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.
dpritsos@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 ave 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 PVBOW vector per Web-document for the corpus.

In order to compare our work to previews works, two different document representation types can be produced form 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 or 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 of character) $t_1, t_2, t_3, ..., t_T$.

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

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

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 sample from the text window and *form a classification task given the Paragraph Vector*. Thus the y of the eq.2 is formed to be $y = b + s(t_1, t_2, t_3, ..., t_{w_{size}})$ 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 splited to a set of paragraphs, as required from PVBOW. To be more specific the paragraphs are sentenses splited from all the document of the whole corpus. Then several models PVBOW feature models are trained for a variaty of parameters and vector dimentions, explained in the experiments section below. After the model has been fitted then one vector for each web-document was inferered from the PVBOW. The final document vectors derived from Distributional Feature Model are given to the open-set learning model explaind 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 equation3) on the *Known* and the *Marked as Unknown samples*.

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

where A_{KS} is the *Known Samples Accuracy* and A_{MUS} is the *Marked as Unknown Samples Accuracy*. The balance parameters $\lambda regulatesthemistakestrade-offontheknownandmarked-unknownsamplesprediction$.

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 of the marked as unknown samples. Thus $g \in 1,2,...,N$ and $g = \emptyset$ when samples is marked as unknown.

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

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 $\{1,2,...,N\}$, 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 keept 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, $C[g]$

$K_i^G, K_{validation^i}^G, U_{validation^i}^G, I^G = Split(G,p,k)$ 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.;

$V^G = U_{validation}^G \cup K_{validation}^G$ 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;

for *each* $i \in I$ **do**

  $D_{VK}^{cos}[i] = COS_D(V_i^G, K_i^G)$ calculating all the Cosine Distances between the web-page of K^G and V^G sets for *every I split combination*;

end

$Ci_A^{min} = argmin(D_{VK}^{cos})$ getting the indices of the closest classes from V;

$Ci_B^{min} = argmin(D_{VK}^{cos})$ getting the indices of the *second closest* classes from V;

$R_V = D_{VK}^{cos}[Di_A^{min}] / D_{VK}^{cos}[Di_B^{min}]$ calculating the Distance Rations R for all the vectors in V

$NA^{max} \leftarrow 0$ initializing *Maximized Normalized Accuracy* with 0 value. $DRT \leftarrow 0$ initializing *Distance Ratio Threshold* with 0 value.

for *each* $drt \in T$ **do**

  for *each* $r,i \in \{R_V, count(R_V)\}$ **do**

    if $r < drt$ **then**

      $vi = Ci_A^{min}[i]$ keep the respective index;

      $Y[i] = G[vi]$ setting the genre's class tag as prediction for this random vector of set V;

    else

      $Y[i] = \emptyset$ setting as none of the known genres or "I don't know";

    end

  end

  $NA_V = NormalizedAccuracy(Y, R_V)$ calculating the Normalized Accuracy as shown in equation 3 for tested threshold drt;

  if $NA_V > NA^{max}$ **then**

    $NA^{max} \leftarrow NA_V$ keeping the maximum NA until the outer for-loop finishes;

    $DRT \leftarrow drt$ keeping the *Distance Ratio Threshold* maximizes the *Normalized Accuracy*;

  else

  end

end

In the optimization procedure the training samples are splited 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 farther splited 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 Rations (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 $g \in 1, 2, ..., N$ 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*.

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

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

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

The NNDR is a open-set classification algorithm, therefore, every random sample will be classified to one of the classes the NNRD 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 pheudocode 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

for *each* $g \in G$ **do**
  $D^{cos}_{C_gX} = COS_D(C[g], X)$ calculating all the Cosine Distances between the random web-page vectors and the class vectors of class g;
end
$Ci^{min}_A = argmin(D^{cos}_{C_gW})$ getting the indices of the closest classes from W;
$Ci^{min}_B = argmin(D^{cos}_{C_gW})$ getting the indices of the *second closest* classes from W;
$R_W = D^{cos}_{C_gW}[Di^{min}_A] / D^{cos}_{C_gW}[Di^{min}_B]$ calculating the Distance Rations R for all the vectors in W
for *each* $r, i \in \{R_W, count(R_W)\}$ **do**
  if $r < DRT$ **then**
    $vi = Ci^{min}_A[i]$ keep the respective index;
    $Y[i] = G[vi]$ setting the genre's class tag as predicition for this random vector fo set W;
  else
    $Y[i] = \emptyset$ setting as none of the known genres or "I don't know";
  end
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 extention 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 $[0, 0.1, ..., 1.0]$. Precision values are interpolated based on the following formula:

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

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

5.2 Corpora

In this paper we study NNRD performace with distributional features derived from a NNet PVBOW coprus. 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 Uknown* (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

she-meshe-me 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-Vocabualry 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) $=\{0.2, 0.5, 0.7\}$ 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 splitted 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 splitted 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 classfications of the marked-unknown-classes as known an 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 dimentions for both algorithm is in all cases a few tousants.

MAX.	STP	SUP	DRT	λ	T.TYPE	DIMs	M*P*	M*R*	M*AUC*	M*F1*
F1	0.7	0.5	0.8	any	C4G	5000	0.664	0.403	0.296	0.502
AUC	0.7	0.5	0.8	any	C4G	5000	0.664	0.403	0.296	0.502
F1	0.7	0.5	0.8	any	W1G	5000	0.691	0.439	0.278	0.537
AUC	0.5	0.5	0.6	0.5	W1G	5000	0.943	0.202	0.191	0.333
F1	0.5	0.5	0.8	0.3 or 1.5	W3G	10000	0.720	0.664	0.417	0.691
AUC	0.5	0.5	0.6	0.5	W3G	5000	0.738	0.604	0.473	0.664

Table 2: Maximum performance of NNDR on TF Features of SANTINIS coprus. STP is the Spliting Training Percentage. SUP is the Spliting Unknown Percentage. DRT is the Distance Ration Threshold. λ is the weigting balance regulation paramters for the Normalised Accuracy. T.TYPE is the Terms Type. DIMs is the features model's dimentions. 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	M*P*	M*R*	M*AUC*	M*F1*
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 threhold of the RFSE. ITER is the number of iterations of the RFSE. T.TYPE is the Terms Type. DIMs is the features model's dimentions. 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.	M*P*	M*R*	M*AUC*	M*F1*
F1	any	any	0.8	any	C4G	50	3	8	0.025	10	0.002	0.829	0.600	0.411	0.696
AUC	any	any	0.8	any	C4G	500	3	3	0.025	10	0.02	0.755	0.602	0.462	0.670
F1	any	any	0.8	any	W1G	50	3	3	0.025	10	0.02	0.733	0.670	0.431	0.700
AUC	any	any	0.8	any	W1G	50	3	8	0.025	10	0.02	0.730	0.623	0.447	0.673
F1	any	any	0.8	any	W3G	100	3	3	0.025	10	0.02	0.827	0.615	0.488	0.706
AUC	any	any	0.8	any	W3G	100	3	3	0.025	10	0.02	0.827	0.615	0.488	0.706

Table 4: Maximum performance of NNDR on Distributional Features of SANTINIS coprus. STP, SUP, DRT, λ, T.TYPE, DIMs are the same as in table 2. MTF is the Minimum Threshold Fequency 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 **??** and **??** 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-precsion 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. Note that the only case where recall with TF features was better compare to distributional features is for W3G (see line 5 of table 2).

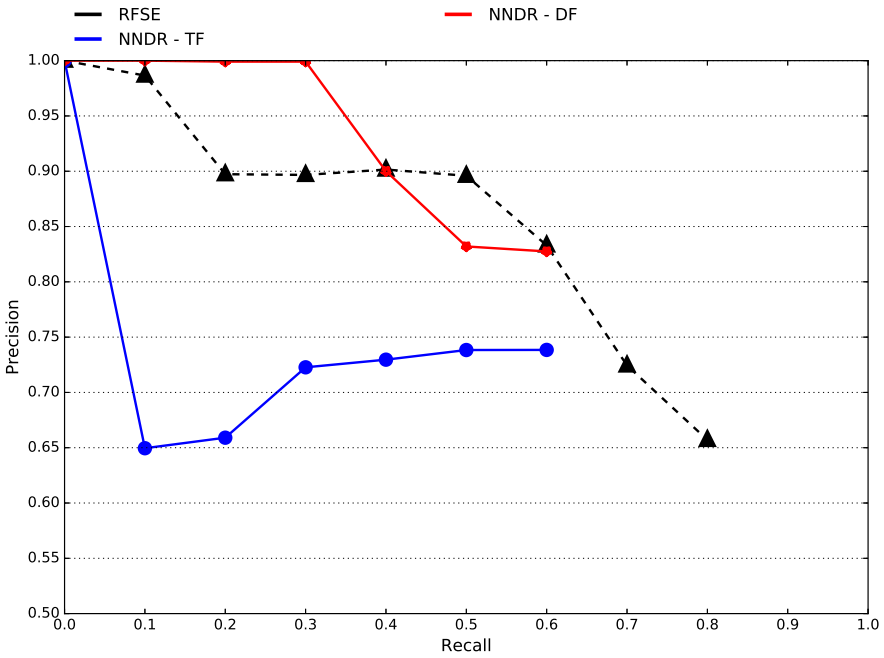

Fig. 1: Precision-Recall Curves of OCSVM and RFSE models on SANTINIS corpus optimized either by AUC or F_1.

7 Conclusions

In this paper we presented an experimental study on WGI focusing on open-set evaluation for this task. In contrast to vast majority of previous work in this area, we adopt the open-set scenario that is more realistic for WGI since it is not feasible to construct a genre palette with all available genres and appropriate samples for each one of them. Moreover, we examined two open-set classification methods and several feature types and similarity measures. To the best of our knowledge, this is the first time the performance of WGI models is evaluated using performance measures and tests specifically designed for open-set classification tasks.

The presented evaluation of open-set WGI covers two basic scenarios. The first is when noise is unstructured, i.e., information about the true genre of pages not belonging to the known genre palette is not available. The second scenario applies when noise is structured, i.e., we actually know the true genre of pages not included in the training classes. For both cases, we propose appropriate evaluation methodologies and present comparative results for the tested models.

In almost all examined cases, RFSE models outperformed the corresponding OCSVM models. This verifies previous work findings about the appropriateness of RFSE for WGI (8). RFSE is able to provide effective models and additionally it is possible to

manage preference on recall or precision, an application-dependent choice, by focusing on optimizing AUC or F_1 respectively. On the other hand, OCSVM proved to be the best-performing method in extreme cases when openness is high. Actually, the restrictions of the available corpora did not allow us to examine cases where openness approaches 1.0. However, it seems that when openness is more than 0.5 OCSVM outperforms RFSE.

As concerns the feature types, in most of the cases W3G and C4G provided the best results. However, the selection of text representation features is a crucial choice that affects performance and it seems to be corpus-dependent. Another crucial parameter of RFSE is the similarity measure. Among the examined measures, MinMax and its combination with cosine similarity provide the most robust results. The choice of similarity measure correlates with feature types. It seems that the combo measure is more effective than MinMax in low openness conditions.

To enhance the evaluation of WGI models in open-set conditions, we need larger corpora including multiple genre labels. New enhanced open-set WGI methods are needed and they should be evaluated using the proposed paradigm. Otherwise, using an evaluation paradigm more appropriate for closed-set tasks, the performance may be over-estimated.

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)