

Deep Learning Distributional Features for Noise Handling in Open-set Web-genre Classification

Dimitrios Pritsos and Efstathios Stamatatos

¹ Dimitrios Pritsos University of the Aegean
Karlovassi, Samos – 83200, Greece.
dpitsos@aegean.gr

² Efstathios Stamatatos University of the Aegean
Karlovassi, Samos – 83200, Greece.
stamatatos@aegean.gr

Abstract. Web genre detection is a task that can enhance information retrieval systems by providing rich descriptions of documents and enabling more specialized queries. Most of previous studies in this field adopt the closed-set scenario where a given palette comprises all available genre labels. However this is not a realistic setup since web genres are constantly enriched with new labels and existing web genres are evolving in time. Open-set classification, where some pages used in the evaluation phase do not belong to any of the known genres, is a more realistic setup for this task. In this case, all pages not belonging to known genres can be seen as noise. This paper focuses on systematic evaluation of open-set web genre identification when the noise is either structured or unstructured. Two open-set methods combined with alternative text representation schemes and similarity measures are tested based on two benchmark corpora. Moreover, we adopt the openness test for web genre identification that enables the observation of effectiveness for a varying number of known/unknown labels.

Keywords: Web Genre Identification · Information Retrieval · Natural Language Processing

1 Introduction

2 Relevant Work

3 Distributional Features Learning

In this study we are using the Doc2Vec out-of-the-box algorithm which is based on the there publications PUBA, PUBB, PUBC while the algorithm can be found at Gensim package <https://github.com/RaRe-Technologies/gensim>. In particular we have implemented a special module inside our package, specialized for HTML preprocessing, named *Html2Vec* (see <https://github.com/dpritsos/html2vec>) where a whole corpus can be fed and matrix of *Bag-of-Words Paragraph Vectors* (PV-BOW) is returned as an output. One PV-BOW vector per Web-document for the corpus.

In order to compare our work to previous works, two different document representation types can be produced from our *Gensim based sub-module*. One is PVBOW Word-n-grams and the other for PVBOW Character-n-grams, which are presented in our following experimental results.

The PVBOW is a *Neural Network* (NNet) where it is formed as a *softmax* multi-class classifier approximating the formula of eq.2. PVBOW is trained using *stochastic gradient descent* where the gradient is obtained via *backpropagation*. The objective function of the NNet is the maximized *average log probability* eq.1, given a sequence of training n-grams (word or character) $t_1, t_2, t_3, \dots, t_T$.

$$\max \frac{1}{T} \sum_{T-k}^{a=k} \log p(t_a | t_{a-k}, \dots, t_{a+k}) \quad (1)$$

$$p(t_a | t_{a-k}, \dots, t_{a+k}) = \frac{e^{y_{t_a}}}{\sum_i e^{y_i}} \quad (2)$$

Particularly for PVBOW, we are using for this study, for each iteration, of the stochastic gradient descent, a *text window* is sampled with size w_{size} . Then a random word is sampled from the text window and *form a classification task given the Paragraph Vector*. Thus the y of the eq.2 is formed to be $y = b + s(t_1, t_2, t_3, \dots, t_{w_{size}})$ where $s()$ is the sequence of words-n-grams or character-n-grams of the sampled window.

In our study we are training a PVBOW Distributional Feature model for the whole corpus. The corpus initially is split to a set of paragraphs, as required from PVBOW. To be more specific the paragraphs are sentences split from all the document of the whole corpus. Then several models PVBOW feature models are trained for a variety of parameters and vector dimensions, explained in the experiments section below. After the model has been fitted then one vector for each web-document was inferred from the PVBOW. The final document vectors derived from Distributional Feature Model are given to the open-set learning model explained below.

4 Nearest Neighbors Distance Ratio

The Nearest Neighbors Distance Ratio (NNRD) algorithm is our variant implementation of the proposed open-set algorithm of Mendes et al. (6). In the original approach euclidean distance has been used because of the variation of data set on which the algorithm has been evaluated. In our approach we are using cosine distance, because in text classification is being confirmed to be the proper choice in hundreds of publications. Moreover, the cosine distance is comparable to the results of the *Random Feature Subspacing Ensemble* algorithm found in (7) where cosine similarity is used for the WGI evaluation.

The NNRD algorithm is an extension of the simple *Nearest Neighbors* NN algorithm where additionally to the sets of training vectors (one set for each class) a threshold is selected by maximizing the *Normalized Accuracy* (NA) as shown in equation 3 on the *Known* and the *Marked as Unknown samples*.

$$NA = \lambda A_{KS} + (1 - \lambda) A_{MUS} \quad (3)$$

where A_{KS} is the *Known Samples Accuracy* and A_{MUS} is the *Marked as Unknown Samples Accuracy*. The balance parameters λ regulate the mistake trade-off on the known and marked-unknown samples prediction.

The optimally selected threshold is the the *Distance Ratio Threshold* (DRT) where NA is maximized. Equation 4 is used for calculating the Distance Ratio (DR) of the two nearest class samples, say s_{c_a} and u_{c_b} , to a random sample r_x under the constrain $c_a c_b$, where c_g is the sample's class.

It is very important to note that the c_g is trained in an open-set framework, therefore, the samples pairs selected for comparison might either be from the known or the marked as unknown samples. Thus $g \in 1, 2, \dots, N$ and $g = \emptyset$ when samples is marked as unknown.

$$DR = \frac{D(r_x, s_{c_a})}{D(r_x, s_{c_b})} \quad (4)$$

where $D(x, y)$ is the distance between the samples where in this study is the *Cosine Distance*.

Therefore, the fitting function of the NN algorithm, described in pseudocode 1.1, is the optimization procedure to find the DRT values for classes respective sets of training samples where NA is maximized.

Algorithm 1.1: *Nearest Neighbor Distance Ratio* training data fitting function

Data: G the set of genre class tags $\{1, 2, \dots, N\}$, p the hyper-parameter regulates the percentage of G tags will be marked as unknown, k the hyper-parameter regulates the percentage of known G tags that will be kept for validation only, T the *Distance Ratio* thresholds set than will test for finding the one which is minimizing the *Normalized Accuracy*, λ regulates the mistakes trade-off on the known and marked-unknown samples prediction (see eq.4), $C[g]$ the matrix of class vector sets one for every genre class tag $g \in G$

Result: DRT the *Distance Ration Threshold* calculated by the NNRD algorithm's fitting function, $C[g]$

```

1  $K_i^G, K_{validation}^G, U_{validation}^G, I^G = Split(G, p, k)$  splitting the  $G$  tags in to
   known/unknown samples combinations using the  $p$  and  $k$  hyper-parameters.
   The amount of split combinations is calculated by the equations 5 and 6.;
2  $V^G = U_{validation}^G \cup K_{validation}^G$  the validation set is the union of the  $I$  splits of the
   known-validation and the marked-as-unknown sets, of the whole training set;
3 for each  $i \in I$  do
4    $D_{VK}^{cos}[i] = COS_D(V_i^G, K_i^G)$  calculating all the Cosine Distances between the
   web-page of  $K^G$  and  $V^G$  sets for every  $I$  split combination;
5 end
6  $C_A^{min} = argmin(D_{VK}^{cos})$  getting the indices of the closest classes from  $V$ ;
7  $C_B^{min} = argmin(D_{VK}^{cos})$  getting the indices of the second closest classes from  $V$ ;
8  $R_V = D_{VK}^{cos}[C_A^{min}] / D_{VK}^{cos}[C_B^{min}]$  calculating the Distance Ratios  $R$  for all the
   vectors in  $V$ 
9  $NA^{max} \leftarrow 0$  initializing Maximized Normalized Accuracy with 0 value.  $DRT \leftarrow 0$ 
   initializing Distance Ratio Threshold with 0 value.
10 for each  $drt \in T$  do
11   for each  $r, i \in \{R_V, count(R_V)\}$  do
12     if  $r < drt$  then
13        $vi = C_A^{min}[i]$  keep the respective index;
14        $Y[i] = G[vi]$  setting the genre's class tag as prediction for this random
       vector of set  $V$ ;
15     else
16        $Y[i] = \emptyset$  setting as none of the known genres or "I don't know";
17     end
18   end
19    $NA_V = NormalizedAccuracy(Y, R_V)$  calculating the Normalized Accuracy as
   shown in equation 3 for tested threshold  $drt$ ;
20   if  $NA_V > NA^{max}$  then
21      $NA^{max} \leftarrow NA_V$  keeping the maximum  $NA$  until the outer for-loop
     finishes;
22      $DRT \leftarrow drt$  keeping the Distance Ratio Threshold maximizes the
     Normalized Accuracy;
23   else
24   end
25 end

```

In the optimization procedure the training samples are split based on their class tags c_x . Then some class tags are *marked as unknown* and some are left being known. Therefore, all the samples of the marked as unknown are used only in the validation subset while the known class tags samples are further split into the classes sets (one for each class) and into the known validation set. Then, samples of the validation sets, both then known and then marked as unknown, are used seamlessly for calculating the set of Distance Ratios (one for each class). Afterwards, a set of DRT values are tested given a range of values $R \in t_1, t_2, t_n$ beforehand where the t_x is selected which is maximizing the NA of the validation set.

The splitting procedure the of the training set is regulated by a hyper-parameter p which defines the percentage of the class tags set $g \in 1, 2, \dots, N$ where they will be marked as unknown. Then the total number of all possible splitting combination are calculated and these split-sets are used for finding the DRT. The combination are found using equations 5 and 6, where eq.6 is the *Binomial Coefficient*.

$$U_{num} = \text{int}(N * p) \quad (5)$$

where N is the size of the class tags set $1, 2, \dots, N$ and p is the percentage regulation parameter for keeping the number of tags to be marked as unknown.

$$S_{num} = \frac{N!}{U_{num}!(N - U_{num})!} \quad (6)$$

The NNDR is a open-set classification algorithm, therefore, every random sample will be classified to one of the classes the NNDR has been fitted or to the unknown when its DR is greater then DRT. While training as explained above the DRT values are tested incrementally until the optimal data fitting for the training function.

In prediction phase the DRT is passed to the NNDR prediction function together with the random samples and the training samples as shown in pseudocode 1.2.

Algorithm 1.2: *Nearest Neighbor Distance Ratio* prediction function

Data: W the vector set of the random web-page to be classified, $C[g]$ the matrix of class vector sets one for every genre class tag $g \in G$, DRT the *Distance Ration Threshold* calculated by the NNRD algorithms fitting function

Result: $Y \in \{G, \emptyset\}$, R the Distance Ratio scores vector, one score for every input vector of the random set W

```

1 for each  $g \in G$  do
2    $D_{C_g X}^{cos} = COS_D(C[g], X)$  calculating all the Cosine Distances between the
   random web-page vectors and the class vectors of class  $g$ ;
3 end
4  $C_A^{min} = argmin(D_{C_g W}^{cos})$  getting the indices of the closest classes from  $W$ ;
5  $C_B^{min} = argmin(D_{C_g W}^{cos})$  getting the indices of the second closest classes from  $W$ ;
6  $R_W = D_{C_g W}^{cos}[D_A^{min}] / D_{C_g W}^{cos}[D_B^{min}]$  calculating the Distance Ratios  $R$  for all the
   vectors in  $W$ 
7 for each  $r, i \in \{R_W, count(R_W)\}$  do
8   if  $r < DRT$  then
9      $vi = C_A^{min}[i]$  keep the respective index;
10     $Y[i] = G[vi]$  setting the genre's class tag as prediction for this random
    vector of set  $W$ ;
11  else
12     $Y[i] = \emptyset$  setting as none of the known genres or "I don't know";
13  end
14 end

```

Our implementation of the above NNRD algorithm can be found at <https://github.com/dpritsos/OpenNNDR>, where it is implemented in Python/Cython and can significantly accelerated using as much as possible CPUs due to its capability for concurrent calculations in C level speed. Since, NNRD is a rather slow classification method, we have seen in practice that there is up to 100 time acceleration from the capability to exploit a cloud service with 32 vCPUs (Xeon) compare to 4-core/8-threads i7 CPU.

5 Experiments

5.1 Open-set Evaluation Methodology

In this study we are measuring the performance of a novel extension of the NN method, designed for open-set classification when the web-documents used as input are *Distributional Encodings of Fixed Size Vectors* derived from an PVBOW NNet model. In particular we are measuring the effect the marked-as-unknown (or marked-as-noise) genre class tags, to the open-set prediction process.

To compensate the potentially unbalanced distribution of web pages over the genres, we are using the macro-averaged precision and recall measures. Than is a modified version of precision and recall for open-set classification tasks proposed by (6). This modification calculates precision and recall only for the known classes (available in the training phase) while the unknown samples (belonging to classes not available during

training) affect false positives and false negatives. To find parameter settings that obtain optimal evaluation performances we use two scalar measures, the *Area Under the Precision-Recall Curve* (AUC) and F_1 . We will show that the appropriate selection of the optimization measure is highly significant in the presence of noise.

Precision-Recall curve is a standard method to visualize the performance of classifiers. In this paper, the Precision-Recall curve is calculated in 11-standard recall levels $[0, 0.1, \dots, 1.0]$. Precision values are interpolated based on the following formula:

$$P(r_j) = \max_{r_j \leq r \leq r_{j+1}} (P(r)) \quad (7)$$

where $P(r_j)$ is the precision at r_j standard recall level.

5.2 Corpora

In this paper we study NNRD performance with distributional features derived from a NNet PVBOW corpus. In particular, the open-set algorithms described above are analytically tested on benchmark corpus already used in previous work in WGI (2; 9; 4; 7), the *SANTINIS* (5) corpus. Details are given in table 1. This is a corpus comprising 1,400 English web pages evenly distributed into 7 genres as well as 80 BBC web pages evenly categorized into 4 additional genres. In addition, it comprises a random selection of 1,000 English web pages taken from the SPIRIT corpus (3). The latter can be viewed as noise in this corpus. In particular in this study SPIRIT tags are considered *Marked as Unknown* (MU) and this is how we measure them.

Note that in the evaluation process we both measuring two kind classification-as-unknown of the open-set algorithm; the *false positive unknown* where are classification of samples which have class tags known to the NNRD model and the *true positive unknown* where they are marked-as-unknown (or marked-as-noise) where they are considered as noise.

5.3 Settings

Our text representation features are based exclusively on textual information from web pages excluding any structural information, URLs, etc. Based on the good results reported in (10; 8; 1) as well as some preliminary experiments. The following document representation schemes are examined: Character 4-grams (C4G), Word unigrams (W1G), and Word 3-grams (W3G).

We use the Term-Frequency (TF) weighting scheme and Distributional Learning scheme. The feature space for TF is defined by a *Vocabulary* which is extracted based on the terms appearing at training set only. The TF-Vocabulary range of dimensions have been tested are $V_{TF} = \{5k, 10k, 50k, 100k\}$. The feature space in the Distributional model is preselected while the deep learning process of PVBOW model. The range of the models fixed dimensions has been tested are $D_{dim} = \{50, 100, 250, 500, 1000\}$ based of previews studies related to the distributional features on Natural Language Processing domain where relatively small dimensions are used, compare to TF scheme.

The DM vocabulary creation process is also driven by an internal terms *vocabulary* which is used for eliminating the terms with lower than a preferred frequency and then

discards the terms from the text window for the PVBOW (see section 3). In this study we have tested as the $TF_{min} = \{3, 10\}$ minimum frequencies set. The text window tested was $P_{win} = \{3, 8, 20\}$ size set. In respect of PVBOW model, other (hyper-)parameter values put to a test were $\alpha = 0.025$, $epochs = \{1, 3, 10\}$ and $decay = \{0.002, 0.02\}$.

Particularly for NNRD then following parameters have been tested: 1) $\lambda = \{0.2, 0.5, 0.7\}$ which is regulating the balance between the known and the unknown classification accuracy risk in the formula 3, 2) DRT threshold selection candidates $DRT = \{0.4, 0.6, 0.8, 0.9\}$. Also two other parameters we have introduced in our implementation; the percentage of the training set to be split internally as Training/Validation sub-sets where $V_{ptg} = \{0.5, 0.7\}$ has been tested. Then the percentage of the validation set which was split as unknown $U_{ptg} = \{0.3, 0.5\}$ has been tested.

As comparative *Baseline* we have employed RFSE. With respect to RFSE, four parameters should be set: the vocabulary size V_F , the number of features used in each iteration FSS , the number of iterations I , and the threshold σ . We examined, $FSS = \{1k, 5k, 10k, 50k, 90k\}$, $I = \{10, 50, 100, 200, 300, 500, 1000\}$ and $\sigma = \{0.5, 0.7, 0.9\}$. Additionally, in this work we are testing three document similarity measures: cosine similarity. All these parameters have been selected as suggested in (7).

Finally, to extract the best possible parameter settings for each classification method we apply grid-search over the space of all parameter value combinations.

Genre	Pages
Blog	200
Eshop	200
FAQ	200
Frontpage	200
Listing	200
Personal Home Page	200
Search Page	200
DIY Mini Guide (BBC)	20
Editorial (BBC)	20
Features (BBC)	20
Short Bio (BBC)	20
Noise (Spirit1000)	1000

Table 1: SANTINIS corpora descriptions and amount of pages per genre.

6 Results

In this study we are evaluating the performance of NNDR algorithm into the open-set framework with the presence of noise, as the notion of noise explained above. That is we are only using the 11 known genre classes of SANTINIS corpus for training, while for evaluation we are using all 12 classes. The true genre class tags for the documents of the in the 12th class is not known and it is only considered as noise. Therefore, we

have to deal with *unstructured noise*. Additionally, since the noise-class is marked then the algorithm is evaluated in both false-positive classifications of the marked-unknown-classes as known and the false-negative classifications of the known-classes as unknown. In both cases the false classifications are causing a penalty to the Macro-F1 and Macro-Precision-Recall Curves we are using for evaluation.

We perform 10-fold cross validation and in each fold we include the full set of 1,000 pages of noise. This evaluation strategy is giving a more realistic evaluation. Since the noise size is greater than the size of any genre included in the given genres collection.

Initially we are comparing the performance of NNDR with RFSE on TF features vocabulary. RFSE returns higher score to NNDR, in tables 2 and 3, for Macro-F1 and Macro-AUC and for any terms-type. Note also that the TF feature model dimensions for both algorithm is in all cases a few thousands.

MAX.	STP	SUP	DRT	λ	T.TYPE	DIMs	MP	MR	MAUC	MF1
F1	0.7	0.3	0.8	any	C4G	5000	0.664	0.403	0.291D	0.502
AUC	0.7	0.3	0.8	any	C4G	5000	0.664	0.403	0.291D	0.502
F1	0.7	0.5	0.8	any	W1G	5000	0.691	0.439	0.348D	0.537
AUC	0.7	0.5	0.8	any	W1G	5000	0.691	0.439	0.348D	0.537
F1	0.5	0.5	0.8	0.3 or 1.5	W3G	10000	0.720	0.664	0.486D	0.691
AUC	0.5	0.5	0.6	0.5	W3G	5000	0.738	0.604	0.437D	0.664

Table 2: Maximum performance of NNDR on TF Features of SANTINIS coprus. STP is the Splitting Training Percentage. SUP is the Splitting Unknown Percentage. DRT is the Distance Ration Threshold. λ is the weighing balance regulation parameters for the Normalized Accuracy. T.TYPE is the Terms Type. DIMs is the features model’s dimensions. MP is the Macro Precision. MR is the Macro Recall. MAUC is the Area Under the Macro PR Curve. MF1 is the F1 score of the Macro Precision and Macro Recall.

MAX.	FSS	σT	ITER.	T.TYPE	DIMs	MP	MR	MAUC	MF1
F1	1000	0.5	100	C4G	50000	0.739	0.780	0.652	0.759
AUC	500	0.5	300	C4G	10000	0.686	0.831	0.722	0.751
F1	10000	0.5	1000	W1G	50000	0.776	0.758	0.657	0.767
AUC	1000	0.5	300	W1G	5000	0.618	0.807	0.673	0.700
F1	1000	0.7	100	W3G	50000	0.797	0.722	0.488	0.758
AUC	10000	0.5	100	W3G	100000	0.657	0.805	0.696	0.723

Table 3: Maximum performance of RFSE on TF Features of SANTINIS coprus. FSS is the Features Subset Selection number. σT is the sigma threshold of the RFSE. ITER is the number of iterations of the RFSE. T.TYPE is the Terms Type. DIMs is the features model’s dimensions. MP is the Macro Precision. MR is the Macro Recall. MAUC is the Area Under the Macro Precision-Recall Curve. MF1 is the F1 score of the Macro Precision and Macro Recall.

MAX.	STP	SUP	DRT	λ	T.TYPE	DIMs	MTF	WS	α	EP.	DEC.	MP	MR	MAUC	MF1
F1	any	any	0.8	any	C4G	50	3	8	0.025	10	0.002	0.829	0.600	0.455D	0.696
AUC	any	any	0.8	any	C4G	500	3	3	0.025	10	0.02	0.755	0.602	0.539D	0.670
F1	any	any	0.8	any	W1G	50	3	3	0.025	10	0.02	0.733	0.670	0.541D	0.700
AUC	any	any	0.8	any	W1G	50	3	3	0.025	10	0.02	0.733	0.670	0.541D	0.700
F1	any	any	0.8	any	W3G	100	3	3	0.025	10	0.02	0.827	0.615	0.564D	0.706
AUC	any	any	0.8	any	W3G	100	3	3	0.025	10	0.02	0.827	0.615	0.564D	0.706

Table 4: Maximum performance of NNDR on Distributional Features of SANTINIS corpus. STP, SUP, DRT, λ , T.TYPE, DIMs are the same as in table 2. MTF is the Minimum Threshold Frequency of the Distributional models Vocabulary. WS is the Windows Size of the text sentence. α is the NNet parameter. EP is the epochs number of the NNet model. DEC is the decay parameter of the NNet model. MP is the Macro Precision. MR is the Macro Recall. MAUC is the Area Under the Macro PR Curve. MF1 is the F1 score of the Macro Precision and Macro Recall.

However, there is a notable difference in the behavior of the two algorithms where it seems to be the reason RFSE outperforms NNDR. NNDR is using the 5000 (with the exception of line 4 of table 3) most frequent features for both training and evaluation phases. RFSE is using about 1000 *randomly selected features* from the 50,000 to 100,000 most frequent features. This observations leading as the conclusion that the textual-genre information is pervasive in several features of the texts irrespectively of their frequency.

In order to research our null hypothesis where the textual-genre information is pervasive in the correlation of the features and not only in their frequency, we employee the distributional features explained above. Comparing tables 2 and 4 there is a notable improvement. Particularly for the Macro-F1 Score the performance reaches 0.7 in all cases. Moreover, the STP, SUP and λ parameters can my any of the set we have tested, on the contrary to the TF features model where only a specific value pare case is returning the maximum performance. Concerning TF model only when W3G terms-type is used we have a comparable $F1$ and AUC value to the respective performance of the distributional model for NNDR.

We have also examined the case where AUC is maximized because it seems in (7) study this measure is maximized when precision is maximized while the recall is kept as max as possible. As an example note line 6 at table 2 where for W3G the precision can reach 0.738 while recall only reduced to 0.604 compare to the 0.664 in the above line where $F1$ is maximized.

The open-set classification framework evaluation design naturally turns our interest to the precision performance of the algorithms. To reason it we have to conceder that a classifier in a binary case should distinguish the *known and the unkown web documents* are not belonging to the target class. In addition it has to recognize the the documents of the target class. Thus 2 out of 3 evaluation score naturally related to the precision score.

In precision perspective it seems NNDR with distributional features outperforms RFSE in all case. In addition the size of features models dimension are 10 to 200 times smaller. NNDR returns 0.829 macro-precision score compare to 0.739 of the RFSE, for C4G terms-type. Moreover, the maximum precision of RFSE is 0.797 with 0.722 recall while the respective recall for them maximum NNDR precision performance is 0.600.

The performance of NNDR (with distributional features) increases for 10% on average for precision and decreases for 21.00% compare to the RFSE. In addition, its performance increases for 7% in precision and 37% in recall compare to it self when TF feature model is used.

We further evaluated the performance of NNDR in respect of precision performance and for this reason we have selected the best performance with the AUC criterion for NNDR using DF model, which in for this corpus is also maximizes $F1$. In figure 1 NNDR with DF, NNDR with TF and RFSE with TF are shown. All curves are showing the performance when W3G is used where it the case NNDR with DF is returning the maximum performance on SANTINIS corpus.

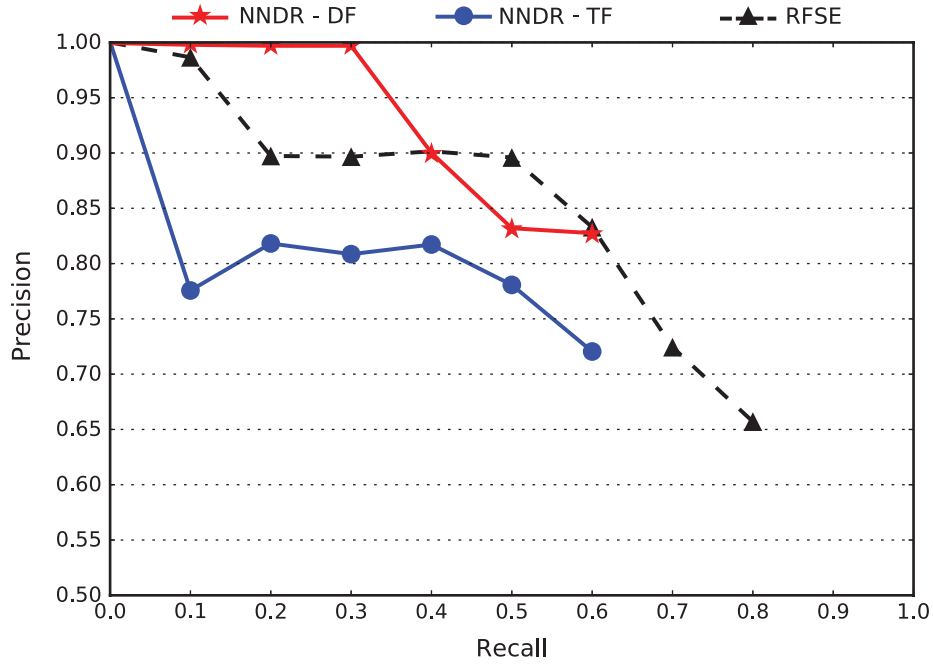


Fig. 1: Precision-Recall Curves of NNDR on SANTINIS corpus with RFSE as Baseline. The curves are for W3G terms-type and for AUC optimization criterion. The 11-recall-levels are shown for each evaluation experiment. The lines are stopping before the 11th recall level due to the open-set framework, i.e. the remaining part after the last mark of each curve the percentage of the corpus tha has been classified as Unknown from the algorithms.

The NNDR-DF is starting higher from all the curves and it remains to 0.99 precision for the 30% of the corpus then it drops to the same level with the RFSE baseline and after 0.50% drops below it. However, it remains over 0.80 up to 60% of the corpus. The the rest of the corpus is classified as unknown. RFSE manages to recognize correctly part of the corpus up to 80% with the precision to drop to 0.70 and then to 0.65. Considering the NNDR-TF is significantly lower in performance compare to the base line and is also has a significant drop in performance for the first 10% of the corpus. Given that the curves are calculated based on the ranked scores from the best performance to the lowest performance it seems that the algorithm significantly affected by the noise of the marked-unknown part. That is the algorithm makes very confident decision for some part of the corpus confusing them with other classes. Its overall performance is over 0.7 precision and still recognize the 60% of the corpus.

To conclude it seems that distributional features are giving a significant increment in performance even for a pure performance and very algorithm as NN, particularly the NNDR, i.e. an open-set version of it. Still RFSE where has successfully used for WGI and other NLP problems, is giving higher over all performance in respect of macro-F1 and AUC. However, in respect of precision which is naturally more important for an

open-set evaluation set the distributional features are offering higher or equal performance with 100 times smaller feature size. Moreover, considering the open-set framework as a filtering-setup for the algorithms one would probably prefer the NNDR-DF algorithm over RFSE.

7 Conclusions

In this paper we presented an experimental study where is focused on the effect of distributional features on an open-set web-page document classification task. In contrast to vast majority of previous work in this area, we adopt the open-set scenario that is more realistic for web-page classification since it is not feasible to construct a genre palette with all available genres and appropriate samples for each one of them. Particularly we are evaluating the possible performance improvement of an open-set algorithm when distributional features are employed, using the SANTINIS corpus.

The open-set algorithm we have evaluated with distributional features input is a novel variation of Nearest Neighbors algorithm named NNDR. A Distance Ratio threshold is calculated while the training procedure for maximizing the *Normalized Accuracy* of the known and the unknown classes.

Our evaluation methodology previews shown to be proper for evaluation open-set scenarios. The natural focus in an open-set evaluation framework is the macro-precision, since always there is a part of the corpus will be classified as unknown. Either because it is unknown (marked-as-unknown) or it is considered as outage while training.

In our evaluation distributional feature are giving a great improvement to the NNDR algorithm compare to it TF features performance. Moreover comparing it to the RFSE algorithm, our baseline, there is also an improvement in respect of macro-precision with some cost to the recall performance. Overall NNDR from a very low performance compare to the baseline it is becoming a comparable choice.

As concerns the terms-types, either for the TF features model of the distributional one in all cases W3G provided the maximum scores.

In this study we concluded that probably the distributional features is a prospective choice for the WGI research community. Thus, a farther research would be fruitful focusing on several important aspects the selection of the similarity measures other than cosine we have used. There are also several variation and parameters for the distributional features one can use, for instance we have used Bag-of-words-paragraph vectors for training on our NNet deep learning model.

Finally it almost became tradition to be underlined that there is a need for a well formed corpus specialized for the WGI task. We need larger corpora including multiple genre labels. In addition there is a need for a constant updating of this corpus to capture the new trends in the genre pallet of the web itself, since new genre classes are added and the current ones are rapidly changing context, as naturally the human communication is reformed in a temporal manner.

Bibliography

- [1] Asheghi, N.R.: Human Annotation and Automatic Detection of Web Genres. Ph.D. thesis, University of Leeds (2015)
- [2] Meyer zu Eissen, S., Stein, B.: Genre classification of web pages. KI 2004: Advances in Artificial Intelligence pp. 256–269 (2004)
- [3] Joho, H., Sanderson, M.: The spirit collection: an overview of a large web collection. In: ACM SIGIR Forum. vol. 38, pp. 57–61. ACM (2004)
- [4] Kanaris, I., Stamatatos, E.: Learning to recognize webpage genres. *Information Processing & Management* **45**(5), 499–512 (2009)
- [5] Mehler, A., Sharoff, S., Santini, M.: *Genres on the Web: Computational Models and Empirical Studies*. Text, Speech and Language Technology, Springer (2010)
- [6] Mendes Júnior, P.R., de Souza, R.M., Werneck, R.d.O., Stein, B.V., Pazinato, D.V., de Almeida, W.R., Penatti, O.A., Torres, R.d.S., Rocha, A.: Nearest neighbors distance ratio open-set classifier. *Machine Learning* pp. 1–28 (2016)
- [7] Pritsos, D., Stamatatos, E.: Open set evaluation of web genre identification. *Language Resources and Evaluation* pp. 1–20 (2018)
- [8] Pritsos, D.A., Stamatatos, E.: Open-set classification for automated genre identification. In: *Advances in Information Retrieval*, pp. 207–217. Springer (2013)
- [9] Santini, M.: Automatic identification of genre in web pages. Ph.D. thesis, University of Brighton (2007)
- [10] Sharoff, S., Wu, Z., Markert, K.: The web library of babel: evaluating genre collections. In: *Proceedings of the Seventh Conference on International Language Resources and Evaluation*. pp. 3063–3070 (2010)