CEUR-WS.org/Vol-2535/paper_11.pdf

A simple and efficient approach for the semi-automated curation for media reviews

Marc Rössler and Florian Hilgenhöner

Unicepta GmbH, Salierring 47-53, 50677 Köln, Germany
Marc.Roessler@Unicepta.com

Abstract. In this paper we discuss a robust and efficient approach to automatically curate news articles into daily media monitoring reports by using document classification. We start with a motivation and a description of the task and work out the characteristics and requirements specific for this use case. Also, we report on initial experiments with simple Naïve Bayes classifiers trained on manually labelled data and discuss them in terms of applicability to the use case. Furthermore, we present the next steps to improve the automated curation without sacrificing efficiency and robustness.

Keywords: Machine Learning, Text Categorization, Media Review Curation, Multi-Label and Multi-Category Classification

1 Introduction

In the business of media monitoring, clienTts expect to be continuously provided with so-called media reviews for them to stay up to date with all topics relevant for their business. A media review is basically a report consisting of a set of curated media articles, sent out weekly, daily or even multiple times a day. Curation in that context includes both the enrichment of the articles with various meta data relevant for that use case and the organization of the content into rubrics or categories. In this paper, we solely focus on the organization of the articles and completely ignore the enrichment steps.

The rubrics to organize the articles are usually hierarchical trees, with a depth of one, two or rarely three levels. Examples of first level rubrics are “corporate news”, “news about corporate products”, “general news within the industry”, while second level rubrics can be e.g. a specific product.

Unicepta GmbH currently produces up to 500 media reviews on a daily basis. They are curated by human decisions in combination with a powerful, client specific filtering based on Boolean searches. The incoming data stream consists of up to 3 million hits per day that are filtered down based on search terms to roughly 50.000 documents. The set of filtered documents is the data pool that is used to populate the approximately 7.000 rubrics used to organize the set of media reviews.

This setup obviously has a potential to be more efficient by delegating some of the human decisions to a Machine-Learning (ML) based algorithm. As the resulting media

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

review involves further human qualification and enrichment (e.g. requesting topic-oriented abstracts for certain articles and topics), the resulting algorithm is not replacing the human decision but aims at increasing the efficiency of the production process.

2 Characteristics of the task, setup and requirements

The task of semi-automated curation of media articles for media review has a set of interesting characteristics:

- It is a multilabel task and at least in theory also a hierarchical task. However, we decided to flatten out all hierarchical aspects. Furthermore, new categories are added almost daily, due to new clients and new setups.
- The data is heavily imbalanced: Certain topics only have a handful of entries even over the course of a year while others have thousands of relevant hits. Also, the next two characteristics add to the imbalance.
- The data is multilingual and contains both European and Asian content while German and English content is dominating.
- The data is prefiltered by search terms which are sophisticated and fine-tuned for some categories while they are unspecific or even flawed for other categories. This is a result of the frequent changes of the search terms, sometimes conducted by users with limited experience with Boolean search syntax.
- News data can be suspected to have a strong topic drift, meaning that the topic of tomorrows articles not necessarily occurred in the past. Furthermore, clients tend to update their briefing on the focus of certain categories from time to time.
- While for certain media reviews, all articles identified by the search-based filter needs to be categorized, other media review or selected categories only are populated by the top-n articles. Furthermore, duplicates and near duplicates of some articles ought to be included or excluded, dependent on the client setup.

The requirements for a system to semi-automatically curate media reviews are as follows: It needs to be robust against the various imbalances of the data, efficient for retraining or online trainable to cope with the topic shift and new or updated categories and furthermore, it needs to run completely self-configured as the team on the ground will not be able to fine tune or configure the system as this would likely eat up all potential efficiencies. On the other side, the system is only supporting human decisions. This means, an incorrect classification simply does not support getting more efficient but does not create an erroneous behavior of the system. Hence, we plan to start with a rather weak but robust approach and to continuously optimize it over time.

3 Related work

Assigning documents into a set of predefined categories is a long-known problem in ML with many successful approaches to solve it. Most approaches are usually based on supervised ML i.e. they require a set of annotated data to train a classifier in order to predict the categories of a document.

Among the ML approaches used are Naïve Bayes, Logistics Regression, Decision Trees, SVMs and most recently also Neural Networks and/or word embeddings created with Neural Networks.

Linear SVMs have a prominent place as they showed [1] and still show superior performance with a reasonable computational effort on this task for a long period of time.

The current paradigm of pre-trained models, methods like BERT [2] and XL-Net [3] outperform or achieve the state of the art in a variety of tasks including question answering, named entity recognition, and natural language inference. Applying this paradigm to text categorization is very interesting [4] despite the computational costs that are significantly higher than any other approach.

Four editions of a challenge on large-scale text classification have been conducted from 2010-2014 [5]. The challenge named LSHTC aimed at assessing the performance of classification systems in large-scale multi-label and hierarchical classification of a large number of categories.

Our task also shares certain characteristics of "extreme multi-label classification" (XMC - see e.g. [6]), though on a smaller scale. XMC refers to the task learn a classifier which can assign a small subset of relevant labels to an instance from a very large set of target labels. An important statistical characteristic of the datasets in XMC is that a large fraction of labels are tail labels, i.e., those which have very few training instances that belong to them.

4 Our approach

The input to process consists of news articles, both online and print content that was OCRed. This input is preprocessed and transformed into a feature vector in the following way. All content (headline, sub headline and content) is combined into a single string that is used for language detection. After removing markup, reducing it to letters and digits, case folding and Umlaut normalization, it is tokenized. All stop words are filtered out and stemming is applied on the resulting tokens.

To better account for duplicates and near duplicates within the training data, articles with identical or very similar content are grouped. The similarity is computed based on keywords extraction as described in [7]. For each group only one article is kept and all labels from the group are assigned to it.

The feature extraction is based on the 5000 most frequent words per language and the TF*IDF is computed as weight per token. Additional features, especially metadata of the articles are currently not reflected in the features set.

We apply a multinomial naive Bayesian classifier in a binary relevance setup and train one classifier per language and label. The training involves random subspace sampling, i.e. multiple classifiers are trained on subsets of documents and features and ADA boost is used to further improve performance. This approach is chosen for its low computational costs, compared to SVMs or even neural networks.

As evaluation metrics, we decided to use the average precision which basically represents the Area Under the Curve in a recall/precision diagram. This metrics is not indicative for setups where only the top-n documents ought to be selected as it takes into account all predictions of the classifier and not just the top n.

As overall metrics, we combine the average precision of all classifiers, weighted by the number of predictions per classifier.

5 Experiments

For our experiments, we have focused on one prototypical media review. We used approximately nine months of data which corresponds to 14.000articles assigned to 31 categories. We did not exclude any category, even though some only had a handful of positive training instances. We hold out 20% of the data as test data for our experiments. For the training of the binary classifiers, all articles assigned to another category was marked as negative instance.

In our experiments, we found that the training time of a Naïve Bayes setup, even with many classes and many more features is still fast and we expect this to enable us to retrain all classifiers multiple times a day on cloud hardware at very reasonable costs.

Table 1. Average precision over all predictions by all classifiers, upper and lower bound (best and worst classifier)

	Average Precision	Upper bound	Lower bound
Naïve Bayes	63.1%	87.4%	0%

We also looked at the impact of the amount of training data and run a set of experiments where we compared the performance starting with 2000 documents up to 14000 documents in steps of additional 2000 documents.

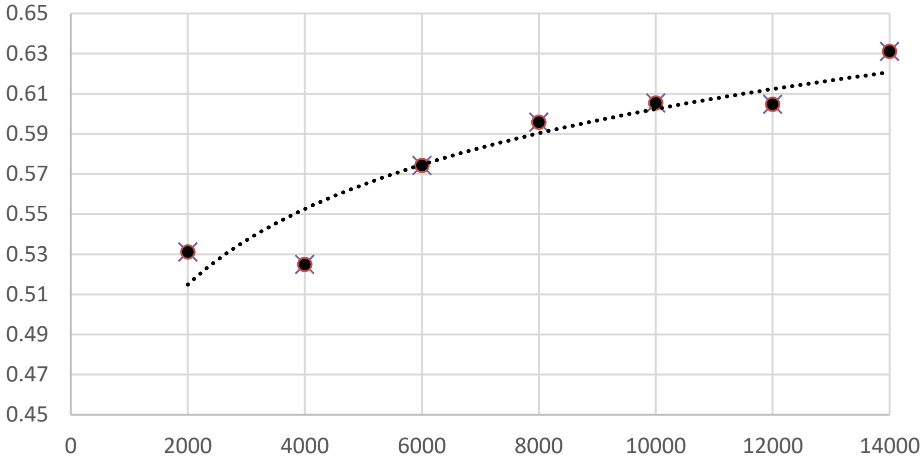

Fig. 1. Average precision over 7 runs with increasing amounts of training data, starting with 2000 and a maximum 14000 training examples. 20% of the data is hold out for testing.

6 Discussion and Outlook

We have shown that a simple and efficiently trainable approach to text categorization brings up reasonable performance that seems at least good enough to support human decisions for curating media reviews. Also, we can see the expected behavior in Fig. 1, that more data leads to better results which comes with the consequence, that new or weak populated rubrics always will suffer from poor performance in terms of classification accuracy. We also studied the results of the individual classifiers to understand the difference in performance. However, besides the observation that more training data leads to better results, we did not identify an obvious pattern that explains the difference in performance.

The results are an encouraging starting point that offers many ways to significantly improve the performance and to increase the efficiency of the production process. When it comes to the feature engineering, it seems attractive to integrate word embeddings as in BERT [2] or Word2Vec[8]. They can be used to extend the feature vector and should especially support classes with very few training instances. Also, we will compare Naïve Bayes with Logistic Regression as the computational costs are comparable. To further address the imbalance of the classes, sampling methods will be evaluated further. Finally, we are also keen to get feedback from our internal production teams. This will help us to understand the variance in performance for the different categorization tasks. In addition, carefully observing the way the teams work will likely bring up information on additional useful features such as phrases, named entities and other meta data and will also help us to better understand how to best integrate automated curation into the production process.

References

1. Joachims T.: Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg. (1998)
2. Devlin J, Chang M., Lee K, and Toutanova K.: BERT: pre-training of deep bidirectional transformers for language under-standing. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186. (2019)
3. Yang Z., Dai Z., Yang Y., Carbonell J.G., Salakhutdinov R., Quoc V. Le. S.: XLNet: generalized autoregressive pretraining for language understanding. arxiv/1906.08237. (2019)
4. Adhikari A., Ram A., Tang R., Lin J.: DocBERT: BERT for Document Class. https://arxiv.org/abs/1904.08398. (2019)
5. Partalas I., Kosmopoulos A., Baskiotis N., Artieres T., Paliouras G., Gaussier E., Androutsopoulos I., Amini M., Galinari P.: LSHTC: A Benchmark for Large-Scale Text Classification. https://arxiv.org/abs/1503.08581 (2015)
6. Babbar, R., Schölkopf, B.: Data scarcity, robustness and extreme multi-label classification. Machine Learning, Volume 108, Issue 8–9, pp 1329–1351. https://doi.org/10.1007/s10994-019-05791-5. (2019)
7. Matsuo Y., Ishizuka M..: Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. In: International Journal on Artificial Intelligence Tools. 13(1), pp 157–169. (2004)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. https://arxiv.org/abs/1301.3781. (2013)