

2018

Validation of Tagging Suggestion Models for a Hotel Ticketing Corpus

Bojan Bozic

Technological University Dublin, bojan.bozic@tudublin.ie

Andre Rios

Technological University Dublin

Sarah Jane Delany

Technological University Dublin, sarahjane.delany@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomcon>

 Part of the [Databases and Information Systems Commons](#)

Recommended Citation

Bozic, B., Rios, A. & Delany, S.J. (2018). Validation of tagging suggestion models for a hotel ticketing corpus. *Proceedings of iiWAS2018 Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services*, Yogyakarta, Indonesia, November 19 - 21, pp.1523. doi:10.1145/3282373.3282386

This Conference Paper is brought to you for free and open access by the School of Computer Science at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, vera.kilshaw@tudublin.ie.

Validation of Tagging Suggestion Models for a Hotel Ticketing Corpus

Bojan Božić
Centre for Applied Data Analytics
Research
Dublin Institute of Technology
Ireland

André Ríos
Centre for Applied Data Analytics
Research
Dublin Institute of Technology
Ireland

Sarah Jane Delany
Centre for Applied Data Analytics
Research
Dublin Institute of Technology
Ireland

ABSTRACT

This paper investigates methods for the prediction of tags on a textual corpus that describes hotel staff inputs in a ticketing system. The aim is to improve the tagging process and find the most suitable method for suggesting tags for a new text entry. The paper consists of two parts: (i) exploration of existing sample data, which includes statistical analysis and visualisation of the data to provide an overview, and (ii) evaluation of tag prediction approaches. We have included different approaches from different research fields in order to cover a broad spectrum of possible solutions. As a result, we have tested a machine learning model for multi-label classification (using gradient boosting), a statistical approach (using frequency heuristics), and two simple similarity-based classification approaches (Nearest Centroid and k -Nearest Neighbours). The experiment which compares the approaches uses recall to measure the quality of results. Finally, we provide a recommendation of the modelling approach which produces the best accuracy in terms of tag prediction on the sample data.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction; Cross-validation; Classification and regression trees;**

KEYWORDS

Tag Prediction, Natural Language Processing, Multi-label Classification, k -Nearest Neighbour

ACM Reference Format:

Bojan Božić, André Ríos, and Sarah Jane Delany. 2018. Validation of Tagging Suggestion Models for a Hotel Ticketing Corpus. In *20th International Conference on Information Integration and Web-based Applications & Services (iiWAS '18)*, November 19–21, 2018, Yogyakarta, Indonesia. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3282373.3282386>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

iiWAS '18, November 19–21, 2018, Yogyakarta, Indonesia

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-6479-9/18/11...\$15.00

<https://doi.org/10.1145/3282373.3282386>

1 INTRODUCTION

The data which is analysed in this paper describes hotel staff inputs in a ticketing system¹. Currently the dataset consists of 2017 manually tagged entries with the following columns: `book_name`, `book_type`, `content`, `id`, `tags`, and `user_id`. The main concern is the prediction of tags based on the content. We have identified suitable models to solve the problem and compared them to each other.

A limitation of the work is due to the size of the data sample, which is relatively small with 2017 observations. This makes prediction more difficult as most approaches need a larger number of examples of correctly tagged text from which to learn. Therefore, predictions for tags that are not frequently used are harder to make.

The aim of the project is to improve the tagging process of the current system. At this point users are manually tagging their entries with hash tags, however user tagging is not consistent (i.e. some users use too many tags, some none at all). Therefore, a future implementation of the tagging service should either automatically tag text entries, with an option to manually correct later (e.g. by an editor) or suggest most likely tags to the user who could select appropriate tags from a set of recommended ones.

This paper addresses two main objectives (i) the exploration of existing sample data, which includes statistical analysis and visualisation of the data to provide an understanding of the data, and (ii) the evaluation of tag prediction approaches. The output is a recommendation of the modelling approach which produces the best accuracy in terms of tag prediction on the sample data.

Figure 1 shows the distribution of tags in our dataset through a heatmap. We can see which tags have been used on the left side of the figure (e.g. `ITproblem`, `TvNotWorking`, `VIP`, etc.) and on the bottom, the number the tags for each text entry (from 1 to 6). Each cell represents how many times a tag is used in each of the 6 cases. For example, "maintenance" in total is used 498 times (entries) of which 204 times is alone (one tag), 215 times appears along with another tag (two tags) and so on. "Electrical", instead, never appears alone, always along with another tag.

The rest of the paper is structured as follows: Section 2 describes related work and the state-of-the-art in tag prediction. Section 3 provides more details on the approach taken, the research method and performance measure used. We describe our process of data exploration in section 4 and how the different approaches work in section 5. Finally, we present the results in section 6 and our conclusions and future work are presented in section 8.

¹<http://www.dmbook.pro/en/demo>

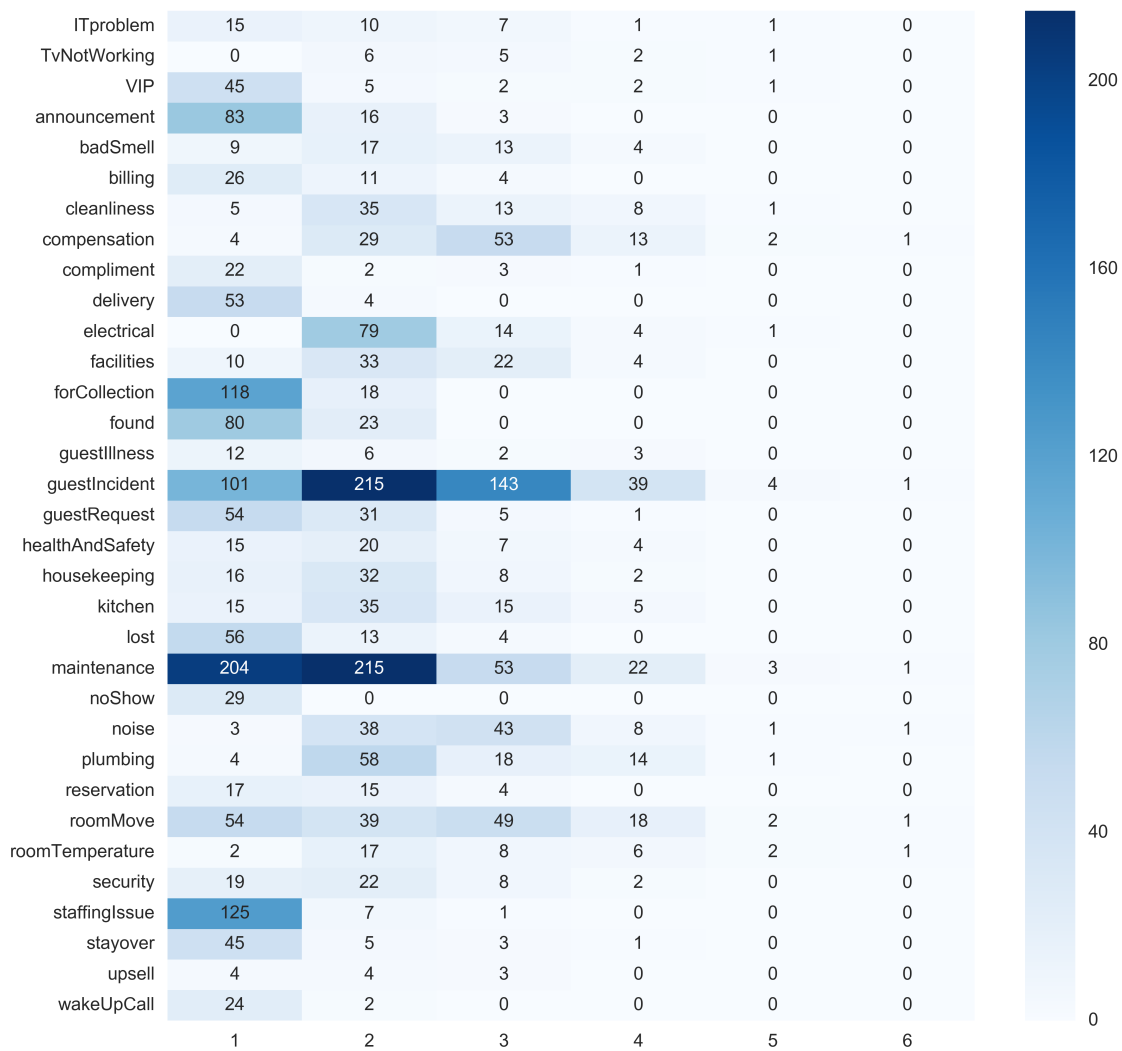


Figure 1: A heatmap showing an overview of used tags and frequencies.

2 RELATED WORK

A tag is a free-text annotation (word, name or phrase) that can be associated with textual data. Tags generally identify or categorise the textual content and can be used to facilitate searching and retrieving information along with revealing groupings and semantic links across information. Tagging is most familiar to us as user generated labels in social media but tags can be applied to any collection of text data. They can be extracted from the text automatically reflecting the semantics of the text or can be recommended from groupings of existing domain specific tags. There are no limits to the number of tags that can be assigned. Tagging gained popularity in Web 2.0 due to the growth of social networking and image sharing websites [20] but has numerous use-cases in a wide range of information management tasks. Identifying and automatically assigning tags in large scale text corpora can provide a means for fast archiving, searching and retrieving information along with

revealing topics and semantic links. In addition tagging can assist in the creation of ontologies. Examples of this include work by [7] which learns a domain ontology from user tagging, building on using WordNet as a backbone ontology and work by [11] which builds ontological structures by using low support association rule mining.

Collaborative tagging is the process of allowing anyone, particularly consumers, to tag content [2]. The process of collaborative tagging can be described as a folksonomy by many researchers as are the social media sites which support collaborative tagging, but the collection of tags resulting from collaborative tagging is a classification system which is more correctly defined as a folksonomy [24]. The challenges of collaborative tagging are identified by many researchers and are listed by [13] as polysemy, where words can have two similar meaning; synonymy, different words with similar or identical meanings; plurals effectively different words with identical meanings; and specificity of tagging, where related

tags that describe an item vary along a scale from very general to very specific.

To automate and improve the user-generated tag assignment process, researchers have developed tag recommender systems for suggesting reliable and relevant tag options. Overviews of methods and approaches to tag recommender systems for social web are available from [5, 6].

The approaches to automatic tagging can be classified according to the source of the tags. Tags can be (i) generated, directly extracted from the text itself or (ii) re-used, identified from data that is already tagged and/or from other external sources. Extracting tags from text uses natural language processing techniques to identify keywords and phrases in the textual content that can be used as tags [9]. Context information associated with the text item can also provide information that can be used to tag including metadata or associated information such as ratings. Research by [1] shows the benefits of structured tagging (providing tags to predefined context or metadata elements) in addition to free-text tags. The extraction of tags from user profile information has been used to solve the 'cold start' problem [14] in tag recommender systems.

Techniques using machine learning approaches have been used also to identify tags from both the source text data using topic modelling approaches [3, 10] and data that is already tagged using tag clustering [4, 19], supervised learning algorithms [22] and graph partitioning algorithms [21].

A variety of approaches for using external sources exist to help in recommending relevant tags for text. Ontology-based tagging involves recommending tags directly from a collaborative user-driven ontology [15]. [23] construct a topic ontology through enriching an existing set of categories automatically from external knowledge resources such as Wikipedia and WordNet. [3] proposes a probabilistic topic model that incorporates DBpedia knowledge.

3 APPROACH AND RESEARCH METHOD

The approach taken in our work on this project was to identify several appropriate methods to predict tags for the text entries and to compare them. In order to address both objectives of **exploration** and **evaluation**, we have preprocessed and cleaned the data in advance of performing the analysis. We used a Bag of Words representation for the text, tokenising on white space and punctuation. To make sure only meaningful words were used as features, stopwords (such as in, and, of, a, the, etc.) were removed and the remaining words were lemmatised.

3.1 Exploration of Sample Data

In the data **exploration** phase, we considered the structure of the sample dataset and identified notable statistics and facts derived from the dataset. This was done by analysing books, tags, and the text content in plotted graphs by quick data visualisation, aggregation, and statistical analysis.

3.2 Evaluation of Tag Prediction Approaches

The second part was the **evaluation** phase, where we identified a set of candidate algorithms to predict tags for the input text. We identified four approaches that have the potential to provide good results according to published research papers and best-practice

definitions. The candidate methods were: a supervised machine learning model for multi-label classification (using gradient boosting), a statistical approach (using frequency heuristics), and two simple similarity-based classification approaches (Nearest Centroid and k -Nearest Neighbour) (detailed descriptions of each approach can be found in section 5). In the final step, we evaluated these methods in order to find the most suitable one for the underlying task. To facilitate evaluation, we used a measure of recall to measure and compare the results. The best performing approach was the basis for our recommendation.

3.2.1 Method. The methodology we used for the evaluation of the approaches is k -fold cross validation, a standard in machine learning, which involves splitting the dataset into training and testing subsets. We used 10-fold cross-validation, partitioning the dataset in 10 parts and iteratively using 9 folds as the training set and the final fold as the test set. The training dataset is used to build the prediction model while the testing dataset is used to test the model. Performance is measured by comparing the predicted tags with the actual tags. The final result is the average over the 10 iterations.

Test Approach. Our evaluation approach included using 10-fold cross-validation to partition training sets and test sets and use average values of the 10 evaluation steps; training the algorithm on the training set (for those approaches which required training); running tests with the predefined test set; comparing the results and measuring recall (as defined below). It is important to use different training and test datasets to ensure that the data does not learn from the examples that are used for testing.

Recall. The performance measure used was recall. For our experiments we defined recall as the ratio of correctly suggested tags over the number of actual tags for a text entry - $\frac{ST}{OT}$ where ST are suggested tags which are correct and OT are original tags in the test set. For example, if a text entry has been tagged with ["maintenance", "noise", "plumbing"] and the system suggests ["maintenance", "plumbing", "cleanliness"], recall would be $\frac{2}{3}$, because 3 tags were in the original dataset of which we got 2 right in our suggested tags. Recall is an appropriate measure for these experiments as it is the most straightforward way of automatically identifying correct tags by comparing with actual tags. It will not allow us to measure situations where other relevant and appropriate tags are suggested but are not in the set of actual tags. To do this would require manual user confirmation of all predicted tags.

4 DATA EXPLORATION

The first step was to investigate the structure and characteristics of the existing dataset. Table 1 shows a sample of data provided by the customer with the following structure:

- **book_name:** Name of the hotel book (main category, i.e. every hotel has one or more books in which they write their tickets or messages, e.g. reception, maintenance, night, log, etc.). In smaller hotels there might be only one book. Synonyms or similar names may be used for the same book (e.g. reception = front desk = front office).

- *book_type*: A simple category for a book. Most probably redundant as input since it may correlate with the book name.
- *content*: The free text entry input by hotel staff. This is the main input for tag prediction.
- *id*: Unique ID of the ticketing system entry.
- *tags*: The field that needs to be predicted. There is no fixed number of tags and there may any number of tags per entry (the provided dataset has between 1 and 6 tags per entry).
- *user_id*: ID of user who created the entry which may also correlate with the book name.

There are three inputs in the dataset that we will use to predict tags: *book_name*, *book_type*, and *content*. The target is the tags field.

The frequency of occurrence of tags is shown in Figure 2. There are currently 33 different tags used in the dataset of 2017 entries. The system should be open and allow new tags, however, due to experience from the customer on their user behaviour and the usage of tags, they expect, over time, tags to stabilise at approximately 40 tags at most. Furthermore, tags could be fixed and set to a given list, if necessary.

Figure 2 also shows that there are two predominant tags (*guestIncident* and *maintenance*) which make up almost a third (33%) of all tags. This highlights the fact that there are some very general tags (such as both mentioned ones) and also some very specific tags like *TvNotWorking* which is only used in 0.46% of the cases. However, if the situation arose in the future where a significantly large number of tags were being used, then it might be useful to create a taxonomy of tags. This would allow comments tagged at lower levels of the tagging hierarchy to be grouped by using tags at higher levels.

An overview of the number of tags per entry is shown in Figure 3. Most entries have only 1 tag but very few have more than 3 tags. This graph shows that most entries seem to be covered sufficiently by 3 tags or less.

In sessions with the customer we identified that recommending additional tags which were not already part of data set was found useful by a human user in certain cases. This would require different approaches (not covered in this paper) which would involve identifying/extracting keywords and phrases from the actual text content. Once used the training set would need to be updated with these tags in order to make them available as recommendations to other users.

Figure 4 shows another view of the relationship between tags and the number of tags per text entry. For each tag it shows the percentage of those entries containing the tag that have that tag alone (represented in blue), or as one of two (green), three (red), etc tags. This is interesting for finding out which tags are mostly used alone and which ones are frequently used in combination with others. Examples for tags that are mostly used alone are *wakeUpCall*, *staffingIssue*, *noShow*, and *delivery*. Tags that are frequently used in combination with other tags are e.g. *plumbing*, *noise*, *roomTemperature*, *electrical*, and *TvNotWorking*. This shows that entries with only one tag tend to be tagged very generally, while very specific tags are accompanied by general tags and indicate that the tagging process has been more precise in these cases.

5 TAG PREDICTION APPROACHES

This section describes four approaches that have been tested and validated for their capacity to predict the actual tags that are associated with the input text. We have accounted for using different approaches from different research fields in order to cover a broad spectrum of possible solutions. As a result, we have tested a supervised machine learning model for multi-label classification), a statistical approach using frequency heuristics, and two simple classification and similarity approaches (Nearest Centroid and k -Nearest Neighbour). The following subsections each describe one of the tested models in more detail.

5.1 Multi-Label Classification

The first method uses a supervised machine learning approach, which is described in more detail in [16]. It uses Gradient Boosting classification² on the training data to build a prediction model/classifier. The strategy uses an ensemble approach building a prediction model or classifier per tag and combines the outputs of all the ensemble member classifiers to decide which tags are appropriate for the test input text. The ensemble approach is a commonly used strategy for multiclass classification.

5.2 Frequency Heuristics

The statistical approach used is based on the frequency of co-occurrence of tags with words in the text entries. This method goes through the training dataset and generates a map of probabilities. It associates a list of tags with each word used in the text entry and calculates the probability of using that tag with that word. As an example, if the word *room* appears three times in texts tagged with *noShow* and ten times in total, the probability of *room* in the list of words for tag *noShow* is 0.3). For each test entry, the tags suggested are those tags associated with the words in the test entry with the highest probability. The top three tags are suggested.

5.3 Nearest Centroid

This method is based on the Rocchio method of classification, which is described in detail in [17]. It creates tag clusters or groupings of all entries in the training set that are tagged with a specific tag and calculates the centroid (the centre point) of the cluster. For each test entry the similarity between it and the centroid of each cluster is measured using cosine similarity. The tags for the top three most similar clusters are returned as the suggested tags.

5.4 k -Nearest Neighbours

k -Nearest Neighbours (k -NN) is a simple algorithm that finds a number (k) of training entries most similar to the test entry and determines what tags to suggest from the tags used by these k nearest neighbours. The value of k can be determined by testing the performance of different k s. The tags to be suggested are those that occur with highest frequency within the k nearest neighbours. We tested for values of k from $k = 1$ to $k = 10$. For $k = 1$ we returned all tags found on the nearest neighbour and for larger values of k we returned the three most frequent tags found on all neighbours.

²<http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>

	book_name	book_type	content	id	tags	user_id
0	DM Log	log	Erna from Pandora called in sick for this ev...	99324	[staffingIssue]	998
1	DM Log	log	928 and 481 are VIP's.	92996	[VIP]	998
2	DM Log	log	had to move Mr Peters room 935 to 838 he is wit...	97855	[roomMove]	998
3	Nights	log	Guest in room 571 no wake up call 2 nd time in...	97507	[guestIncident, wakeUpCall]	998
4	Maintenance	log	!Urgent! - Please replace outside balcony door...	97411	[maintenance]	998
5	10.30am Briefing Notes	log	Welcome Inspection tomorrow, all staff in full ...	97284	[internal]	998

Table 1: Data sample.

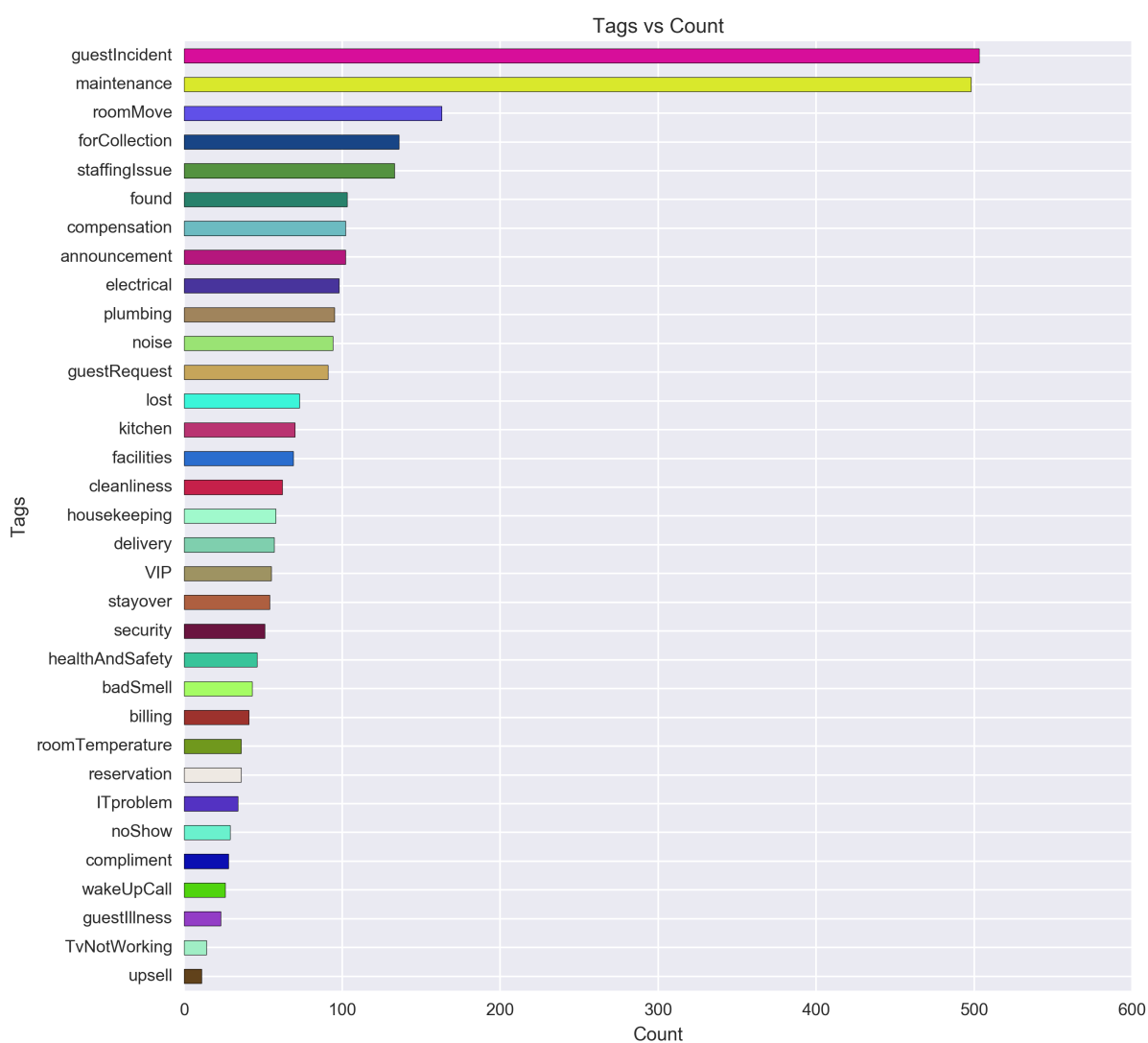


Figure 2: All tags used in the data set based on their frequency.

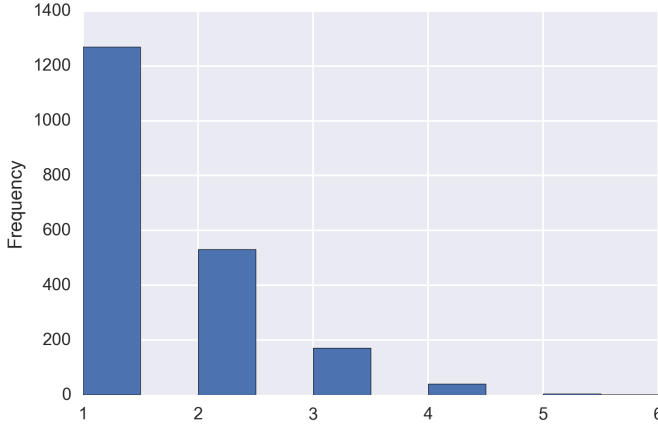


Figure 3: Number of tags per entry.

Method	Recall		k
	with book names	without book names	
Multi-label Classification	0.735	0.707	-
Frequency Heuristics	-	0.709	-
Nearest Centroid	0.764	0.772	-
k-Nearest Neighbour	0.986	0.979	1
k-Nearest Neighbour	0.987	0.981	2
k-Nearest Neighbour	0.990	0.983	3
k-Nearest Neighbour	0.991	0.986	4
k-Nearest Neighbour	0.992	0.990	5
k-Nearest Neighbour	0.994	0.991	6
k-Nearest Neighbour	0.995	0.992	7
k-Nearest Neighbour	0.997	0.993	8
k-Nearest Neighbour	0.997	0.993	9
k-Nearest Neighbour	0.997	0.993	10

Table 2: 10-fold cross-validated recall for all methods with and without book names, and different sizes of k for k -NN.

The sklearn implementation and description of different flavours and use cases for k -NN can be found in [18].

6 EXPERIMENT RESULTS

We calculated the performance of each method against the test dataset by measuring recall as defined in section 3. Table 2 shows recall for each method with and without book names as an additional input (except for Frequency Heuristics which does not provide the option to add additional inputs).

The table shows that k -NN clearly provides the best recall. Since k -NN turned out to provide the best results, we have performed further experiments with values of k from 1 to 10. This helped us to observe the impact of the number of neighbours used on the performance and to identify the best value for k .

While the size of k used does not seem to have a big impact on overall performance, results suggest that at $k = 8$ performance peaks and raising the value above 8 does not have a beneficial

impact. However, we would need a significantly larger dataset to validate which size of k might be preferred. Overall, our experiment measuring the recall leads us to the conclusion that k -NN is a good approach for predicting tags on this dataset and the value of k should be confirmed in experiments with larger datasets.

6.1 Book names

A comparison of performance with and without book names as input indicates that including book names in the input text potentially improves results; significantly for multi-label classification, slightly for k -NN, but has a rather negative impact on Nearest Centroid. While adding additional inputs may have a slight impact on training time, the impact of adding an additional word to the input should be negligible. The results show that performance generally tends to improve when including book names in the input data.

6.2 Runtime

Another comparison took into regard the runtime of the various methods. We have used an Intel®Core™i7-7600U CPU @ 2.80GHz processor configuration with 16GB RAM for the tests. We organised the tests in two separate categories: one for all methods which do not require training and detect tags on the fly and one for those which need to run training first so we have included training time and runtime for tag detection.

Figure 5 shows the runtime of methods which require no training and are hence much faster. Here we can clearly see that the method for frequencies with heuristics performs best and that k -NN runtime splits in two groups. The k -NN group with booknames as additional feature has better runtime than the group of k -NN without booknames. Also we can see that the value of k does not have much influence on the runtime in our scope.

In comparison, Figure 6 shows methods with training times. The graph shows that the multi-label classifier is much faster than nearest centroid and does not make much difference whether booknames are used as an additional feature or not. In the case of nearest centroid, the version with booknames is clearly faster than the one without, but both are outperformed by the multi-label classifier.

Our conclusion is that the methods with the best runtime are frequencies with heuristics for methods without training and multi-label classifier for methods with training. However, since frequencies with heuristics provides the worst results (see Section 6 EXPERIMENT RESULTS) and the performance of k -NN is not significantly worse, we recommend the use of k -NN for methods without training. For methods with training, we recommend the multi-label classifier, since it has only slightly worse results, but much better performance than nearest centroid. In both cases though, we recommend the use of booknames as additional feature as they do not negatively influence runtime (in some of our cases even on the contrary) and provide better results in tag detection.

7 FURTHER CONSIDERATIONS

The experiments performed on the sample dataset explored the feasibility of the automatic prediction of tags for the text entry test cases and results of the experiments have indicated that performance is good for certain approaches. It is important to note that

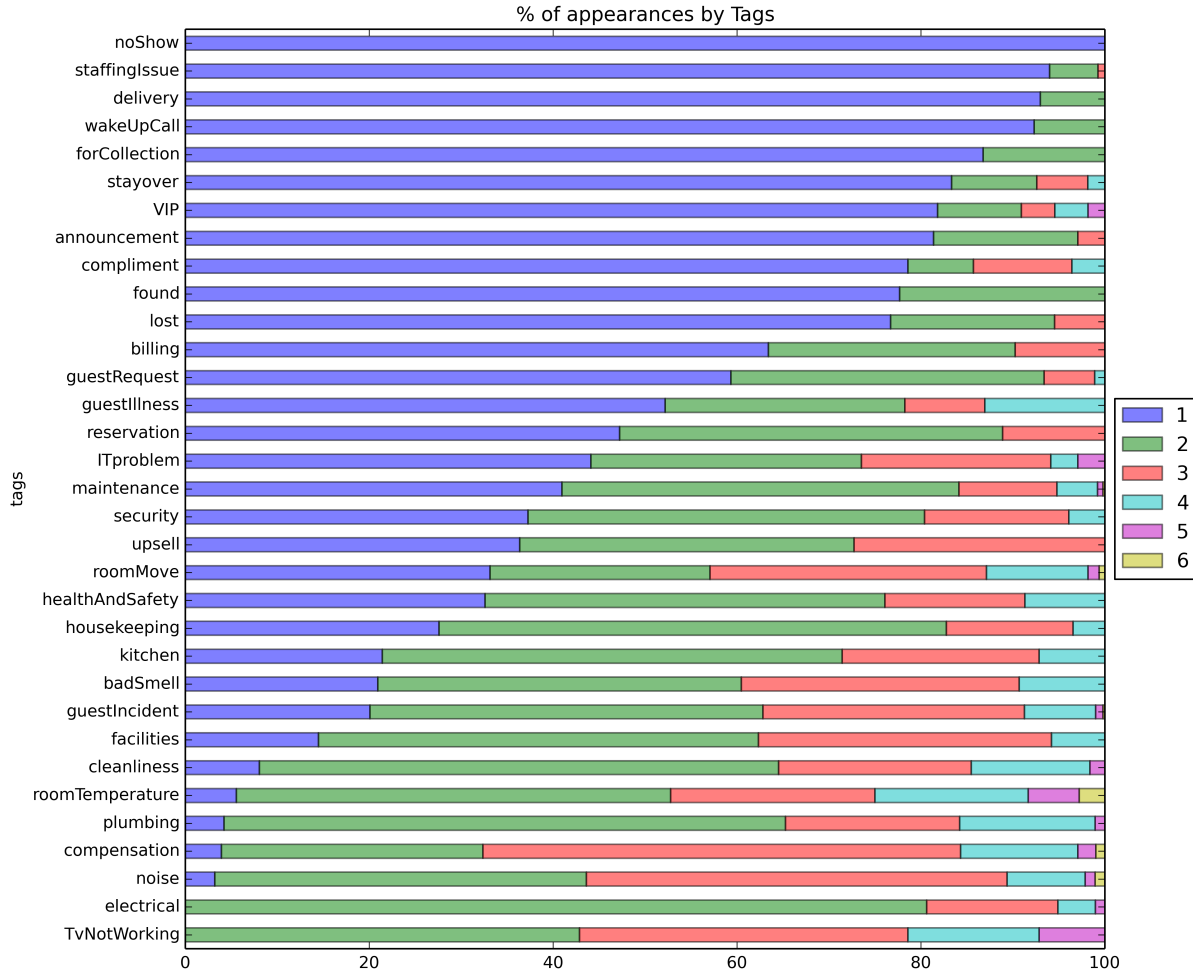


Figure 4: Proportion map of all entries and the number of total tags they appear in.

to implement any of these approaches in a solution in a real-world environment there are additional factors that must be considered.

Each of the approaches evaluated above require training data to learn from, and most require an offline training process to produce a prediction model that is used to predict new book entries in real-time. The model captures/learns the current ‘concept’ from the training data. The assumption is that the training data used to build the model is representative of all types of book entries and tags used. Over time when new examples of entries come about or new tags are used, the model will not know about these and will suffer from what is known as *concept drift* [8]. All prediction scenarios need to consider how to manage concept drift and keep the prediction model up to date with all representative examples of the problem it has been built to tackle.

Generally handling concept drift requires periodic re-training on an updated training set that incorporates the new examples and tags not previously seen. The resulting new model is then swapped into the real-time process to be used for prediction. Three of the four techniques explored fall under this scenario. *k*-NN, however,

is different; it does not require an offline training process to build a model which is then used for prediction. It uses the training examples at prediction time and it determines the most appropriate examples to learn from at the time of prediction (through identifying the most similar neighbours). Managing concept drift simply means adding new examples to the training set as they occur. With *k*-NN there is a trade-off between ease of managing concept drift and prediction time. Finding the most similar examples (the nearest neighbours) for a new example involves comparing with each training example. As the training set grows with the new training data, the time taken to predict can increase. This limitation means that the training data needs to be periodically edited and pruned to produce a smaller, leaner set to work with. There are a number of editing algorithms available that facilitate this, e.g. [12].

8 CONCLUSION AND FUTURE WORK

The work in this paper involved investigating a specific application or use-case of tag prediction from text which involved identifying how to accurately predict tags for a particular dataset in order to

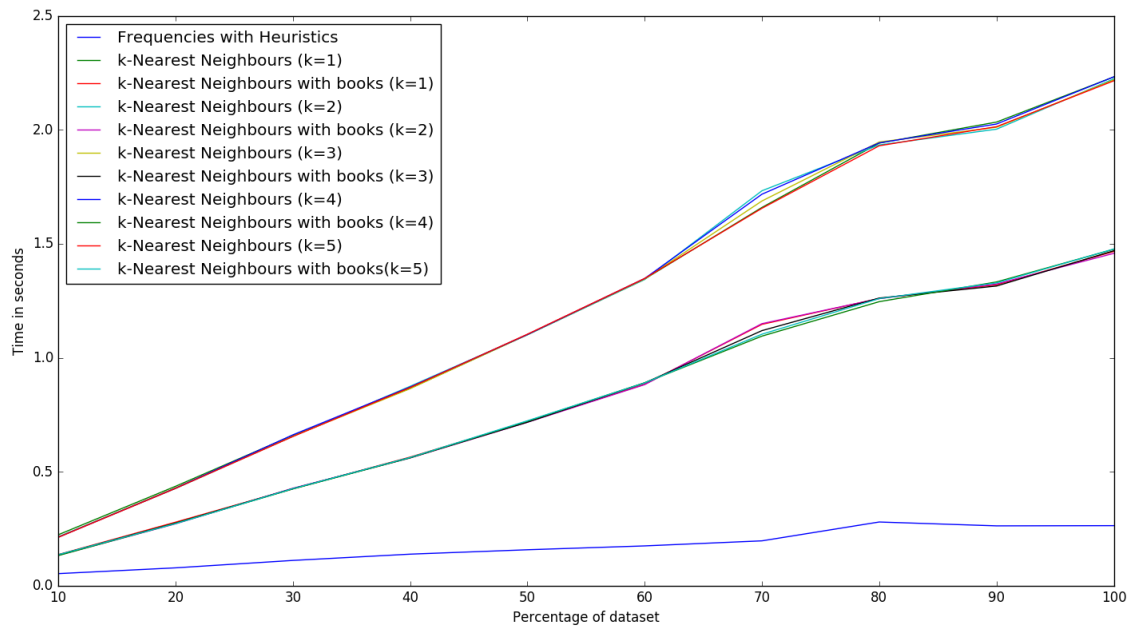


Figure 5: Runtime of methods which require no training.

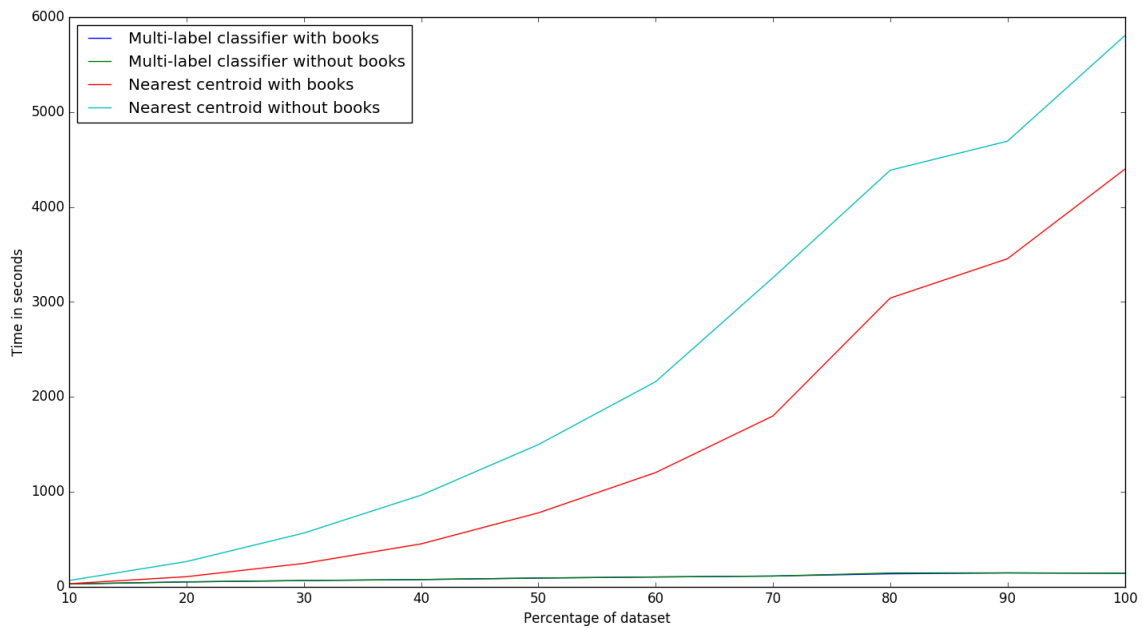


Figure 6: Runtime of methods which require training.

assist a company to use their data more effectively. The research involved the exploration of the dataset to understand the data and experimentation on appropriate techniques to identify the best performing approach for the given dataset. Our recommendation, based on exploration of the given dataset, is to use the k -NN approach to predict tags using both the entry content and the book names as input to the process. While k -NN shows the best performance in terms of recall, it also provides some benefits in terms of managing concept drift. There will be a need, however, to periodically edit the training set to maintain acceptable prediction times. Further experimentation would be required to determine how and when this would be required.

The next steps of the project will have two main goals:

- (1) Testing of the recommended approach: This involves performing extended test scenarios for feature extraction and training to find the most appropriate parameters for the model, as well as measuring performance on large datasets and testing scalability of the model. It will also involve identifying how to handle concept drift over time.
- (2) Implementation of a prototype solution: In the next step, we will implement a working prototype providing a usable service for the company that can be tested and evaluated on real-time data.

ACKNOWLEDGMENTS

We would like to thank DmBook Pro³ for providing manually tagged sample data which our research was based.

REFERENCES

- [1] 2006. Structured vs. unstructured tagging – A case study. *International Journal* Edinburgh (2006). Archived proceedings of the WWW Tagging Workshops.
- [2] Scott A. Golder and Bernardo Huberman. 2005. The Structure of Collaborative Tagging Systems. 32 (09 2005).
- [3] M. Allahyari and K. Kochut. 2016. Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*. 63–70. <https://doi.org/10.1109/ICSC.2016.34>
- [4] Grigory Begelman, Philipp Keller, and Frank Smadja. 2006. Automated Tag Clustering: Improving search and exploration in the tag space. In *Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006*. Edinburgh, Scotland. http://pui.ch/phred/automated_tag_clustering/
- [5] Fabiano M. Belém, Jussara M. Almeida, and Marcos A. Gonçalves. 2017. A Survey on Tag Recommendation Methods. *J. Assoc. Inf. Sci. Technol.* 68, 4 (April 2017), 830–844. <https://doi.org/10.1002/asi.23736>
- [6] Mattia Campana and Franca Delmastro. 2017. Recommender Systems for Online and Mobile Social Networks: A survey. *Volumes 34–4* (10 2017), 75–97.
- [7] Endang Djuana, Yue Xu, and Yuefeng Li. 2011. Learning Domain Ontology for Tag Recommendation. *Science And Technology* (2011). <http://eprints.qut.edu.au/45499/>
- [8] João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Comput. Surv.* 46, 4 (2014), 44:1–44:37. <https://doi.org/10.1145/2523813>
- [9] Kazi Saidul Hasan and Vincent Ng. 2014. Automatic Keyphrase Extraction: A Survey of the State of the Art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1262–1273. <http://www.aclweb.org/anthology/P14-1119>
- [10] Ralf Krestel and Peter Fankhauser. 2009. Tag recommendation using probabilistic topic models. *ECML PKDD Discovery Challenge 2009* (2009), 131.
- [11] Huai ren Lin, Joseph Davis, and Ying Zhou. 2009. An Integrated Approach to Extracting Ontological Structures from Folksonomies. In *The Semantic Web: Research and Applications*, Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 654–668.
- [12] Stewart Massie, Susan Craw, and Nirmalie Wiratunga. 2006. Complexity Profiling for Informed Case-Base Editing. In *Advances in Case-Based Reasoning*, Thomas R. Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 325–339.
- [13] Alireza Noruzi. 2006. Folksonomies: (Un)Controlled Vocabulary? 33 (01 2006).
- [14] Christine Preisach, Leandro Balby Marinho, and Lars Schmidt-Thieme. 2010. Semi-supervised tag recommendation - Using untagged resources to mitigate cold-start problems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6118 LNAI, 1 (2010), 348–357.
- [15] Roman Prokofyev, Alexey Boyarsky, Oleg Ruchayskiy, Karl Aberer, Gianluca Demartini, and Philippe Cudré-Mauroux. 2012. Tag recommendation for large-scale ontology-based information systems. *Proceedings of the 11th international conference on The Semantic Web - Volume Part II* (2012), 325–336.
- [16] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine learning* 85, 3 (2011), 333.
- [17] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press. 292–297 pages.
- [18] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press. 297–301 pages.
- [19] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin D. Burke. 2008. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23–25, 2008*, Pearl Pu, Derek G. Bridge, Bamshad Mobasher, and Francesco Ricci (Eds.). ACM, 259–266.
- [20] Gene Smith. 2007. *Tagging: People-powered Metadata for the Social Web* (first ed.). New Riders Publishing, Thousand Oaks, CA, USA.
- [21] Yang Song, Lu Zhang, and C. Lee Giles. 2011. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web* 5, 1 (2011), 1–31.
- [22] Clayton Stanley and Michael Byrne. 2013. Predicting tags for stackoverflow posts. In *Proceedings of the International Conference on Cognitive Modelling, ICCM 2013*. 414–419.
- [23] V. Subramaniaswamy and S. Chentur Pandian. 2012. Effective Tag Recommendation System Based on Topic Ontology Using Wikipedia and WordNet. *Int. J. Intell. Syst.* 27, 12 (2012), 1034–1048. <https://doi.org/10.1002/int.21560>
- [24] J Trant. 2009. Studying Social Tagging and Folksonomy: A Review and Framework. *Journal of Digital Information* 10, 1 (2009).

³<http://www.dmbook.pro/en>