mentioned languages) as shown in table 2 and table 3 for SGD classifier and for MLP classifier respectively. If we observe the results both the models performed well on English compared to other languages. Motivated from this our future work will be to improve the accuracy on Indian languages. MLP classifier outperformed SGD in almost all tasks. If we talk

Lang.	Acc.	P	R	F1
English	0.77	0.77	0.77	0.77
Bengali	0.66	0.70	0.68	0.66
Gujarati	0.60	59	0.6	0.58
Hindi	0.43	0.5	0.42	0.43
Malayalam	0.44	0.47	0.38	0.36
Marathi	0.5	0.5	0.48	0.44
Tamil	0.45	0.44	0.5	0.39
Telugu	0.54	0.6	0.51	0.52
English(CS)	0.62	0.64	0.62	0.63

Table 3: Classification Report for MLP Classfier
English(CS) is fine-grained classification task for Computer Science Domain in English
Acc:Accuracy P:Precision R:Recall F1:F1-score

about fine grained technical domain identification, SGD outperformed MLP classifier. Comparatively Malayalam and Tamil got less scores in both the algorithms. From all the experiments we can conclude that we can use MLP classifier for Technical Domain Identification but still there is a huge need of improving or coming up with new algorithms for morphologically rich Indian languages.

5 Conclusion & Future Work

we are in the process of exploring many different algorithms for Technical Domain Identification. In the future we want to work on other possible languages for possible technical domains. In this paper we showed two machine learning algorithms(SGD and MLP). TFIDF doesn't depend on any language or domain specific resources hence, we preferred TFIDF as feature extraction method for both the ML algorithms presented in the experiments. From the results we can conclude that Multi Layer Perceptron is performing better on these technical domains for the provided languages.

References

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pages 200–209.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Jasleen Kaur and Jatinderkumar R Saini. 2015. A study of text classification natural language processing algorithms for indian languages. *The VNSGU Journal of Science Technology*, 4(1):162–167.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Kamal Nigam, John Lafferty, and Andrew McCallum. 1999. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. Stockholom, Sweden.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.