Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)

Multi-Label Classification of Blurbs with SVM Classifier Chains

Franz Bellmann, Lea Bunzel, Christoph Demus, Lisa Fellendorf, Olivia Gräupner, Qiuyi Hu, Tamara Lange, Alica Stuhr, Jian Xi, Dirk Labudde, Michael Spranger
University of Applied Sciences Mittweida
Technikumplatz 17
09648 Mittweida
spranger@hs-mittweida.de

Abstract

Due to the increasing digitalisation multi-label classification gains in importance in many areas. In this paper we propose a method to classify blurbs into eight basic book genre using an ensemble of classifier chains composed of radial support vector machines using word embeddings and author information as features. Five models were tested using different implementations and features, as well as different numbers of chains. The best model reached a performance of a micro average F1 of 0.841.

1 Introduction

The increasing digitalisation often requires the integration of data and print media. In most cases, the first step is the classification of the datasets, or more specifically the documents, into a taxonomy, for example in order to assign these to different fields. For instance, before a bank approves a credit request, all the information given by the applicant has to be assigned to different categories to assess whether all required documents have been handed in or, when necessary, to inform the responsible official. Additional applications might be in forensics to classify evidential documents or in the library system. For the latter, the task is to sort the books into a thematic library taxonomy based on the short summary given on the back of a book's cover (blurbs). The difficulty, compared to a simple classification task, is the multinomially mapping of the books to the labels in a taxonomy, meaning each book can be assigned to more than one category which can belong to different taxonomy levels (Remus et al., 2019).

The GermEval 2019 shared task addresses this task for the German language with two subtasks, whereas the second one focuses on the different taxonomy levels. The data consists of blurbs of German books, which are provided by the publisher Random House. These blurbs and several meta information for instance the title and author have to be categorized into the most common writing genres and subgenres of German literature. For the first task these are only the taxonomy entries of the first level, namely: *Literatur & Unterhaltung*, *Ratgeber*, *Kinderbuch & Jugendbuch*, *Sachbuch*, *Ganzheitliches Bewusstsein*, *Glaube & Ethik*, *Künste*, and *Architektur & Garten*. In this paper an approach based on chained SVM models is presented and tested for the first task. The paper is organized as follows: First, some related work is presented in Section 2. Then an overview is given of the data and the methods in Sections 3 and 4. The results are presented in Section 5 before we conclude with Section 6.

2 Related Work

In the last two decades a lot of research has been conducted in the field of multi-label text classification whereas over time research has focused on several approaches. The most obvious approach is to adapt classifiers, such as kNN (Zhang and Zhou, 2005) or neural networks, to the multi-label task. Yet, typically, such classifiers have some shortcomings such as a high algorithmic complexity, which may lead to high computational costs. Another possibility is the transformation of the problem into several binary classification problems, also known as binary relevance approach. In this case two approaches exist. The one-versus-all approach trains one binary classifier per label and the all-versus-all one for each possible label combination. While the complexity of the one-versus-all approach grows linearly, the complexity of the all-versus-all approach increases approximately quadratically. However, the disadvantage of the one-versus-all approach is that it does not consider possible label correlations. One way to address this problem is to build a final classifier

with two stages. In the first stage each documents is classified using the binary classifier and then, in the second stage, the decisions are added to the input vector before the classifier is trained again, so that, in fact, the classifiers do not work independently anymore (Godbole and Sarawagi, 2004). Based on the aforementioned paper Read et al. (2009; 2011) developed classifier chains and subsequently ensembles of them which address the problem that different classifier orders result in different decisions.

Another important part is the representation of the documents. Usually, they are modelled as a term frequency vector (bag of words) in a Vector Space as described in Salton et al. (1975), yet, the resulting document vectors are high dimensional and sparse. Subsequently, vectors representing words and their relations in low dimensional space can be used as introduced in Mikolov et al. (2013a). Additionally, methods like Global Vectors (Pennington et al., 2014) and FastText (Joulin et al., 2017) were developed but with these approaches alone complete texts cannot be represented as low dimensional vectors. Addressing this problem Mikolov et al. (2013b) described that the addition of this word vectors produces meaningful results. Furthermore Yin and Jin (2015) hypothesize that the sum of word vectors of all words in a document results in a meaningful document vector. Although they apply the idea only to skip gram models Chilakapati (2018) generalized this to all word embeddings.

3 Data

The data used in this paper was provided by the organizers of the first task for the GermEval2019 and consisted of blurbs from 20,784 books from the publisher Random House. The models described in this paper were trained on the training set containing 14,548 blurbs and evaluated with the validation set (2079 blurbs). For the submitted model both, the training set as well as the validation data set, were used.

Each document consists of title, blurb, author, URL, ISBN, release date and associated labels. As can be seen in Table 1 the dataset is unbalanced with the category *Literatur & Unterhaltung* with the largest amount of books having 7817 books compared to 128 in the category *Architektur & Garten*.

Furthermore, some anomalies could be observed while exploratively analysing the data, which, however, only concerned about one percent of the data. For example, for some books no authors were available, while for others the blurbs were missing. In the second case, the data samples would have a null vector as a document vector and, consequently, would not have been assigned to a category. However, as additional author information was used as one feature some documents with a missing body were simply assigned to the author's genre. Books with no author information were handled in the same way as those books for which no information was available in the created author database (see Section 4.3).

Additionally, a few special books were found in the dataset. Their ISBN starts with a four, whereas usually the ISBN begins with the digit nine. After a short overall inspection concerning these objects, it was discovered that a few so called fan products were placed in the data set. The first idea was to delete them, but because of an existing body, it was decided to keep them. In fact, it turned out that most of them were assigned to the correct genre. Another problem was that the category *Ratgeber* has no equivalent category on `www.randomhouse.de`. This problem could not be solved. As a result the category *Ratgeber* was not taken into account for the additional feature based on the writing genre of the authors.

Genre	**# blurbs**
Literatur & Unterhaltung	7817
Sachbuch	2201
Kinderbuch & Jugendbuch	1987
Ratgeber	1862
Ganzheitliches Bewusstsein	803
Glaube & Ethik	598
Künste	146
Architektur & Garten	128

Table 1: Number of blurbs in each genre.

4 Methods

To solve the given classification task, an ensemble classifier chain as described by Read et al. (2009; 2011) was used. Global Vector as well as FastText representations of the texts combined with the information in which genre the respective author mostly publishes their work were considered as features. Due to the unbalanced nature of the

data the micro average F1-measure was used as the main evaluation criterion.

4.1 Encoding multi-class labels

For this task each possible category combination a book can belong to is encoded in one single number in order to handle these multi label categories easier. Hence, the category vector is considered as a bit pattern of the size eight. Each position in this pattern is related to one genre, as shown in Table 2.

Encoding	Genre
1	Literatur & Unterhaltung
2	Sachbuch
4	Kinderbuch & Jugendbuch
8	Ratgeber
16	Ganzheitliches Bewusstsein
32	Glaube & Ethik
64	Künste
128	Architektur & Garten

Table 2: Encoding for each genre.

Consequently, multi-label categories can be represented by adding up those bit values, so every combination has its own unique number. For example, if a document was classified as *Literatur & Unterhaltung* and *Glaube & Ethik* at the same time, the binary code would be 00100001 (in decimal: $1 + 32 = 33$). Therefore, its (multi label) category would be 33.

4.2 Preprocessing

As the data was provided by the organizers in an XML format, in a first step, it was converted into CSV data sets. Then, the data was filtered in order to get rid of the fine-grained categories. Furthermore, all columns except of *title*, *author* and *body* were removed. Afterwards, the blurbs contained in the body were tokenized into words and normalized. This included the conversion to lower case as well as lemmatizing and POS tagging using TreeTagger (Schmid, 1994). For further process steps only adjectives, nouns, verbs and adverbs were used.

4.3 Feature modelling

Vectorization

Next, the blurbs were vectorized, by converting each blurb into a low dimensional document vector. To do so, a pre-trained Global Vector (GloVe) with 300 dimensional word vectors based on a bag of words (BOW) containing a corpus with about 850,000 words of the German Wikipedia was downloaded from Pietsch et al. (2018). In order to create low dimensional and non-sparse document vectors the method described in Yin and Jin (2015) was used. The authors show that taking word vectors into account leads to more meaningful results which is basically the idea of word embeddings.

Subsequently, each document representation is simply the sum of these vectors weighted by the term frequency of the respective word, as show in Equation 1, were n is the number of words in the corpus, t_j is the frequency of each word w_i in the specific document, and v_j represents the word vector of each word w_i.

$$\vec{d_i^*} = \sum_{j=1}^{n} (t_j \cdot v_j) \tag{1}$$

For comparison, we built a new 230 dimensional FastText model (continues bag of words), which was trained on about 29,000 blurbs, consisting of the blurbs in the provided data and additional blurbs crawled from `www.randomhouse.de`.

At the end, the document vectors were standardized to the euclidean length 1.

Including author information

As mentioned above, information about the author was included as an additional feature. The necessary information was crawled from the publisher's website `www.randomhouse.de`. More precisely, information about 15,717 authors was crawled to find out how many books an author wrote in each of the eight categories. Then, this information was transformed into one vector a for each author containing eight elements $a_1, ..., a_8$, one for each category, each including the publishing frequency of the author in this category. As already discussed before, no information was available for the category *Ratgeber*. Further, each vector was normalized in order to get comparable results as well as weighted to achieve higher values if the author is very active as shown in Equation 2.

$$a_i' = \frac{a_i}{\sum_{j=1}^{n} a_j} \log(a_i + 1) \tag{2}$$

Finally, the author vector was appended to the document vector to include it in the model.

4.4 Classifier Chains

Classifier chains are first described in Read et al. (2009) and consist of several binary classifiers linked together. It is a method which turns a multi-label classification problem into n binary ones, whereas n is the number of possible categories. In this specific case, there is one binary classifier for each of the eight categories. The advantage of this method is that the classifiers do not work independently, but take into account the other classifiers' results. Thereby, correlations between the labels will have an influence on the final result. This is especially valid if there exist interdependencies between categories as is the case in highly branched taxonomies. Even though this is obviously not the case for the coarse classification task, it can not be completely excluded and it is definitely relevant for the fine-grained classification task (classification including subgenres). The models in this paper use these classifier chains as described in Read et al. (2009; 2011) and an overview is given in Figure 1.

The input for the classifier chain is a document's feature vector $\vec{f}$ containing the word features w_i from the document vector $\vec{d_i^*}$ and the author features $a_1, ..., a_8$. In a specified order each binary classifier predicts one category and forwards $[\vec{f}, c_i]$ as an extended document vector, whereas c_i is the predicted result of this classifier. The additional dimensions represent the multi-label classification of the document.

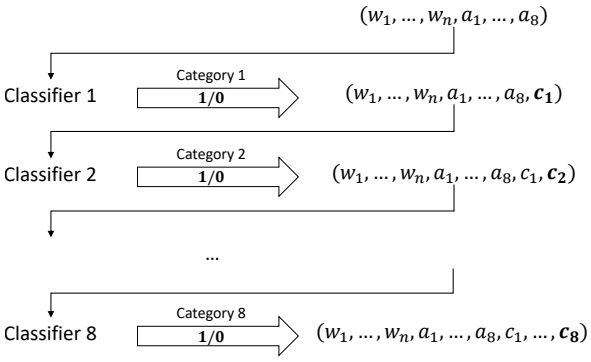

Figure 1: Structure of a classifier chain. The input for the chain is a document vector consisting of word features w_i and author features a_i. Each linked classifier adds its decision as feature c_i to that vector. The resulting vector serves as the input for the next classifier.

Basically, the order of the classifiers in the chain can be chosen arbitrarily or randomly. If there is an inherent order between categories then this order should be resembled in the order of the classifiers. As already described there is no such inherent order in the coarse classification task. Therefore, a random order was chosen. Any hidden dependencies can be considered by classifying repeatedly in different orders as described in Section 4.5.

The binary classifiers represent the core of a classifier chain. For the used models a Support Vector Machine with a one-versus-all technique was chosen because it works well with high dimensional vectors. The Support Vector Machine finds the optimal hyper-plane in the feature space in order to separate the categories and then classifies new vectors by mapping them into the feature space (Lee et al., 2011).

In this study, two different implementations of Support Vector Machines were tested. Both were used with an Radial Basis Function kernel which maps the vectors in a non linear way into the feature space. Therefore, it can handle non linear correlations between the features. Additionally, the classes were weighted because the dataset is very unbalanced (Hsu et al., 2016).

Both implementations differ in the way to train the Support Vector Machine. Using LibSVM (SVM-C) (Chang and Lin, 2001) requires the user to set the parameters C and γ manually. In this study the standard parameters $C = 1$ and $\gamma = 1/|\vec{f}|$ were chosen. In contrast, the caret package (version 6.0-84) for R implements a radial Support Vector Machine that tries to optimize its parameters using a given number (in this case 20) of randomly chosen parameter sets (Kuhn, 2019).

4.5 Ensemble Classifier Chains

The ensemble classifier chains are a method to overcome limitations of classifier chains and to improve their performance. As mentioned before, the performance of a classifier chain may depend on the order of the single classifiers and may lead to different results. In a learning ensemble this problem is minimized by grouping together a number of classifier chains each with a different order. It was used as described in Read et al. (2009; 2011).

Subsequently, a function is needed to determine the number of chains that need to come to the same voting in order to determine the overall category for a given document. Read et al. (2011) suggest to use the method by Tsoumakas and Katakis (2007) to calculate a threshold value as a lower bound

for the number of chains that need to be in agreement. In more detail, they use the label cardinality $LCard$ which is the average number of labels L per document over all categories (see Equation 3), whereas $L_{i,j}$ is the label assigned to the i-th document for the j-th category.

$$\text{LCARD} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{|L|} L_{i,j} \quad (3)$$

The number of the voting classifier chains v that minimizes the difference between the $LCard$ of the training set and the $LCard$ of the test set is considered to be the optimal number of of voters $\hat{v}$ which have to be in agreement (see Equation 4).

$$\hat{v} = \arg\min_{v} |LCard^{train} - LCard^{test}| \quad (4)$$

5 Experimental Results

The models used in the study were trained on the training set containing 14,548 blurbs. For the evaluation the provided validation set containing 2079 blurbs and the Python script was used (Aly et al., 2019). The models were named according to which method or feature was used. An overview of the abbreviations is given in Table 3. Each name thus consist of the SVM implementation used (L or C), the vectorization method (G or F) and whether or not the author information was used as an additional feature (A or nothing).

method/feature	value	abbrev.
SVM	LibSVM / Caret	L / C
vectorization	GloVe / FastText	G / F
author	yes/no	A

Table 3: Explanation of abbreviations for naming the models.

5.1 Comparison of different Models

Overall, five different models were compared with each other, whereas for each model 5 chains were used to prevent anomalies of single chains. The results can be found in Table 4. It can be seen that the CGA-model has the best performance with an F-score of 0.8291, as well as the best results for precision and recall. Furthermore, it can be noted that both models using FastText perform worse than the models using GloVe. The reason could be the vectorization method itself, yet, it should also be considered that for GloVe pre-trained vectors were used whereas FastText was trained on the blurbs, a much smaller corpus.

Additionally, it can be seen that those models containing the author information as a feature perform better than those not considering this information. This can be easily explained. Most authors wrote books in only one of the eight categories and correspondingly, about 93% of all books only belong to a single category. Hence, using the additional information can improve the results.

Model	Precision	Recall	F1
CFA	0.7758	0.7619	0.7688
CF	0.6515	0.6596	0.6555
CGA	**0.8429**	**0.8157**	**0.8291**
CG	0.7656	0.7794	0.7724
LG	0.7690	0.7955	0.7820

Table 4: Evaluation results for different models each containing five chains.

5.2 Influence of the number of chains

The influence of the number of chains in an ensemble classifier chain was analysed using the example of the CG-model. The results are shown in Table 5. It can be seen that the performance only slightly increases with the number of chains and that the best F1-score is reached with 10 chains. However, the difference to the F1-score of one chain is only minimal. The results confirm the assumption that there is no clear interdependency between the categories at the upper level.

# chains	Precision	Recall	F1
1	**0.8083**	0.7395	0.7724
2	0.7756	0.7704	0.7730
3	0.7793	0.7614	0.7702
4	0.7743	0.7722	0.7732
5	0.7656	0.7794	0.7724
6	0.7748	0.7776	0.7762
7	0.7748	0.7776	0.7762
8	0.7796	0.7709	0.7752
9	0.7778	0.7785	0.7781
10	0.7802	**0.7897**	**0.7849**

Table 5: Performance of the CG model with different numbers of chains.

Moreover, when a single chain is used, the difference between precision and recall is greater than when more chains are used. This can be explained with the threshold value. That means, the more chains need to be in agreement, the more decreases the probability to assign a certain category erroneously, but the more decreases the probability to assign that category generally. In other words, the classifier chain is more conservative in assigning a label. As a consequence less categories are predicted which leads to a higher precision but a lower recall.

5.3 Combination of different models

As was discussed in Section 5.1, the models differ in their performance depending on what features are used. Hence, one idea was to combine the two best performing models (CGA and LG) in order to get better results. Thus, the models were combined by using different numbers of chains from each model. However, no further improvement could be noticed. The best F1-scores are still reached when only the CGA model is used. This indicates that both models misclassify approximately the same books. Consequently, these books cannot be classified correctly, even when combining two models.

5.4 Performance of the different categories

In this section the performance of each category is presented using the example of the CGA-model containing 10 chains and using a minimal consensus of four voters. This model is the one with the best overall performance, reaching an F1-score of 0.841.

We found out that the categories (1, 2, 4, 8, 16, 32, 64, 128) perform very good if they occur as the only category for a book (single label). The category with the best results is category one (*Literatur & Unterhaltung*) with an F-score of 0.931, whereas the category with the worst results is 16 (*Ganzheitliches Bewusstsein*) with an F-score of 0.625. The first category is also the category with the most books in the corpus; about 50% of all books. Thus, it is not surprising that this category has the best classification results. It is also the reason for the good overall performance of the model. In contrast, the performance is much worse when categories are combined (multi label). Then all F-scores are below 0.3333 and most of them are even 0. In all those cases precision as well as recall are very low. Nevertheless, it does not affect the overall performance much because it is evaluated using the micro average F-score and the multi label classes only rarely appear (about 7%).

6 Conclusion

This paper deals with multi-label classification of blurbs using ensembles of classifier chains. The best result was achieved with an ensemble of 10 classifier chains, whereas for each classifier a Support Vector Machine with a radial kernel function was used as well as a global vector representation combined with the author information as one feature with an micro F1-score of 0.841. It was shown that neither the number of chains nor the combination of different models had an important influence on the performance. Furthermore, the performance of each category was analysed and the results show that the best performing category is *Literatur und Unterhaltung*, whereas the worst on is *Ganzheitliches Bewusstsein*.

References

Rami Aly, Steffen Remus, and Chris Biemann. 2019. Codalab's evaluation script for germeval-2019 task 1: Shared task on hierarchical classification of blurbs. Language Technology Lab, Universität Hamburg.

Chih-Chung Chang and Chih-Jen Lin. 2001. *LIBSVM: a library for support vector machines*. Department of Computer ScienceNational Taiwan University, Taipei, Taiwan.

Ashok Chilakapati. 2018. Word Embeddings and Document Vectors: Part 1. Similarity. http://xplordat.com/2018/09/27/word-embeddings-and-document-vectors-part-1-similarity/. Accessed 11.06.2019.

Shantanu Godbole and Sunita Sarawagi. 2004. Discriminative methods for multi-labeled classification. In *Advances in Knowledge Discovery and Data Mining*, pages 22–30, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. 2016. A practical guide to support vector classication. Technical report, Department of Computer Science, National Taiwan University.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, pages 427–431.

Max Kuhn. 2019. *Package 'caret' Version 6.0-84*.

Lam Hong Lee, Chin Heng Wan, Rajprasad Rajkumar, and Dino Isa. 2011. An enhanced support vector machine classification framework by using euclidean distance function for text document categorization. *Applied Intelligence*, 37(1):80–99.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *Computing Research Repository*, arXiv:1301.3781. Version 3.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Malte Pietsch, Timo Möller, and Milos Rusic. 2018. Pretrained German Word Embeddings. deepset GmbH, https://deepset.ai/german-word-embeddings. GloVe of german Wikipedia Corpus, accessed 25.05.2019.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2009. Classifier chains for multi-label classification. In *Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer Berlin Heidelberg.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359.

Steffen Remus, Rami Aly, and Chris Biemann. 2019. Germeval-2019 task 1: Shared task on hierarchical classification of blurbs. In *Proceedings of the GermEval 2019 Workshop*.

Gerard Salton, Alice Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Helmut Schmid. 1994. *TreeTagger - a part-of-speech tagger for many languages*. Institute for Computational Linguistics of the University of Stuttgart.

Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13.

Yanping Yin and Zhong Jin. 2015. Document sentiment classification based on the word embedding. In *Proceedings of the 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering 2015*, pages 456–461. Atlantis Press.

Min-Ling Zhang and Zhi-Hua Zhou. 2005. A k-nearest neighbor based algorithm for multi-label classification. In *IEEE International Conference on Granular Computing*, volume 2, pages 718 – 721.