08-16-2018

METHOD AND RESULTS

1. APPROACHES

The process of automatically clustering publisher entities could be reduced to a simpler classification problem, where we build a predictive model using supervised machine learning techniques. Such a model is easier to evaluate.

Our work involves the following steps:

Creating a gold standard dataset

More details in the later sections

Representation of strings

Each string in the feature set was reduced to a numeric value using some technique.

Feature Selection

We evaluated the performance of different features in predicting the clusters, and the detailed report could be found in the following sections.

Classification

Each instance was categorized into one of the 4 clusters, each cluster representing a publisher entity.

2. CREATING A GOLD STANDARD DATASET

Two researchers clustered the publisher entities based on the names of the publisher. Also, a previously created gold standard was used to identify different publisher names which are the same entity.

The author came up with 4 cluster acronyms for identifying publishers:

1) Lerner
2) McGraw-Hill (MGH)
3) Random House (RH)
4) Others

Previous work by Connaway et al. (2012) used 7 clusters for MARC records. The name of the cluster (as identified by the author of this report) and the corresponding acronyms are as follows:

1) Lerner

2) Lerner Sports (Lernersports)
3) McGraw-Hill (MGH)
4) McGraw-Hill Ryerson (MGHR)
5) Random House (RH)
6) Random House, New Zealand (RHNZ)
7) Ediciones Lerner (Elearner)
8) Others

Comparison of the annotations by both the authors show a match of 976 out of 1002 instances.

	Ghosh								
Connaway		Lerner	Lernersports	MGH	MGHR	RH	RHNZ	Elearner	Others
	Lerner	**319**	0	0	0	0	0	0	4
	Lernersports	4	**0**	0	0	0	0	0	0
	MGH	0	0	**337**	0	0	0	0	0
	MGHR	0	0	2	**0**	0	0	0	0
	RH	0	0	1	0	**321**	0	0	2
	RHNZ	0	0	0	0	10	**0**	0	0
	Elearner	0	0	0	0	0	0	**0**	0
	Others	2	0	0	0	0	0	0	**0**

Table 1. Inter-annotator Agreement

Kappa value = (977 - 326.15)/ (1002 - 326.15) = 0.963

Using a different annotator, the kappa statistic obtained was 0.997. This highlights the strong agreement between annotators, showing the reliability of the obtained gold standard data.

3. REPRESENTATION OF STRINGS

In our work, all the strings were reduced to numeric features using three techniques:

3.1. TF-IDF

Term frequency, inverse document frequency scores which awards rare words present in a document for many times. Each feature column was treated as a collection of the document while individual instances were treated as documents.

There was a total of 38 features, not including the predicted variable.

3.2. HASHING

Every string was represented by hashing them to an n-dimensional vector, where the value of n was determined based on the feature.

The final number of dimensions was 232, not including the predicted variable.

3.3. BAG OF WORDS

Every string was represented using a bag-of-words, where the string was represented by 1-hot vector for the words in the collection.

The final number of dimensions was 212, not including the predicted variable.

For each of the three representations, we used two different techniques: one involving the removal of stopwords, and the other without stopword removal.

Features	Dimensions		
	TF-IDF	HASHING	BAG-OF-WORDS
ft_01_isbn	1	5	5
ft_02_isbn_pubname	1	5	5
ft_03_isbn_countryname	1	5	5
ft_04_lccn	1	1	1
ft_10_title	1	20	20
ft_11_statement_of_responsibility	1	5	5
ft_12_publisher_place	1	20	20
ft_13_publisher_name	1	30	30
ft_14_publication_date	1	1	1
ft_15_content_code	1	5	3
ft_16_content_type_term	1	5	5
ft_17_content_source	1	5	5
ft_18_media_code	1	5	1
ft_19_media_type_term	1	5	5
ft_2_physical_desc	1	5	5
ft_20_media_source	1	5	5
ft_21_carrier_code	1	5	4
ft_22_topical_term	1	5	5
ft_23_form_subdivision	1	5	5
ft_24_heading_source	1	5	5
ft_25_authority_record_control_num	1	5	5
ft_26_geographical_subdivision	1	5	5
ft_27_geo_name	1	5	5
ft_28_geo_source_of_headingwterm	1	5	5
ft_29_geo_arn	1	5	5
ft_3_place	1	5	5
ft_30_genre	1	5	1
ft_31_host_item	1	5	5
ft_32_series_title	1	5	5
ft_33_series_volume	1	5	5
ft_34_form_data	1	5	5
ft_35_source_of_term	1	5	5
ft_36_index_arn	1	5	5
ft_4_language	1	5	3
ft_5_catalogue_source	1	5	1
ft_8_personal_name	1	5	5
ft_9_relation	1	5	2

Table 2. String representation – feature dimensions

4. FEATURE SELECTION

Following techniques were used for determining the importance of features:

1. **Linear Regression model**
2. **L2 regularization** (called ridge regression for linear regression) adds the L2 norm penalty to the loss function. Since the coefficients are squared in the penalty expression, **Ridge regression** forces regressions coefficients to spread out similarly between correlated variables. It is useful for feature interpretation: a predictive feature will get a non-zero coefficient.
3. **Lasso** picks out the top performing features while forcing other features to be close to zero. It is useful when reducing the number of features is required.
4. **Stability selection** applies a feature selection algorithm on different subsets of data and with different subsets of features. After repeating the process, some times, the selection results can be aggregated, for example by checking how many times a feature ended up being selected as important when it was in an inspected feature subset. We can expect strong features to have scores close to 100% since they are always selected when possible. Weaker, but still relevant features will also have non-zero scores since they would be selected when stronger features are not present in the currently selected subset, while irrelevant features would have scores (close to) zero since they would never be among selected features.
5. **Recursive feature elimination** is a greedy optimization based on the idea to repeatedly construct a model and choose either the best or worst performing feature setting the feature aside and then repeating the process with the rest of the features. We have constructed the model using Linear Regression.
6. **The random forest** consists of a number of decision trees. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. Random forest's impurity-based ranking is typically aggressive in the sense that there is a sharp drop-off of scores after the first few top ones. Tree based methods can model non-linear relations well and don't require much tuning. For a forest, the impurity decrease from each feature can be averaged, and the features are ranked according to this measure.
7. Selecting **the k-best** features using chi-squared for classification.
8. With a **linear correlation** (Lin. corr.), each feature is evaluated independently, and we measure the linear relationship between each feature and the response variable.
9. **Extra Trees Classifier**: Another tree-based classifier

(Refer to files: feature-analysis-[TFIDF/HASHING/CV]-[NOSTOPREM] for feature-based scores)

The scores generated using the above techniques were averaged to arrive at the final score. The top-10 features (along with the mean score) is shown in Tables 3-8. In each of the tables, we have included the publisher name (ft_13_publisher_name) obtained from MARC records to compare its importance to the other features.

4.1. HASHING

Feature Name	Mean Score
ft_04_lccn	0.18
ft_21_carrier_code	0.18
ft_20_media_source	0.15
ft_19_media_type_term	0.14
ft_18_media_code	0.13
ft_29_geo_arn	0.13
ft_15_content_code	0.12
ft_17_content_source	0.12
ft_24_heading_source	0.12
ft_36_index_arn	0.12
ft_13_publisher_name	**0.11**

Table 3. Top features using hashing (stopwords removed)

Feature Name	Mean Score
ft_18_media_code	0.21
ft_04_lccn	0.19
ft_15_content_code	0.19
ft_20_media_source	0.16
ft_16_content_type_term	0.14
ft_17_content_source	0.14
ft_19_media_type_term	0.14
ft_21_carrier_code	0.14
ft_24_heading_source	0.13
ft_25_authority_record_control_num	0.13
ft_13_publisher_name	**0.08**

Table 4. Top features using hashing (stopwords not removed)

4.2. BAG OF WORDS

Feature Name	Mean Score
ft_20_media_source	0.23
ft_16_content_type_term	0.21
ft_17_content_source	0.2
ft_19_media_type_term	0.19
ft_13_publisher_name	**0.16**

ft_15_content_code	0.15
ft_18_media_code	0.15
ft_31_host_item	0.15
ft_21_carrier_code	0.14
ft_12_publisher_place	0.13

Table 5. Top features for bag-of-words (stopwords removed)

Feature Name	Mean Score
ft_17_content_source	0.22
ft_16_content_type_term	0.2
ft_20_media_source	0.19
ft_19_media_type_term	0.17
ft_13_publisher_name	**0.16**
ft_18_media_code	0.16
ft_15_content_code	0.14
ft_21_carrier_code	0.14
ft_12_publisher_place	0.13
ft_5_catalogue_source	0.13
ft_31_host_item	0.12

Table 6. Top features for bag-of-words (stopwords not removed)

4.3. TF-IDF

Feature Name	Mean Score
ft_13_publisher_name	**0.4**
ft_19_media_type_term	0.27
ft_12_publisher_place	0.24
ft_02_isbn_pubname	0.18
ft_5_catalogue_source	0.18
ft_4_language	0.17
ft_11_statement_of_responsibility	0.16
ft_18_media_code	0.16
ft_31_host_item	0.16
ft_34_form_data	0.15

Table 7. Top features for tf-idf (stopwords removed)

Feature Name	Mean Score
ft_13_publisher_name	**0.31**
ft_19_media_type_term	0.29
ft_12_publisher_place	0.26
ft_11_statement_of_responsibility	0.16
ft_2_physical_desc	0.16

ft_31_host_item	0.16
ft_18_media_code	0.15
ft_34_form_data	0.15
ft_4_language	0.15
ft_02_isbn_pubname	0.14

Table 8. Top features for tf-idf (stopwords not removed)

As TF-IDF used scores to represent every feature, it was not lossless in nature. However, it was extremely efficient in terms of computational time and resources. As is evident from Tables 5 and 6, when using TF-IDF, the publisher name (as obtained from MARC records) is most informative. One of the possible reasons could be the multiword nature of the feature, as is observed for publisher place, physical description, and statement of responsibility.

5. CLASSIFICATION

Classification of the instances was done using a k-NN algorithm with 3,5 and 7 clusters. The dataset was split into 70% for training and 30% for testing.

The process of classifying the instances was repeated 10 times, with different selection of training and test sets. The final accuracy for different configurations are reported in Table 9.

(Refer to files: predictions- [TFIDF/HASHING/CV]-[NOSTOPREM] for feature prediction-based scores)

String Representation	Stopwords Removed	Classifier	Features Selected	Accuracy
TF-IDF	Yes	k-NN (k=3)	All	0.937
		k-NN (k=5)	All	0.942
		k-NN (k=7)	All	0.939
	No	k-NN (k=3)	All	0.927
		k-NN (k=5)	All	0.925
		k-NN (k=7)	All	0.918
Hashing	Yes	k-NN (k=3)	All	0.927
		k-NN (k=5)	All	0.924
		k-NN (k=7)	All	0.927
	No	k-NN (k=3)	All	0.921
		k-NN (k=5)	All	0.926
		k-NN (k=7)	All	0.925
Bag-of-words	Yes	k-NN (k=3)	All	0.978
		k-NN (k=5)	All	0.979
		k-NN (k=7)	All	0.977
	No	k-NN (k=3)	All	0.978
		k-NN (k=5)	All	0.977
		k-NN (k=7)	All	0.976

Table 9. Classification Performance

The confusion matrix is presented in Table 10 for the classification results using Bag-of-words approach, no stopword removal and all features. The total number of possible clusters were 4,

	Others	Lerner	MGH	RH
Others	0	2	0	1
Lerner	0	103	1	0
MGH	0	3	84	1
RH	0	0	0	106

Table 10. Confusion Matrix

We have used 'Others' to identify publisher names like Turnaround [distributor], Publishers Group UK [distributor], and Random Century. The test set looks evenly distributed with Lerner, McGraw-Hill, Random House containing 104, 88, and 106 instances respectively. Our classifier failed to recognize the class 'Other' as there were fewer instances in the Training set.

6. DISCUSSIONS

6.1 How useful is ISBN?

Most important features if the ISBN information is removed from the dataset:

1) ft_17_content_source
2) ft_20_media_source
3) ft_13_publisher_name
4) ft_16_content_type_term
5) ft_15_content_code

The above features were consistent with and without stopwords.

The accuracy of the classifier varied between 97-98%, which clearly highlights that ISBN is not one of the most important features for classifying publisher entities.