Fine-grained Opinion Mining from Mobile App Reviews with Word Embedding Features

Mario Sänger[1], Ulf Leser[1], and Roman Klinger[2]

[1] Department of Computer Science, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
{saengerm, leser}@informatik.hu-berlin.de
[2] Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Pfaffenwaldring 5 b, 70569 Stuttgart, Germany
klinger@ims.uni-stuttgart.de

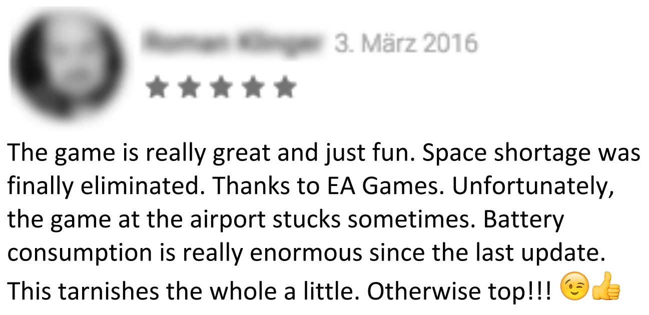
Abstract. Existing approaches for opinion mining mainly focus on reviews from Amazon, domain-specific review websites or social media. Little efforts have been spent on fine-grained analysis of opinions in review texts from mobile smart phone applications. In this paper, we propose an aspect and subjective phrase extraction model for German reviews from the Google Play store. We analyze the impact of different features, including domain-specific word embeddings. Our best model configuration shows a performance of 0.63 F_1 for aspects and 0.62 F_1 for subjective phrases. Further, we perform cross-domain experiments: A model trained on Amazon reviews and tested on app reviews achieves lower performance (drop by 27 percentage points for aspects and 15 percentage points for subjective phrases). The results indicate that there are strong differences in the way personal opinions on product aspects are expressed in the particular domains.

Keywords: Sentiment Analysis, Reviews, German, App Reviews, Opinion Mining

1 Introduction

The analysis of sentiment expressions and opinions in text gained a lot of attention within the last decade [23]. Studied types of texts include product reviews, Twitter messages or blog posts [36]. The analysis of mobile applications (also known as *apps*) and their user reviews in app stores, such as the Apple App Store[3], Google Play Store[4], BlackBerry World[5] or Windows Store[6], has only gained very limited attention so far. However, app reviews offer interesting characteristics which

[3] https://itunes.apple.com/us/genre/ios/id36?mt=8
[4] https://play.google.com/store/
[5] https://appworld.blackberry.com/webstore/
[6] https://www.microsoft.com/en-us/windows/apps-and-games


[redacted] 3. März 2016

★★★★★

The game is really great and just fun. Space shortage was finally eliminated. Thanks to EA Games. Unfortunately, the game at the airport stucks sometimes. Battery consumption is really enormous since the last update. This tarnishes the whole a little. Otherwise top!!! 😉👍

Fig. 1. Example of a user review for a mobile application. The review contains useful information and feedback for app developers, *e.g.* that game in general is *just fun*. However, the *battery consumption*, which is an aspect of the application, is *really enormous*.

deserve special investigation: On the one side, they share properties with Tweets and other social media texts, *e.g.*, comparably short and informal language [6]. On the other side they are similar to product reviews from other domains or platforms, *e.g.*, reviews about household appliances, consumer electronics or books on Amazon, as they typically describe the user's opinion about specific aspects. In the example user review in Figure 1, the task would be to detect for instance the aspects "game" with the evaluation "great" and "fun". It also highlights that the aspect "battery consumption" is evaluated negatively, as the word "enormous" indicates.

The analysis of app reviews is also interesting from a commercial point of view. The reviews form a rich resource of information, since they hold the user's opinions about the application. Moreover the reviews often contain complaints about problems and errors of the app as well as mentions of desired features. Incorporating this feedback into the developement process can have a huge influence on the success of the application [22]. However, the overwhelming amount of user reviews challenges app developers. An application can get hundreds or thousands of reviews each day, which make a manual inspection and analysis very time consuming and impractical. An automated analysis of the reviews would be benefical for app users as well since this would enable them to analyze the advantages and disadvantages of one or multiple applications more easily. For example, they could compare two fitness trackers according to specific aspects like the accuracy of the tracked route or the visualization of the training progress.

With this paper, we present an evaluation of different features in a linear-chain conditional random field to detect aspect phrases and subjective phrases in German user reviewers from the Google Play Store. Specifically we investigate the following research questions: (1) Which performance can be achieved on German app reviews with a model which only takes textual features into account? (2) How does the performance change if training is performed on Amazon product reviews (still testing on app reviews)? (3) Under the assumption that performance in such cross-domain model application setting drops: Can the use of word embedding based features dampen the negative effect?

The rest of the paper is structured as follows: Section 2 highlights related work in app mining research. We present our model in Section 3, followed by the description of the word embedding features (Section 4). The evaluation of the approach is given in Section 5. We conclude with Section 6 and mention promising future steps and investigations.

2 Related Work

Recent year's work in opinion mining produced a vast number of approaches [16, 31, 34]. The majority of approaches focuses on the study of product reviews [5], Twitter messages [32] and blog posts [18]. Only very few approaches investigate mobile applications and user reviews in app stores.

A early approach is done by Harman et al. [14]. They analyze the price, customer rating and the rank of app downloads of apps in the BlackBerry App Store. Evaluation results show a strong correlation between the customer rating and the rank of app downloads. In contrast, Iacob and Harrison [17] automatically detect feature requests in app reviews. They use a corpus of 3,279 reviews from different applications in the BlackBerry App Store and manually create a set of 237 linguistic patterns (*e.g.* "*Adding <request> would be <POSITIVE-ADJECTIVE>*"). Fu et al. [8] focus on negative reviews and the identification of reasons which lead to poor ratings. For this purpose the utilize Latent Dirichlet allocation (LDA) [2] to extract topics from negative reviews and compare the main reasons for poor reviews of applications from different categories. Chen et al. [3] employ different topic models in a semi-supervised classifier to distinguish informative and non-informative reviews. A different approach to the analysis of app reviews is followed in [13]. They extract application features based on noun, verb and adjective collocations. They use SentiStrength [33], a sentence based opinion mining method, to determine the user opinions about the extracted features. Moreover, the recognized features will be combined to more general topics using LDA.

Other approaches in this area include fraud detection [9], classification of app reviews to identify bug reports and feature requests [24], and coarse-grained sentiment analysis [11, 13]. Further research investigates topic and keyword identification methods [10, 37] as well as review impact analysis [28]. Table 1 summarizes work in this area. The majority of the approaches is based on manually created English corpora which aren't available to the research community. For other languages only a few data sets exist, *e.g.* for German Maalej and Nabil [24] make their review data available but only provide document level annotations. Sänger et al. [30] recently published a corpus of German app reviews annotated with aspects, subjective phrases and polarity. However, they only provide results for a baseline model. In this paper, we perform further experiments on this resource. To the best of our knowledge, this is the only corpus available which contains such fine-grained annotations from the domain.

Table 1. Overview of existing work on app store review mining and analysis. For each approach the overall objective, 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.

Authors	**Objective**	**Store**	**#Apps**	**#Reviews**
Harman et al. [14]	Identification of correlations between price, rating and download rank	B	32,108	—
Iacob, Harrison [17]	Pattern-based detection of feature requests	B	270	137,000
Galvis et al. [10]	Identification of topics and keywords using LDA	G	3	327
Fu et al. [8]	Analysis of negative reviews and their reasons	G	171,493	13,286,706
Pagano, Maalej [28]	Analysis of the impact of app user reviews	A	1,100	1,100,000
Guzman, Maalej [13]	Extraction of application features / characteristics	A,G	7	32,210
Chen et al. [3]	Identification of informative and non-informative reviews	G	4	241,656
Vu et al. [37]	Identification of topic and keywords using topic modeling techniques	G	95	2,106,605
Maalej, Nabil [24]	Classification of app review into bugs, feature requests and simple appraisals	A,G	1,140	1,303,182

The use of word embedding based features has shown considerable impact on the performance on a variety of NLP tasks, for instance chunking [4] or named entity recognition [35]. Existing approaches either use word embeddings directly [32] or derive discrete features [7] from them. For example, Guo et al.[12] perform k-means clustering to get binary feature vectors. In constrast, Turian et al.[35] utilize the intervals in which the values of the vector components lie to generate discrete features. We are not aware of any previous work that has investigated word embeddings and features based on word embeddings in the context of app reviews, especially in a cross-domain setting.

3 Baseline Model

We model the recognition of subjective phrases and application aspects as sequence labeling task, *i.e.*, every word of a review text is assigned a category label from the set $L = \{\text{O}, \text{B-Subj}, \text{I-Subj}, \text{B-Asp}, \text{I-Asp}\}$. We use a linear-chain conditional random field [21] and the MALLET toolkit [25] to implement the model. To learn the parameters of the model, the maximum likelihood method is applied. Inference is performed using the Viterbi algorithm [29].

Our baseline model takes lexical, morphological and grammatical features from each word (*e.g.* the token itself, part-of-speech tag, capitalization, 3-character pre- and suffix) into account to capture the characteristics of application aspects and evaluative phrases. The features are inspired by [19]. We further integrate negation word detection as well as smiley and emoticon recognition. For negation word detection we manually compiled a list of German terms, which imply the absence of certain matters or carry out a negation of an actual situation, and match them with the review text. We use a manually assembled list of smileys and emoticons for recognition based on the lists GreenSmilies (`http://www.greensmilies.com/smilie-lexikon/`) and Smiley lexicon (`http://home.allgaeu.org/cwalter/smileys.html`).

In addition to the textual features of the currently considered token, the characteristics of the context words are taken into account. For this purpose, all features of the words with a distance of two positions before and after the current token are added to the feature vectors. Each feature will be marked with the distance to the currently considered token. All features of our model are represented as boolean values.

4 Word Embedding Model

We generate features from word embeddings to enrich our model. We opt for derivation of discrete features to be able to gain insights about the impact and effectiveness of such features. The features are inspired by previous work [15, 38].

4.1 Synonym Expansion

The first feature category that is based on embeddings represents the use of synonyms and semantically related words. More formally, for a word w up to 10 other words w' from the vocabulary V with a cosine-similarity greater than a threshold t (according to their embeddings $v(w)$ and $v(w')$) are added as synonym features.

$$\operatorname{syn}(w) = \{w' | w' \in V \setminus \{w\} \wedge \operatorname{sim}(v_w, v_{w'}) \geq t\} .$$

We set $t = 0.8$ empirically based on a hold out set. Similar words are likely to represent the same or similar concepts and should therefore get the same label. For instance, if the term *app* is recognized as an indicator of an aspect, it is likely that terms such as *application*, *program* or *tool* should also be considered as aspects since they describe similar concepts.

4.2 Clustering

Synonym features only model relationships implicitly between groups of similar words. To make this explicit, we perform hierarchical clustering of the word embeddings and add the index of the most similar cluster center to the current word as well as the full path and all path prefixes in the cluster hierarchy. Using the path prefixes enables the model to take varying levels of granularity into account and thus test different abstraction layers and cluster sizes.

Figure 2 shows the procedure exemplarily. As with the synonym expansion, the aim of the clustering is to check the presence of groups of words rather

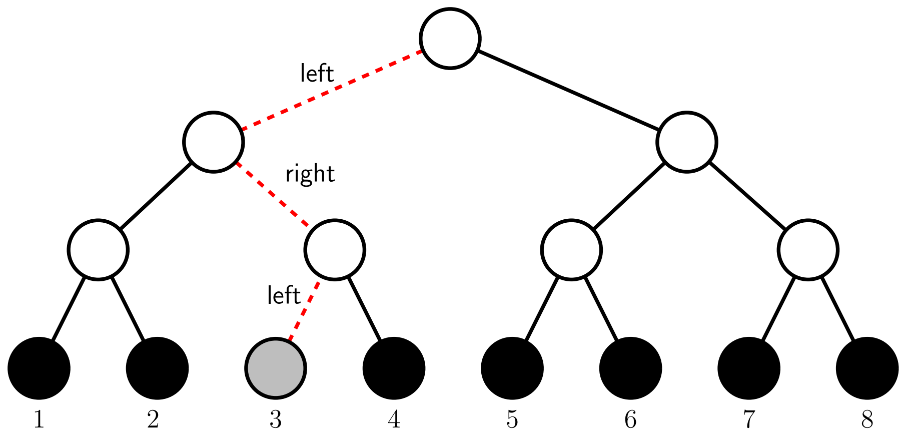


Fig. 2. Example for clustering-based feature extraction. The grey leave corresponds to the closest cluster to a considered word. Features for the path from the root are therefore *left*, *left-right*, *left-right-left*, and *cluster-id=3*.

than individual words and thus achieve a higher recall. For example, subjective expressions like *exciting*, *fascinating* or *wonderful*, potentially used to describe the user interface of an app, should be treated equivalently and therefore should get the same label. To build the cluster tree we apply a recursive top-down approach. At the beginning all word embeddings form one cluster, which is divided into two sub-clusters using the k-means clustering algorithm and cosine similarity. Each sub-cluster is then recursively divided into two sub-clusters until a depth of 10 layers is reached.

5 Experiments and Results

We perform two experiments to evaluate the performance of our approach. Firstly, we conduct an in-domain experiment for app reviews. Secondly, we train on Amazon product reviews and test on the same set of app reviews. With this experiment we are able to determine the impact of the training domain on the performance of our model.

5.1 Data

As a dataset, we use the Sentiment Corpus of App Reviews (SCARE, [30]) to perform the in-domain experiment. This corpus consists of 1,760 annotated German application reviews from the Google Play Store with 2,487 aspects and 3,959 subjective phrases in total. The reviews belong to popular apps from 11 categories (*e.g.* fitness trackers, instant messengers, games and newspaper apps) and represent a variety of use-cases of mobile applications. For the cross-domain experiment, we utilize the Bielefeld University Sentiment Analysis Corpus for German and English (USAGE, [20]). The corpus contains 611 German Amazon reviews about products from eight different categories (*e.g.* coffee machines,

Table 2. Comparison of the SCARE [30] and the USAGE [20] review corpus. The SCARE corpus consists of German app reviews from the Google Play Store. In contrast, the USAGE corpus comprises Amazon product reviews about household appliances (e.g. dishwashers or coffee machines).

	SCARE (App Reviews)	**USAGE** (Amazon Reviews)
Documents	1,760	611
⌀ Sentences / Doc.	1.87	4.82
⌀ Tokens / Doc.	19.02	89.90
Subj. Phrases	3,959	5,086
⌀ Subj. Phrases / Doc.	2.50	8.32
Aspects	2,487	6,340
⌀ Aspects / Doc.	1.41	10.38

microwaves and dishwashers) with overall 6,340 annotated product aspects and 5,086 subjective phrases. Table 2 compares the two data sets according to basic statistics. The figures especially highlight the different review text lengths (19 vs. 90 token) of the two corpora.

5.2 Experimental Setup

We first examine the baseline version of our model. Subsequently the features described in Section 4 and combinations of them are added. In the in-domain experiment we split the SCARE corpus by randomly sampling 1,408 reviews (80% of the corpus) as training set. The remaining reviews are used to test our model. For the cross-domain experiment we train our model on the complete USAGE corpus and evaluate on the same SCARE reviews as in the in-domain experiment. For each experiment, we report F_1 scores regarding exact and partial matches. An exact match requires that the text boundaries of a predicted subjective phrase or aspect must exactly match those of the goldstandard. When considering partial matches only an overlap of at least one token between a predicted and a goldstandard text span must exist.

As part of the evaluation of the word embedding features we make use of two different text corpora to learn the embeddings. To build general-purpose embeddings we utilize a German Wikipedia dump [1]. Moreover, we use a corpus of 800,000 app reviews we collected in [30] to learn domain-specific word embeddings. We apply the CBOW Model [26, 27] with a vector size of 50. All other hyperparameters of the method are left on their default.

5.3 Evaluation Results

The results of our in- and cross domain experiments are given in Table 3 resp. 4.

Our *baseline model* reaches an F_1 score of 0.62 for aspect detection and 0.61 for recognition of subjective phrases. Taking partial matches into account the

Table 3. Evaluation results as F_1 measure of the in-domain experiment. We distinguish aspect and subjective phrase detection as well as exact and partial matches for each experiment. Furthermore, the impact of each feature according to the baseline model is given in parentheses. Bold figures mark the highest result of a column.

		Subj. Phrases		Aspects	
		Exact	Partial	Exact	Partial
	Baseline model	0.605	0.769	0.620	0.685
App Review Embeddings	+ Synonym Expansion	0.610 (+ 0.8%)	0.777 (+ 1.0%)	0.628 (+ 1.3%)	**0.702 (+ 2.5%)**
	+ Clustering Features	0.615 (+ 1.7%)	**0.783 (+ 1.8%)**	0.616 (- 0.6%)	0.691 (+ 0.9%)
	+ All	0.615 (+ 1.7%)	0.782 (+ 1.7%)	**0.634 (+ 2.2%)**	0.698 (+ 1.9%)
Wikipedia Embeddings	+ Synonym Expansion	0.606 (+ 0.2%)	0.767 (- 0.2%)	0.626 (+ 0.9%)	0.685 (+/- 0%)
	+ Clustering Features	**0.618 (+ 2.1%)**	0.781 (+ 1.6%)	0.623 (- 0.5%)	0.688 (+ 0.4%)
	+ All	**0.618 (+ 2.1%)**	0.782 (+ 1.7%)	0.620 (+/- 0%)	0.683 (- 0.3%)

recognition of subjective phrases achieves clearly better values (0.77 vs. 0.69). In general, the figures are comparable to validations results of other models in product domains [20], which proves the suitability of our approach.

To test *cross-domain*, we train our model on Amazon product reviews and test on app reviews. This decreases performance considerably. The model reaches 0.46 F_1 for subjective phrase recognition and 0.35 F_1 for aspect detection. This is a performance decrease of 24% resp. 44% in comparison to the in-domain experiment. Considering also partial matches as true positives lowers these values to 8% resp. 27%. The results indicate that there are strong differences in the way personal opinions on product aspects are expressed in the particular domains.

Performance improvements can be observed with the inclusion of *word embedding* based features. We accomplish the best overall performance by using both features, synonym expansion and clustering, based on domain-specific word embeddings in the in-domain setting. The model achieves an F_1 score of 0.62 (+1.7 %) for recognition of subjective phrases and 0.63 (+2.2 %) for aspects.

In the cross-domain experiment the recognition of subjective phrases can benefit from embedding features. Here, improvements up to 7.0 % regarding exact matches resp. 5.5% for partial matches are reached. However, the detection of application aspects suffers from embedding features: Using the complete feature set in conjunction with the domain-specific embeddings, the performace decreases by 11.9 %. That is remarkable because in the in-domain setting these embeddings show better results than the Wikipedia-based embeddings.

Table 4. Evaluation results as F_1 measure of the cross-domain experiment. We distinguish aspect and subjective phrase detection as well as exact and partial matches for each experiment. Furthermore, the impact of each feature according to the baseline model is given in parentheses. Bold figures mark the highest result of a column.

		Subj. Phrases		Aspects	
		Exact	Partial	Exact	Partial
	Baseline model	0.457	0.707	0.350	**0.504**
App Review Embeddings	+ Synonym Expansion	0.456 (- 0.2%)	0.713 (+ 0.8%)	**0.355 (+ 1.4%)**	0.497 (- 1.4%)
	+ Clustering Features	0.444 (- 2.8%)	0.703 (- 0.6%)	0.345 (- 1.1%)	0.463 (- 8.1%)
	+ All	0.445 (- 0.7%)	**0.746 (+ 5.5%)**	0.316 (- 9.7%)	0.444 (- 11.9%)
Wikipedia Embeddings	+ Synonym Expansion	0.463 (+ 1.3%)	0.707 (+/- 0%)	0.350 (+/- 0%)	0.489 (- 3.6%)
	+ Clustering Features	**0.489 (+ 7.0%)**	0.718 (+ 1.6%)	0.337 (- 3.7%)	0.485 (- 3.8%)
	+ All	0.483 (+ 5.7%)	0.721 (+ 2.0%)	0.353 (+ 0.9%)	0.496 (- 1.6%)

6 Conclusion

In this paper, we presented a fine-grained sentiment analysis model for German for app reviews from the Google play store. The model is based on conditional random fields and takes lexical, morphological and grammatical features as well as domain-specific characteristics into account to extract subjective expression and application aspects from the user review texts. To model relationships between words and groups of words we enrich our approach with discrete features based on word embeddings.

The evaluation of the model shows competitive figures according to results of similar extraction approaches developed on other product domains. Furthermore, we illustrate that the performance of our model can be improved by 2 % with features based on domain-specific word embeddings. A cross-domain experiment revealed that there are clear differences in the way personal opinions and product aspects are expressed in app reviews in contrast to Amazon product reviews. This proves the necessity of domain specific models for fine-grained app review mining which take the linguisitic peculiarities of the short and informal review texts into account. Our approach represents a first step towards more detailed analysis of reviews which will support application developers as well as app customers to analyze and compare the advantages and drawbacks of one or multiple apps.

Future work will include the evaluation of the model on other sentiment data sets (*e.g.* Tweets or blog posts) as well as reviews from other languages. Moreover, we will compare the discretization of the word embeddings as done in this work with directly integrating them in our model. Another interesting research direction will be to take into account domain adaptation methods to improve the generalization of our model as well as to investigate other analysis methods (*e.g.* neural network based approaches which learn embeddings in a task specific manner). Beyond the optimization of our proposed apporach the integration of further information extraction methods can improve the usefulness of the model. For example, our current model is not designed to automatically infer the polarity (positive, negative or neutral) of an subjective expression. The extraction of relations between subjective expressions and the application aspects they are actually targeting would be benefical, too. Furthermore, the assignment of the extracted application aspects to a particular feature (group) or topic will enable further analysis of the extracted results.

Acknowledgments. We thank Christian Scheible, Peter Adolphs and Steffen Kemmerer for their valuable feedback and fruitful discussions.

References

1. Al-Rfou, R., Perozzi, B., Skiena, S.: Polyglot: Distributed word representations for multilingual nlp. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning. pp. 183–192. Association for Computational Linguistics, Sofia, Bulgaria (August 2013)
2. Blei, D., Ng, A.Y., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
3. Chen, N., Lin, J., Hoi, S.C., Xiao, X., Zhang, B.: Ar-miner: mining informative reviews for developers from mobile app marketplace. In: Proceedings of the 2014 International Conference on Software Engineering. pp. 767–778. Hyderabad, India (2014)
4. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research 12(Aug), 2493–2537 (2011)
5. Cui, H., Mittal, V., Datar, M.: Comparative experiments on sentiment classification for online product reviews. In: Proceedings of the Eighteenth Conference on Innovative Applications of Artificial Intelligence. vol. 6, pp. 1265–1270. Boston, MA, USA (2006)
6. Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Petrak, J., Bontcheva, K.: Analysis of named entity recognition and linking for tweets. Information Processing & Management 51(2), 32–49 (2015)
7. Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C., Smith, N.: Sparse overcomplete word vector representations. In: Proceedings of Association for Computational Linguistics. Beijing, China (2015)
8. Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., Sadeh, N.: Why people hate your app: making sense of user feedback in a mobile app store. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1276–1284. Association for Computing Machinery, Chicago, USA (2013)
9. Gade, T., Pardeshi, N.: A survey on ranking fraud detection using opinion mining for mobile apps. International Journal of Advanced Research in Computer and Communication Engineering 4(12) (2015)

10. Galvis Carreno, L., Winbladh, K.: Analysis of user comments: an approach for software requirements evolution. In: Proceedings of the 2013 International Conference on Software Engineering. pp. 582–591. San Francisco, CA, USA (2013)
11. Gu, X., Kim, S.: What parts of your apps are loved by users? In: Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering. pp. 760–770. IEEE, Lincoln, USA (2015)
12. Guo, J., Che, W., Wang, H., Liu, T.: Revisiting embedding features for simple semi-supervised learning. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 110–120. Doha, Qatar (2014)
13. Guzman, E., Maalej, W.: How do users like this feature? a fine grained sentiment analysis of app reviews. In: Proceedings of the 22nd International Requirements Engineering Conference. pp. 153–162. Karlskrona, Sweden (2014)
14. Harman, M., Jia, Y., Zhang, Y.: App store mining and analysis: Msr for app stores. In: Proceedings of the 9th IEEE Working Conference on Mining Software Repositories. pp. 108–111. Zurich, Switzerland (2012)
15. Hintz, G., Biemann, C.: Delexicalized supervised german lexical substitution. In: Proceedings of GermEval 2015: LexSub. pp. 11–16 (2015)
16. Hutto, C.J., Gilbert, E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Eighth International AAAI Conference on Weblogs and Social Media. Ann Arbor, MI, USA (2014)
17. Iacob, C., Harrison, R.: Retrieving and analyzing mobile apps feature requests from online reviews. In: Proceedings of the 10th IEEE Working Conference on Mining Software Repositories. pp. 41–44. San Francisco, CA, USA (2013)
18. Jakob, N., Gurevych, I.: Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 1035–1045. Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
19. Klinger, R., Cimiano, P.: Joint and pipeline probabilistic models for fine-grained sentiment analysis: extracting aspects, subjective phrases and their relations. In: IEEE 13th International Conference on Data Mining Workshops. pp. 937–944. Dallas, TX, USA (2013)
20. Klinger, R., Cimiano, P.: The usage review corpus for fine grained multi lingual opinion analysis. In: Proceedings of the Nineth International Conference on Language Resources and Evaluation. pp. 2211–2218. Reykjavik, Iceland (2014)
21. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning. Morgan Kaufmann, Williamstown, MA, USA (2001)
22. Liang, T.P., Li, X., Yang, C.T., Wang, M.: What in consumer reviews affects the sales of mobile apps: A multifacet sentiment analysis approach. International Journal of Electronic Commerce 20(2), 236–260 (2015)
23. Liu, B.: Sentiment analysis: Mining opinions, sentiments, and emotions (2015)
24. Maalej, W., Nabil, H.: Bug report, feature request, or simply praise? on automatically classifying app reviews. In: Proceedings of the IEEE 23rd International Requirements Engineering Conference. pp. 116–125. IEEE, Karlskrona, Sweden (2015)
25. McCallum, A.: Mallet: A machine learning for language toolkit (2002), `http://mallet.cs.umass.edu` (Online; Last access 08.02.2017)
26. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at International Conference on Learning Representations. Scottsdale, AZ, USA (2013)

27. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119. South Lake Tahoe, NV, USA (2013)
28. Pagano, D., Maalej, W.: User feedback in the appstore: An empirical study. In: Proceedings of the 2013 21st IEEE International Requirements Engineering Conference. pp. 125–134. IEEE, Rio de Janeiro, Brazil (2013)
29. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
30. Sänger, M., Leser, U., Kemmerer, S., Adolphs, P., Klinger, R.: SCARE - the sentiment corpus of app reviews with fine-grained annotations in german. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). Portorož, Slovenia (2016)
31. Täckström, O., McDonald, R.: Discovering fine-grained sentiment with latent variable structured prediction models. In: Advances in Information Retrieval, pp. 368–374. Springer (2011)
32. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. pp. 1555–1565. Baltimore, MD, USA (2014)
33. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology 61(12), 2544–2558 (2010)
34. Titov, I., McDonald, R.: A joint model of text and aspect ratings for sentiment summarization. In: Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Columbus, OH, USA (2008)
35. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 384–394. Uppsala, Sweden (2010)
36. Vinodhini, G., Chandrasekaran, R.: Sentiment analysis and opinion mining: a survey. International Journal 2(6) (2012)
37. Vu, P.M., Nguyen, T.T., Pham, H.V., Nguyen, T.T.: Mining user opinions in mobile app reviews: a keyword-based approach. In: Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering. pp. 749–759. IEEE, Lincoln, NE, USA (2015)
38. Yu, M., Zhao, T., Dong, D., Tian, H., Yu, D.: Compound embedding features for semi-supervised learning. In: Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics. pp. 563–568. Atlanta, GA, USA (2013)