Towards Automatic Topical Classification of LOD Datasets

Robert Meusel
Data and Web Science Group
University of Mannheim
B6 26, Mannheim, Germany
robert@dwslab.de

Blerina Spahiu
Department of Computer Science, Systems and Communication
University of Milan Bicocca
Viale Sarca, 336 20126 Milano
spahiu@disco.unimib.it

Christian Bizer
Data and Web Science Group
University of Mannheim
B6 26, Mannheim, Germany
chris@dwslab.de

Heiko Paulheim
Data and Web Science Group
University of Mannheim
B6 26, Mannheim, Germany
heiko@dwslab.de

ABSTRACT

The datasets that are part of the Linking Open Data cloud diagramm (LOD cloud) are classified into the following topical categories: media, government, publications, life sciences, geographic, social networking, user-generated content, and cross-domain. The topical categories were manually assigned to the datasets. In this paper, we investigate to which extent the topical classification of new LOD datasets can be automated using machine learning techniques and the existing annotations as supervision. We conducted experiments with different classification techniques and different feature sets. The best classification technique/feature set combination reaches an accuracy of 81.62% on the task of assigning one out of the eight classes to a given LOD dataset. A deeper inspection of the classification errors reveals problems with the manual classification of datasets in the current LOD cloud.

Keywords

Linked Open Data, Topic Detection, Data Space Profiling

1. INTRODUCTION

The Web of Linked Data offers a rich collection of structured data provided by hundreds of different data sources that use common standards such as dereferencable URIs and RDF. The central idea of Linked Data is that data sources set RDF links pointing at other data sources – e.g., `owl:sameAs` links – so that all data is connected into a global data space [3, 8]. In this data space, agents can navigate from one data source to another by following RDF links, thereby discovering new data sources on the fly.

Since the proposal of the Linked Data best practices in 2006, the Linked Open Data cloud (LOD cloud) has grown to roughly 1 000 datasets (as of April 2014) [15]. The datasets cover various topical domains, with social media, government data, and metadata about publications being the most prominent areas [15].

The most well-known categorization of LOD datasets by topical domain is the coloring of the LOD cloud diagram.[1] Up till now, the topical categories were manually assigned to the datasets in the cloud either by the publishers of the datasets themselves via the `datahub.io` dataset catalog or by the authors of the LOD cloud diagram. In this paper, we investigate to which extent the topical classification of new LOD datasets can be automated for upcoming versions of the LOD cloud diagram using machine learning techniques and the existing annotations as supervision.

Beside creating upcoming versions of the LOD cloud diagram, the automatic topical classification of LOD datasets can be interesting for other purposes as well: Agents navigating on the Web of Linked Data should know the topical domain of datasets that they discover by following links in order to judge whether the datasets might be useful for their use case at hand or not. Furthermore, as shown in [15], it is interesting to analyze characteristics of datasets grouped by topical domain, so that trends and best practices that exist only in a particular topical domain can be identified.

In this paper, we present – to the best of our knowledge – the first automatic approach to classify LOD datasets into the topical categories that are used by the LOD cloud diagram. Using the data catalog underlying the recent LOD cloud, we train machine learning classifiers with different sets of features. Our best classification technique/feature set combination reaches an accuracy of 82%.

The rest of this paper is structured as follows. Section 2 introduces the methodology of our experiments, followed by a presentation of the results in Section 3 and a discussion of remaining classification errors in Section 4. Section 5 gives an overview of related work. We conclude with a summary and an outlook on future work.

WWW2015 Workshop: Linked Data on the Web (LDOW2015).

[1] `http://lod-cloud.net`

2. METHODOLOGY

In this section, we first briefly describe the data corpus that we use for our experiments and the different feature sets we derive from the data. We than briefly introduce the classification techniques that we considered and sketch the final experimental setup that was used for the evaluation.

2.1 Data Corpus

In order to extract our features for the different datasets which are contained in the LOD cloud, we used the data corpus that was crawled by Schmachtenberg et al. [15] and which was used to draw the most recent LOD cloud diagram. Schmachtenberg et al. used the *LD-Spider* framework [9] to gather Linked Data from the Web in April 2014. The crawler was seeded with URIs from three different sources: (1) dataset descriptions in *lod-cloud* group of the `datahub.io` dataset catalog, as well as other datasets marked with Linked Data related tags within the catalog; (2) a sample of the *Billion Triple Challenge* 2012 dataset[2]; and (3) datasets advertised on the `public-lodw3.org` mailing list since 2011. The final crawl contains data from 1 014 different LOD datasets.[3] Altogether 188 million RDF triples were extracted from 900 129 documents describing 8 038 396 resources. Figure 1 shows the distribution of the number of resources and documents per dataset contained in the crawl.

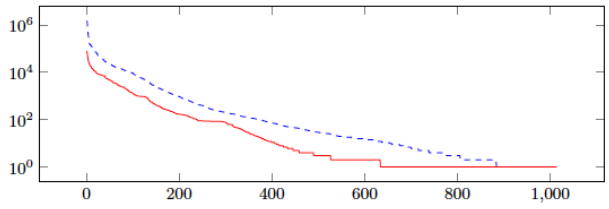

Figure 1: Distribution of the number of resources (– – –) and documents (——) (log scale) per dataset contained in the crawl.

In order to create the 2014 version of the LOD cloud diagram, newly discovered datasets were manually classified into one of the following categories: *media, government, publications, life sciences, geographic, social networking, user-generated content,* and *cross-domain*. A detailed definition of each category is available in [15].

Figure 2 shows the number of datasets per category contained in the 2014 version of the LOD cloud. As we can see, the LOD cloud is dominated by datasets belonging to the category *social networking* (48%), followed by *government* (18%) and *publications* (13%) datasets. The categories *media* and *geographic* are only represented by less than 25 datasets within the whole corpus.

2.2 Feature Sets

For each of the datasets, we created the following eight feature sets based on the crawled data.

Vocabulary Usage (VOC): As many vocabularies target a specific topical domain, e.g. `bibo` bibliographic information, we assume that the vocabularies that are used by a dataset form a helpful indicator for determining the topical category of the dataset. Thus, we determine the vocabulary of all terms that are used as predicates or as the object of a type statement within each dataset. Altogether we identified 1 439 different vocabularies being used by the datasets (see [15] for details about the most widely used vocabularies).

[2]`http://km.aifb.kit.edu/projects/btc-2012/`

[3]The crawled data is publicly available: `http://data.dws.informatik.uni-mannheim.de/lodcloud/2014/ISWC-RDB/`

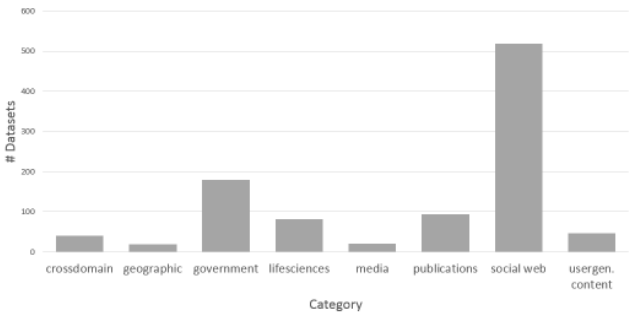

Figure 2: Number of datasets per category contained in the LOD cloud.

Class URIs (CUri): As a more fine-grained feature, the `rdfs:` and `owl:classes` which are used to describe entities within a dataset might provide useful information to determine the topical category of the dataset. Thus, we extracted all the classes that are used by at least two different datasets, resulting in 914 attributes for this feature set.

Property URIs (PUri): Beside the class information of an entity, information about which properties are used to describe the entity can be helpful. For example it might make a difference, if a person is described with `foaf:knows` statements or if her professional affiliation is provided. To leverage this information, we collected all properties that are used within the crawled data by at least two datasets. This feature set consists of 2 333 attributes.

Local Class Names (LCN): Different vocabularies might contain synonymous (or at least closely related) terms that share the same local name and only differ in their namespace, e.g. `foaf:Person` and `dbpedia:Person`. Creating correspondences between similar classes from different vocabularies reduces the diversity of features, but on the other side might increase the number of attributes which are used by more than one dataset. As we lack correspondences between all the vocabularies, we bypass this, by using only the local names of the type URIs, meaning `vocab1:Country` and `vocab2:Country` are mapped to the same attribute. We used a simple regular expression to determine the local class name checking for `#`, `:` and `/` within the type object. By focusing only on the local part of a class name, we increase the number of classes that are used by more than one dataset in comparison to CUri and thus generate 1 041 attributes for the LCN feature set.

Local Property Names (LPN): Using the same assumption as for the LCN feature set, we also extracted the local name of each property that is used by a dataset.

This results in treating `vocab1:name` and `vocab2:name` as a single property. We used the same heuristic for the extraction as for the LCN feature set and generated 3 493 different local property names which are used by more than one dataset, resulting in an increase of the number of attributes in comparison to the PUri feature set.

Text from `rdfs:label` (LAB): Beside the vocabulary-level features, the names of the described entities might also indicate the topical domain of a dataset. We thus extracted all values of `rdfs:label` properties, lower-cased them, and tokenized the values at space-characters. We further excluded tokens shorter than three and longer than 25 characters. Afterward, we calculated the TF-IDF value for each token while excluding tokens that appeared in less than 10 and more than 200 datasets, in order to reduce the influence of noise. This resulted in a feature set consisting of 1 440 attributes.

Top-Level Domains (TLD): Another feature which might help to assign datasets to topical categories is the top-level domain of the dataset. For instance, government data is often hosted in the `gov` top-level domain, whereas library data might be found more likely on `edu` or `org` top-level domains.[4]

In & Outdegree (DEG): In addition to vocabulary-based and textual features, the number of outgoing RDF links to other datasets and incoming RDF links from other datasets could provide useful information for classifying the datasets. This feature could give a hint about the density of the linkage of a dataset, as well as the way the dataset is interconnected within the whole LOD cloud ecosystem.

We were able to create all features (except LAB) for 1 001 datasets. As only 470 datasets provide `rdfs:label`s, we only use these datasets for evaluating the utility of the LAB feature set.

As the total number of occurrences of vocabularies and terms is heavily influenced by the distribution of entities within the crawl for each dataset, we apply two different normalization strategies to the values of the vocabulary-level features VOC, CUri, PUri, LCN, and LPN: On the one hand side, we create a *binary* version (*bin*) where the feature vectors of each feature set consist of 0 and 1 indicating presence and absence of the vocabulary or term. The second version, the *relative term occurrence* (*rto*), captures the fraction of vocabulary or term usage for each dataset.

The following table shows an example of the two different feature set versions for the terms t_i:

	Feature Vector			
Feature Set Version	t_1	t_2	t_3	t_4
Term Occurrence	10	0	2	8
Binary (*bin*)	1	0	1	1
Relative Term Occurrence (*rto*)	0.5	0	0.1	0.4

[4]We restrict ourselves to top-level domains, and not public suffixes.

2.3 Classification Approaches

We evaluated the following three classification techniques on our task of assigning topical categories to LOD datasets.

***k*-Nearest Neighbor:** *k*-Nearest Neighbor (*k*-NN) classification models make use of the similarity between new cases and known cases to predict the class for the new case. A case is classified by its majority vote of its neighbors, with the case being assigned to the class most common among its k nearest neighbors measured by the distance function. In our experiments we used a k equal to 5 with *Euclidean*-similarity for non-binary term vectors and *Jaccard*-similarity for binary term vectors.

J48 Decision Tree: A decision tree is a flowchart-like tree structure which is built top-down from a root node and involves some partitioning steps to divide data into subsets that contain instances with similar values. For our experiments we use the *Weka* implementation of the C4.5 decision tree [12]. We learn a pruned tree, using a confidence threshold of 0.25 with a minimum number of 2 instances per leaf.

Naive Bayes: As a last classification method, we used Naive Bayes (NB). NB uses joint probabilities of some evidence to estimate the probability of some event. Although this classifier is based on the assumption that all features are independent, which is violated in many use cases, NB has shown to work well in practice [14].

2.4 Experimental Setup

In order to evaluate the performance of the three classification methods, we use 10-fold cross-validation and report the average accuracy in the end.

As the number of datasets per category is not equally distributed within the LOD cloud, which might influence the performance of the classification models, we also explore the effect of balancing the training data. We used two different balancing approaches: (1) we *down sample* the number of datasets used for training until each category is represented by the same number of datasets; this number is equal to the number of datasets within the smallest category; and (2) we *up sample* the datasets for each category until each category is at least represented by the number of datasets equal to the number of datasets of the largest category. The first approach, reduces the chance to overfit a model into the direction of the larger represented classes, but it might also remove valuable information from the training set, as examples are removed and not taken into account for learning the model. The second approach, ensures that all possible examples are taken into account and no information is lost for training, but by creating the same entity many times can result in emphasizing those particular data points. For example a neighborhood based classifier might look at the 5 nearest neighbors, which than could be one and the same data point, which would result into looking only at *the* nearest neighbor.

3. RESULTS

In the following, we first report the results of our experiments using the different feature sets in separation. Afterward, we report the results of experiments combining attributes from multiple feature sets.

Table 1: Results of different single feature sets. Best three single and average results are marked in bold.

Classification	VOC		CUri		PUri		LCN		LPN				
Approach	bin	rto	bin	rto	bin	rto	bin	rto	bin	rto	LAB	TLD	DEG
Major Class	51.85	51.85	51.85	51.85	51.85	51.85	51.85	51.85	51.85	51.85	33.62	51.85	51.85
k-NN (no sampling)	77.92	76.33	76.83	74.08	**79.81**	75.30	76.73	74.38	**79.80**	76.10	53.62	58.44	49.25
k-NN (down sampling)	64.74	66.33	68.49	60.67	71.80	62.70	68.39	65.35	73.10	62.80	19.57	30.77	29.88
k-NN (up sampling)	71.83	72.53	64.98	67.08	75.60	71.89	68.87	69.82	76.64	70.23	43.97	10.74	11.89
J48 (no sampling)	78.83	79.72	78.86	76.93	77.50	76.40	**80.59**	76.83	78.70	77.20	63.40	67.14	54.45
J48 (down sampling)	57.65	66.63	65.35	65.24	63.90	63.00	64.02	63.20	64.90	60.40	25.96	34.76	24.78
J48 (up sampling)	76.53	77.63	74.13	76.60	75.29	75.19	77.50	75.92	75.91	74.46	52.64	45.35	29.47
Naive Bayes (no sampling)	34.97	44.26	75.61	57.93	78.90	75.70	77.74	60.77	78.70	76.30	40.00	11.99	22.88
Naive Bayes (down sampling)	64.63	69.14	64.73	62.39	68.10	66.60	70.33	61.58	68.50	69.10	33.62	20.88	15.99
Naive Bayes (up sampling)	77.53	44.26	74.98	55.94	77.78	76.12	76.02	58.67	76.54	75.71	37.82	45.66	14.19
Average (no sampling)	63.91	66.77	77.10	69.65	**78.73**	75.80	**78.35**	70.66	**79.07**	76.53	52.34	45.86	42.19
Average (down sampling)	62.34	67.34	66.19	62.77	67.93	64.10	67.58	63.38	68.83	64.10	26.38	28.80	23.55
Average (up sampling)	75.30	64.81	71.36	66.54	76.22	74.40	74.13	68.14	76.36	73.47	44.81	33.92	18.52

3.1 Results for Single Feature Sets

Table 1 shows the accuracy that is reached using the three different classification algorithms with and without balancing the training data. *Majority Class* is the performance of a default baseline classifier always predicting the largest class: *social networking*.

As a general observation, the vocabulary-based feature sets (VOC, LCN, LPN, CUri, PUri) perform on a similar level, where DEG and TLD alone show a relatively poor performance and in some cases are not at all able to beat the majority class baseline. Classification models based on the attributes of the LAB feature set perform on average (without sampling) around 20% above the majority baseline, but predict still in half of all cases the wrong category. Algorithm-wise, the best results are achieved using the decision tree (J48) without balancing (maximal accuracy 80.59% for LCN_{rto}) and the k-NN algorithm, also without balancing for the PUri_{bin} and LPN_{bin} feature sets. Comparing the two balancing approaches, we see better results using the up sampling approach for almost all feature sets (except VOC_{rto} and DEG). In most cases, the category-specific accuracy of the smaller categories is higher when using up sampling. Using down sampling the learned models make more errors for predicting the larger categories. Furthermore, when comparing the results of the models trained on unbalanced data with the best model trained on balanced data, the models on the unbalanced data are more accurate except for the VOC_{bin} feature set. Having a closer look at the confusion matrices, we see that the balanced approaches are in general making more errors when trying to predict datasets for the larger categories, like *social networking* and *government*.

3.2 Results for Combined Feature Sets

For our second set of experiments, we combine the available attributes from the different feature sets and train again our classification models using the three described algorithms. As before, we generate a *binary* and *relative term occurrence* version of the vocabulary-based features. In addition, we create a second set (*binary* and *relative term occurrence*), where we omit the attributes from the LAB feature set, as we wanted to measure the influence of this particular set of attributes, which is only available for less than half of the datasets. Furthermore we created a combined set of attributes consisting of the three best performing feature sets from the previous section.

Table 2 reports the results for the five different combined feature sets:

ALL_{rto}: Combination of the attributes from all eight feature sets, using the *rto* version of the vocabulary-based features.

ALL_{bin}: Combination of the attributes from all eight feature sets, using the *bin* version of the vocabulary-based features.

NoLab_{rto}: Combination of the attributes from all feature, without the attributes of the LAB feature set, using the *rto* version of the vocabulary-based features.

NoLab_{bin}: Combination of the attributes from all feature, without the attributes of the LAB feature set, using the *bin* version of the vocabulary-based features.

Best3: Includes the attributes from the three best performing feature sets from the previous section based on their average accuracy: PUri_{bin}, LCN_{bin}, and LPN_{bin}.

We can observe that when selecting a larger set of attributes, our model is able to reach a slightly higher accuracy of 81.62% than using just the attributes from one feature set (80.59%, LCN_{bin}). Still the trained model is unsure for certain decisions and has a stronger bias towards the categories *publications* and *social networking*.

4. DISCUSSION

In the following, we look at the best performing approach (Naive Bayes trained on the attributes of the NoLab_{bin} feature set using up sampling). Table 3 shows the confusion matrix of this experiment, where on the left side we list the predictions by the learned model, while the head names the actual category of the dataset. As observed in the table, there are three kinds of errors which occur more frequently than 10 times.

The most common confusion occurs for the *publication* domain, where a larger number of datasets are predicted to belong to the *government* domain. A reason for this is that government datasets often contain metadata about government statistics which are represented using the same vocabularies and terms (e.g. *skos:Concept*) that are also used in the publication domain. This makes it challenging for a vocabulary-based classifier to distinguish those two categories apart. In addition, for example the `http://mcu.es`

Table 2: Results of combined feature sets. Best three results in bold.

Classification Approach	Accuracy in %				
	ALL_{bin}	ALL_{rto}	$NoLab_{bin}$	$NoLab_{rto}$	Best3
k-NN (no sampling)	74.93	71.73	76.93	72.63	75.23
k-NN (down sampling)	52.76	46.85	65.14	52.05	64.44
k-NN (up sampling)	74.23	67.03	71.03	68.13	73.14
J48 (no sampling)	**80.02**	77.92	79.32	79.01	75.12
J48 (down sampling)	63.24	63.74	65.34	65.43	65.03
J48 (up sampling)	79.12	78.12	79.23	78.12	75.72
Naive Bayes (no sampling)	21.37	71.03	**80.32**	77.22	76.12
Naive Bayes (down sampling)	50.99	57.84	70.33	68.13	67.63
Naive Bayes (up sampling)	21.98	71.03	**81.62**	77.62	76.32

Table 3: Confusion matrix for the $NoLab_{bin}$ feature set, with Naive Bayes classification model, balanced by up sampling.

	True Category							
Prediction	social networking	crossdomain	publications	government	lifesciences	media	usergen. content	geographic
social networking	489	4	5	10	2	4	11	1
crossdomain	1	10	3	1	1	0	1	1
publications	8	10	54	9	4	4	2	2
government	3	4	14	151	1	2	0	2
lifesciences	5	3	12	0	72	2	5	5
media	6	3	4	1	1	7	2	0
usergen. content	6	1	1	2	0	2	26	0
geographic	1	5	1	5	1	0	0	8

dataset – the Ministry of Culture in Spain – was manually labeled as *publication* within the LOD cloud, whereas the model predicts *government* which turns out to be a borderline case in the gold standard. A similar frequent problem is the prediction of *life sciences* for datasets in the *publications* category. This can be observed, e.g., for the `http://ns.nature.com/publications/`, which describe the publications in *Nature*. Those publications, however, are often in the life sciences field, which makes the labeling in the gold standard a borderline case.

The third most common confusion occurs between the *usergenerated content* and the *social networking* domain. Here, the problem is in the shared use of similar vocabularies, such as *foaf*. At the same time, labeling a dataset as either one of the two is often not so simple. In [15], it has been defined that *social networking* datasets should focus on the presentation of people and their interrelations, while *usergenerated content* should have a stronger focus on the content. Datasets from personal blogs, such as `www.wordpress.com`, however, can convey both aspects. Due to the labeling rule, these datasets are labeled as *usergenerated content*, but our approach frequently classifies them as *social networking*.

In summary, while we observe some true classification errors, many of the mistakes made by our approach actually point at datasets which are difficult to classify, and which are rather borderline cases between two categories.

5. RELATED WORK

Topical profiling has been studied in the data mining, database, and information retrieval communities. The resulting methods find application in domains such as documents classification, contextual search, content management and review analysis [1, 11, 2, 16, 17].

Although topical profiling has been studied in other settings before, only a small number of methods exist for profiling LOD datasets. These methods can be categorized based on the general learning approach that is employed into the categories *unsupervised* and *supervised*. Where the first category does not rely on labeled input data, the latter is only applicable for labeled data.

Ellefi et al. [5] try to define the profile of datasets using semantic and statistical characteristics. They use statistics about vocabulary, property, and datatype usage, as well as statistics on property values, like string lengths, for characterizing datasets. For classification, they propose a feature/characteristic generation process, starting from the top discovered types of a dataset and generating property/value pairs. In order to integrate the property/value pairs they consider the problem of vocabulary heterogeneity of the datasets by defining correspondences between features in different vocabularies. The authors have pointed out that it is essential to automate the feature generations and proposed the framework to do so, but do not evaluate their approach on real-world datasets. In our work, we draw from their ideas of using schema-usage characteristics as features for the topical classification, but focus on LOD datasets.

An approach to detect latent topics in entity-relationship graphs is introduced by Böhm et al. [4]. Their approach works in two phases: (1) A number of subgraphs having strong relations between classes are discovered from the whole graph, and (2) the subgraphs are combined to generate a larger subgraph, which is assumed to represent a latent topic. Their approach explicitly omits any kind of features based on textual representations and solely relies on the exploitation of the underlying graph. Böhm et al. used the *DBpedia* dataset to evaluate their approach.

Fetahu et al. [6] propose an approach for creating dataset profiles represented by a weighted dataset-topic graph which is generated using the category graph and instances from DBpedia. In order to create such profiles, a processing pipeline that combines tailored techniques for dataset sampling, topic extraction from reference datasets, and relevance ranking is used. Topics are extracted using named-entity-recognition techniques, where the ranking of the topics is based on their normalized relevance score for a dataset.

While the mentioned approaches are *unsupervised*, we employ *supervised* learning techniques as we want to exploit the existing topical annotation of the datasets in the LOD cloud.

6. CONCLUSION AND FUTURE WORK

In this paper, we investigate to which extent the topical classification of new LOD datasets can be automated using machine learning techniques. Our experiments indicate that

vocabulary-level features are a good indicator for the topical domain, yielding an accuracy of around 82%.

The analysis of the limitations of our approach, i.e., the cases where the automatic classification deviates from the manually labeled one, points to a problem of the categorization approach that is currently used for the LOD cloud: All datasets are labeled with exactly *one* topical category, although sometimes two or more categories would be equally appropriate. One such example are datasets describing life science publications, which can be either labeled as *publications* or as *life sciences*. Thus, the LOD dataset classification task might be more suitably formulated as a *multi-label classification* problem [18, 10].

A particular challenge of the classification is the heavy imbalance of the dataset categories, with roughly half of the datasets belonging to the *social networking* domain. Here, a two-stage approach might help, in which a first classifier tries to separate the largest category from the rest, while a second classifier then tries to make a prediction for the remaining classes. When regarding the problem as a multi-label problem, the corresponding approach would be *classifier chains*, which make a prediction for one class after the other, taking the prediction of the first classifiers into account as a feature for the remaining classifications [13].

In our experiments, RDF links have not been exploited beyond dataset in- and out-degree. For the task of web page classification, *link-based classification techniques*, that exploit the contents of web pages linking to a particular page, often yields good results [7] and it is possible that such techniques could also work well for classifying LOD datasets.

Acknowledgements

This research has been supported in part by FP7/2013-2015 COMSODE (under contract number FP7-ICT-611358).

7. REFERENCES

[1] C. C. Aggarwal and C. Zhai. A survey of text clustering algorithms. In *Mining Text Data*, pages 77–128. 2012.

[2] T. Basu and C. A. Murthy. Effective text classification by a supervised feature selection approach. In *12th IEEE International Conference on Data Mining Workshops, ICDM Workshops, Brussels, Belgium, December 10, 2012*, pages 918–925, 2012.

[3] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[4] C. Böhm, G. Kasneci, and F. Naumann. Latent topics in graph-structured data. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 2663–2666, 2012.

[5] M. B. Ellefi, Z. Bellahsene, F. Scharffe, and K. Todorov. Towards semantic dataset profiling. In *Proceedings of the 1st International Workshop on Dataset PROFIling & fEderated Search for Linked Data co-located with the 11th Extended Semantic Web Conference, PROFILES@ESWC 2014, Anissaras, Crete, Greece, May 26, 2014.*, 2014.

[6] B. Fetahu, S. Dietze, B. P. Nunes, M. A. Casanova, D. Taibi, and W. Nejdl. A scalable approach for efficiently generating structured dataset topic profiles. In *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, pages 519–534, 2014.

[7] L. Getoor and C. P. Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 2005.

[8] T. Heath and C. Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.

[9] R. Isele, J. Umbrich, C. Bizer, and A. Harth. LDSpider: An open-source crawling framework for the web of linked data. In *Proc. ISWC '10 –Posters and Demos*, 2010.

[10] B. Liu, X. Li, W. S. Lee, and P. S. Yu. Text classification by labeling words. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, pages 425–430, 2004.

[11] J. Nam, J. Kim, E. Loza Mencía, I. Gurevych, and J. Fürnkranz. Large-scale multi-label text classification - revisiting neural networks. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II*, pages 437–452, 2014.

[12] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[13] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009.

[14] I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.

[15] M. Schmachtenberg, C. Bizer, and H. Paulheim. Adoption of the linked data best practices in different topical domains. In *The Semantic Web–ISWC 2014*, pages 245–260. Springer, 2014.

[16] P. Shivane and R. Rajani. A survey on effective quality enhancement of text clustering & classification using metadata.

[17] G. Song, Y. Ye, X. Du, X. Huang, and S. Bie. Short text classification: A survey. *Journal of Multimedia*, 9(5):635–643, 2014.

[18] G. Tsoumakas and I. Katakis. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.