SCARE – The Sentiment Corpus of App Reviews with Fine-grained Annotations in German

Mario Sänger, Ulf Leser  
Department of Computer Science  
Humboldt-Universität zu Berlin  
Rudower Chaussee 25  
12489 Berlin, Germany  
{saengerm,leser}@informatik.hu-berlin.de

Steffen Kemmerer, Peter Adolphs  
Neofonie GmbH  
Robert-Koch-Platz 4  
10115 Berlin, Germany  
{steffen.kemmerer,peter.adolphs}@neofonie.de

Roman Klinger  
Institut für Maschinelle Sprachverarbeitung  
Universität Stuttgart  
Pfaffenwaldring 5b  
70569 Stuttgart, Germany  
klinger@ims.uni-stuttgart.de

Abstract

The automatic analysis of texts containing opinions of users about, e.g., products or political views has gained attention within the last decades. However, previous work on the task of analyzing user reviews about mobile applications in app stores is limited. Publicly available corpora do not exist, such that a comparison of different methods and models is difficult. We fill this gap by contributing the Sentiment Corpus of App Reviews (SCARE), which contains fine-grained annotations of application aspects, subjective (evaluative) phrases and relations between both. This corpus consists of 1,760 annotated application reviews from the Google Play Store with 2,487 aspects and 3,959 subjective phrases. We describe the process and methodology how the corpus was created. The Fleiss-κ between four annotators reveals an agreement of 0.72. We provide a strong baseline with a linear-chain conditional random field or word-embedding features with a performance of 0.62 for aspect detection and 0.63 for the extraction of subjective phrases. The corpus is available to the research community to support the development of sentiment analysis methods on mobile application reviews.

Keywords: sentiment analysis, app reviews, German corpus, conditional random field, word embedding

1. Introduction

Mobile devices, such as smartphones and tablets, are widespread in our society. Applications for these devices (also known as apps) become increasingly popular and gain a lot of attention in our daily lives. Applications are typically downloaded via app stores, i.e., application distribution platforms, such as the Apple App Store\(^1\), Google Play Store\(^2\), BlackBerry World\(^3\) or the Windows Store\(^4\). These platforms offer their users the functionality to assess applications with a 5-star rating and a textual review. An example for such review is depicted in Figure 1.

These reviews form a rich resource of information for app developers, since they hold the user’s opinions about the application itself and important aspects, like design and usability. Moreover the reviews often contain complaints about problems and errors of the application as well as desired feature requests. Incorporating this feedback in the development may have influence on the success of the app. However, one challenge for the developers is to deal with the overwhelming amount of reviews. Applications can have hundreds of thousands or even millions of reviews. A manual inspection and analysis of all these reviews is very time consuming and impractical. The app stores themselves offer only basic analysis capabilities.

The analysis of opinions in reviews has been widely investigated within the last decade (Pang and Lee, 2008) and is typically referred to as sentiment analysis or opinion mining (Liu, 2012). However, previous work in the area of analyzing user reviews in app stores is limited. Unlike product reviews of other domains, e.g. household appliances, consumer electronics or movies, application reviews offer a couple of peculiarities which deserve special treatment: The way in which users express their opinion in app reviews is shorter and more concise than in other product reviews. Moreover, due to the frequent use of colloquial words and a flexible use of grammar, app reviews can be considered to be more similiar to Twitter messages (“Tweets”) than reviews of products from other domains or platforms like Amazon. To the best of our knowledge, no corpus of annotated application reviews is available to the research community. Therefore, the comparison of existing methods is difficult. With this paper, we contribute to this situation: We publish the first annotation with fine-grained sentiment information (i.e., evaluating subjective phrases, aspects, and their relation) of German mobile app reviews from the Google Play Store. The corpus is available for future research\(^5\).

2. Previous Work

A plethora of approaches for opinion mining has been proposed in the last decades. Approaches employ rules (Reckman et al., 2013) or dictionaries (Waltinger, 2010b) to extract or are based on statistically trained, supervised models (Klinger and Cimiano, 2013b; Li et al., 2010) or incorporate weakly supervised machine learning techniques (Titov and McDonald, 2008; Täckström and McDonald, 2011). Many of the existing approaches are focused on the study of prod-

---

2https://play.google.com/store/
3https://appworld.blackberry.com/webstore/
5The corpus and further information, including the complete app list and annotation guidelines, are available at http://www.romanklinger.de/scare/
3.2. Corpus Selection

We select eleven application categories, which represent typical use-cases of mobile applications. The categories are instant messengers, fitness tracker, social network platforms, games, news applications, alarm clocks, navigation and map applications, office tools, weather apps, sport news and music players. We further choose 10–15 widely-used applications from each category, leading to 148 applications in total. A complete list of these applications and categories is available at the corpus website.6

3.1. Corpus Selection

Table 1: Overview of existing work on app store review mining and analysis. For each approach the number of applications and reviews used as well as the app store (Apple App Store (A), Google Play Store (G) or BlackBerry World (B)) they originate from are given. All approaches use English language reviews.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Store</th>
<th>#Apps</th>
<th>#Reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vasa et al. (2012)</td>
<td>A</td>
<td>17,330</td>
<td>8,701,198</td>
</tr>
<tr>
<td>Harman et al. (2012)</td>
<td>B</td>
<td>32,108</td>
<td>—</td>
</tr>
<tr>
<td>Iacob and Harrison (2013)</td>
<td>B</td>
<td>270</td>
<td>137,000</td>
</tr>
<tr>
<td>Khalid (2013)</td>
<td>A</td>
<td>20</td>
<td>6,390</td>
</tr>
<tr>
<td>Galvis Carreno et al. (2013)</td>
<td>G</td>
<td>3</td>
<td>327</td>
</tr>
<tr>
<td>Fu et al. (2013)</td>
<td>G</td>
<td>171,493</td>
<td>13,286,706</td>
</tr>
<tr>
<td>Pagano and Maalej (2013)</td>
<td>A</td>
<td>1,100</td>
<td>1,100,000</td>
</tr>
<tr>
<td>Chen et al. (2014)</td>
<td>G</td>
<td>4</td>
<td>241,656</td>
</tr>
<tr>
<td>Khalid et al. (2014)</td>
<td>A</td>
<td>20</td>
<td>6,390</td>
</tr>
<tr>
<td>Guzman and Maalej (2014)</td>
<td>A,G</td>
<td>7</td>
<td>32,210</td>
</tr>
<tr>
<td>Vu et al. (2015)</td>
<td>G</td>
<td>95</td>
<td>2,106,605</td>
</tr>
<tr>
<td>Martin et al. (2015)</td>
<td>B</td>
<td>15,095</td>
<td>2,729,103</td>
</tr>
<tr>
<td>Maalej and Nabil (2015)</td>
<td>A,G</td>
<td>1,140</td>
<td>1,303,182</td>
</tr>
</tbody>
</table>

Further research in this area includes topic (Galvis Carreno and Winbladh, 2013) and feature detection (Guzman and Maalej, 2014), keyword extraction (Vu et al., 2015), and review impact analysis (Pagano and Maalej, 2013). Table 1 provides an overview of app store mining approaches and the review corpora used within them.

For fine-grained sentiment analysis and opinion mining in other domains than app reviews, a plethora of manually annotated corpora is available (Lakkaraju et al., 2011; Nakov et al., 2013; Wiebe et al., 2005, for instance). Hu and Liu (2004) constructed a corpus of Amazon reviews annotated with aspects, subjective phrases and a polarity score for each sentence. Spina et al. (2012) provide a data set containing 9,396 Tweets annotated with offsets for aspect mentions (of predefined categories) and evaluative phrases. Annotated blog posts about cars and cameras are the content of the JDPA sentiment corpus from Kessler et al. (2010).

A few corpora exist for other languages than English. Examples are a corpus of sentences from German web texts with subjectivity and polarity annotations on sentence, phrase and word level (Clematide et al., 2012). Klinger and Cimiano (2014) published the USAGE corpus containing the annotation of product aspects and evaluative phrases in German and English Amazon reviews.

In the following, we present the Sentiment Corpus of App Reviews (SCARE) consisting of mobile application reviews annotated with aspects, subjective (evaluating) phrases, polarities and their relation.

Figure 1: Example review for the app MAPS.Me (personally identifiable information blurred for depiction).

http://www.romanklinger.de/scare

6.5

Einer wirklich tolle App! Sie ist einfach zu bedienen, die Karten werden schnell dargestellt und durch die Nutzung von Open Street Maps sind die meisten Karten sehr aktuell.

3. A Corpus for Fine-grained Sentiment Analysis of Mobile Application Reviews

3.1. Corpus Selection

We select eleven application categories, which represent typical use-cases of mobile applications. The categories are instant messengers, fitness tracker, social network platforms, games, news applications, alarm clocks, navigation and map applications, office tools, weather apps, sport news and music players. We further choose 10–15 widely-used applications from each category, leading to 148 applications in total. A complete list of these applications and categories is available at the corpus website.6

http://www.romanklinger.de/scare
We use the Android Market API\textsuperscript{7}, a programming interface for the Google Play Store, to retrieve reviews of mobile applications. This API provides a sub-sample with up to 5,000 reviews per application which contains both the latest and older reviews of an application. Repeating requests allows for collecting more reviews, if available. We retrieved all reviews for the selected list of applications in the time period of December 2014 to June 2015 leading to over 800,000 German app reviews.

### 3.2. Annotation Guidelines

We distinguish the classes aspect and subjective (evaluative) phrase. An aspect is part of an app or related to it, e.g. separate features of the application, usability, design, price, required authorizations or displayed advertisement. Additionally, we regard the whole application itself as well as errors and feature requests as aspects. Beyond that, we annotate the relationship of the aspect to the main application discussed in the review. Aspects which refer to an app or an aspect of an app other than the app in discussion are marked as “foreign”. This is often the case in cross-application comparisons. All other aspects are “related”.

Subjective phrases express opinions and statements of a personal evaluation regarding the app or a part of it, that are not based on (objective) facts but on individual opinions of the reviewers. Each subjective phrase is assigned a polarity (positive, negative, neutral) and may have a set of aspects it refers to.

The following sentences (with aspects and subjective phrases) illustrate the entity classes and annotation guidelines:

- **Sehr gute und übersichtliche App**.
  (Very good and well-arranged app.)
  - App is a target of Sehr gute and übersichtliche. Both evaluations are positive.

- **Die Verbindungsanzeige funktioniert nicht**.
  (The connection indicator does not work.)
  - Verbindungsanzeige is a target of funktioniert nicht, which represents a negative evaluation.

- **Die App ist cool, aber das Design ist schrecklich**.
  (The app is cool, but the design is terrible.)
  - **cool** is a positive evaluation of App, **schrecklich** represents a negative opinion for Design.

In addition, the annotators were instructed to annotate aspects and subjective phrases as fine-grained as possible and to avoid overlapping annotations. The annotations should be as short as possible, as long as the meaning is understandable if only the annotations were given (without the sentence itself).

We performed a two-step annotation process. Firstly, the actual annotation is performed by the annotator. Secondly, the annotator checks and improve the annotations created by himself in the first step and examines the review text for more aspects, subjective phrases and relations between them. The complete guidelines are available on the corpus website.

### 3.3. Annotation Process

Annotation was performed by four annotators using the program Brat\textsuperscript{8} in version v1.3 (Stenetorp et al., 2012). One of the annotators is an author of this paper. The group was composed exclusively of men aged 25 to 35 years. The training of the four annotators and optimization of the guidelines has been conducted in three iterations. To quantify the inter-annotator agreement, Fleiss’ $\kappa$ has been measured (Fleiss, 1981). In each iteration, 20 reviews were randomly sampled from the complete review corpus and given to the annotators.

The agreement between the annotators reached a $\kappa$-value of 0.57 (on token level using an in-out classification scheme) in the first annotation round. An analysis of the pairwise results showed that three of the annotators had a relatively high agreement (average $\kappa$-value of 0.66). The agreement between the three annotators and the fourth was comparably low ($\kappa$-value of 0.48). Within a meeting with all four annotators, problems and ambiguities in the annotations of the first round were discussed. In the second iteration, an agreement of 0.76 has been achieved. The differences between individual annotators did not reoccur in this iteration. To confirm the result a third iteration was carried out, which revealed an agreement of 0.72.

After completion of this training phrase, the actual annotation was performed. The annotation took place in June and July 2015 over a period of four weeks. For each application category, we randomly sampled 160 reviews from the complete corpus and gave them to the four annotators. The distribution of reviews was designed such that $\approx$20% of the reviews of each category were annotated by two annotators. This enabled us to monitor the development of the agreement. Each annotator worked on 36 reviews per category. Further, one annotator (one of the authors of this paper) annotated another 52 reviews per category. To build the final corpus, we harmonized the annotations from reviews which were annotated by more than one annotator by considering all identical annotations as well as the intersection of all overlapping but not completely identical annotations, if the meaning was still understandable in a manual process.

### 4. Analysis

In the following, we provide an overview of the corpus. Furthermore, we provide a prediction baseline for future models to be evaluated using SCARE.

#### 4.1. Collected App Reviews

As mentioned in Section 3.1., the overall corpus consists of 800,000 German app reviews, retrieved in six months. The number of reviews varies greatly between the individual application categories. Only 20,000 reviews were collected for office tools and alarm clocks. In contrast, over 150,000 reviews were retrieved for games and instant messengers. These differences result from different degrees of popularity:

\[\text{http://brat.nlplab.org/}\]
Table 2: Statistics of the full corpora as well as separated into different app categories. In total, the corpus contains 6,446 entities (aspects and subjective phrases) and 1,969 relations between them. The provided average pairwise $F_1$ measures refer to exact matches between the annotations of two annotators.

For instance, the instant messenger Threema is installed on one to five million devices compared to 500,000 to 1,000,000 installations of Smart Office 2 (office tools, numbers as of February 27, 2016).9

The collected reviews are very short in contrast to other product domains (Pollach, 2006) with an average length of 17 tokens. Negative reviews, i.e., reviews with a 1 or 2 star rating, are generally longer with 25 tokens on average than positive ones (4 or 5 stars, 13 tokens on average).

The average star rating of the reviews is 3.75 (on a scale between 1 and 5). The average values for the different categories vary from 3.46 (instant messenger) to 4.22 (alarm clocks). Nearly 70% of all reviews have a minimal or maximal star rating. In contrast, ratings with three stars, which supposedly reflect a neutral or mixed assessment, are rare (about 8%). This distribution pattern is common in user reviews and has as well been observed in other product domains (Filatova, 2012).

4.2. Annotations

Table 2 summarizes the annotated corpus. The corpus consists of 1,760 annotated application reviews. On average, a review contains 3.66 entities. The number of aspects is in total 2,487. The most frequent aspects are referring to the whole application itself. Less frequent aspects are often more specific. The majority of the aspects are directly related to the application in discussion. Only 130 aspects were marked as “foreign”. There are 3,959 subjective phrases from which 2,463 are positive, 1,433 are negative, and 63 are neutral. Subjective phrases are, with an average length of 1.83 tokens per phrase, longer than the annotated aspects (avg. 1.24 tokens). The most frequent subjective expressions are “Super”, “Top”, “Gut” (good), “Sehr gut” (very good) and “cool”.

The number of subjective phrases within the different application categories is relatively constant. In contrast, the number of application aspects (from 149 to 287) and relations (104 to 237) varies much more. Especially reviews for social network applications contain less application aspects (149) and relations (104) than reviews of other categories. This indicates that reviews of this category are more often simple appraisals which do not outline detailed information about what features or properties of the application they concerned about, e.g. “Sehr gut” (Very good) or “Einfach nur lächerlich” (Just ridiculous).

The inter-annotator agreement for the final corpus is the same as during the training phase with a Fleiss $\kappa$ of 0.72. The highest agreement holds within weather and fitness apps. The annotation of games and music players shows the lowest agreement. Reviews from social networks contain less application aspects, more slang words and are generally more often plain praising or blaming of the application, e.g. “Facebook sucks!” or “Ich liebe instagram!!” (I love insta-
We use the following features to capture the characteristics of aspects and subjective phrases in our IOB (inside, outside, begin) sequence prediction setting (inspired by previous work (Klinger and Cimiano, 2013a)).

We perform two experiments: 10-fold cross-validation on the full corpus including all reviews from all application categories. Cross-validation is performed on the document level (not on sentence-level) to ensure that no characteristics of one review is shared between the respective training and validation set.

2. Cross-category validation: training on the reviews from all but one application category and test on reviews of the hold-out category. This setup is performed for each application category. The goal of this evaluation is to get insights on how homogeneously opinions and application aspects are expressed within different categories and how easy a model trained on app reviews of certain categories can be transferred to reviews of a new application category.

Table 3: Overview of the used polarity lexicons. The first six are previously published, general-purpose polarity lexicons. The last two are domain-specific lexicons build on the collected app reviews. The figures represent the actual determined word numbers from the freely available files.

<table>
<thead>
<tr>
<th>Lexicon/Authors</th>
<th>#Pos</th>
<th>#Neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>SentiWS (Remus et al., 2010)</td>
<td>13,910</td>
<td>13,825</td>
</tr>
<tr>
<td>GermanLex (Clematide and Klenner, 2010)</td>
<td>2,812</td>
<td>4,677</td>
</tr>
<tr>
<td>German Sentiv Spin (Waltinger, 2010b)</td>
<td>42,276</td>
<td>63,284</td>
</tr>
<tr>
<td>German Subj. Clues (Waltinger, 2010a)</td>
<td>3,336</td>
<td>5,742</td>
</tr>
<tr>
<td>German Polarity Clues (Waltinger, 2010a)</td>
<td>2,994</td>
<td>5,749</td>
</tr>
<tr>
<td>German Polarity Clues (Waltinger, 2010a)</td>
<td>17,627</td>
<td>19,962</td>
</tr>
<tr>
<td>App-Domain-1vs5 (AD-1vs5)</td>
<td>1,101</td>
<td>326</td>
</tr>
<tr>
<td>App-Domain-12vs45 (AD-12vs45)</td>
<td>3,449</td>
<td>975</td>
</tr>
</tbody>
</table>

Figure 2: Example for feature extraction based on word embeddings. The grey leaf corresponds to the closest cluster to the word for which features are extracted. Features for the path from the root are therefore left, left-right, left-right-left, and cluster-id=3.

gram), than reviews of other categories. On the contrary, reviews for weather and fitness apps consist of clear and recognizable expressions of sentiments, application aspects and links between both. In particular, many reviews of the two categories are focus on the accuracy (weather forecast or tracked running route) of the application.

Pairwise $F_1$ scores can serve as an upper bound for automatic analysis tools: If the agreement between two human annotators is lower than between a tool and a human, the result should be interpreted critically. We observe a higher agreement on negative contexts and 5 resp. 4 and 5 star ratings as positive contexts.

4.3. Prediction Baseline

In the following, we provide baseline results on SCARE, achieved with a linear-chain conditional random field model (CRF, Lafferty et al. (2001)). We use the MALLET toolkit\textsuperscript{10} for implementation.

We use the following features to capture the characteristics of aspects and subjective phrases in our IOB (inside, outside, begin) sequence prediction setting (inspired by previous work (Klinger and Cimiano, 2013a)):

- **Token-based features:** Each token is represented by a set of features, i.e., the token itself, its POS tag, the combination of the token and the POS tag, the token in lower case letters as well as whether the token contains numbers or non-ASCII characters. In addition, we check whether the token is an emoticon, smiley or negation word based on manually created lists\textsuperscript{11}.

- **Polarity lexicons:** We use eight polarity lexicons to detect positive or negative words, each leading to a separate feature (cf. Table 3). Two out of these eight are domain-specific dictionaries we compiled based on their pointwise mutual information with respect to the star-rating\textsuperscript{12}. This procedure has been performed on the full crawled corpus, but without taking advantage of the annotations. Therefore, for unseen data, these dictionaries can be adapted analogously.

- **Word embeddings:** To capture characteristics of infrequent terms (like typos or slang words), we opted for the creation of word embeddings-based features (Turney and Pantel, 2010). For each token, all other tokens with a cosine-similarity greater than 0.8 are added as features. In addition, the index of the most similar cluster center of a hierarchical clustering is added as well as the full path and all (path) prefixes in the cluster hierarchy. An example for this procedure is shown in Figure 2.

We use the CBOW model of Word2Vec\textsuperscript{13} (Mikolov et al., 2013a; Mikolov et al., 2013b) to estimate word embeddings with a context size of 5 on the complete corpus of collected app reviews. We omit tokens with less than 10 occurrences. All other parameters of the model are set to the default values.

- **Context features:** To capture the context of a token, all features of tokens in a left and right-window of 2 are taken into account.

We perform two experiments:

1. **10-fold cross-validation on the full corpus** including all reviews from all application categories. Cross-validation is performed on the document level (not on sentence-level) to ensure that no characteristics of one review is shared between the respective training and validation set.

2. **Cross-category validation:** training on the reviews from all but one application category and test on reviews of the hold-out category. This setup is performed for each application category. The goal of this evaluation is to get insights on how homogeneously opinions and application aspects are expressed within different categories and how easy a model trained on app reviews of certain categories can be transferred to reviews of a new application category.

\textsuperscript{10}http://mallet.cs.umass.edu/

\textsuperscript{11}The created lists can be found on our corpus website at http://www.romanklinger.de/scare/

\textsuperscript{12}For AD-1vs5 and AD-12vs45 we treated 1 resp. 1 and 2 star ratings as negative contexts and 5 resp. 4 and 5 star ratings as positive contexts.

\textsuperscript{13}https://code.google.com/archive/p/word2vec/
<table>
<thead>
<tr>
<th>Cross-category validation</th>
<th>Aspects</th>
<th>Subjective phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exact</td>
<td>Partial</td>
</tr>
<tr>
<td></td>
<td>$P$</td>
<td>$R$</td>
</tr>
<tr>
<td>Alarm Clocks</td>
<td>0.69</td>
<td>0.56</td>
</tr>
<tr>
<td>Fitness Tracker</td>
<td>0.74</td>
<td>0.58</td>
</tr>
<tr>
<td>Games</td>
<td>0.73</td>
<td>0.59</td>
</tr>
<tr>
<td>Instant Messenger</td>
<td>0.68</td>
<td>0.57</td>
</tr>
<tr>
<td>Navigation / Maps</td>
<td>0.65</td>
<td>0.51</td>
</tr>
<tr>
<td>News Apps</td>
<td>0.65</td>
<td>0.58</td>
</tr>
<tr>
<td>Music Player</td>
<td>0.67</td>
<td>0.58</td>
</tr>
<tr>
<td>Office Tools</td>
<td>0.69</td>
<td>0.53</td>
</tr>
<tr>
<td>Social Networks</td>
<td>0.67</td>
<td>0.57</td>
</tr>
<tr>
<td>Sport news</td>
<td>0.67</td>
<td>0.53</td>
</tr>
<tr>
<td>Weather Apps</td>
<td>0.66</td>
<td>0.55</td>
</tr>
<tr>
<td>10-fold cross-val.</td>
<td>0.69</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Table 4: Evaluation results reached by our CRF-based model. “10-fold cross-val.” refers to a 10-fold cross-validation experiment on the full corpus. In contrast, the “cross-category validations” correspond to a evaluation setting in which the model is trained on all reviews except for the application category indicated in the table. The reviews of the ‘left-out’ category are used for testing, the results of which are shown in the table. We further distinguish aspect and subjective phrase prediction as well as exact and partial matches for each experiment.

The evaluation results are shown in Table 4.3. We report precision, recall and $F_1$ measures and further distinguish exact and partial matches between prediction and annotation. In exact mode the predicted text spans of aspects and subjective phrases must exactly match those of the gold standard. A partial match true positive holds if gold and prediction overlap by at least one token.

The results of the 10-fold cross-validation experiment are similar to the figures of other sentiment analysis systems on reviews of other product domains (Klinger and Cimiano, 2014). Considering only exact matches the model achieves similar values on the extraction of application aspects ($F_1$ score of 0.62) and subjective evaluations (0.63). Taking partial matches into account, the model reaches higher results on the detection of subjective expressions (0.80) than aspects (0.69). Precision is higher than recall throughout the experiments. In order to achieve a better understanding and comparability of the results, we further run the model described in Klinger and Cimiano (2013b), expanded to include the polarity lexicons and word embeddings of our model, on SCARE. The model from Klinger and Cimiano (2013b) achieves a slightly lower performance ($F_1$ score of 0.67 on the aspect extraction and 0.78 on subjective evaluations regarding partial matches) on SCARE than the CRF-based model.

The results of the cross-category experiment show relatively homogeneous performance for aspect detection and extraction of subjective phrases. Therefore, we expect a good adaptability to novel domains, unseen at the time of estimating the model.

5. Summary and Conclusion

We present, to the best of our knowledge, the first manually annotated resource for fine-grained sentiment analysis of German mobile application reviews. The reviews are annotated with aspects, evaluative (subjective) phrases and relations between them. The corpus consists of 1,760 annotated application reviews containing 2,487 aspects and 3,959 subjective phrases. During annotation we could reach an inter-annotator agreement of 0.72 (Fleiss’ $\kappa$). We further provide a strong prediction baseline by applying a CRF-based model on the corpus resulting in an $F_1$ score of 0.62 for aspect detection and 0.63 for the extraction of subjective phrases.

During the construction of the corpus, we further collected a data set of over 800,000 reviews from apps of 11 different application categories, which is (as far as we known) the first German corpus in this domain available. These data will motivate and enable an array of novel research questions to be investigated and foster the development of sentiment analysis methods on mobile application reviews and, in general, on German text.

Acknowledgments

We thank Heiko Ehrig, Hilmi Yildirim and Abdoulaye Dramé for their annotation work and feedback during the optimization of the annotation guidelines.

6. Bibliographical References


Klinger, R. and Cimiano, P. (2013b). Joint and pipeline probabilistic models for fine-grained sentiment analysis: Extracting aspects, subjective phrases and their relations. In IEEE International Conference on Data Mining Workshops (ICDMW), Dallas, TX, USA.


In Proceedings of the 12th IEEE Working Conference on Mining Software Repositories, pages 123–133. IEEE.


