university of groningen

faculty of arts

Event Prediction in Dutch Literature

Fine-tuning BERTje to investigate a correlation between canonicity and use of events

Björn Overbeek

Bachelor thesis
Information Science
Björn Overbeek
s4415213
July 21, 2023

ABSTRACT

This bachelor thesis explores the correlation between prestige and the occurrence of events in Dutch literature using natural language processing techniques. The study involves fine-tuning the BERTje language model for event detection in Dutch literary texts and achieves impressive results with an f1-score of 0.820 and 0.804 on two different train/test splits. The analysis reveals a significant correlation between novel canonicity and the use of non-realis events, suggesting that novels with higher prestige tend to utilize fewer events that do not actually occur in the story. The findings contribute valuable insights into the factors influencing a novel's canonical status, shedding light on the relationship between language and literary canonicity.

CONTENTS

PREFACE

This thesis is the culmination of three years of studying Information Science at the Rijksuniversiteit Groningen. I really enjoyed myself, but am also happy I will be done after this, since the workload increased as I got further into the study. I am in need of a good vacation.

I started off strong working on the thesis, but my motivation quickly decreased when I started struggling with Label Studio. By the time all the Label Studio issues were fixed and resolved, the deadline was too close to make the first deadline. Thanks to Huub and Cain for sticking with me through the Label Studio issues and still annotating data for me to complete my thesis in time. I think it turned out quite good in the end, and I am very satisfied with the result.

I would also like to thank my supervisor, Andreas van Cranburgh, for sticking with me through all the issues I had. Thanks to Niels, Ocar, and Dennis for providing me with some very useful last minute feedback.

Lastly I would also like to thank Marije and the rest of my family for being so supportive and sticking with me when I had so little time for them because I was so, so busy.

Björn Overbeek

Groningen, July 2023

1 INTRODUCTION

What elements contribute to a novel being canonical? This is a much discussed question in the literary studies, and not much quantitative studies have been performed to answer this question. Most research that has been done in this field is qualitative, such as foregrounding (van Peer and Hakemulder, 2006). The problem with quantitative research on literature in the past was that it takes a lot of time to perform. Recent advancements in natural language processing (NLP) have made it easier to focus on more quantitative features of novels, without sacrificing on the textual features we can investigate.

One of the features that we can investigate are events: an action that takes place in the novel. In this paper we will distinguish between two types of events: realis and non-realis events. Realis events being the events that actually took place in the story world, and non-realis are the events that we as the reader are not sure took place (see chapter 3.2 for more information). Events play a central role in the progression of a story, and by stringing these together a plot is created. Plots can be organized and created in very different ways, starting with the events (Brooks, 1992).

Previous work on event detection has mostly been focused on the news domain, for instance project NewsReader (Agerri et al., 2014) and the Automatic Content Extraction (ACE) program (Doddington et al., 2004). While it is very useful to detect events in news for tasks like text summarization and content extraction, these models are less effective on a domain like literature.

The role of events in literature is dissimilar to news articles that report factual events. Literary texts are generally longer than news articles, and since they are more creative, they do not have as clear of a causal structure as news articles.

A recent work by Sims et al. (2019) describes the training of machine learning models specifically to detect events in English literature. To show one of the advantages of creating such a model, they used the best performing model to predicted events in a set of novels selected based on prestige (Underwood, 2019). Using these predictions they find a correlation between the ratio of events and the prestige of the novel. An event prediction model can be used to find correlations like these and expand our literary knowledge.

The model by Sims et al. (2019) performs quite well on English novels, but there is not yet any model that predicts events on Dutch literature. With high-performance Dutch language models like BERTje (de Vries et al., 2019), a Dutch version of BERT (Devlin et al., 2019), it should be possible to create a similar model for Dutch literature. Therefore, the main research question we would like to answer in this paper is the following:

> Is it possible to fine-tune BERTje to predict events and their realis and non-realis modality in Dutch literature?

The analysis by Sims et al. (2019) was limited by using only realis events, and thus was not able to explore the difference in effect realis and non-realis events have on prestige. Using the model we will create, we will also try to answer the following research question:

> Is there a difference in the way novels with different levels of canonicity use realis and non-realis events in Dutch literature from 1850 to 1950?

To answer these questions, we will annotate events in Dutch literature and use these annotations to fine-tune BERTje. With this model we will predict events in novels that have a canonical indicators. Using these predictions we will be able to determine if there is a correlation between events and canonicity.

Our hypothesis for training the model is that we can fine-tune BERTje to achieve an f1-score of 0.8 for predicting events, because of the quality of the pre-trained model, and its performance on similar down-stream tasks. As for the correlation between canonicity and events, we expect there to be significantly more non-realis events in more canonical literature, since more non-realis events can increase the complexity of the story, and a certain level of complexity is associated with canonical novels.

2 RELATED WORK

Previous experiments on event detection were mostly done on media other than literature, such as the Automatic Content Extraction program (ACE) by Doddington et al. (2004) and the NewsReader project by Agerri et al. (2014). The latter also being able to predict events in Dutch news. Sims et al. (2019) trained several models to predict events in English literature, ranging from a simple to more advanced. They used BERT word embeddings (Devlin et al., 2019) for their more advanced models, but did not fine-tune the pre-trained model to predict events. Yang et al. (2019) explore the use of fine-tuning pre-trained models to predict events using less annotated data, or even use automatically generated examples.

de Vries et al. (2019) provide a Dutch transformer-based pre-trained language model that was able to outperform multilingual BERT (Devlin et al., 2019) on several downstream tasks, such as part-of-speech tagging, named-entity recognition, and sentiment analysis. BERTje was created using the same architecture and parameters as BERT, but was trained on Dutch text only. Noteworthy for our research is that a large portion (4.4GB) of the training data (12GB total/2.4 billion tokens) for BERTje consists of novels, compared to multilingual BERT which is only trained on Wikipedia text[1]. We assume that since our target medium is literature, this will improve the performance of our final model, since it will be more familiar with the structure and patterns commonly found in literary texts.

There are several works that incorporate novel prestige or canonicity. Algee-Hewitt et al. (2016) looked for textual features that are predictive of author prestige, where prestige was defined as times an author was included in the Oxford *Dictionary of National Biography*. The study also included author popularity as a measure, where popularity was measured by the amount of times a work was reprinted. Underwood (2019) defined author prestige as the number of times the author's works were reviewed by elite literary journals, and defined author popularity by the number of times their works can be found on historical bestseller lists. Underwood (2019) finds a correlation between high prestige fiction and the Harvard General Inquirer categories of `NATURAL OBJECTS` and `KNOWLEDGE AND AWARENESS`.

"The Riddle of Literary Quality"[2] is a project that investigates the literary quality of Dutch novels under the assumption that formal characteristics of a text influence the literary quality of the text. Several publications have been made contributing to this research.

To determine whether a novel is canonical, van Cranenburgh et al. (2022) created a corpus consisting of 1346 novels from DBNL[3]. The publication dates range of the novels from 1800 to 2000. The feature that sets this dataset apart from the other novel databases is the metadata of the novels, which also contain metrics relating to canonicity. These canonicity metrics include, but are not limited to: the number of times the author or novel is referred to in scholarly texts, the number of copies the Dutch libraries hold and lend of the novel, and the number of times the author is mentioned on Wikipedia[1].

[1] https://nl.wikipedia.org/wiki/Hoofdpagina

[2] https://literaryquality.huygens.knaw.nl/

[3] https://www.dbnl.org/

3 DATA AND MATERIAL

3.1 COLLECTION

For training the model, we used the OpenBoek corpus (van Cranenburgh and van Noord, 2022). The corpus consists of 9 fragments of Dutch literary texts as well as translated novels from 1860 to 1918. All the texts are public domain texts from Project Gutenberg. Each fragment is at least 10,000 tokens long, and the total corpus has 103,000 tokens. The corpus has already been annotated with coreference, entities, quotes, and POS tags. It has also contains output from Alpino (Bouma et al., 2001), a Dutch wide-coverage computational analyzer, and spelling normalized versions of the tokenized texts using the Oudeboeken Alpino extension (van Noord, 2023).

3.2 ANNOTATION

What exactly the definition of en event is will vary depending on one's background. To keep these annotations relatively simple, we have decided on the following definition:

> An event is the primary action or state expressed by the main verb in a sentence.

This means that we will not annotate auxiliary verbs for this task. If a verb is an event, we assign either a 'realis' or 'non-realis' tag to the verb, based on several conditions. These conditions are based on the ACE 2005 (Linguistic Data Consortium, 2005) and the Light ERE guidelines (Aguilar et al., 2014).

- **Tense**
  If the verb has a future or imperative tense form, we as the reader do not know whether the event actually happened, and thus the event is non-realis.
- **Polarity**
  If the verb has a negative polarity it will be marked as a non-realis event. Negative polarity can occur when negating words are present, like 'niet', 'nooit', or 'geen'. If any of these words alter the meaning of the verb, resulting in absence of the event, this verb will be tagged as a non-realis event.

  Any feeling of doubt will also result in a non-realis event.
- **Example**
  If the event is part of an example, and did not actually happen, it will also be tagged as non-realis. A common example of an example is the use of the word 'als'.
- **Question**
  If the verb is part of a question, it will also be considered a non-realis event, as we do not know whether the event happened.

If none of these conditions are met, we tag the event as realis.

The full annotation task also included agent/patient tagging, and a flag for if an event has a separate verb particle. Since we do not use these annotations for this study, we will not include information on these annotations here. The full guidelines, including more examples, the rules for the other annotations, and an annotation flowchart, can be found in the GitHub repository (Overbeek, 2023).

During the annotation of the text, we decided to only show one sentence at a time, and show all the sentences in a random order. This way, the sentences would be evaluated on their own, and not in the context of the rest of the story. This was to make the annotation task easier and more consistent.

3.3 ANNOTATION PROCESS

We decided to annotate the data (5709 lines) with three people, which resulted in 1903 sentences to annotate per person. In the end we only managed to annotate 4806 lines of the corpus.

To annotate the data we each set up a local instance of Label Studio (Tkachenko et al., 2020-2022), since a centralized server did not work good enough with the current version of Label Studio.

After updating the annotation guidelines several times, we arrived at an inter-annotator f1-score of 0.848 for events only over the 50 annotations we created separately. The precision and recall were 0.851 and 0.845 respectively. For a task that is relatively complex, we find the agreement rate satisfactory.

Annotating 1903 lines took around 20 hours actively annotating sentences. Note that this was not only events, but also their agents/patients, and separate verb particle flags.

The final annotations resulted in 10,158 events annotated, of which 7,582 realis, and 2,572 non-realis. See Figure 1 for a visualization of the distribution. On average, each annotation had 2.114 tagged events.

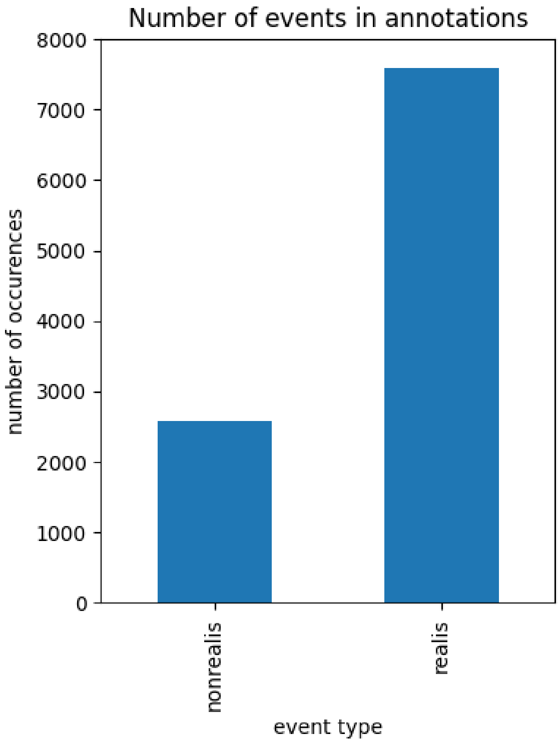

Figure 1: The event counts in all the annotations.

3.4 PROCESSING

To process the Label Studio output, we created an Annotation class that would take the Label Studio JSON file as input, and store all the useful information. When the annotations were then required for a new task, only a new export method would have to be written for the class. There are currently two export methods: one for the inter-annotator agreement, and one that exports the events for fine-tuning BERTje.

To merge the separate annotations into one file we created a small script. This script also checks for duplicate annotations and missing annotations.

4 EVENT DETECTION

4.1 MODEL

To detect events in Dutch literature, we will fine-tune BERTje for the downstream task of event detection. Since BERTje was already trained on literary texts, we decided not to introduce more literary texts, and directly fine-tune the model for event detection.

Using the Hugging Face transformers library (Wolf et al., 2020), we were able to fine-tune the model without too much code.

First we converted the annotations into a dataset (Lhoest et al., 2021) for optimal compatibility with the other classes in the transformers library. By using a seed when creating the train/test split in the dataset we ensure reproducibility. We created two splits to test and compare performance of the model. One model is trained and evaluated by a 90/10 train/test split, and the other model is created and evaluated by a more robust 75/25 train/test split.

We used the transformers AutoTokenizer loaded with the BERTje tokenizer to convert the tokens into BERTje WordPiece embeddings (Wu et al., 2016). Using subword tokens will help with complex words, and could possibly avert issues with old Dutch words that BERTje is not familiar with.

We used seqeval to evaluate the model predictions, both during fine-tuning and to evaluate the model on the test set at the end. This is a commonly used evaluation method in other token classification tasks, such as NER tagging, so we expected this evaluation to be a good fit for event detection.

After some experimenting we decided to leave most of the hyperparameters default, as this provided a good result without needing to train the model repeatedly with small differences in hyperparameters. We noticed noticeable differences in performance across runs because of the randomness of the model initialization. This was confirmed in a study by Orr et al. (2018). This resulted in setting one important parameter: a function for the initialization of the model, in which we set a seed to create reproducible results.

It took 3 epochs to fine-tune the BERTje model on our annotations.

4.2 RESULTS

The results of the two different models were slightly different. The results from the model with a 90/10 train/test split can be seen in Figure Table 1. The overall f1 score of 0.820 is quite close to the golden standard inter-annotator f1 score of 0.848. With a test set consisting of 10% of the data, there were only 989 events to base these results on.

The results of the model with a train/test split of 75/25 are shown in Table 2. Like expected, all the scores are slightly lower than the model with more training data, but it still performs quite well with an f1 score of 0.804.

label	f1-score	number	precision	recall	accuracy
realis	0.851	736	0.843	0.860	
non-realis	0.725	253	0.761	0.692	
overall	0.820	989	0.824	0.817	0.972

Table 1: Scores per event type on 10% test set by 90% model

label	f1-score	number	precision	recall	accuracy
realis	0.838	1878	0.852	0.824	
non-realis	0.711	674	0.733	0.690	
overall	0.804	2552	0.821	0.788	0.968

Table 2: Scores per event type on 25% test set by 75% model

4.3 DISCUSSION

For the amount of annotations we fine-tuned the model with, we are very satisfied with the result. The 90/10 model has an f1-score that is close to the inter-annotator score, which we assumed was the limit of performance on the model. The f1-score of the 75/25 model is only 0.016 lower than the 90/10 model. This shows that the model is able to acquire new functionality by fine-tuning on a relatively small dataset.

Acknowledging that the event detection task in the experiment by Sims et al. (2019) is slightly different, we can loosely compare the results to see that our model's performance is comparable to their best model. This model is a Bidirectional LSTM (Huang et al., 2015) trained on BERT word embeddings (Devlin et al., 2019) and achieved an f1-score of 0.755 on their test set.

Noticeably, the non-realis events have lower scores than the realis events. This is most likely due to the underrepresentation compared to the realis events. The overrepresentation of realis events has caused the models achieve a better fit on realis events, and the fit on non-realis is lacking. This means that when predicting events later on, the model will be less accurate when predicting non-realis events, and we should be mindful of this known flaw.

5 EVENTS & CANONICITY

To show one of the use cases of the Dutch event prediction model, we will look if there is a correlation between novel prestige and the use of events in the novel.

Using the canonicity metrics provided in the dataset by van Cranenburgh et al. (2022), we can define our own canonicity score. Since we care mostly about the text of the novel itself, and less about the author who wrote it, we decided to use the amount of secondary references to the novel as canonicity score. This is column `DBNLSecRefsTitle`. As the boxplot in Figure 2 shows, the majority of secondary references is quite low: the third quartile is at three references. The average is 2.83, but this number is not very representative of the dataset because of the many outliers, the highest of which is the novel Max Havelaar by Multatuli with a canonicity score of 204.

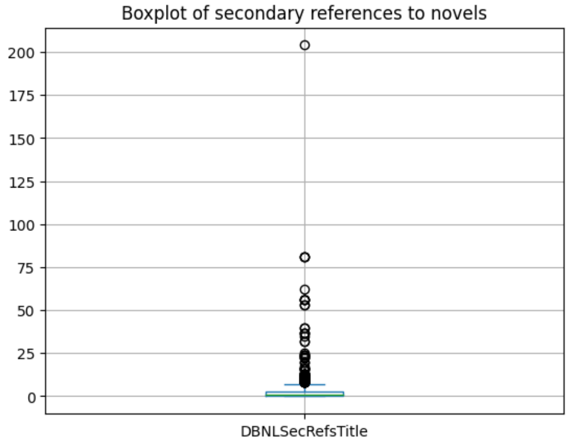

Figure 2: The event counts in all the annotations.

To determine what novels we would use for this experiment, we first filtered the novels on a time frame from 1850 up to, and including, 1950. This time frame roughly corresponds to the time frame in which the Openboek novels were published. This was to ensure that our model would not face unfamiliar (modern) texts to predict events on, thus resulting in inaccurate predictions. We also filtered out any novels that overlap with the Openboek Corpus, so the model would not encounter novels it was fine-tuned on.

To create the list of canonical novels, we took the novels that had a canonicity score higher than 3, and filtered out the outliers. We decided to select novels with a score of 3 or higher because this is where the third quartile of all filtered novels ends, so we select the top 25% of novels. We filtered out the outliers by selecting only the novels within on one standard deviation above and below the third quartile and first quartile respectively. We decided to only allow one author per category, to minimize the influence a single author had over the results. If more than one novel per author was present at this stage, we picked the novel with the highest canonicity. After filtering we were left with a dataset of 122 novels.

For the non-canonical set of novels we selected the same amount of novels as the canonical dataset, but in this set each novel had a canonicity score of 0. Again, one novel was allowed per author, but for the non-canonical dataset the novel for the author was randomized. To ensure reproducibility, we set a seed for the sampling of novels and for deciding which novel was selected per author.

We ran the best performing model on each of the novels, and performed several test. The first metrics we use are based on the research by Sims et al. (2019). We computed the density of the events by dividing the amount of events by the total amount of tokens, resulting in a token ratio. We also computed the distance events had from each other by dividing the amount of tokens by the amount of events. Since we created different annotations for realis and non-realis events, we can not only compute these metrics for all events, but also distinguish between realis and non-realis events. The last metric we will look at is the ratio of realis to non-realis events, which will give us another insight into how different novels use events. We compute this measure by dividing the amount of realis events by the amount of non-realis events.

Figure 3 shows the ratio (top row) and distance (bottom row) metrics for all events on the right hand side, and to the left of those we can see the metrics only for realis and non-realis events. Figure 4 shows the ratio of realis to non-realis events in novels. A clear visual difference can be observed for the non-realis events in Figure 3, and in Figure 4.

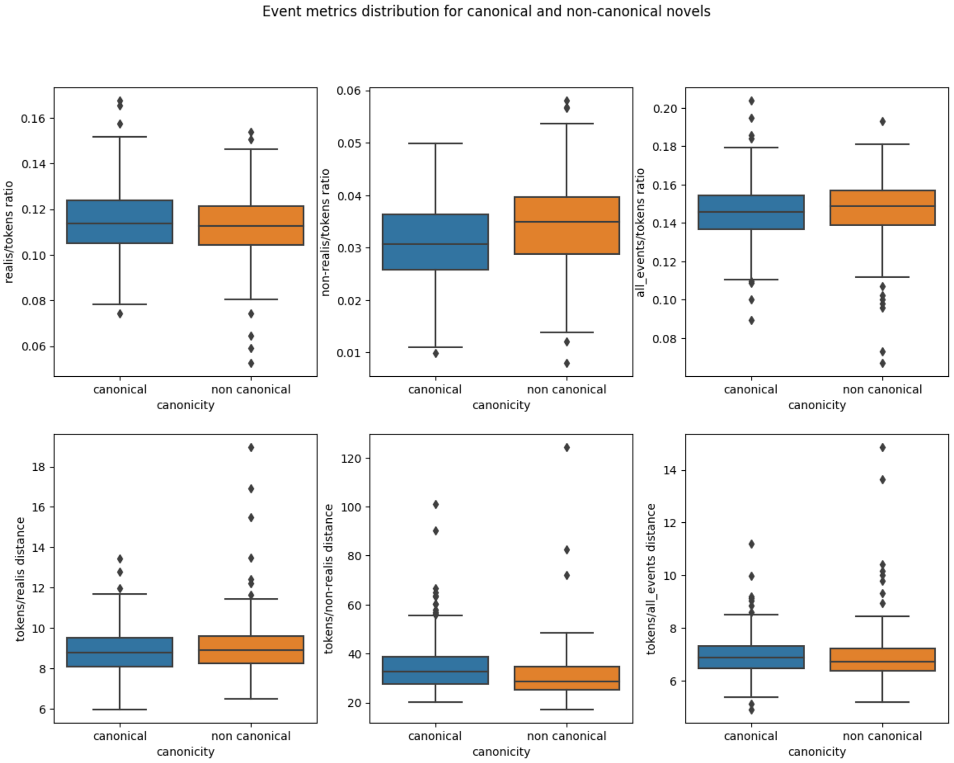

Figure 3: The ratio and distance metrics for all event types

To determine whether there is a correlation between the use of events and the canonicity, we used a point biserial correlation test (Kornbrot, 2014). This test is mathematically equivalent to the Pearson correlation test, and it allows us to compute the correlation between a continuous variable (the metric) and a binary variable (the canonicity). We chose to use point biserial correlation instead of biserial correlation since the data underlying the canonicity scores is not normally distributed. the canonicity score is skewed to the right, with the majority of novels having a low canonicity score. The results of running this test on every metric is shown in Table 3.

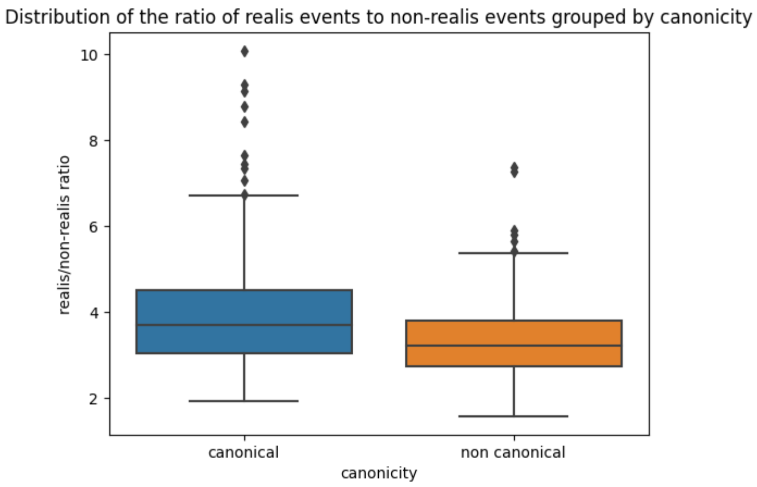

Figure 4: The distribution of the ratio of realis to non-realis events in novels grouped by canonicity

metric	correlation	p-value
realis/tokens ratio	0.1148	0.0735
non-realis/tokens ratio	-0.2354	0.0002
all_events/tokens ratio	-0.0115	0.8577
tokens/realis distance	-0.1198	0.0616
tokens/non-realis distance	0.1764	0.0057
tokens/all_events distance	0.0177	0.7836
realis/non-realis ratio	0.2556	0.0001

Table 3: Point biserial correlations between different metrics and canonicity

Using an α of 0.05, the only significant correlations are the non-realis/tokens ratio, the tokens/non-realis distance, and the realis/non-realis ratio. This means that the non-realis events are the only events that have a significant correlation with novel canonicity. All metrics suggest that the less non-realis events occur in a novel, the higher its canonicity.

We can go one step further and try to predict the novel canonicity score based on the metrics. For this we use a simple linear regression model. The results for fitting the model on all the different metrics are shown in Table 4.

Using an α of 0.05, the realis/token ratio model is significant, as well as the non-realis/tokens ratio model, the tokens/non-realis distance model, and the realis/non-realis ratio model. The latter is displayed in Figure 5. Here we can see, like the table suggests, that the higher the canonical score, the higher the realis/non-realis ratio. This means that less non-realis events get used as the canonicity score goes up. This is supported by the plot in Figure 6 that shows a descending slope in a plot of the regression line.

metric	slope	p-value
realis/tokens ratio	23.7923	0.0194
non-realis/tokens ratio	-61.6952	0.0009
all_events/tokens ratio	4.2757	0.6338
tokens/realis distance	-0.2395	0.0277
tokens/non-realis distance	0.0320	0.0114
tokens/all_events distance	0.1100	0.4561
realis/non-realis ratio	0.4848	0.0000

Table 4: Linear regression between different metrics and canonicity score

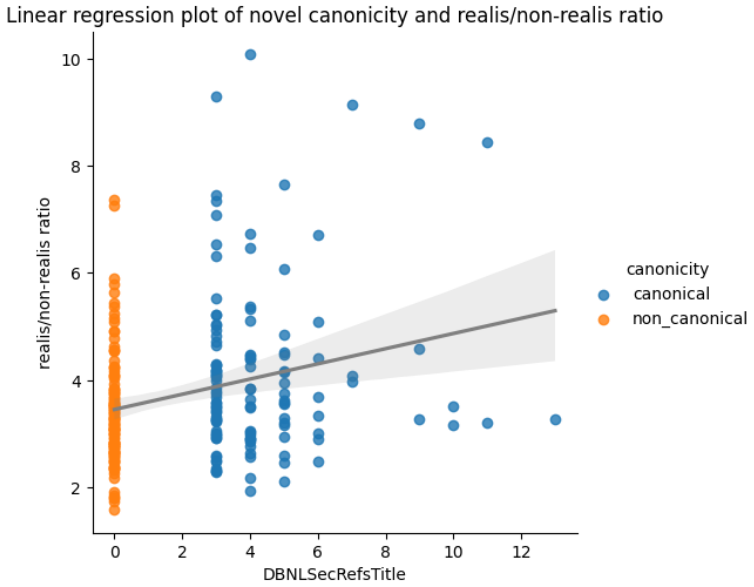

Figure 5: Linear regression plot of novel canonicity and realis/non-realis ratio

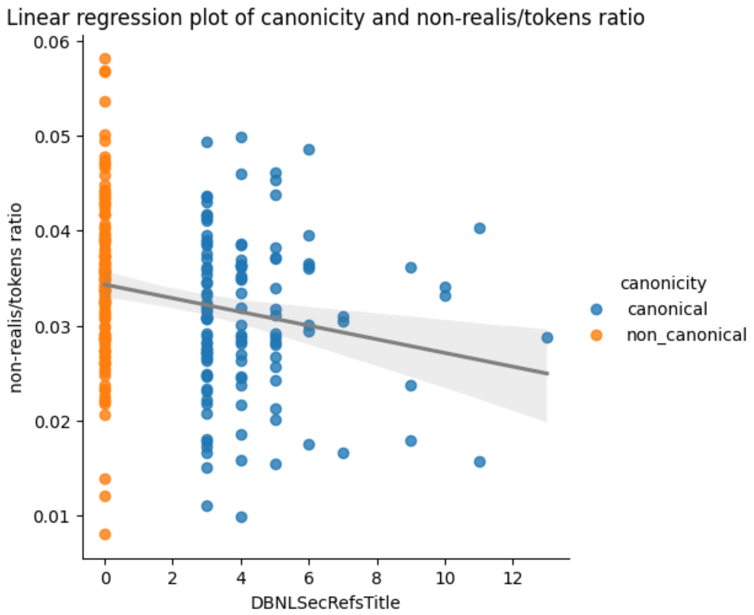

Figure 6: Linear regression plot of novel canonicity and non-realis/tokens ratio

While the p-value of these experiments suggest that these results are significant, we should always keep in mind that the model used to predict these events was not perfect. Because of the skewed dataset, the results for non-realis events were less accurate with a maximum f1-score of 0.725. While this is decent, we would need to increase the performance of this model to definitively conclude that there is a correlation between canonicity and non-realis events.

To increase the performance of the model we could, in the future, revise the guidelines to increase the inter-annotator agreement. Annotating more data will likely also result in better model performance, especially for non-realis events that do not have nearly as many examples.

Using event predictions of this model with increased performance we could then run this experiment again, and see if the outcome has changed. If the performance of this model is good enough, and the results point to a correlation once again, we can definitively conclude that there is a correlation between the use of events in a novel and its canonicity.

6 CONCLUSION

In conclusion, this paper investigated the correlation between prestige and the use of events in Dutch literature. We fine-tuned the BERTje language model for event detection in Dutch literary texts. The model achieved impressive results with an f1-score of 0.820 and 0.804 on two different train/test splits, showing its ability to detect events effectively. We can confirm our hypothesis that fine-tuning BERTje to achieve an f1-score exceeding 0.80 is possible.

Furthermore, we explored the relationship between novel canonicity and the occurrence of events. The analysis revealed that non-realis events, events that do not actually occur in the story, showed a significant correlation with novel canonicity. Contrary to our hypothesis, novels with higher canonicity scores tended to use fewer non-realis events, suggesting that the presence of such events may influence a novel's canonicity.

The results highlight the importance of events in literary narratives and their potential role in shaping a novel's canonical status. By using natural language processing techniques, we provided quantitative insights into previously qualitative questions about literary canonicity. The findings contribute to a better understanding of the factors that contribute to a novel being considered canonical.

Future work in the field of event prediction on Dutch literature should include more recent training data to ensure the model can predict events on recent novels. This can also be done by using the Oudeboeken spelling normalization tool before annotating. The annotation guidelines should also be updated and improved to achieve a better gold standard for the model. Introducing a new category of 'descriptive' verbs could make the model focus more on actual events by excluding descriptive main verbs. Training the model on whole novels instead of individual sentences opens up context for the model and might increase performance. Future work could also look into the reason as to why canonical novels use more realis events, since we were not able to explore this here.

Code to support this work can be found on the GitHub repository (Overbeek, 2023), together with a guide on how to use the code and reproduce the results.

BIBLIOGRAPHY

Rodrigo Agerri, Eneko Agirre, Itziar Aldabe, B.A.Z. Beloki, Egoitz Laparra, German Rigau, A. Soroa, and Ruben Urizar. 2014. Newsreader project. 53:155–158.

Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. 2014. A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 45–53, Baltimore, Maryland, USA. Association for Computational Linguistics.

Mark Andrew Algee-Hewitt, Sarah Allison, Marissa Gemma, Ryan Heuser, Franco Moretti, and Hannah Walser. 2016. Canon/archive : large-scale dynamics in the literary field.

Gosse Bouma, Gertjan van Noord, and Robert Malouf. 2001. Alpino: Wide-coverage computational analysis of dutch.

P. Brooks. 1992. *Reading for the Plot: Design and Intention in Narrative*. Harvard University Press.

Andreas van Cranenburgh and Gertjan van Noord. 2022. Openboek: A corpus of literary coreference and entities with an exploration of historical spelling normalization. *Computational Linguistics in the Netherlands Journal*, 12:235–251.

Andreas van Cranenburgh, Sara Veldhoen, and Michel De Gruijter. 2022. Textual features and metadata for DBNL novels 1800-2000.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan Noord, van, and Malvina Nissim. 2019. Bertje: A dutch bert model. *ArXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Diana Kornbrot. 2014. *Point Biserial Correlation*. John Wiley Sons, Ltd.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor

Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linguistic Data Consortium. 2005. ACE (Automatic Content Extraction) English annotation guidelines for events. `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf`.

Gertjan van Noord. 2023. Event prediction. `https://github.com/gertjanvannoord/oudeboeken`.

Walker Orr, Prasad Tadepalli, and Xiaoli Fern. 2018. Event detection with neural networks: A rigorous empirical evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 999–1004, Brussels, Belgium. Association for Computational Linguistics.

Björn Overbeek. 2023. Event prediction. `https://github.com/bbjoverbeek/event-prediction`.

Willie van Peer and Frank Hakemulder. 2006. Foregrounding. *The Pergamon Encyclopaedia of Language and Linguistics*.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3623–3634, Florence, Italy. Association for Computational Linguistics.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2022. Label Studio: Data labeling software. Open source software available from https://github.com/heartexlabs/label-studio.

T. Underwood. 2019. *Distant Horizons: Digital Evidence and Literary Change*. University of Chicago Press.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.