Hierarchical multi-label classification of Blurbs (coarse)

Robin Meier, Julian D. Dommers, Sophie Friedl, Michelle Liebold
HS Mittweida
{rmeier3, jdommers, sfriedl, mliebold}@hs-mittweida.de

Carolin Diener, Leon Schulz, Maria-T. Eger, Lukas Kallert
HS Mittweida
{cdiener, lschulz3, meger, lkallert}@hs-mittweida.de

Abstract

The goal of Hierarchical multi-label classification of Blurbs is the classification of German books into different genres. Therefor we used the Random Forest classifier and trained it with various features. Our programme has a greatest F1 score of 80.89 percent.

1 Introduction

Nowadays there exist a lot of documents on the internet because of the digitisation. For this reason it is necessary to support the classification of documents. This classification simplifies the dealing with the documents. Therefore books are classified in different genres on the basis of their blurbs.

One possibility for this task is the Hierarchical multi-label classification of Blurbs, which is used to classify labels for a short text. Moreover each label has a hierarchical structure of different categories.

The association of blurbs happens with a programme which uses different features. This task pays attention to the classification of German books into eight genres: *Literatur & Unterhaltung, Ratgeber, Kinderbuch & Jugendbuch, Sachbuch, Ganzheitliches Bewusstsein, Glaube & Ethik, Künste, Architektur & Garten.*

Today there exist some new causes for improving the Hierarchical multi-label classification. For example the increasing digital documents and the necessary fine categories. For these reasons the traditional multi-label text classification must be attempted and we like to support a new approach with our ideas for a Hierarchical multi-label classification of blurbs (coarse).

2 Material and Methods

The basis of our programme "Hierarchical multi-label classification of blurbs (coarse)" is the programming language "Python". Furthermore the libraries "sklearn", "TextBlobDE", "spacy"and "pandas" were used. The library "spacy" is utilised for the lemmatisation. "Sklearn" is a library for machine learning. A tool for the management and analysis of data is "pandas". "TextBlobDE" is a library for processing textual data.

The data for training and testing was provided by "Language Technology Group, Universität Hamburg". These files are available at `https://competitions.codalab.org/competitions/21226`.

The dataset of German books consists of: blurb, genres, title, author, URL, ISBN and date of publication. The books are crawled from the Random House page.

There exist a lot of different classifiers for this task. Therefore we tested an "Ensemble" with five classifiers. These were Random Forest, Radius Neighbor, Gaussian Naive Bayes, Multinominal Naive Bayes, Bernoulli Naive Bayes. After data pre-processing the five classifiers used a majority decision to get a score. But the analysis showed that the Random Forest classifier singularly is a preferable choice for our problem. Because all other classifiers had functional difficulties and at this moment one of our features, the dictionary, was not implemented correctly.

For a solid and competent programme we created a plan of the procedure. At first we fed all trainings and test data into our programme. These files are available at `https://competitions.codalab.org/competitions/21226`. Secondly we isolated the blurb, author, titel, all categories, ISBN and the first category of each data set. In the third place we fed a list of stop words into our programme. That list is available in "Python" in the package of "NLTK". Furthermore we added some words on our own.

After that we created dictionaries for each genre. "Textblob"was used for the language analysis. In addition to that we formatted each word into lower

case letters. If the word is a noun, verb or adjective, then the programme must put it in the most suitable dictionary. In case of pre-existing words in the dictionary, we increase the number by one. Furthermore we used n-grams for the creation of the dictionaries. N-grams are the result of breaking a text into fragments. The text is decomposed and contiguous fragments are summarised as n-grams. The fragments can be nouns, verbs or adjectives.

To improve the dictionaries we tested three so called "dictionary-functions". For the first one we utilised parts of the "Gompertz-function": $df1(x)=\frac{\ln(1{,}5 \cdot counter)}{e \cdot e^{-1 \cdot e^{-\frac{2}{e} \cdot x}}}$. In the second place parts of the "Sigmoid-function" were harnessed: $df2(x)=\frac{ln(1{,}5 \cdot counter)}{\frac{1}{1+e^{-x}}}$. Finally we used a logarithm for the third "dictionary-function": $df3=\frac{\ln(counter)}{number\ of\ dictionaries\ with\ the\ word}$.

Because of the greatest functionality of the third "dictionary-function" we used it for all further runs.

Features have an enormous influence on the classifier. For this reason we established the following features: number of words in one sentence, number of sentences in one blurb, relative frequency of nouns, verbs and adjectives in a blurb, number of selected symbols (? # $ % & ¨ : ; , -) and the minimal gap between a genre and a feature.

Beyond we constructed a self-written feature, which operates as described: The feature compiles for each genre an ideal average of all features. As a result the programme created an optimal book for each genre. Besides the distance of every feature to every ideal book was measured.

In addition to that we created another feature, the "genre dictionary accordance rate". This rate is calculated by the addition of the values of the optimal book and the dictionary(both of one genre). Moreover this results were divided by the number of the counter.

With the assistance of our features an array was issued, which we utilised for the DataFrame. This DataFrame contained on the one hand all features and on the other hand all genres. With this DataFrame our classifier Random Forest trained and predicted. For the training we constructed 100 decision-trees with a maximal depth of 50 and a minimal sample leaf of one.

Subsequently we tested three functions for the multi-label classification. The first one worked with a recursive function which started with a variable threshold. If a genre was above this threshold, then the function will define the genre as correct. If no genre can exceed the threshold, it runs again with a reduced one.

Secondly we created a function which constructed a "barrier". This gate was made by the subtraction of the value of the maximal genre and a fixed variance. Every genre above this threshold was defined as a correct result.

The third function operated with a fixed threshold. Based on the sorted probabilities of the genres, our function filtered the highest probabilities to reach the fixed threshold.

Finally the dataset of our results was printed. This file possesses the following format: A table shows in the first column the ISBN and in the second column the genres of the particular book.

3 Results

To represent our programme of Hierchical multi-label classification of Blurbs (coarse) we like to introduce our results for the 10-fold cross-validation.

The following table shows different runs of our programme with various parameters. This table has four columns, which are Method, Precision, Recall and F1 Score. "Method" describes the features we tested for this special run. The other three columns contain the results of precision, recall and F1 Score for this run.

The best result of this evaluation is the F1 Score with 80.89 percents.

The following abbreviations and explanations support the comprehension of the table:

CO describes a cut off value which sets all dictionary entries to zero if they are smaller than the cut off value.
DF1/DF2/DF3 are the dictionary Functions, where **DF1** uses the Gompertz-fuctions, **DF2** uses the Sigmoid-function and **DF3** uses a simple logarithmic function. The **MO1/MO2/MO3** functions are used to get multiple outputs based on their probabilities. For all runs we utilised one genre per book to train with. We added the title to the blurb to get more words. All features where used inside a mean-feature for each genre:

- **SI** - sentence information (words per sentence and number of sentences)
- **RFWC** - relative frequencies of word classes (noun, verb, adjective)
- **#S** - number of chosen Symbols (? # $ % & ¨ : ; , -)

- **GDAR-1** - genre dictionary accordance rate (1Grams)
- **GDAR-15** - genre dictionary accordance rate (1-5Grams)
- **1GDAR-15** - genre dictionary accordance rate (1-5Grams) all in Grams in one wordbook
- **MF** - mean features (distance for each feature to each optimal genre of the feature)

Method	Precision	Recall	F1-Score
SI + RFWC + #S + GDAR-15 + MF + MO2(0.12) + DF1 + CO(1)	0.8023	0.7991	0.8007
SI + RFWC + #S + GDAR-15 + MF + MO2(0.12) + DF2 + CO(1)	0.7910	0.7955	0.7872
SI + RFWC + #S + GDAR-15 + MF + MO2(0.12) + DF3 + CO(1)	0.7064	0.7283	0.7172
SI + RFWC + #S + GDAR-15 + MF + MO2(0.1) + DF1 + CO(0)	0.8187	0.7897	0.8039
SI + RFWC + #S + GDAR-15 + MF + MO2(0.1) + DF1 + CO(1)	0.8084	0.7964	0.8023
SI + RFWC + #S + GDAR-15 + MF + MO1(0.7,0.1) + DF1 + CO(0)	0.8264	0.7789	0.8019
SI + RFWC + #S + GDAR-15 + MF + MO3(0.6) + DF1 + CO(0)	0.7868	0.8076	0.7971
SI + RFWC + #S + GDAR-1 + MF + MO3(0.7) + DF1 + CO(0)	0.7604	0.8296	0.7935
SI + RFWC + #S + GDAR-1 + MF + MO1(0.7,0.1) + DF1 + CO(0)	0.8252	0.7789	0.8014
SI + RFWC + #S + GDAR-15 + MO2(0.1) + DF1 + CO(0)	0.7806	0.7946	0.7876
SI + RFWC + #S + 1GDAR-15 + MF + MO2(0.1) + DF1 + CO(0)	0.8175	0.7874	0.8022
SI + RFWC + #S + 1GDAR-15 + MF + MO2(0.1) + DF1 + CO(0.56)	0.8162	0.7946	0.8053
SI + RFWC + #S + 1GDAR-15 + MF + MO2(0.1) + DF1 + CO(0.56)	0.8186	0.7951	0.8066
MultiLabelInput + SI + RFWC + #S + 1GDAR-15 + MF + MO2(0.1) + DF1 + CO(0.56)	0.8071	0.8108	0.8089
Lemmatisation + MultiLabelInput + SI + RFWC + #S + 1GDAR-15 + MF + MO2(0.1) + DF1 + CO(0.56)	0.8061	0.8090	0.8075

Table 1: Evaluation results

Subsequently, three runs of our model using the Random Forest classifiers for the Hierarchical multi-label classification of Blurbs were submitted as:

-
-
-

4 Discussion

Lemmatisation is a method to map a word into its dictionary form. This method can be used to reduce the amount of words with a similar meaning. The function of lemmatisation in our programme is the improvement of the "genre dictionary accordance rate" (GDAR), which is not implemented properly. The stage of development of lemmatisation needs some new innovations to improve the work with german texts. Furthermore the time required is 20-times greater than without lemmatisation. With the use of this method our results deteriorated minimally.

Moreover the dictionary functions are used as weighting functions. If one word is more often in several dictionaries, then the weighting function will be lesser. The three dictionary functions are constructed as follows:

- The numerator of the three functions is a natural logarithm which connects the number of words with the weakness of the gradient.
- The denominator of df1 connects the number of dictionaries in which one word occurs with the scale of the whole value.
- The denominator of df2 describes the change of the linear reduction dependent on the frequency of one word in different dictionaries.
- The denominator of df3 is similar to the denominator of df1 with additional parameters.

In addition to that the mapping of the dictionary function one is located between zero and one. Furthermore the dictionary function three maps between zero and the Euler number.

The results show that the greatest functions is the df3. The presumably reasons is the use of parameters.

The meaning of the cut off is the separation of less relevant words in dictionaries because of the sparsely weighting. Different tests shows that the value of 0.56 is the greatest. All overhead values are relevant.

Thresholds are used for the multi-label output. The second multi-label function provides the best results.

Finally we establish an oversampling which arises from the unweighted data. This problem is not resolved.

5 Conclusion

Our programme describes the "Hierarchical multi-label classification of blurbs (coarse)" with the use of the classifier Random Forest. In connection with that we created different features like the dictionaries.

To sum up, one could say that the "Hierarchical multi-label classification of blurbs (coarse)" is a complex and challenging problem. For this reason the problem needs an advanced development. Suggestions for our programme are the avoidance of oversampling and the improvement of the german lemmatisation. The problem of this lemmatisation is the necessary compound decomposition which should be advanced.

References

Rami Aly, Steffen Remus and Chris Biemann 2019. *GermEval 2019 Task 1 – Shared task on hierarchical classification of blurbs.* Language Technology group of the University of Hamburg, Germany.