Prediction of Happy Endings in German Novels based on Sentiment Information

Albin Zehe, Martin Becker, Lena Hettinger, Andreas Hotho, Isabella Reger, and Fotis Jannidis

University of Würzburg, 97074 Würzburg, Germany
{zehe,becker,hettinger,hotho}@informatik.uni-wuerzburg.de,
{isabella.reger,fotis.jannidis}@uni-wuerzburg.de

Abstract Identifying plot structure in novels is a valuable step towards automatic processing of literary corpora. We present an approach to classify novels as either having a happy ending or not. To achieve this, we use features based on different sentiment lexica as input for an SVM-classifier, which yields an average F1-score of about 73%.

1 Introduction

Every child knows that stories are supposed to have a happy ending. Every adult knows that this is not always true. In fact, in the course of the 19th Century a happy ending became a sign of popular literature, while high literature was marked by a preference for the opposite. This makes happy endings an interesting point of research in the field of digital literary studies, since automatically recognizing a happy ending, as one major plot element, could help to better understand plot structures as a whole.

To achieve this, we need a representation of plot that a computer can work with. In digital literature studies, it has been proposed to use emotional arousal as a proxy [5]. But can we just use existing data mining methods in combination with sentiment features and expect good results for happy ending classification?

In this work, we tackle the problem of identifying novels with a "happy ending" by training a classifier. Our goal is not to present the best method for doing so, but to show that it is generally possible. We introduce our proposed approach, which already yields results considerably above a random baseline, and point out some problems and their possible solutions. Our method uses sentiment lexica in order to derive features with respect to semantic polarity and basic emotions. To account for the structural dynamics of happy endings, these features are built by considering the relation of different sections of the novels. We are able to train a support vector machine (SVM) which yields an average F1 score of 0.73 on a corpus with over 200 labelled German novels. To the best of our knowledge, our work is the first to cover happy ending classification.

The remainder of this paper is structured as follows: related work and background information is presented in Sections 2 and 3. The features and data we use are described in Sections 4 and 5. Then we present our results (Section 6).

In: P. Cellier, T. Charnois, A. Hotho, S. Matwin, M.-F. Moens, Y. Toussaint (Eds.): Proceedings of DMNLP, Workshop at ECML/PKDD, Riva del Garda, Italy, 2016.
Copyright © by the paper's authors. Copying only for private and academic purposes.

2 Related Work

Recently, a lot of attention has been paid to sentiment analysis in the Digital Humanities community. In this section we cover publications constructing features that are useful to our task, but have not actually been used to recognize happy endings.

Matthew Jockers proposed, in a series of blog posts, to use the "analysis of the sentiment markers" as *a novel method for detecting plot* [5, 7, 8]. The basic idea of representing plot by emotions was well received, but the following discussion showed his approach to use Fourier Transformation (FT) and a low-pass filter to smooth the resulting curves is not reasonable, since FT assumes periodicity of the signal [6, 14].

Elsner constructs a representation of the plot in a story using sentiment values, among other features, in [2]. He cites other works stating that sentiment is a very important part of plot development and is therefore critical to automatic understanding of plot.

Mohammad builds emotional representations like ours in [9]. In [1], similar representations are used to automatically compose music from written text.

In [3], Goyal et al. present AESOP, a system that can identify *plot units*. AESOP is partially based on affect states, which are closely related to sentiments.

3 Background

We refer to a novel as having a "happy ending", if the situation of the main characters in the novel improves towards the end of the story or is constantly favourable. In this paper, we propose a method for automatically predicting whether novels have a happy ending or not, based on features derived from sentiment analysis. We start by formally defining the task of "happy ending classification" and introduce some concepts of sentiment analysis which are relevant for our features.

Happy ending classification. We formally define "happy ending classification" as a simple classification task: Given a corpus C, we aim to learn a function $f : C \to \{0, 1\}$, where $f(c) = 1$ iff a novel $c \in C$ has a "happy ending". In this work, we use a support vector machine (SVM) to train and test the classification function f based on a labelled gold standard. The SVM model requires a feature vector for each novel (cf. Section 5). We mostly use sentiment based features as introduced in Section 4.

Sentiment analysis. Since plot construction, and in particular happy endings, are tightly coupled with sentiments [2], sentiment analysis provides a solid basis for our classification. The goal of sentiment analysis is to determine the *polarity* and *emotions* a human reader would associate with a given word, sentence or other element of a text. In this work, we focus on word-level sentiment analysis.

Polarity denotes if a word has a positive (e.g. friend) or negative (e.g. war) connotation. It can be expressed as a ternary value (-1, 0, or 1). A word can also be associated with a set of *basic emotions*. There are many definitions for basic emotions, as discussed in [10]. Plutchik et al. define a set of eight basic emotions in [12]: joy, trust, fear, surprise, sadness, disgust, anger and anticipation.

Generally, polarities and emotions are collected in sentiment lexica. Each lexicon contains a set of words which it associates with a number of *sentiment values* according to a set of dimensions (such as polarity or different emotions). In Section 4 we derive different features for each novel based on such a lexicon and in Section 5 we introduce the sentiment lexicon we use in our study.

4 Features

For "happy ending classification" we derive feature vectors based on a set of text *segments* which we combine to form *sections*. The final feature vectors are derived based on certain characteristic values of these segments and sections (e.g. the polarity of the final segment or the difference between the polarity of the first and the last section).

Negation detection. Our features are based on sentiments. However, sentiments can be negated (e.g. "not happy"). For considering negations, we apply the relatively simplistic technique presented in [11]: we add a negation marker to any word between a negation word and the following punctuation, inverting its sentiment score. Following the textblob implementation,[1] we multiply negated sentiments by 0.5, improving results slightly.

Segments. Given a corpus of novels $\mathcal{C}$, we first split each novel $C \in \mathcal{C}$ into n segments, $C = \{S_1, \ldots, S_n\}$. We evenly split by word count resulting in segments of size $\frac{\|C\|}{n}$, where $\|C\|$ denotes the number of words in novel C. Note that the last segment may be shorter due to the length of the novel and the number of segments.[2]

Sentiment values for segments. Now we derive a set of characteristic values for each segment. Given a fixed lexicon L with several dimensions (e.g., the polarity or an emotion), let $v_d(w)$ denote the value lexicon L associates with word w according to dimension d. For example the word "death" is strongly associated with the dimension "sadness", that is, $v_{\text{sadness}}(\text{"death"}) = 1$.

For each segment S_i and each dimension d in the lexicon L, we calculate the characteristic value $\bar{v}_d(S_i)$ as follows:

$$\bar{v}_d(S_i) = \frac{\sum\limits_{w \in L_i} v_d(w)}{|L_i|} \tag{1}$$

where L_i denotes the words in S_i which are covered by lexicon L.

[1] `https://pypi.python.org/pypi/textblob-de/`. The sentiment analysis in textblob is not fully ported to German and does not include basic emotions, so we did not use it directly.

[2] The novels were split into words using textblob-de. Words were lemmatized iff the lexicon used in the respective experiment contained lemmatized forms.

Sections. We merge consecutive segments into sections. We consider two prominent sections, *main* and *final*. The main section $\mathcal{S}_{\text{main}} = \{S_1, \ldots, S_m\}$ covers the majority (75 up to 98%, depending on the experimental setup) of the segments starting from the beginning. The final section $\mathcal{S}_{\text{final}} = \{S_{m+1}, \ldots, S_n\}$ covers the remaining segments and represents the "ending" of the novel. Additionally, we consider a third section, the *late-main* section $\mathcal{S}_{\text{late}} = \{S_{2m-n+1}, \ldots, S_m\}$, which covers the last part of $\mathcal{S}_{\text{main}}$. This section is introduced in order to better capture the sentiment development at the end of the novel. For example, there may be a catastrophic event shortly before the end which is then resolved, leading to a happy ending. Since, in our experiments, the late and the final section are always of the same length, all sections are defined by specifying the number of segments in the main section m.

Sentiment values for sections. For each of these sections, we calculate the characteristic value averages based on the covered segments. In particular, given the segments of a novel, $C = \{S_1, \ldots, S_n\}$, and a section $\mathcal{S}$, we calculate the average characteristic value by extending $\bar{v}_d$ to sections:

$$\bar{v}_d(\mathcal{S}) = \frac{\sum\limits_{S_i \in \mathcal{S}} \bar{v}_d(S_i)}{|\mathcal{S}|} \tag{2}$$

Features. Based on these characteristic values, we finally define the features for each novel. Given n segments, a main section of size m, and a lexicon L, the feature vector contains the following values for each dimension d: (1) the characteristic value of the final section $f_{d,\text{final}} = \bar{v}_d(\mathcal{S}_{\text{final}})$, (2) the characteristic value of the last segment $f_{d,\text{n}} = \bar{v}_d(S_n)$, (3) the difference between the main and the final section $f_{d,\text{main-final}} = \bar{v}_d(\mathcal{S}_{\text{main}}) - \bar{v}_d(\mathcal{S}_{\text{final}})$ and (4) the difference between the late-main and the final section $f_{d,\text{late-final}} = \bar{v}_d(\mathcal{S}_{\text{late}}) - \bar{v}_d(\mathcal{S}_{\text{final}})$. The change in sentiment values towards the end of the novel is characterized by the two differences. The difference was used to ensure that generally sad novels that had a significant improvement in the final segments can still be classified as having a happy ending or, in reverse, a drop in positive emotions towards the end of a generally happy novel can be recognized as a sad ending.

5 Dataset

In this section, we describe our annotated corpus, as well as the sentiment lexicon we derive our features from.

Annotated novels. Our dataset consists of 212 German novels compiled from the TextGrid Digital Library[3] and the Projekt Gutenberg[4], mostly written between 1750 and 1920. The number of words in the novels ranges from less than 20,000 words up to more than 300,000. These novels have been manually labelled

[3] `https://textgrid.de/digitale-bibliothek`
[4] `http://gutenberg.spiegel.de`

Table 1. Examples for entries in the NRC lexicon

Word	pos.	neg.	anger	antic.	disg.	fear	joy	sadn.	surp.	trust
Entführung	0	1	1	0	0	1	0	1	1	0
verachten	0	1	1	0	1	0	0	0	0	0
Bewunderung	1	0	0	0	0	0	1	0	0	1

by domain experts as either having a happy ending or not,[5] based on sources like the Kindler,[6] Wikipedia[7] or by reading relevant parts of the novel. Half of the novels (106) are annotated as having a happy ending. The annotated data can be made available upon request.

Sentiment lexica. In this work, we employ the German version of the NRC sentiment lexicon,[8] which is provided by the original author of the English version [10]. It encompasses the following semantic dimensions: if a word is positive (0 or 1) or negative (0 or 1), and if a word is associated with some basic emotion (each 0 or 1). We also add another dimension, i.e., the "polarity", which is the negative value subtracted from the positive value.

After removing duplicates and all-zero entries, which would not help in our task, the lexicon contains 4597 entries, as exemplified in Table 1. We also evaluated our approach on SentiWS [13] (polarity scores $\in \{-1, 0-1\}$) and GPC (German Polarity Clues) [15] (polarity scores $\in [-1, 1]$), however, achieving inferior results.

6 Results and Discussion

In this section we train a support vector machine (SVM)[9] for classifying happy endings of novels on the annotated corpus introduced in Section 5 using the features presented in Section 4. For the SVM we use an RBF kernel and the parameters $C = 1$ and $\gamma = 0.01$. A linear kernel gave slightly worse results, grid search for parameter selection did not lead to an improvement. We standardize the features using the sklearn StandardScaler[10] before classification. All tests were run with 10-fold cross-validation.

Baselines. Since our dataset is equally divided in novels with and without a happy ending, the random baseline as well as the ZeroR classifier (which assigns every novel to the largest class) reach 50% accuracy. Dropping the notion of sections, and only using the average sentiment values of the entire novel (i.e.,

[5] Because this is a simple task, each novel was labelled by only a single domain expert.
[6] `http://www.derkindler.de`
[7] `de.wikipedia.org`
[8] `http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm`
[9] We tried some other classifiers as well, with Random Forests and Naive Bayes reaching about the same score, while k-NN and Decision Trees performed worse.
[10] `http://scikit-learn.org/stable/modules/preprocessing.html`

Table 2. Results for an overall segment count of $n = 75$, a main section containing $m = 71$ segments, and the NRC lexicon averaged over 20 iterations, each using 10-fold cross validation.

happy ending	precision	recall	f1-score	support
False	0.72	0.73	0.73	106
True	0.73	0.72	0.73	106
avg/**total**	0.73	0.73	0.73	**212**

one value for each dimension in L) yields an F1 score of 0.54, which is slightly above the random baseline. Adding the scores of the last segment improves the F1 score to about 0.66 for the best performing segment count $n = 75$. This suggests that the final segment of the novel is indeed an important feature for classifying happy endings.

Best parameter configuration. Our method requires to choose a sentiment lexicon and the number of segments n, as well as the length of the main section m (in segments). We compared different configurations and found that working with the NRC as introduced in Section 5 using $n = 75$ segments with a main section of $m = 71$ segments worked best. Other lexica containing only polarity scores (cf. Section 5) performed worse, suggesting that the combination of basic emotions represents a more accurate picture of the overall mood in a novel than polarity alone. Table 2 shows the results with the best configuration, accumulated over 20 iterations.[11]

Influence of segmentation and section size. In this paragraph, we describe how changing the number of segments n and the percentage of segments assigned to the main section $\mathcal{S}_{\text{main}}$, that is $\frac{|\mathcal{S}_{\text{main}}|}{n}$, influences the results. Figure 1 shows the average F1-score over 20 test runs based on the NRC lexicon. Each line corresponds to a segment count n. From a larger set of segment counts we chose the 4 best performing ones. The x-axis corresponds to the percentage of segments in the main section. The y-axis shows the F1-score achieved with the respective configuration. It can be seen that splitting the novels into 50 segments mostly works very well, but is outperformed by 75 segments with a main section of 71 segments (about 95%). Furthermore, most segmentations perform best when using about 5% or 10% of the segments as the final section.

Limitations. Here, we list some limitations of our work and suggest possible solutions.

One variation we tried was to limit our features to sentences containing explicit references to the main character of the novel. The intuition behind this is that the concept of happy ending is closely related to the fate of protagonists of a story. Using a domain-adapted named entity recognition (NER) toolkit [4], we selected the main character as the one being explicitly named most often. After

[11] The results vary slightly between iterations, with total F1-scores mostly between 71% and 75%. Averaging over 20 iterations yields stable results.

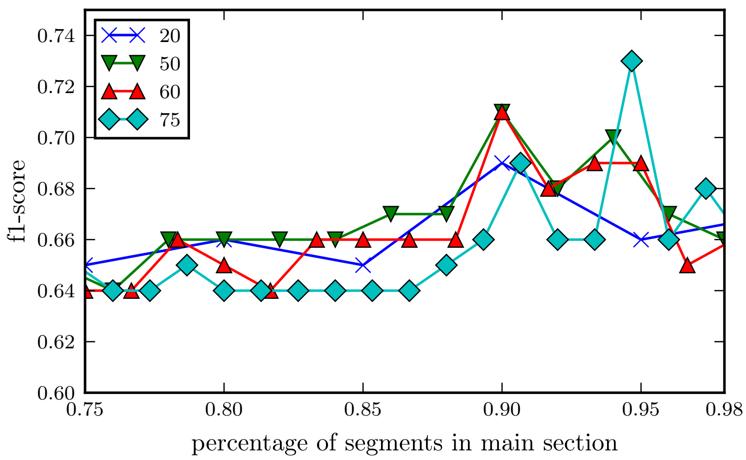

Figure 1. Plot of different segmentation configurations. Segment count n is represented as different lines, the x-axis corresponds to the number of segments in the main section m and the y-axis is the F1-score.

segmenting the novel, we then removed all sentences not mentioning this character. Contrary to our expectation, this did not improve results but indeed led to a significant drop in accuracy. The reason for this might be our decision to (for now) avoid error-prone co-reference resolution or our strict choice of focusing only on a single main character.

Employing more sophisticated sentiment analysis would likely improve our results. For example, while our relatively crude negation detection only led to slight improvements, considering a more advanced set of sentiment shifters should help to get better results.

We also did not take into account that some stories are not told in chronological order. Those stories are difficult to our system, as the happy ending may happen at some arbitrary point in the text. Working around this problem would require a way to identify corresponding scenes in different novels.

Finally, we are currently working on a way to choose the length of the main section individually for each novel, instead of passing it to the model as a fixed hyperparameter.

7 Conclusion

In this work, we have presented an SVM classifier for identifying novels with happy endings. Our approach is based on features derived from sentiment lexica and exploits structural dynamics by comparing different sections of the novels.

We consider the F1-score of 0.73 to be a good starting point for future work, such as evaluating our method on more extensive labelled datasets. Additionally, it is interesting to investigate if the same parameters yield good results for different novel collections, or different languages.

References

1. Davis, H., Mohammad, S.: Generating music from literature. In: Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL). pp. 1–10. Association for Computational Linguistics, Gothenburg, Sweden (April 2014)
2. Elsner, M.: Abstract representations of plot struture. LiLT (Linguistic Issues in Language Technology) 12 (2015)
3. Goyal, A., Riloff, E., Daumé III, H.: Automatically producing plot unit representations for narrative text. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 77–86. Association for Computational Linguistics (2010)
4. Jannidis, F., Krug, M., Puppe, F., Toepfer, M., Weimer, L., Reger, I.: Automatische Erkennung von Figuren in deutschsprachigen Romanen (2015)
5. Jockers, M.L.: A novel method for detecting plot (Jun 2014), `http://www.matthewjockers.net/2014/06/05/a-novel-method-for-detecting-plot/`
6. Jockers, M.L.: Requiem for a low pass filter (Apr 2015), `http://www.matthewjockers.net/2015/04/06/epilogue/`
7. Jockers, M.L.: The rest of the story (Feb 2015), `http://www.matthewjockers.net/2015/02/25/the-rest-of-the-story/`
8. Jockers, M.L.: Revealing sentiment and plot arcs with the syuzhet package (Feb 2015), `http://www.matthewjockers.net/2015/02/02/syuzhet/`
9. Mohammad, S.: From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. In: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 105–114. Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
10. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon 29(3), 436–465 (2013)
11. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 79–86. Association for Computational Linguistics (2002)
12. Plutchik, R.: A general psychoevolutionary theory of emotion. Theories of emotion 1, 3–31 (1980)
13. Remus, R., Quasthoff, U., Heyer, G.: Sentiws – a publicly available german-language resource for sentiment analysis. In: Proceedings of the 7th International Language Resources and Evaluation (LREC'10). pp. 1168–1171 (2010)
14. Schmidt, B.M.: Commodius vici of recirculation: the real problem with syuzhet (Apr 2015), `http://benschmidt.org/2015/04/03/commodius-vici-of-recirculation-the-real-problem-with-syuzhet/`
15. Waltinger, U.: Sentiment analysis reloaded: A comparative study on sentiment polarity identification combining machine learning and subjectivity features. In: Proceedings of the 6th International Conference on Web Information Systems and Technologies (WEBIST '10). Valencia, Spain (April 2010)