# *Judging a book by its (back) cover*

**Erica Cau**
e.cau@studenti.unipi.it
Student ID: 545126

**Andrea Failla**
a.failla@studenti.unipi.it
Student ID: 627098

**Federico Mazzoni**
f.mazzoni6@studenti.unipi.it
Student ID: 524324

## ABSTRACT

What can the back cover of a book, or rather, the summary thereby contained – reveal about the book itself? In this project, we attempt to classify a large collection of books with respect to their genre by only exploiting their summary. Additionally, we test and extract the main features of each genre to assess whether the quality of a book is related to the clarity of its summary. While some algorithms performed sub-optimally during these tasks, the results were both interesting and satisfying overall.

## 1 INTRODUCTION

Research in NLP has provided a great variety of algorithms and models which can be used to process and analyze text. In this paper, we will employ some of these techniques – from the traditional *Word2Vec* to the cutting-edge *Zero-Shot Learning* – in order to answer the following question: can a book be judged by its (back) cover? More specifically, can it be judged by *the summary* on the back cover? In order to answer this question, we will check:

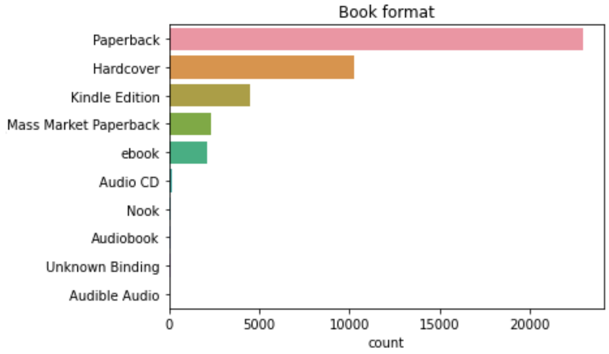
- whether a given book can be *judged by its genre* by an algorithm. As such, the question was turned into a classification problem;
- after collecting a set of books of a given genre, whether it is possible to exctract semantically relevant information about said genre, e.g., exploiting NER strategies;

The rest of the report is organized as follows. First, we do a preliminary data exploration, detailing the dataset and underlining an inherent problem w.r.t. the definition of *genre*. Then, we move on to the genre recognition task, exploiting a wide range of techniques in order to classify book summaries. Lastly, we extract and compare the named entities of the summaries to unveil genre-dependent traits in book summaries.

## 2 DATA EXPLORATION

The publicly-available *Goodreads' Best Books Ever*[1] dataset was compiled in 2019 and it includes the best **54301** reviewed books at the time, with 12 features for each of them. However, the dataset contains many duplicated books, e.g. the same book had two different records for a *Paperback* and a *Hardcover* edition. Additionally, the dataset includes non-English books, which were discarded for the purposes of this project, along with duplicated titles. We then focused on genres, our target variable. The dataset featured a list of genres for each book. This is due to *Goodreads*'s user-defined genre system. For example, at the time of writing, 15,579 users defined *The Hitchhiker's Guide to the Galaxy* as *Science Fiction*, 11,682 users as *Fiction* and only 5,777 users as *Humor*. Additionally, those labels are not mutually exclusive (i.e., a user could have defined the book both as *Science Fiction* and *Humor*). The list provided by the dataset was ordered w.r.t. the most-common labels (i.e., the *Guide* had *Science Fiction* at the top of the list). Overall, we had more than 25000 genres, with some of them non-traditional or too specific (like *Dungeons and Dragons* – a macro category for the books related to the homonymous tabletop role-playing game – and *Sequential Art*, a category for comics[2] and drawing handbooks). As such, we decided to focus on the top 10 genres: *Young Adult*, *Fantasy*, *Classics*, *Historical*, *Science Fiction*, *Fiction*, *Nonfiction*, *Romance*, *Mystery* and *Sequential Art*. Since most of the books had more than a single genre according to the users, we arbitrarly assigned as genre the first on we found in the genres list that was also among our 10 labels.
It was also created another smaller dataset, containing only

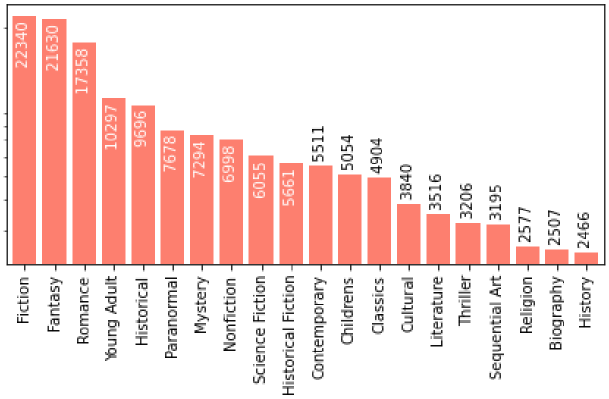
**Figure 1: Book format**



[1] *Goodreads' Best Books Ever* dataset on Kaggle: https://www.kaggle.com/datasets/meetnaren/goodreads-best-books

[2] As defined in Scott McCloud's *magnum opus*, *Understanding Comics*. Amusingly, McCloud makes it also clear that *Sequential Art* is a *medium*, and not a kind of *literary genre*.

**Figure 2: Top genres (original dataset)**



the books having as first genre one of the genres in our 10-genres list.

## 3 WHAT IS A *GENRE*?

Traditionally, there are two contradictory definitions of (literary) genre.

(1) a *narratological* definition sees a genre as a set of narrative rules and events, involving characters and places belonging to defined archetypes. For example, we expect the characters of two lovers to have a key narrative role in a *Romance* book. While they might appear in a *Mystery* book, in that case their relationship would probably play a minor role;

(2) alternatively, a genre can be seen as a set of books sharing the same overall *aesthetic*. The aforementioned *The Hitchhiker's Guide to the Galaxy* and Isaac Asimov's corpus of works have narratively very little in common, but they share the same overall *Sci-Fi* aesthetic involving robots, space travels, aliens and foreign planets.

For instance, *A Game of Thrones*, the first book in George R. R. Martin's *A Song of Ice and Fire*, is narratively a *mystery* book (the character of Ned Stark is the detective investigating the hidden plots of House Lannister), but it esthetically belongs to the *fantasy* genre.

For the purpose of our analysis, it is worth noticing that half of the top 10 genres we identified does not belong to any of these definitions. Indeed, *Young Adult* identifies the *target audience*, *Classical* addresses the book's role in the literary canon, *Fiction* and *Non-Fiction* the kind of text (with *Fiction* technically being a superset of *Fantasy*, *Science Fiction*, *Romance* and *Mystery*), and we already noted the problematic nature of the *Sequential Art* label. Additionally, while *Historical* might identify a common aesthetic, it fails to describe a set of narrative rules.

Therefore, our dataset lacks a coherent definition of genre, which may explain some of the sub-optimal results we obtained during the classification task.

For the NER task – and for the Zero-Shot Learning – in order to correctly identify the characteristics of a given genre, we decided to limit the scope of our analysis to the 4 genres belonging to both of the definitions we presented, i.e. *Fantasy*, *Science Fiction*, *Romance* and *Mystery*.

## 4 MULTILABEL CLASSIFICATION

In this section we model the problem as a multiclass classification. The target variable, *genre*, was defined by chosing as distinctive label for each book, the most frequent genre among the 10 most common genres in all the dataset.

In order to test different algorithms and techniques to extract textual features, the dataset was divided into a training and a test set using a 70/30 split.

### Traditional Machine Learning algorithms

At first, we approached the task through vectorization and traditional Machine Learning algorithms, i.e. *Support Vector Machines*, *Decision Trees*, *Random Forests* and *Logistic Regression*.

We set a custom baseline using a Logistic Regression without performing any parameter tuning or text cleaning step. As can be seen in Table 1, we obtained a base accuracy of 65%. By looking at the single genres, we observed that this predictor recognized particularly well some genres, namely *Sequential Art* and *Sci-fi* (over 80% of precision).

Moving forward, we tested Linear SVMs which, in their Scikit-learn implementation, employ by default a One-Vs-One approach for multiclass problems.

First of all, we built a pipeline with the aim to clean, tokenize and extract the features from the text. As it regards the data cleaning step, it was performed using *SpaCy* and brought us to remove extra spaces and new lines; we also detected and removed – using regular expressions – hashtags, URLs, stopwords and punctuation. Then, we proceeded with the extraction of lemmas, bigrams and trigrams, which were also the input for the `TfidfVectorizer`, which allowed us to create the features vectors.

The SVM was implemented using a *pipeline*, that performed four different steps: at first, it applied the `TfidfVectorizer`, then the selection of the most relevant features according to the *chi-squared test* and their weighting. The last step was the application of the classification algorithm, that ran both on train and test set, to better evaluate the model. While the results were good on the training set, with a 95% of accuracy, the same cannot be said for the results on the test set, since they were below the baseline (63% of accuracy).

Then, we tried to improve the performance using a GridSearch with a *10-fold cross validation.* At the end of the hyperparameter tuning phase, we ran the algorithm again with the best parameters ($C = 0.1$ and $K = 5000$). Using this configuration we achieved a 65% of accuracy. We noticed that the label *Sequential Art* was the most recognizable among all the other nine, with an accuracy equal to 74%.

After some very undeperforming tests with *Decision Trees*, we moved onto their ensemble counterparts, namely *Random Forest.* The model was fine-tuned using a *GridSearch* with a *5-fold cross-validation* but the results were nevertheless below the baseline.

An improvement was seen using the Logistic Regression, which returned more balanced results, both on the training and test: on the former, it performed better (78%), while on the latter the accuracy was 67%, slightly over the baseline.

A last attempt was to use a smaller dataset[3], but still, the results were below or almost equal to the baseline.

### Word2Vec and Doc2Vec

We also tried the same task exploiting *Word2Vec* and *Doc2Vec*, to represent documents and words as vectors – although obtaining generally poorer results.

We employed Doc2Vec in its *Distributed Memory* variant[4]. We created document vectors with 300 features, ignoring words appearing less than 5 times setting, drawing 5 noise words (the lowest amount among the suggested range) and with a window of 10. The model was then trained for 50 epochs.

As for Word2Vec, after the train-test-split and the preprocessing, we created word vectors with 300 features, excluding words appearing less than 3 times and with a window of 10. Subsequently, from the average of all the word vectors in a sentence, we generated the respective *sentence vectors*, setting 100 features for each vector. Results were slightly better than those of *Doc2Vec*, especially with the *Logistic Regressor.*

### BERT

BERT is a transformer-based pre-trained model originally developed by Google and published in 2018 [1]. The models provides for tokenization and classification tasks, and is best suited for masked language prediction, question answering and, more generally for sequence classification. BERT comes in two variants, namely a *base* version, which exploits 12 encoders with 12 attention heads, and a *large* version, with double the encoders and 16 attention heads. In this work, we exploited both large and base, although obtaining similar results, with just a $\sim$ 2% increase in accuracy of the latter over the former. The results are the best so far, showing that large(r) training data/parameter sets/models can overcome (to an extent) the structural biases in our data. We also tested the capabilities of RoBERTa, an optimized BERT-based model which employs dynamic masking and a larger training set [4]. Overall, the differences in performance between the four models are not particularly relevant, thus we deem the last model to not be worth the extra time needed for fine-tuning.

## 5 BINARY CLASSIFICATION

Moving from the considerations in Section 2, we reshape the classification problem as binary, taking into account two mutually-exclusive genres, namely *Fiction* and *Nonfiction.* Results highlight that there is indeed information about the book's genre embedded in its summary, thus we conclude that the difficulties that have arisen in the previous task might be addressed by compiling a set of more rigorous labels genre-wise.

### TF-IDF vectorization and binary classification

At first, we addressed the binary classification task using the same TF-IDF vectorization already discussed in section 4. The classifiers performed undoubtedly better, as shown in Table 2. The most accurate model turned out to be the Logistic Regression – that achieved an accuracy equal to 90% – but also SVM and Random Forest returned interesting results (see Table 2).

### Word2Vec & Doc2Vec

Using the same parameters detailed before with the binary dataset resulted in higher performances. However, the models trained with both Word2Vec and Doc2Vec were still worse than the others.

### BERT

As all other models, BERT and RoBERTa performed way better in the binary classification task, and they still turn out to be the most accurate. All four BERT-based models reach $\sim$ 95% of accuracy on the test set, with similar scores during the validation phase. RoBERTa Large slightly outperforms the others. by just 1%. Once again, we wish to stress the high time and space complexity of these models, which require hours of training even on recent GPUs.

## 6 ZERO SHOT LEARNING

Zero-shot learning is a classification method that allows to link the inner meaning of textual data (but also images) to a set of labels given in input without tuning a model on a training set [3]. We implemented a Zero-Shot classifier using the

[3] The dataset is composed only by books having as a primary genre, a genre among the top 10-genres list.

[4] Preliminary tests with the *Distributed Bag of Words* variant provided very poor results

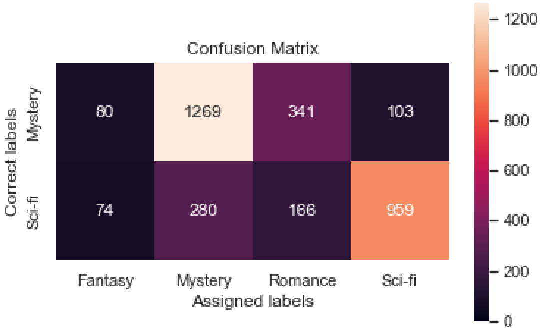
**Table 1: Multilabel classification results**

| | TF-IDF | | | | Doc2Vec | | | Word2Vec | | | BERT | | RoBERTa | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Baseline | LR | SVM | Random Forest | LR | SVM | Random Forest | LR | SVM | Random Forest | Base Unc. | Large Unc. | Base | Large |
| Accuracy | 0.65 | 0.67 | 0.65 | 0.56 | 0.53 | 0.53 | 0.46 | 0.59 | 0.57 | 0.52 | 0.75 | 0.77 | 0.75 | 0.78 |
| Weighted F-1 | 0.63 | 0.67 | 0.64 | 0.52 | 0.54 | 0.54 | 0.41 | 0.58 | 0.54 | 0.49 | 0.75 | 0.77 | 0.75 | 0.78 |

**Table 2: Binary classification results**

| | TF-IDF | | | Doc2Vec | | | Word2Vec | | | BERT | | RoBERTa | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LR | SVM | Random Forest | LR | SVM | Random Forest | LR | SVM | Random Forest | Base Unc. | Large Unc. | Base | Large |
| Accuracy | 0.90 | 0.89 | 0.88 | 0.84 | 0.84 | 0.74 | 0.87 | 0.84 | 0.82 | 0.95 | 0.95 | 0.95 | 0.96 |

**Figure 3: Zero-Shot Learning Confusion Matrix**



`zero-shot-classification` pipeline from HuggingFace[5] and the pre-trained `bart-large-mnli` model to perform the summary classification. We applied this approach to only two of the four subsets created for the NER task, namely *Sci-fi* and *mystery* books: this was necessary since this task took an high computational time to be completed.

The Zero-Shot Lerning model took as input the summaries and a set of four genres: *Mystery*, *Sci-Fi*, *Fantasy* and *Romance*. Then, it assigned to each summary three possible tags.

The results were interesting, since the model was precise in the recognition of the two labels, in particular with regard to *Sci-Fi* books, where it achieved a 0.90 (0.82 on *Mystery* books). We also noticed that the most common misleading labels were *Romance* and *Mystery* w.r.t. *Sci-fi* (see Figure 3): these results may be explained if we assume that books of various genres may be characterized also by *romance* and *mystery* elements in their storyline.

## 7 SUMMARY ANALYSIS

In this section, we exploit the Flair Python library to extract *named entities*, i.e. people, location or organization names with a proper name.

### Named entity recognition

The NER extraction was carried out using a pre-trained model among the ones offered by Flair: we chose the *English NER in Flair*[6], that allowed to recognize four different entities: **PER** (person name), **LOC** (location name), **ORG** (organization name) and **MISC** (other name). The model was applied in the four thematic subsets with books falling into a correct definition of genre, as discussed in Section 3.

Results are consistent with what is generally expected from each genre. *Romance* stories have character-driven narratives, while *Fantasy* and *Science Fiction* books feature a greater focus on world-building – with more importance given to the various places (or in *Science Fiction*'s case, the various organizations) featured, or – under the *MISC* category – the names of fictitious historical events (like ancient wars), races or artifacts (like the eponymous Ring in *The Lord of the Ring* saga). Both characters and places play major roles in Mystery books – understanding the roles of multiple characters in multiple places (and indeed, times) is, after all, the role of the detective.

It should be noted that the *MISC* category also features non-English character names (like *Felurian* from *The Name of the Wind*), or real world prizes and publications talking about the book.

## 8 CONCLUSIONS

Surely, book genre recognition is not an easy task. We believe the main issue lies with the definition of the set of genres

[5] Pipeline link: https://huggingface.co/facebook/bart-large-mnli

[6] Flair documentation: https://github.com/flairNLP/flair/blob/master/resources/docs/TUTORIAL_2_TAGGING.md

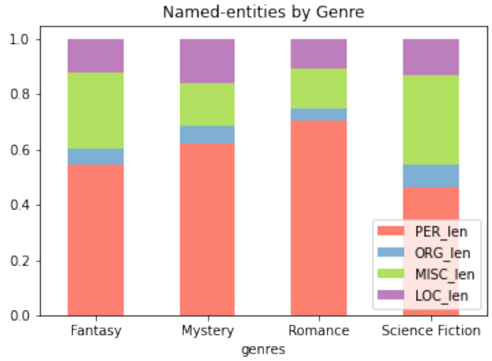


**Figure 4: Named-entities distribution**

to be recognized. For optimal solutions, such set should be composed of mutually-exclusive items, but in reality this is not always the case. In fact, books often belong to different genres (and subgenres), and as such have multiple peculiar traits that may mislead even state-of-the-art models. Another issue, although case-specific, is related to the user-defined nature of the genres in our data, as there is virtually no limitation to the amount of labels that users may create. Also, as noted in Section 3, this can lead — and has, in fact, led — to labels that do not actually correspond to literary genres, but rather to media (e.g., *Sequential Art*) or a vague aesthetics (e.g., *Historical*). *Fiction* also overlaps with several actual genres.

Nonetheless, we managed to achieve good results in the multilabel classification task, especially with transformer-based models such as BERT and RoBERTa. Better results were achieved in the binary classification task, which is understandably easier, for it was restricted to two mutually-exclusive genres, namely *Fiction* and *Nonfiction*.

We also exploited Zero-Shot Learning for the classification of *Science Fiction* and *Mystery* summaries obtaining interesting results, both w.r.t. precision and error-analysis, as we found out that some plot elements may be peculiar features of other genre (e.g. mystery traits in a Sci-Fi book).

Lastly, we focused on studying how genre affects the books' summaries in terms of *Named Entities*: insightful results are the high number of characters identified in the *Romance* summaries – a distinguishing trait of the genre's focus on characters and introspection – and the prevalence of fictitious organizations, historical events, races and items in *Fantasy* and *Sci-fi* books – two genres offering rich fictitious lores.

## 9 ACKNOWLEDGEMENTS

We wish to thank our friends Pippo and Alessandro. A very special thanks to John Muir.

## REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[2] Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. 2020. Task Aware Representation of Sentences for Generic Text Classification. In *COLING 2020, 28th International Conference on Computational Linguistics*.

[3] H. Larochelle, D. Erhan, and Yoshua Bengio. 2008. Zero-data Learning of New Tasks. In *AAAI*.

[4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]