Content Enrichment Using Linked Open Data for News Classification

Hsin-Chang Yang, Yu-Chih Wang

Dept. Information Management, National University of Kaohsiung, Taiwan
yanghc@nuk.edu.tw, m1053302@mail.nuk.edu.tw

Abstract

In the Web era, people tend to rely on the Web to receive news instead of traditional ways such as newspapers. However, the amount of news generated online is enormous that prohibits people from obtaining their interested news. Most of the common newswire sites still classified the news manually that costed a lot of human effort and may receive unstable result. In the past decades, text classification has been a hot topic and received attention from many scholars in areas such as natural language processing, information retrieval, and machine learning, etc. Various classification algorithms and models have been developed to tackle this problem. In the meantime, Tim Berners-Lee proposed the concept of linked data in 2006. Linked open data (LOD) were constructed prevalently since then. In this study, we try to incorporate LOD into the news classification system. We collected four datasets in order to evaluate the accuracy in various text lengths with or without incorporating LOD. Three classification algorithms, namely K nearest neighbors, support vector machines, and decision trees, were used to classify the news. The experimental results show that the linked open data can improve the accuracy in news classification, especially for short texts or small datasets.

Keywords: Link open data, Machine learning, Text classification, News classification

1 Introduction

Online news services have emerged in past decades due to the pervasion of mobile devices and Internet access. Many readers, especially youngsters, relied on online news services rather than traditional sources such as newspapers, TV, and radio. A survey in 2017 reported that 41 percent of Internet users from the United States named TV as their main source of news, whereas 44 percent stated the internet (incl. social media) was their main news source.[1] Similar shares were also applied across countries worldwide.

Such emergence of online news makes it difficult for users to receive their interested news among a sea of news generated constantly. Therefore, newswires gave class labels or tags to categorize the news articles to allow users to access their interested news easily. However, such tasks were tedious and subjective, and may produce questionable result which may diminish the user experience and satisfaction. Research on automatic news classification was thus attract attention from researchers from various areas. The aim of automatic news classification is to assign a predefined class to each news article according to its similarity to the class representative measured in some metric. A major source for such similarity measurement is the content of the news articles, which is mostly textual. Therefore, techniques of text classification or text mining were commonly adopted to tackle the news classification tasks.

Text classification methodologies rely on the comprehension of the semantics behind the texts to produce precise classification. In generally, the semantics of a piece of text is realized as the set of important keywords occurred in this piece of text. Several approaches on the identification and assessing of the keywords have been suggested, e.g. the vector space model [1]. However, these techniques suffered from the sparsity problem due to the high dimensionality of data, i.e. large number of keywords occurred in the text. The problem gets worse when the text is short, e.g. social messages. Therefore, one major challenge of text classification is to determine the semantics of short texts. A possible approach is to extend the short text with related terms. Such enrichment may increase the fairness of similarity measurement between two short texts since common terms with similar meaning should likely occur in both texts. An important issue on the plausibility of such enrichment is its reliability. A simple technique is to extend the keywords with their synonyms defined in some thesaurus. However, this approach often fails due to word ambiguity such as homonymy and polysemy. Acquisition of reliable related terms is then crucial in short text enrichment.

*Corresponding Author: Hsin-Chang Yang; E-mail: yanghc@nuk.edu.tw
DOI: 10.3966/160792642020092105015
[1] https://www.statista.com/statistics/198765/main-source-of-international-news-in-selected-countries/

Linked open data (LOD), which is a subset of linked data [2], is commonly recognized as the best practice of the Semantic Web vision coined by Berners-Lee [3]. LOD consists of a set of open data interlinked by semantic attributes. The LOD is composed by open data which can be accessed freely. The data items in the LOD were interlinked with semantic attributes that reveal the semantic relationships between data items. These two major characteristics, namely openness and semantic interlinking, make the LOD a rich source of knowledge across wide-ranged domains. According to a dump of LOD Cloud[2] in 2017 [4], there are more than 28 billion unique RDF triples from 650K datasets covered in LOD Cloud. The LOD Cloud centered at DBpedia [5-6], which is a linked data implementation of Wikipedia. The latest version of DBpedia contains 9.5 billion RDF triples. Lots of work have used DBpedia as their information source to tackle tasks such as recommendation, information retrieval, semantic discovery, etc.

In this work, we try to solve the problem of short text classification by incorporating information from LOD, viz. DBpedia, and apply to the news classification task. News articles of different lengths will be collected and classified with and without content enrichment from DBpedia. After collecting the news, we first enriched them using related information from DBpedia. We then classified these news using three different classifiers, namely *k*-nearest neighbors (KNN), support vector machines (SVM), and decision trees (DT). The results of the original set and enriched set were compared to reveal the effectiveness of the proposed approach.

This article is divided into the following sections. Section 2 will briefly summarize some related work. The proposed scheme will be addressed in Section 3. Section 4 shows the experimental results and their evaluation. Finally, we will give the conclusions and future work in the last section.

2 Related Work

2.1 News Classification

News classification could be considered as a sub-domain of text classification. Khan et al. [7] summarized earlier works on text classification, focused mainly on text representation and machine learning techniques. Dalal and Zaviri [8] explained the general strategy and surveyed some techniques of automatic text classification using machine learning approaches. The main steps involved in automatic text classification are (i) document pre-processing, (ii) feature extraction/selection, (iii) model selection, (iv) training and testing the classifier. They stated that pre-processing and feature selection steps play a crucial role in the size and quality of training input given to the classifier, which in turn affects the classifier accuracy. Jindal et al. [9] also gave a more comprehensive review of the techniques for text classification, focusing on various steps involved in text classification process viz. document representation methods, feature selection methods, data mining methods and the evaluation technique used by each study to carry out the results on a particular dataset. Kowsari et al. [10] summarizes the state-of-the-art text classification algorithms. They gave comparisons on various models, such as Rocchio algorithm, boosting, bagging, logistic regression, Naïve Bayes classifier, k-nearest Neighbor, Support Vector Machine, decision tree, conditional random field, random forest, as well as deep learning approaches. Advantages and pitfalls for each approach were also discussed.

Generally, a news article is a semi-structural or unstructural piece of text which may contain structural information such as titles, authors, date/time, news body, or even tags. Classification based on such limited structural information was generally achieved by applying some kinds of text mining process to discover the semantical relations between textual elements in various granularity, e.g. keywords, sentences, or entire articles. Based on the general text classification task, Kaur and Bajaj [11] gave a brief review on the techniques of news classification. Although news classification tasks have been tackled using various approaches, e.g. information retrieval [12], support vector machines [13], and Näive Bayes classifier [14], etc., the general methodologies resemble those for text classification.

2.2 Short Text Classification

Short text classification is a sub-task of text classification and receives much attention in recent years due to emerging social network services that constantly produce enormous amount of short texts, i.e. messages. Difficulties arose in classifying such short texts since extracting semantics from them are considerably difficult due to their characteristics such as sparseness, large-scale, immediacy, and non-standardization [15]. The major difficulty of short text classification should be the sparsity of the features extracting from short texts. To diminish such difficulty, two approaches were common adopted, namely semantic extraction approach and semantic expansion approach. The former tries to discover the concept, inner structure, and correlation of texts to obtain their implicit semantics which are more expressive and subjective. Techniques such as latent semantic analysis (LSA) [16-18] and Latent Dirichlet Allocation (LDA) [19] were applied to discover the semantics of texts. On the other hand, the latter tries to expand the content of short texts with semantically relevant information and produce longer texts that may reduce the feature

[2] https://lod-cloud.net/

sparseness. Many of works of this approach used queries on some keywords of a short text to retrieve contents and expand this short text. For example, Sun [20] queried a set of pre-labeled short texts using selected keywords from a short text and predicted its class by majority vote of the search results. Wang et al. [21] proposed a unified framework to expand short texts based on word embedding clustering and convolutional neural network. Yang et al. [22] proposed a topic model based approach which combines both lexical and semantic features by employing a background knowledge repository, namely Wikipedia, to learn topics with respect to all target categories. They mapped each word occurrence to a particular topic and then represented a short text with these mapped topics whose dimensionality is low. Ma et al. [23] adopted the word embedding approach which learns the embeddings also from Wikipedia. Shen et al. [24] first found the word clustering centers and used them to expand word vectors with words in the same cluster. Wang et al. [25] combined explicit and implicit representations of short text using convolutional neural networks for classification. They first conceptualized a short text as a set of relevant concepts using Probase [26] and then obtained the embedding of short text by coalescing the words and relevant concepts on top of pre-trained word vectors.

3 The Proposed Method

Figure 1 depicts the processing flow of our method. Conceptual descriptions of each step will be addressed in the following subsections. Details of implementation will be mentioned in Sec. 4.

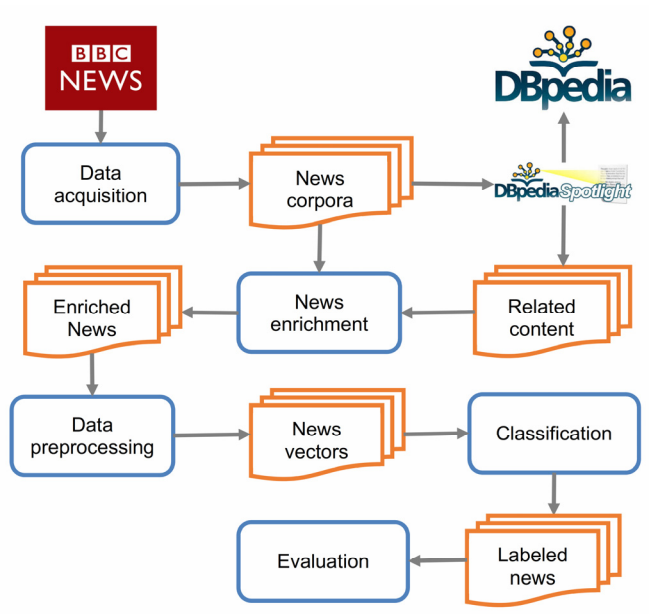

Figure 1. The flowchart of our method

3.1 Data Acquisition

We acquired news from the major newswire BBC for its popularity and rich categorization structure. To verify our approach, we collected news of various granularity into different corpora. Four corpora, namely 'News Titles', 'News Descriptions', 'Short News', and 'Regular News', were assembled. These corpora contain news articles in different lengths, reflecting the amount of information they carried. We should expect that shorter articles are composed by less words and are harder to be classified correctly due to possible vagueness of their meaning. In this work, we conducted experiments on these corpora to verify such conjecture as well as the effect of enrichment of short news from exterior information. Table 1 summarizes the characteristics of these corpora. Figure 2 depicts an example of 'News Titles' and 'News Descriptions' elements in a news article of BBC News[3]. The other two corpora, i.e. 'Short News' and 'Regular News', consist of the full content of a news article whose length meets the threshold.

Table 1. Characteristics of various corpora

ID	Title	Description	Length
C_T	News Titles	The titles of news articles	< 20
C_D	News Descriptions	Brief descriptions of the news	< 50
C_S	Short News	News containing few words	< 350
C_R	Regular News	News containing many words	≥ 350

Figure 2. Elements of news articles. The content of this news is not shown here and could be retrieved by clicking the title

3.2 News Enrichment

In this work, the contents of news articles are expanded using information gathered from DBpedia which is the centroid of linked open data. The DBpedia could be considered, in short, the linked data version of Wikipedia. It contains fruitful resources linked together by their semantical relationships and makes it an ideal knowledge base. DBpedia provided a tool, DBpedia Spotlight [27], to allow users to annotate their contents with resources of DBpedia. We adopted DBpedia Spotlight API[4] to enrich the news corpora. Figure 3 depicts an example of annotated news. Underlined

[3] http://www.bbc.com/
[4] https://www.dbpedia-spotlight.org/api

terms were annotated words or phrases that have related resources in DBpedia. Following the annotated links, we can retrieve resources regarding the keywords/phrases in DBpedia. An example is shown in Figure 4 which depicts the DBpedia resource of 'Sicily' by following the hyperlinks of the annotated keyword 'Sicilian' in Figure 3. A news article was then expanded with information extracted from such resources to enrich its content. In this work, we only extracted the attribute `rdfs:comment` to enrich the news since it provided rich textual descriptions on the resource. On the other hand, DBpedia also provides many elements which themselves contain semantical meaning, such as `rdf:type`, `rdf:seeAlso`, `owl:sameAs`, etc. However, we did not encompass such semantical attributes for they did not contain enough textual information for further processing. Although some of the attributes, e.g. `rdf:seeAlso`, could provide textual information by following the links of their values, we did not encompass such second-order information in this work due to their wide diversity. After expansion, a piece of news article, n_i, $i \in (1, N)$, becomes an enriched news article, $n_i^{'} = \bigcup R(n_i)$, where N is the number of news articles in the corpus and $R(n_i)$ is the set of resources annotated by DBpedia Spotlight in n_i.

The Diciotti docked in the Sicilian port of Catania on Monday.

The Italian government is refusing to most of the migrants disembark until a solution is reached regarding their relocation with other European states.

Prosecutors have now opened an inquiry into possible illegal confinement, illegal arrest, and abuse of power.

Mr Salvini responded to the news by shrugging of the threat, saying those still on board would be allowed off "in the coming hours".

Rice ball protest in Italian migrant boat row
Matteo Salvini: Can Italy trust this man?
Who is responsible for helping out at sea?
According to Mr Salvini, they would be taken in "by bishops who are opening their doors, their hearts and their wallets".

So far only small groups have been allowed in, including unaccompanied children and the sick, two of whom are said to have tuberculosis.

Figure 3. Excerpt of a news annotated using DBpedia Spotlight. We presented the output in HTML format for better comprehension. In the meantime, there are other output formats, e.g. JSON, *n*-tuples, turtle, etc., available provided by DBpedia Spotlight API

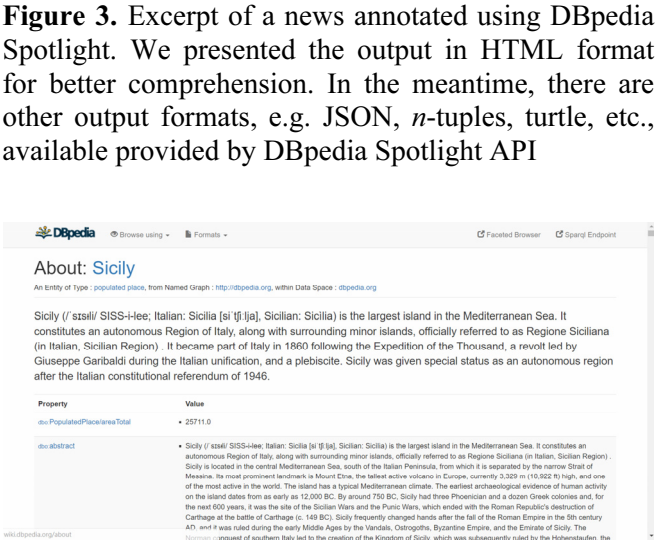

Figure 4. DBpedia resource about 'Sicily' which was annotated with 'Sicilian' in the news in Figure 2

3.3 Data Preprocessing

The news articles went through several standard text preprocessing steps to be transformed into proper form for later classification. First, a news article n_i or $n_i^{'}$ was segmented into a set of words. Stemming and stopword elimination steps were further applied to these words to reduce the redundancy and number of words. Let $n_i = \{k_{ij} \mid j \in (1, t_i)\}$, where k_{ij} and t_i denote the j-th unique word in n_i and the number of unique words in n_i, respectively. Similar denotations were applied to $n_i^{'}$, i.e. $n_i^{'} = \{k_{ij}^{'} \mid j \in (1, t_i^{'})\}$. We can then collected all unique words appeared in all news article and obtained the vocabulary V of this corpus, i.e. $V = \bigcup_i n_i$. A news article n_i was then transformed into a vector $\mathbf{n}_i = [w_{ij}]^T$, $j \in (1, |V|)$, where w_{ij} denotes the weight of the j-th vocabulary word in n_i and is defined using classical *tf-idf* scheme in vector space model [1] as follow:

$$w_{ij} = \frac{c_{ij}}{\sum_j c_{ij}} \log \frac{d_j}{N}, \quad \textbf{(1)}$$

where c_{ij} denotes the number of occurrence of k_{ij} in n_i and d_j denotes the number of news articles containing k_{ij}. Similar denotations were applied to $n_i^{'}$ to obtain V' and $\mathbf{n}_i^{'} = [w_{ij}^{'}]^T$, $j \in (1, |V^{'}|)$.

3.4 News Classification

The news article will be classified using three different classifiers, namely *k*-nearest neighbors [28], support vector machines [29], and decision trees [30]. These classifiers were chosen for their popularity and competitive performance. For each corpus in Table 1, both original and expanded news articles were used to train the classifiers. Each corpus was divided into training and test sets for training and validation purpose. The `scikit-learn` toolkit[5] [31] was adopted for implementation of these classifiers.

3.5 Performance Evaluation

We evaluated the effectiveness of our method by 10-fold cross validation. Each corpus in Table 1 was validated with and without content enrichment. A total of eight corpora were used in the experiments to train 24 classifiers. We evaluated the result by using accuracy, recall, precision, and F1 measure.

[5] http://scikit-learn.org/stable/

4 Experimental Result

4.1 Corpora Construction and Preparation

We collected the news titles and descriptions from BBS News using BBC News API[6] during April 2018 and May 2018. The news articles were returned in JSON format. For example, the return result regarding the article in Figure 2 is shown below:


```
{
   -"source": {
    "id": "bbc-news",
    "name": "BBC News"
    },
    "author": "BBC News",
    "title": "Minister investigated
       over migrant stand-off",
    "description": "Sicilian
       prosecutors are investigating
       Italy's Matteo Salvini as
       migrants are kept on a rescue
       boat.",
    "url":
       "http://www.bbc.co.uk/news/wor
       ld-europe-45310479",
    "urlToImage":
       "https://ichef.bbci.co.uk/news
       /1024/branded_news/18409/produ
       ction/_103173399_048834412.jpg
       ",
    "publishedAt": "2018-08-
       25T23:12:30Z"
},
```


We then extracted the news titles and descriptions from the '`title`' and '`description`' elements of the JSON files. These extracted titles and descriptions were then assembled into the two corpora C_T (for 'News Titles') and C_D (for 'News Descriptions'), respectively. For the other two corpora, i.e. C_S (for 'Short News') and C_R (for 'Regular News'), we retrieved the body of news articles by crawling the BBC News site using Google site search during February and September, 2017 and categorized them into corresponding corpus according to their lengths. If the length of a news article exceeds a predefined threshold, it will be categorized as 'Regular News'; otherwise, it will be categorized as 'Short News'. The threshold was set to 350 in our experiments.

Each news article was categorized by BBC. The categorization label could be resolved from the URL of the news article. For example, the URL of the news in Figure 2 is `http://www.bbc.co.uk/news/world-europe-45310479`. The categorization label of this article is 'world-europe' at the end of this URL, where 'world' and 'europe' denote the first and second level categories, respectively. There are 14 first-level categories among the retrieved articles, namely 'Blogs', 'Business', 'World', 'Science', 'Technology', 'UK', 'Magazine', 'Entertainment', 'Disability', 'Health', 'Education', 'Election', 'Explainers', and 'Health'. However, some of the news URL did not contain such categorization labels.

Some of the retrieved news were video news containing negligible number of words and were discarded. We also discarded those news without proper category labels. Table 2 summarizes the sizes of each corpus.

Table 2. Sizes of various corpora

Corpus ID	Title	# of retrieved news	# of news in experiments
C_T	News Titles	8702	8301
C_D	News Descriptions	8809	8053
C_S	Short News	1589	874
C_R	Regular News	4036	3215
Total		23136	20443

These corpora were then enriched by DBpedia Spotlight. DBpedia Spotlight will identify words or phrases that are relevant to some resources in DBpedia and return information regarding the resources. The returned information includes the following items [32]:
URI The URI of the DBpedia resource.
support The minimum number of inlinks the DBpedia resource has to have in order to be annotated.
types The types of this resources.
surfaceForm The surface form of the resource.
offset The offset of the annotating word or phrase to the beginning of the article.
similarityScore The similarity score between the resource vector and the context surrounding the surface form.
percentageOfSecondRank The relative difference in topic score between the first and the second ranked resource.

Many resources could be annotated in a news article. In our experiments, we only allowed resources with similarity scores greater than 0.5 to be retrieved since resources with low similarity scores may not be so relevant to the news. The threshold of support was set to 20 to discard those resources with few references. The URIs of the annotated resources were then used to retrieve the DBpedia resources. We extracted the value of `rdfs:comment` attribute to expand the news. Figure 3 depicts the result of DBpedia Spotlight annotation of an example news and Figure 4 depicts one of the resources annotating to the news. Note that the annotated resources in this example may not fit the similarity requirement of our experiments.

The collected corpora as well as their enriched versions were then preprocessed using a series of

[6] https://newsapi.org/s/bbc-news-api

standard text processing procedures. First, each article was segmented using Natural Language Toolkit (NLTK) [33] and Stanford CoreNLP [34]. We discarded punctuation marks and numbers for they carry less meaning and are ambiguous. Stopwords listed in the NLTK Stopwords corpus which contained 2400 stopwords for 11 languages were also removed. The remaining words were further stemmed using CoreNLP stemmer to remove morphologically similar words. We then collected all unique words appeared in a corpus to its vocabulary. Table 3 shows the statistics of words and vocabulary in each corpus. Figure 5 depicts the histograms of word counts for each corpus.

Table 3. Statistics of number of words in each corpus

Corpus ID	Max # of words	Min # of words	Average # of words	Size of vocabulary
C_T	16	2	8	11057
$C_T^{'}$	292	5	87	34109
C_D	56	7	18	15037
$C_D^{'}$	365	10	112	39890
C_S	348	50	270	18333
$C_S^{'}$	1376	79	654	45394
C_R	1993	350	751	52719
$C_R^{'}$	4174	392	1523	117120

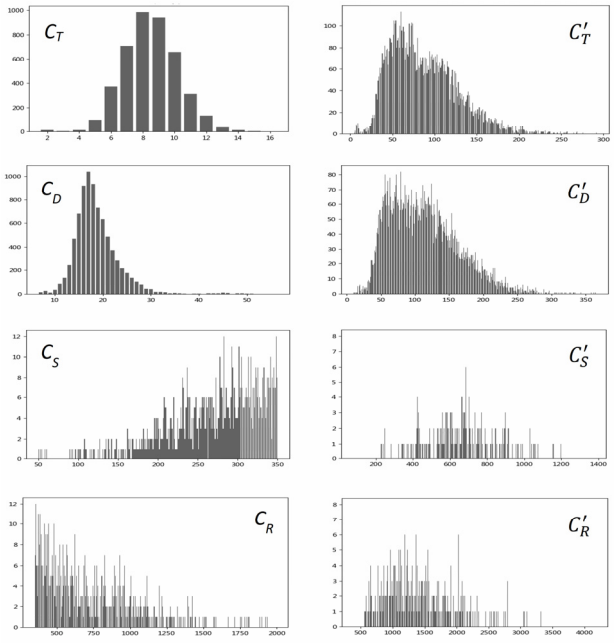

Figure 5. The histograms of word counts for each corpus. The horizontal axes are the word counts of news articles in a corpus. Vertical axes represent the numbers of news articles in each corpus

The retrieved news articles belong to 14 categories. Table 4 summarizes the category distributions of all corpora. The news articles were not evenly distributed in categories since we collected all news articles spanning a period of time. Each news article was then transformed into a vector using classical *tf-idf* scheme of vector space model as described in Sec. 3.3. These news vectors were then segmented into two subsets for training and testing purpose.

Table 4. Category distributions of each corpus

Category	Max # of words	Min # of words	Average # of words	Size of vocabulary
UK	2563	2590	265	729
Business	357	369	69	351
World	1536	1318	396	1202
Sport	3017	3011	0	0
Science	90	91	6	154
Health	111	108	7	180
Technology	172	148	72	224
Blogs	96	92	19	69
Entertainment	317	280	33	135
Education	34	39	2	62
Disability	7	6	0	8
Election	1	1	1	47
Magazine	0	0	4	53
Explainers	0	0	0	1
Total	8301	8053	874	3215

4.2 Classification

In this work, we performed the experiments using 10-fold cross validation which divided each corpus into 10 subsets and used 9 of them for training and the remaining 1 for testing. The training set was used to train three different classifiers, namely *k*-nearest neighbors, support vector machines, and decision trees. We adopted the implementations of these classifiers in `scikit-learn` toolkit. Various versions of these classifiers were implemented in the toolkit. Basically, default parameters of these classifiers were used with some customization. Some major parameters are addressed below. For KNN, we used the default value of the number of neighbors, i.e. 5, and the Minkowski metric as the distance metric. For SVM, we adopted the `linearSVC` implementation. The declarations and the default parameter values of these classifiers are shown below:

```
class
    sklearn.neighbors.KNeighborsClass
    ifier(n_neighbors=5, weights=
    'uniform', algorithm='auto',
    leaf_size=30, p=2, metric=
    'minkowski', metric_ params=None,
    n_jobs=1, **kwargs)
class
    sklearn.svm.LinearSVC(penalty='l2',
    loss='squared_hinge', dual=True,
    tol=0.0001, C=1.0, multi_class=
    'ovr', fit_intercept=True,
    intercept_scaling=1, class_weight=
    None, verbose=0, random_state=
    None, max_iter=1000)
```

```
class
    sklearn.tree.DecisionTreeClassifi
    er(criterion='gini',
    splitter= 'best', max_depth=None,
    min_samples_split=2, min_samples_
    leaf=1, min_weight_fraction_
    leaf=0.0, max_features=None,
    random_state=None, max_leaf_
    nodes=None, min_impurity_
    decrease=0.0, min_impurity_split=
    None, class_weight=None, presort=
    False)
```

The declarations and parameters of these classifiers could be found in `scikit-learn` documentation[7] .

4.3 Evaluation Result

Each corpus was evaluated using 10-fold cross validations. Each corpus was divided into 10 partitions which 9 of them were used for training and the remaining 1 was used for testing. Thus, 10 experiments were conducted for each corpus. The evaluation results of these experiments were then averaged as the final evaluation. In this work, 4 metrics, namely accuracy, recall, precision, and F1 measure, were used for evaluation. Table 5 shows the evaluation results of each corpus.

Table 5. Evaluation results of each corpus. The terms 'Acc', 'Rec', and 'Pre' stands for Accuracy, Recall, and Precision, respectively

Corpus ID	Classifier	Acc	Rec	Pre	F1
C_T	KNN	0.72	0.47	0.37	0.40
	SVM	0.78	0.57	0.39	0.43
	DT	0.73	0.48	0.36	0.39
C'_T	KNN	0.72	0.44	0.39	0.40
	SVM	0.79	0.52	0.43	0.45
	DT	0.74	0.45	0.39	0.41
C_D	KNN	0.73	0.45	0.33	0.36
	SVM	0.78	0.55	0.36	0.40
	DT	0.67	0.34	0.29	0.30
C'_D	KNN	0.71	0.41	0.35	0.37
	SVM	0.80	0.46	0.43	0.44
	DT	0.69	0.31	0.30	0.30
C_S	KNN	0.67	0.51	0.48	0.48
	SVM	0.79	0.58	0.46	0.49
	DT	0.60	0.39	0.39	0.38
C'_S	KNN	0.70	0.44	0.45	0.43
	SVM	0.82	0.57	0.47	0.50
	DT	0.62	0.36	0.37	0.35
C_R	KNN	0.68	0.54	0.52	0.51
	SVM	0.80	0.72	0.58	0.61
	DT	0.56	0.39	0.37	0.38
C'_R	KNN	0.70	0.62	0.58	0.58
	SVM	0.81	0.74	0.63	0.66
	DT	0.54	0.37	0.36	0.36

We will give some discussions on various aspects of the evaluation results. First, we will compare the performance of different classifiers. Figure 6 depicts the comparisons of the results among different classifiers. In this figure, we averaged the results of each classifier over all corpora. It is clear that SVM performed best among these classifiers. On the other hand, decision trees were the worst classifiers among all.

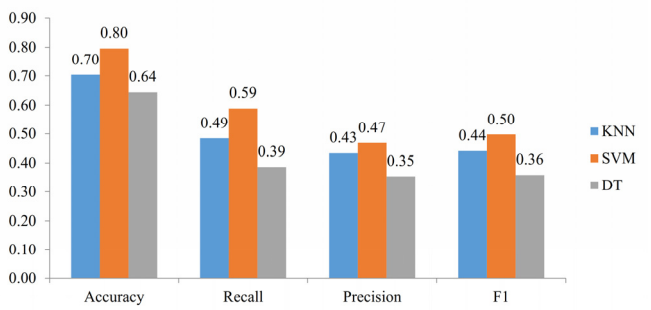

Figure 6. Comparisons of different classifiers with various metrics. The metrics were averaged over all corpora

To reveal the effect of article length, we compared the results of different corpora as shown in Figure 7 and Figure 8 for original and enriched corpora, respectively. In both sets of corpora, the KNN and SVM performed similarly that longer articles tended to produce better result. Decision trees, however, did not follow the same rule. It is also interesting that C_D performed worse than C_T in all classifiers. This contradiction may cause by the additional content in the descriptions that were written in somewhat lyrical or exaggerated manner while the topics tended to indicate the main themes of the news.

Finally, we compared the contributions of using enrichment from LOD. Figure 9 depicts the effects of applying enrichment. The differences of each metric for each corpus were shown. A positive value indicates the enriched corpus outperformed the original corpus, and vice versa. For example, a value of 0.01 for accuracy metric means that the accuracy was improved by 1% using the enriched corpus. The values of 'ALL' corpus were average of the four types of corpora used in our experiments. We can observe that all corpora, except C_D, produced better average results by using enriched corpora when using KNN and SVM. The improvements were even significant for SVM for it improved all corpora. It is noteworthy that the values of recall metric were decreased in most of the corpora, indicating that using LOD enrichment tends to misclassify some news articles. The reason behind this could due to some of the enriched content may spread across many topics and confuse the classifiers.

[7] http://scikit-learn.org/stable/user_guide.html

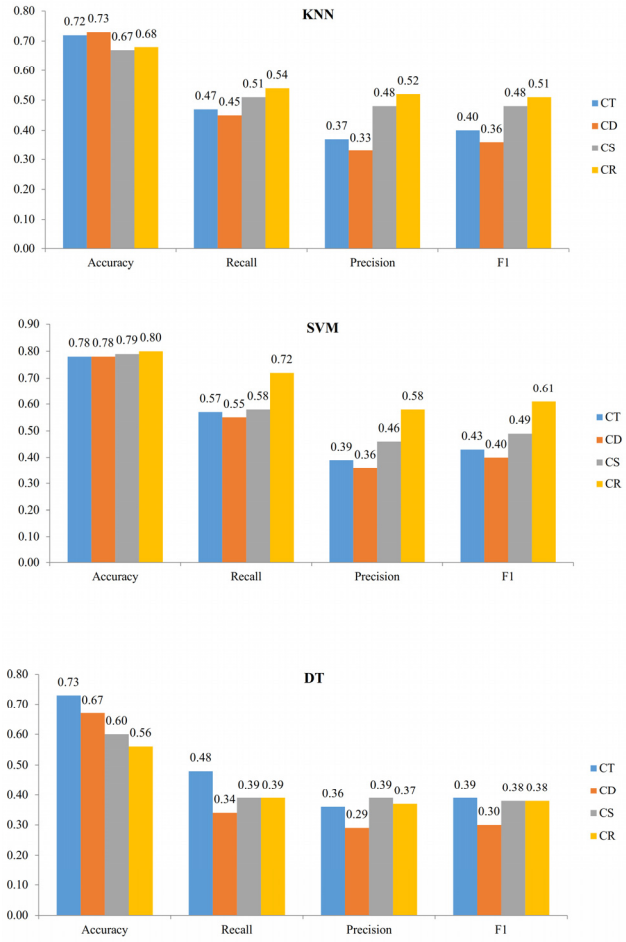

Figure 7. Results of corpora of different article lengths with various metrics using the three classifiers. The legends CT, CD, CS, and CR stand for C_T, C_D, C_S, and C_R, respectively

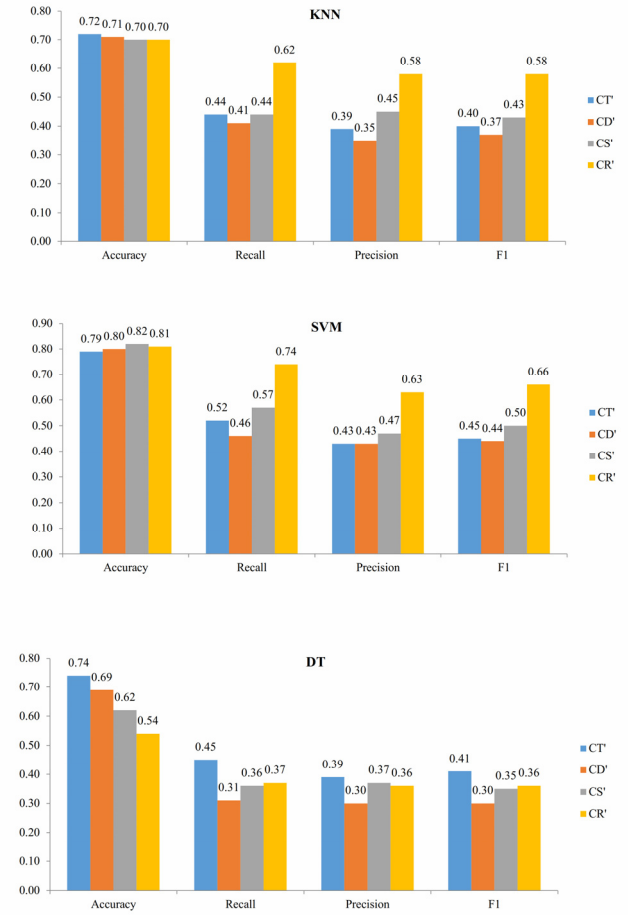

Figure 8. Results of enriched corpora of different article lengths with various metrics using the three classifiers. The legends CT', CD', CS', and CR' stand for C_T', C_D', C_S', and C_R', respectively

The effect of enrichment may be affected by the volume of corpus since larger corpus may contain richer information and deteriorate the effect of enrichment. To verify above hypothesis, we further constructed several subsets of various sizes for each corpus. We then conducted experiments using the same approach in previous experiments. Figure 10 depicts the results. In each graph, we showed the accuracies of both original and enriched corpora, as well as their differences, using various-sized subsets. Generally, the enrichments produced greater improvements in smaller corpora. However, such improvements decay when the article length increases. For C_R and C'_R, which contain lengthy articles, small amounts of articles still produced comparable results to larger corpora. On contrary, we can observe that other corpora improved their accuracies when the sizes of corpora increase. Moreover, C_T, which contains extreme short texts, obtained better improvements when its volume getting smaller.

5 Conclusions and Discussions

In this work, we proposed an approach for news classification by incorporating related content discovered from the linked open data knowledge base DBpedia. For each news article, we tried to find related information regarding topics occurred in the article from DBpedia. These information were combined into the original article for enrichments to amend the cold start and sparseness problems often encountered in short text classification tasks, e.g. news classification. Series of experiments were conducted to verify the effectiveness of the proposed approach. The experimental results demonstrated the following observations. First, the SVM classifiers outperform the other two classifiers, i.e. KNN and DT, which conforms to many previous reports. Second, the enrichment process does improve the effectiveness of the classification, especially for shorter articles. Finally, our method produced better improvements when the volume of corpora is small. Therefore, our approach may apply to the classification tasks of small datasets composed of short texts.

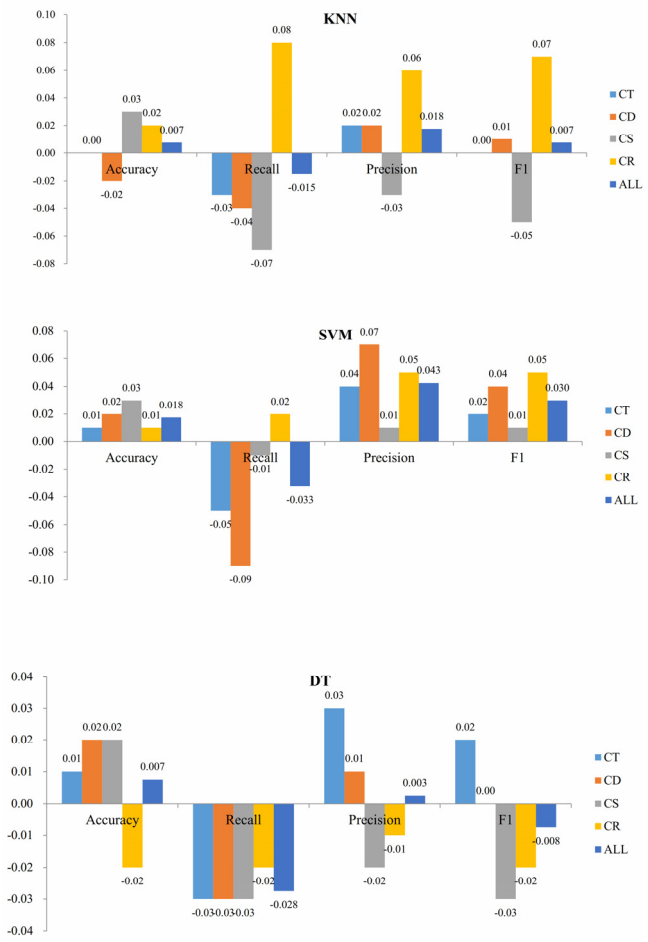

Figure 9. Comparisons of results using original and enriched corpora

In this work, we augmented the text using words (features) coming from linked open data. These features may not appear in the original text and carry semantic information regarding the text. Such semantic augmentation is not possible for traditional methods.

Acknowledgments

The work is supported by the Ministry of Science and Technology under Grant No.: MOST 107-2410-H-390-008.

References

[1] G. Salton, M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[2] T. Berners-Lee, *Linked Data*, http://www.w3.org/DesignIssues/LinkedData.html.

[3] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, *Scientific American*, Vol. 284, No. 5, pp. 29-37, May, 2001.

[4] J. D. Fernández, W. Beek, M. A. Martínez-Prieto, M. Arias, LOD-a-lot: A Queryable Dump of the LOD Cloud, *International Semantic Web Conference*, Vienna, Austria, 2017, pp. 75-83.

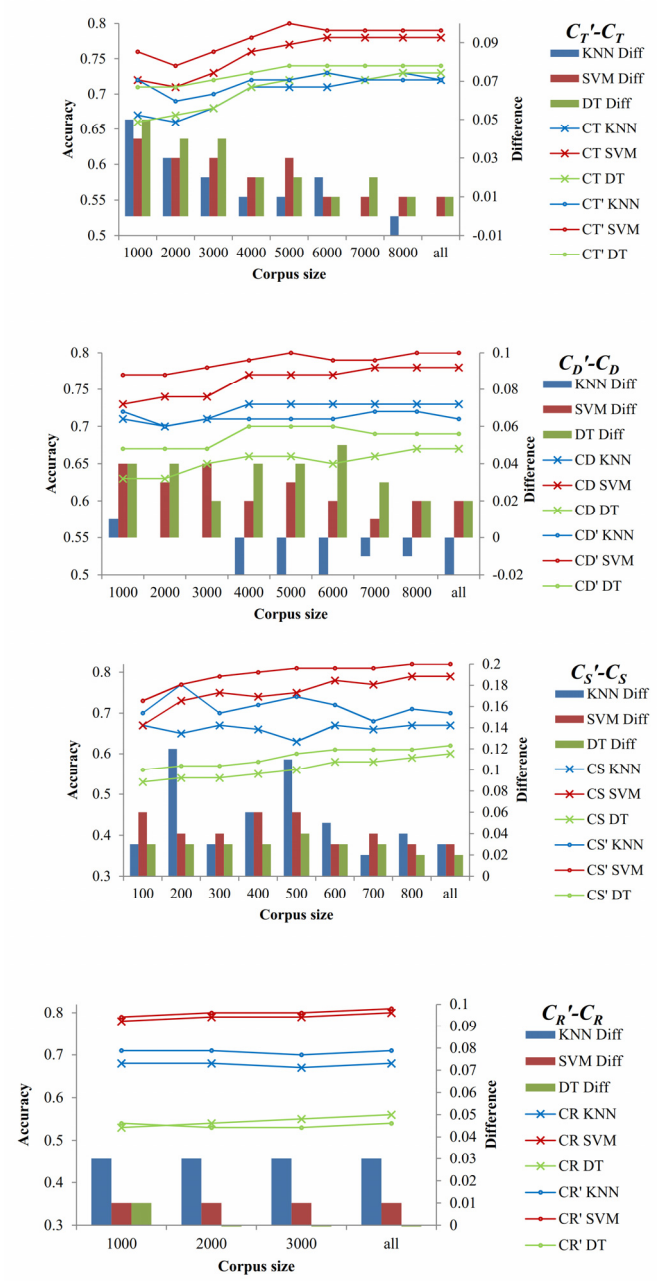

Figure 10. Comparisons of results using various corpus sizes

[5] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data, in: K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, P. Cudré-Mauroux (Eds.), *The Semantic Web*, Springer, 2007, pp. 722-735.

[6] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia, *Semantic Web Journal*, Vol. 6, No. 2, pp. 167-195, 2015.

[7] A. Khan, B. Baharudin, L. H. Lee, K. Khan, A Review of Machine Learning Algorithms for Text-Documents Classification, *Journal of Advances in Information Technology*, Vol. 1, No. 1, pp. 4-20, February, 2010.

[8] M. K. Dalal, M. A. Zaveri, Automatic Text Classification: A Technical Review, *International Journal of Computer Applications*, Vol. 28, No. 2, pp. 37-40, August, 2011.

[9] R. Jindal, R. Malhotra, A. Jain, Techniques for Text Classification: Literature Review and Current Trends, *Webology*, Vol. 12, No. 2, Article 139, December, 2015.

[10] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, Text Classification Algorithms: A Survey, *Information*, Vol. 10, no. 4, 150, April, 2019.

[11] G. Kaur, K. Bajaj, News Classification and Its Techniques: A Review, *IOSR Journal of Computer Engineering*, Vol. 18, No. 1, pp. 22-26, January-February, 2016.

[12] P. Kroha, R. Baeza-Yates, A Case Study: News Classification Based on Term Frequency, *16th International Conference on Database and Expert Systems Applications (DEXA 2005)*, Copenhagen, Denmark, 2005, pp. 428-432.

[13] I. Dilrukshi, K. De Zoysa, A. Caldera, Twitter News Classification using SVM, *8th International Conference on Computer Science & Education (ICCSE)*, Colombo, Sri Lanka, 2013, pp. 287-291.

[14] A. Selamat, H. Yanagimoto, S. Omatu, Web News Classification Using Neural Networks Based on PCA, *the 41st SICE Annual Conference*, Osaka, Japan, 2002, pp. 2389-2394.

[15] G. Song, Y. Ye, X. Du, X. Huang, S. Bie, Short Text Classification: A Survey, *Journal of Multimedia*, Vol. 9, No. 5, pp. 635-643, May, 2014.

[16] S. Zelikovitz, F. Marquez, Transductive Learning for Short-Text Classification Problems Using Latent Semantic Indexing, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 19, No. 2, pp. 143-163, March, 2005.

[17] Q. Pu, G.-W. Yang, Short-Text Classification Based on ICA and LSA, *International Symposium on Neural Networks*, Chengdu, China, 2006, pp. 265-270.

[18] X.-H. Phan, L.-M. Nguyen, S. Horiguchi, Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-Scale Data Collections, *International Conference on World Wide Web*, Beijing, China, 2008, pp. 91-100.

[19] M. Chen, X. Jin, D. Shen, Short Text Classification Improved by Learning Multi-granularity Topics, *International Joint Conference on Artificial Intelligence(IJCAI)*, Barcelona, Spain, 2011, pp. 1776-1781.

[20] A. Sun, Short Text Classification Using Very Few Words, *International ACM SIGIR conference on Research and Development in Information Retrieval*, Portland, Oregon, 2012, pp. 1145-1146.

[21] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, H. Hao, Semantic Expansion Using Word Embedding Clustering and Convolutional Neural Network for Improving Short Text Classification, *Neurocomputing*, Vol. 174, pp. 806-814, January, 2016.

[22] L. Yang, C. Li, Q. Ding, L. Li, Combining Lexical and Semantic Features for Short Text Classification, *Procedia Computer Science*, Vol. 22, pp. 78-86, 2013.

[23] C. Ma, W. Xu, P. Li, Y. Yan, Distributional Representations of Words for Short Text Classification, *Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado, 2015, pp. 33-38.

[24] Y. Shen, Q. Zhang, J. Zhang, J. Huang, Y. Lu, K. Lei, Improving Medical Short Text Classification with Semantic Expansion Using Word-Cluster Embedding, *International Conference on Information Science and Applications*, Hong Kong, China, 2018, pp. 401-411.

[25] J. Wang, Z. Wang, D. Zhang, J. Yan, Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification, *International Joint Conference on Artificial Intelligence (IJCAI-17)*, Melbourne, Australia, 2017, pp. 2915-2921.

[26] W. Wu, H. Li, H. Wang, K. Q. Zhu, Probase: A Probabilistic Taxonomy for Text Understanding, *ACM SIGMOD International Conference on Management of Data*, Scottsdale, Arizona, 2012, pp. 481-492.

[27] J. Daiber, M. Jakob, C. Hokamp, P. N. Mendes, Improving Efficiency and Accuracy in Multilingual Entity Extraction, *International Conference on Semantic Systems (I-Semantics)*, Graz, Austria, 2013, pp. 121-124.

[28] N. S. Altman, An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, *The American Statistician*, Vol. 46, No. 3, pp. 175-185, August, 1992.

[29] C. Cortes, V. Vapnik, Support-Vector Networks, *Machine learning*, Vol. 20, No. 3, pp. 273-297, September, 1995.

[30] J. R. Quinlan, Induction of Decision Trees, *Machine Learning*, Vol. 1, No. 1, pp. 81-106, 1986.

[31] 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, E. Duchesnay, scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830, February, 2011.

[32] P. N. Mendes, M. Jakob, A. Garcia-Silva, C. Bizer, DBpedia Spotlight: Shedding Light on the Web of Documents, *International Conference on Semantic Systems (I-Semantics)*, Graz, Austria, 2011, pp. 1-8.

[33] S. Bird, E. Klein, E. Loper, *Natural Language Processing with Python*, *1st Edition*, O'Reilly Media, Inc., 2009.

[34] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, D. McClosky, The Stanford CoreNLP Natural Language Processing Toolkit, *Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, Maryland, 2014, pp. 55-60.

Biographies

Hsin-Chang Yang received his M. Sc. and Ph.D. degrees from National Taiwan University, Taiwan, in 1990 and 1996, respectively. He is a professor at National University of Kaohsiung, Taiwan. His research interests include text mining, neural networks, semantic discovery, information retrieval, and linked open data.

Yu-Chih Wang received his M.S. degree from National University of Kaohsiung, Taiwan, in 2018. He is a system architect in Bank SinoPac, Taiwan. His researcth interests include text mining, machine learning, and data analysis.