Document Classification Using Wikidata Properties

Sheeban Wasi [1](BBDU) Madhurendra Sachan [2](BBDU) Manuj Darbari [3](BBDNITM)

[1,2,3]Lucknow, India

[1]sheeban@tikaj.com

[2]madhurendra@tikaj.com

[3]manuj_darbari@acm.org

Abstract. The recent advancements in computer science and technology, additionally lead to the generation of a prodigious amount of documents in digital form which are perpetually incrementing. They are widespread in structured as well as semi-structured formats. The documents are interrelated and convey the same message but broadcasted from different sources. In this paper we have endeavored to enhance the document classification by using classification algorithms integrating with external knowledge base from Wikidata and classifies the document of same type in appropriate categories using Wikidata properties as features.

Keywords: Wikidata, tf-idf, learning algorithms, feature selection, VSM, tokenization, classification.

1 Introduction

Classification is necessity of any system which process documents, a fundamental step is to build knowledge base for a model by training certain algorithms, but as we move towards social media or news media data, we cannot limit system to certain categories. The news data is growing at a rapid rate and getting piled up in digital repositories it is difficult to access the information efficaciously so one way to solve this quandary is automatic categorization [1].

Classification is an imperative technique to label document based on the relevant content of the document. Eg. Apple iPod can be classified in 'MP3 player category' and 'Apple Products'. In growing social data, a classifier needs to constantly use a knowledgebase system which is improved daily. eg: if someone is talking about "xyz" which is newly launched music, it should know about "xyz" as music. Some of the classifiers which we used are Multinomial Naive Bayes Classifier, Support Vector Machines, Logistic Regression, Ridge Classifier etc. But, before using these algorithms pre-processing is required which also helps in improving the classification significantly [2, 3]. We also used Vector space model which is an important part of this research.

Vector space model (VSM) is an approach which is widely used for representation of the document as a point in space. The value is decided by the frequency of the terms in a particular document. These words are then used as features which are called bag-of-words. But the challenge here is dimension because the documents contain high dimensional texts which extend the processing time. So, feature extraction needs to be done so that only the important terms get trained reducing the high dimensionality for increasing the efficacy of the classification process. Classification results can be affected by the redundant data termed as noisy data so preprocessing is required in order to remove it. Stop words removal, stemming are performed. To find the weight of the word in the document and convert the document in a structured format a scheme called tf-idf (term frequency- inverse document frequency) is used, the tf-idf value is directly proportional to the weight of the terms in the collection of documents, but is offset by the frequency of the words in the corpus, which helps to determine words that are more useful or common as compared to others. This can amend categorisation as it provides a semantic background knowledge to document. Semantic background knowledge can additionally be extracted through WordNet etc.

In this paper, we propose a way to amend semantic knowledge by utilising Wikidata. We endeavour different coalescences to enrich documents, utilising wikidata properties. Our experiment shows that utilising only Wikidata as our knowledge base (without tf-idf) gave remarkable results.

2 Previous Work

There are numerous researches conducted in the field of digital documents classification and a lot of algorithms have been implemented in order to create a classifier. In the previous approach for categorisation the features from within the textual document are extracted and used for performing classification [4, 5]. For effectively determining the class relationship between the article and the category ; words, phrases, sequences and concepts perform a crucial role. But this is limited and cannot be used as a core-idea for classification algorithm. So in order to overcome this limitation numerous researches proposed an idea to introduce an external knowledge base to enrich the document representation [3, 6, 7]. Researchers used Open Directory Project (OPD) [4], WordNet[4] to enhance the knowledge and enrich the document representation. OPD fails to give acceptable results due to unorganized hierarchy, its extremely unbalanced. WordNet[4] is the thesaurus of English language and in various models the general concepts are extracted from WordNet.

3 Proposed Method

Training a classifier and use Wikidata as knowledge base, primarily we need to pre-process the data. We used standard datasets used by research community so that we can compare the quality of our work .The Datasets we used is BBC dataset to further check the performance of our models.

3.1 Pre-Processing

The most imperative part is to pre-process the data so that further implementations can be done. The basic processing steps followed is listed which helps in making the process efficient by reducing the process time.

Tokenization

Text mining research usually involves words or sentences which is split word by word to process it any further. So in this all the words will be split and the punctuations will also be discarded since it cannot represent any category.

Example :
Input : Donald Trump Is Elected President
Output : [Donald] [Trump] [Is] [Elected] [President]

Stop Word Removal

Stop-words removal has been used by many researchers in the field of text mining for information retrieval. This algorithm removes the words which are not necessary to define the category of the article or we can say unimportant words like conjunctions and prepositions that only increases the process time and making the calculations complicated . This removal process uses the stop word dictionary.

Example : ‘a’, ‘an’, ‘the’ , ‘I’, ‘an’, ‘will’ or any other prepositions and conjunctions.
Input: Donald Trump is the elected President,
Output: ['Donald', 'Trump', 'Elected', 'President']

n-gram Creation

In order to handle multi-words entities and reduce false lookups we used ngram where n=1,2,3,4. We excluded sub sets if certain set was discovered in knowledgebase.

Input : ['Donald', 'Trump', 'Elected', 'President', 'Stunning', 'Repudiation', 'Establishment']
Output: (Partial):

- donald-trump-elected-president, donald-trump-elected, donald-trump Donald, trump, trump-elected, trump, elected.

Above processing would exclude "donald" and "trump" if "donald-trump" is found in knowledgebase. Complexity of search method is

$$O(n)= \boldsymbol{n!} \quad (1)$$

Where n is number of words, the search space could be further reduced by not looking up word pairs which had stop words in between.

Term Frequency Inverse Document Frequency

It is a statistical measure to calculate how important is a word in a corpus. Term frequency is the number of times a word can be found in an article and idf is the computation from log of the inverse probability of word being found in any article.

$$idf_i = log \frac{|D|}{|\{j:t_i \varepsilon d_j\}|} \quad (2)$$

|D| : cardinality of D, or the total number of documents in the corpus |{ j:ti ∈ dj }| : number of documents where the term t_i appears (viz. the document frequency) (that is $n_{i,j}$ 0) . If the term is not in the corpus it will lead it to division-by-zero. Therefore it's common to use

$$1 + | \{j : t_i \varepsilon d_j\} | \quad (3)$$

Donald Trump is the president of United States

1. Indian president and Israel president was a part of the international summit.
2. Elections are five years ahead from now.

-
- Let's assume we want to calculate the weight of the term "president" in the document in #1 and #2 . As we can see in the list that 'president' comes once in #1 and twice in #2 and the term is written in two documents. So, the calculation can be seen as follows:

- Tf-idf(learning,1) = 1 * log (3/2) = 0.23
- Tf-idf(learning,2) = 2 * log (3/2) = 0.46

POS Tagging

Part-Of-Speech tagging is a method to annotate words in a sentence as Noun, Verb, Adjective etc. For this work we use Maxent tagger which is provided as part of Stanford Core NLP, with default model i.e left3words-wsj-0-18.tagger

Example:

Input: Donald Trump Is Elected President in Stunning Repudiation of the Establishment

Output: [('Donald', 'NNP'), ('Trump', 'NNP'), ('Is', 'VBZ'), ('Elected', 'VBN'), ('President', 'NNP'), ('in', 'IN'), ('Stunning', 'NNP'), ('Repudiation', 'NNP'), ('of', 'IN'), ('the', 'DT'), ('Establishment', 'NN')]

Here NNP, VBZ, VBN, IN, DT are standard part of speech tags.

3.2 Classification Algorithm

Classification algorithms is a supervised learning method in which the model learns form the data input given and uses the learning to classify new observation. We used classification algorithms to improve the model. SVM gave much better results as compared to other classification algorithms. SVM algorithm was proposed by Vapnik Sebastiani (2002). in order to get a solution for two-class problem.

Basically, SVM finds a separation between hyperplanes which is defined by classes of data. It is trained using the pre-classified documents and gives good results as compared to Naive Bayes and other alternatives when the entire

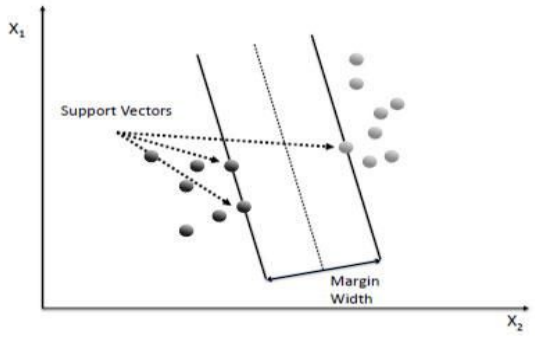

Fig 1. SVM Hyperplane.

vocabulary was used as the feature set because it can operate in fairly large data sets as it measures the margin of separation rather than matches on features.

Different semantic functions can be used as kernels K: radial basis and polynomial basis kernels which affects the accuracy of the algorithm.

3.3 Wikidata Integration

Wikidata[10] is an ontological representation of wikipedia documents, which are generated dynamically but can also be manually updated. The data is consist of document - where each item represents an object or a property.

In a classification problem some contextual information can help in classification by adding the property specific to that article/document [11-13].

Example:

- A business news can contain more items which are organization, company, product, CEO, founders, Investors etc.
- An entertainment article would contain entities related to movies, actors etc.
- A technology related cluster would talk more about applications, mobile phones, development etc. related items.

Some examples to items which can be found during analysis :
https://www.wikidata.org/wiki/Q5284
https://www.wikidata.org/wiki/Q189490
https://www.wikidata.org/wiki/Q95
https://www.wikidata.org/wiki/Q22686
As initial processes we imported Wikidata to MongoDB which is document based - schema less database. It contained *45386255* documents.

4 Methodology

The pipeline consisted of methods to reduce the search space . The pipeline had following stages for:

1. Load All “en” valued labels & aliases from Wikidata into memory. Normalize each value to a base form.
   e.g:*Donald,Trump:-donald-trump*
   Store all references found in a hashmap
2. POS tag each word in the input text, Extract all proper nouns in order.
   e.g: *['Donald', 'Trump', 'President', 'Stunning', 'Repudiation']*
3. Create N gram for extracted list of proper nouns, using sliding window approach. *N=4,3,2,1*
   e.g:['donald-trump-president-stunning','donald-trump-president’,’donald-trump’,’donald’,’trump’,’trump-president’,’trump’,’trump-president-stunning’,’trump-president’,’trump’,’president’...] and so on
4. Normalize the pairs to a base form and look for their presence in loaded

hashmap.
Words found in Wikidata:
donald-trump(Q5295230, Q23001025, Q22686, Q5295229, Q27947481),
president ('Q1255921' 'Q7241206',...)
stunning (Q7628762)
repudiation ('Q19358049', 'Q21071664')

5. Use frequency of occurrence of Wikidata properties in found words as feature in classifier.
   In preceding example following:
   We count occurrences of *Q5295230, Q23001025, Q22686,Q27947481, Q1255921, Q7241206* in given text input
6. Removed noisy features with negligible variance.
7. Since there are over 4000+ Wikidata properties we applied Latent Dirichlet Allocation (LDA) for feature reduction with N = 6 (N - Number of topics)
8. Train Multiclass SVC classifier.

5 Results

We used BBC dataset for training and testing, Since BBC dataset is very huge, we created a sub dataset of 3000 samples, where we used 80% for training and 20% for testing.
We used Jaccard Similarity Score for generating accuracy score, which is basically size of intersection of actual classification & predicted results for each index divided by size of union of actual classification and predicted.

Table 1. Results - SVM with tf-idf.

Category	Precision	Recall	F1-scores
Business	0.99	0.96	0.97
Entertainment	0.96	0.97	0.96
Politics	0.95	0.97	0.96
Sports	1.00	0.99	1.00
Technology	0.96	0.99	0.97
Avg/Total	0.98	0.98	0.98

Accuracy: 95.9551

Table 2. Results with proposed method**.**

Category	Precision	Recall	F1-scores
Business	0.94	0.94	0.94
Entertainment	0.97	0.95	0.96
Politics	0.92	0.91	0.91

Sports	0.95	1.00	0.98
Technology	0.96	0.94	0.95
Avg/Total	0.97	0.97	0.97

Accuracy : 96.8492

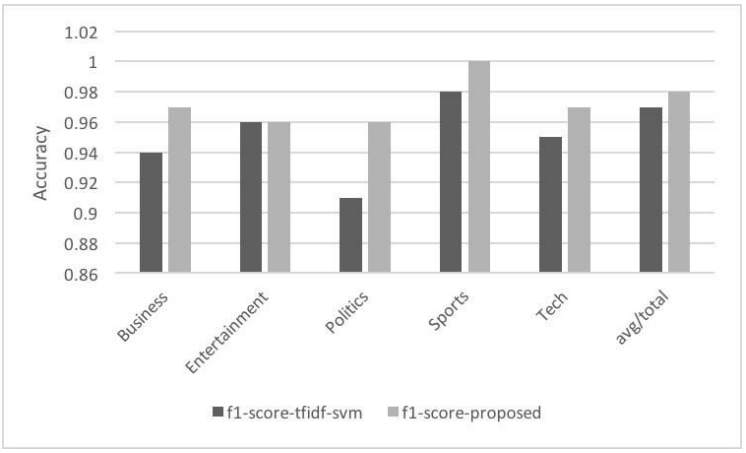

Fig 2. F1 score comparison

Above results clearly reflect that the knowledge base affects the accuracy of prediction, depending on category.

6 Conclusion & Future Work

The proposed method has increased the accuracy of data classification to 96.84% (about 2% above the conventional method). At the same time the method reduces the dependency on training data sets and uses an external knowledge base. The method by utilizes a dynamic data-set enhances the classification of real world news and articles.

This method can in future be evolved to autonomously create clusters when being trained on generic parameters. Future researchers can also use this method to automatically classify and create clusters of data from varied sources in real-world.

References

1. Ittoo, A., Nguyen, L.M., van den Bosch, A.: Text analytics in industry: Challenges, desiderata and trends. Comput. Ind. 78, 96–107 (2016).
2. Talib, R., Kashif, M., Ayesha, S., Fatima, F.: Text Mining: Techniques, Applications and Issues. ijacsa. 7, (2016).

3. Khan, M.T., Durrani, M., Ali, A., Inayat, I., Khalid, S., Khan, K.H.: Sentiment analysis and the complex natural language. Complex Adapt Syst Model. 4, 15 (2016).
4. Elberrichi, Z., Rahmoun, A., Bentaalah, M.A.: Using WordNet for Text Categorization. International Arab Journal of Information Technology (IAJIT). 5, (2008).
5. Sun, A., Lim, E.-P., Ng, W.-K.: Web classification using support vector machine. In: Proceedings of the fourth international workshop on Web information and data management - WIDM '02 (2002).
6. Tripathy, A., Agrawal, A., Rath, S.K.: Classification of sentiment reviews using n-gram machine learning approach. Expert Syst. Appl. 57, 117–126 (2016).
7. Muliono, Y., Tanzil, F.: A Comparison of Text Classification Methods k-NN, Naïve Bayes, and Support Vector Machine for News Classification. jpit. 3, 157–160 (2018).
8. Osiński, S., Weiss, D.: Conceptual Clustering Using Lingo Algorithm: Evaluation on Open Directory Project Data. In: Intelligent Information Processing and Web Mining. pp. 369–377 (2004).
9. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys. 34, 1–47 (2002).
10. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM. 57, 78–85 (2014).
11. Hassan, S., Rafi, M., Shaikh, M.S.: Comparing SVM and naïve Bayes classifiers for text categorization with Wikitology as knowledge enrichment. In: 2011 IEEE 14th International Multitopic Conference (2011).
12. Wang, P., Hu, J., Zeng, H.-J., Chen, L., Chen, Z.: Improving Text Classification by Using Encyclopedia Knowledge. In: Seventh IEEE International Conference on Data Mining (ICDM 2007) (2007).
13. Zhang, Z., Lin, H., Li, P., Wang, H., Lu, D.: Improving Semi-supervised Text Classification by Using Wikipedia Knowledge. In: Lecture Notes in Computer Science. pp. 25–36 (2013).