Hierarchical classification of Blurbs (coarse)

Robin Meier, Julian D. Dommers, Sophie Friedl, Michelle Liebold
HS Mittweida
rmeier3, jdommers, sfriedl,
mliebold, cdiener, lschulz3, meger, lkallert @hs-mittweida.de

Carolin Diener, Leon Schulz, Maria T. Eger, Lukas Kallert
HS Mittweida

Abstract

This document contains the instructions for preparing a camera-ready manuscript for the proceedings of KONVENS 2018. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used for both papers submitted for review and for final versions of accepted papers. Authors are asked to conform to all the directions reported in this document.

1 Introduction

Hierarchical multi-label classification of Blurbs is used to classify labels for a short text. Moreover each label has a hierarchical structure of different categories. Today there exist some new causes for improving this classification. For example the increasing digital documents and the necessary fine categories. For these reasons the tradditional 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 classification of blurbs (coarse)" is the programming language "Python". Furthermore the libraries "json", "argparse", "random", "timeit", "spacy", "pandas" and "numpy" were used. In addition to that we imported single functions from different libraries. These were: the function "Pool" from the library "multiprocessing", "TextBlobDE" from "textblob_de", "RandomForestClassifier" from "sklearn.ensemble", "metrics" from "sklearn" and the functions "GridSearchCV", "cross_val_score" and "cross_val_predict" from "sklearn.model_selection".

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

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 Random Forest classifier singularly is a preferable choice for our problem.

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 an 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. To improve the dictionaries we tested three functions. The first one was the sigmoid-function: $sig(x)=\frac{1}{1+e^{-x}}$. In the second place we used a logarithm: $\frac{\ln(counter)}{number\ of\ dictionaries\ with\ the\ word}$. Finally the "Gompertz-function": $gomp(x) = \frac{\ln(1{,}5 \cdot counter)}{e \cdot e^{-1 \cdot e^{-\frac{2}{e} \cdot x}}}$ was benefitted. Because of the finest functionality of this role we used it for all further runs.

Features have an enormous influence on the clas-

sifier. 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 Nearest Zentroid as another feature. This classifier compiled 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 ”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 fixed threshold of If a genre was above this threshold, then the function will define the genre as correct.

Secondly we created a function which constructed a ”barrier”. This gate was made by the subtraction of the maximal genre and the variance of 0,1. Every genre above this threshold was defined as a correct result.

The third function operated with a fixed threshold of . 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 columm the ISBN and in the second columm the genres of the particular book.

3 Results

3.1 Fonts

For reasons of uniformity, Adobe's **Times Roman** font should be used. The style file has been adjusted (2015-06-16) so that the above commands are included.

Please remove any command to use the `times` package from your LaTeX source.

```
% remove lines like:
\usepackage{times}
```

Type of Text	Font Size	Style
paper title	15 pt	bold
author names	12 pt	bold
author affiliation	12 pt	
the word “Abstract”	12 pt	bold
section titles	12 pt	bold
document text	11 pt	
captions	11 pt	
abstract text	10 pt	
bibliography	10 pt	
footnotes	9 pt	

Tabelle 1: Font guide.

3.2 The First Page

Center the title, author's name(s) and affiliation(s) across both columns. Do not use footnotes for affiliations. Do not include the paper ID number assigned during the submission process. Use the two-column format only when you begin the abstract.

Title: Place the title centered at the top of the first page, in a 15-point bold font. (For a complete guide to font sizes and styles, see Table 1) Long titles should be typed on two lines without a blank line intervening. Approximately, put the title at 2.5 cm from the top of the page, followed by a blank line, then the author's names(s), and the affiliation on the following line. Do not use only initials for given names (middle initials are allowed). Do not format surnames in all capitals (e.g., use “Schlangen” not “SCHLANGEN”). Do not format title and section headings in all capitals as well except for proper names (such as “BLEU”) that are conventionally in all capitals. The affiliation should contain the author's complete address, and if possible, an electronic mail address. Start the body of the first page 7.5 cm from the top of the page.

The title, author names and addresses should be completely identical to those entered to the electronical paper submission website in order to maintain the consistency of author information among all publications of the conference. If they are different, the publication chairs may resolve the difference without consulting with you; so it is in your own

interest to double-check that the information is consistent.

Abstract: Type the abstract at the beginning of the first column. The width of the abstract text should be smaller than the width of the columns for the text in the body of the paper by about 0.6 cm on each side. Center the word **Abstract** in a 12 point bold font above the body of the abstract. The abstract should be a concise summary of the general thesis and conclusions of the paper. It should be no longer than 200 words. The abstract text should be in 10 point font.

Text: Begin typing the main body of the text immediately after the abstract, observing the two-column format as shown in the present document. Do not include page numbers.

Indent when starting a new paragraph. Use 11 points for text and subsection headings, 12 points for section headings and 15 points for the title.

3.3 Footnotes

Footnotes: Put footnotes at the bottom of the page and use 9 points text. They may be numbered or referred to by asterisks or other symbols.[1] Footnotes should be separated from the text by a line.[2]

3.4 Graphics

Illustrations: Place figures, tables, and photographs in the paper near where they are first discussed, rather than at the end, if possible. Wide illustrations may run across both columns. Color illustrations are discouraged, unless you have verified that they will be understandable when printed in black ink.

Captions: Provide a caption for every illustration; number each one sequentially in the form: "Figure 1. Caption of the Figure." "Table 1. Caption of the Table." Type the captions of the figures and tables below the body, using 11 point text.

Acknowledgments

The acknowledgments should go immediately before the references. Do not number the acknowledgments section. Do not include this section when submitting your paper for review.

[1]This is how a footnote should appear.

[2]Note the line separating the footnotes from the text.

4 Discussion

5 Conclusion

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.