Enhancing classification of Research papers to categories of Computer Science domain using Wikipedia articles

No Author Given

No Institute Given

Abstract. We propose a novel method for enhancing classification performance of Research papers to Computer Science categories using Wikipedia articles of the categories. We use state-of-the-art representation learning methods for embedding documents followed by Learning to Rank method for classification. Given the abstracts of research papers from Citation network dataset our method outperforms the state-of-the-art by **3%** accuracy. Our method is also faster to train as compared to state-of-the-art for text classification.

Keywords: Text classification; Representation Learning; Learning to Rank

1 Introduction

Finding categories of research papers automatically from its abstract is a relatively tough problem due to limited context and content presented at a relatively higher level of abstraction than the rest of the paper. This problem comes under standard text classification problem. Traditional methods for text classification works by representing document as human curated features like tf-idf features and than applying a linear classifier like SVM on top of it. Due to high dimensionality of the tf-idf features, use of dimensioanlity reduction techniques like LSA, LDA followed by a classifier is also popular. One limitation of these dimensionality reduction methods is that they are unable to capture semantic meaning of words and phrases in the document.

Recent advancements in Deep Learning (Ref. Bengio) have proven to be more efficient than traditional Deep learning techniques on supervised classification task in Computer Vision and Speech Processing (Ref. Representation Learning for speech and vision). In NLP domain also these Deep Learning inspired representation learning methods like: Word2vec (Ref. Mikolov) and Paragraph2vec have proven to be more effecitve than more traditional text classification tasks (Ref. Mikolov doc2vec). One limitation with these methods is that they don't use any kind of external knowledge base in their model. We overcome this limitation by using Wikipedia as external Knowledge base and linking categories to their correspoding Wikipedia pages and combining both of them to build a

classification model. We show that our method outperforms recent State-of-the-art method for text classification by **3%** on ACM papers from citation network dataset (Ref.)

2 Approach

In this section we briefly describe about representation learning methods and learning to rank method. we describe two recently popular representation learning methods : Word2vec and Paragraph2vec for representation of document in semantic space.

2.1 Word2vec

Given a word sequence (w1,w2,...,wn), the objective of word2vec with skip-gram model is to maximize the following log probability:

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{j=-c}^{c} logp(w_{t+j|w_t})$$

In the skipgram model $p(w_(t+j)|w_t)$ is defined as :

$$p(w_O|w_I) = \frac{exp(v_{'wo)}}{\sum_{w=1}^{W} exp(v_w w_w I)}$$

where v_W and $v_W^{'}$ are input and output representation of words. Word representations are learned during optimization of this objective function.

2.2 Paragraph2vec

Let (w1, w2, ..., wn) be the words in a document, the objective of the paragraph2vec model is to maximize the average log probability defined as follows:

$$\frac{1}{N}\sum_{n=1}^{N} logp(w_n|w_1, ..., w_{n-1}, s, t)$$

where s is the size of context window and t is the title. To generate the paragraph vector, the model simply treats the title as a special word and generates the representation of title, which can be used as representation for the document.

2.3 Learning to Rank

Learning to rank can be defined as learning a function H over set of queries (q1, q2, ..., qm), set of documents (d1, d2,...., dn) with labels R, which can be binary or real valued depending upon the task. In our case R is the set $0, 1$ which represents relevant or non-relevant. The aim of h then is to rank relevant

document higher than irrelevant documents with respect to the query. Formally, h can be defined as (Ref LTR MSR) :

$$h(w, \psi(q, d)) \rightarrow R$$

where ψ is a function which gives a combined representation of query and document. Learning to rank can be broadly divided into 3 categories:

Pointwise Approach It takes triplets (q, D, R) as input and trains a classification/regression model depending on type of R

Pairwise Approach This model takes (q, D) pairs as input and ranks correct pair higher than incorrect pair (generated by random sampling).

Listwise Approach It treats a query with its list of candidates as single learning instance, thus captures considerably more information about the ground truth ordering

3 Experiments

3.1 Our Approach

We use pointiwse approach, since its easier to train and requires less number of data points than pairwise and listwise approaches. (Ref SIGIR LTR paper).

For each paper d_p and its ACM categories l_p, we find the Wikipedia page with title matching the category d_{+w} (positive sample). For each paper d_p we also sample a negative category and its corresponding Wikipedia article d_{-W}. We define a function g which representation of the document by embedding it using word2vec or paragraph2vec in a d-dimensional space.

We follow Bag-of-Words approach for finding representation of the document when using word2vec for representation of the document (average of the word vectors). For Out oF Vocabulary words we generate a uniformly randomly sampled vector and replace the embedding of the work with this vector. For paragraph2vec, we find its representation by using inference method described in the paper (Ref. Mikolov). We define ψ in our case as:

$$\psi(d_p, d_w) = [R(d_p); R(d_W)]$$

Where R(d) is the representation of the document in d-dimensional space. For function h, we select Logistic Regression with implementation available in sklearn.

3.2 Dataset

We use citation network dataset (Ref.) which contains 247543 papers with each article categorised into one of 24 ACM categories. Since for our method we are assuming that labels of the paper have some Wikipedia article with same title, we select only those categories which have some Wikipedia page with same title. We use Wikipedia dump of Oct. 24 (Footnote). We found out that out of 24 ACM categories, 23 categories have a corresponding Wikipedia page.

That leaves us with 236565 articles. We randomly split these articles into 80% training instances and 20% testing instances. So training data contains 189290 papers and testing data contains 47275 papers.

Method	Accuracy
Tf-Idf + SVM Baseline	58.8281%
CNN baseline	69.07827%
Our model	**71.2512%**

Results and Discussion: For benchmarking we select recent state-of-the-art in word embedding based text classification method (REf. CNN for text clas). Since this baseline uses CNN which requires fixed length input sequence, we convert the papers into fixed length documents by padding them till average sequence length in training corpus. We use the code made available by authors (footnote) and ran it on our dataset with the best settings reported.

We present the result of our method on Table 1. For evaluating performance we use accuracy as measure. It is clear that our method outperforms the baseline in accuracy by **2.17 %**. Since abstracts are short texts, adding external information to the model clearly gives us an advantage over current methods. Advantage over simple Bag-of-Words model is clear since their h

References

1. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001) @inproceedingsjoachims1998text, title=Text categorization with support vector machines: Learning with many relevant features, author=Joachims, Thorsten, booktitle=European conference on machine learning, pages=137–142, year=1998, organization=Springer
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
3. National Center for Biotechnology Information, `http://www.ncbi.nlm.nih.gov`