TOWARDS MAKING SENSE OF ONLINE REVIEWS BASED ON STATEMENT EXTRACTION

Michael Rist, Ahmet Aker and Norbert Fuhr

Department of Computer Science and Applied Cognitive Science,
University Duisburg-Essen, Duisburg, Germany

ABSTRACT

Product reviews are valuable resource for information seeking and decision making purposes. Products such as smart phone are discussed based on their aspects e.g. battery life, screen quality, etc. Knowing user statements about aspects is relevant as it will guide other users in their buying process. In this paper, we automatically extract user statements about aspects for a given product. Our extraction method is based on dependency parse information of individual reviews. The parse information is used to learn patterns and use them to determine the user statements for a given aspect. Our results show that our methods are able to extract potentially useful statements for given aspects.

KEYWORDS

Aspect-based opinion extraction, dependency parse trees, dependency patterns

1. INTRODUCTION

Product reviews are a valuable resource for information seeking and decision making purposes. Online shops such as Amazon allow customers and users to add reviews to every product in their online catalogue. Certain reviews are direct feedback about a product and can guide new customers in their buying process. However, in most cases due to the immense number of reviews it is impossible for a customer to digest all of them. Several automatic processing methods of such reviews have been proposed. One intensively studied approach is the extraction of sentiment from reviews and summarization of positive and negative instances within the review set [6,7]. For instance, for the following review the sentiment approach would say that the review is positive:

The display is bright, colourful and has a high resolution.

However, sentiment extraction has a clear limitation in this case because it is not able to provide why the review is positive. Obviously, this review provides some pieces of evidence about a product and claims implicitly that the product is good because of those premises. Thus, a better approach than sentiment extraction would be a pipeline that is able to extract those premises and provide a quality judgment or claim about the product based on the extracted premises such as: *The product X is good because many think it has an excellent display.*

We present such a pipeline in Figure 1. Our pipeline shows the process of evaluating all reviews from one certain product. In the first step, it extracts premises which entails aspects and personal

Natarajan Meghanathan et al. (Eds) : NLP, JSE, CST, SIP, ARIA - 2018
pp. 01– 12, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80201

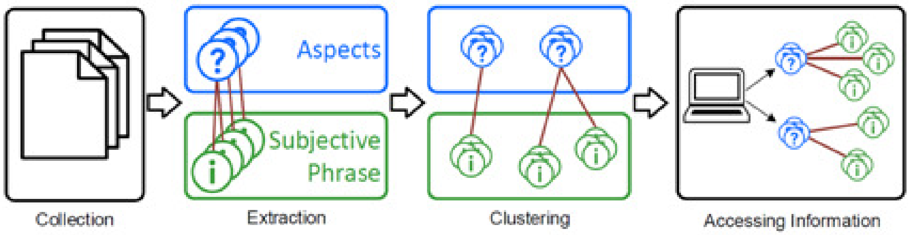

Figure 1. Information extraction pipeline

statements. Next, because the same aspect can be expressed in different ways (display, screen) it groups the different expressions of an aspect together to a broader one (e.g. display). The same is performed for all statements provided about an aspect. In the final step, it generates a summary about the product based on the aspects and statements. Our goal is to have a pipeline such that product reviews from arbitrary categories can be summarized. In this work, we focus at step one (extraction of premises) of the pipeline and leave the remaining steps for future work.

In our case a premise consists of an aspect and one or more personal statements. For instance, for the earlier example we have the aspect display and three statements: *bright*, *colourful* and *high resolution*. We assume in this work that aspects within reviews are already known and focus only on the automatic extraction of subjective phrases. Our statement extraction method is based on dependency parse trees. From the parse tree, we obtain generalized patterns that highlight the boundaries of statements and link them to an aspect within a review.

Patterns generated from dependency parse trees have been already investigated for extracting information from well-formed text [8,9,10] as well as in combination with aspect-based opinion mining [4,11,14]. However, to the best of our knowledge such patterns have not been applied to extract statements for given aspects.

The remainder of the paper is structured as follows. First, we take a short look at other approaches and methods used to process reviews for information extraction. After that we introduce the data we work with. Section 4 presents our technical solution to aspect relevant subjective phrase extraction, followed by Section 5 describing our experimental settings. Results are described and discussed in Section 6. We conclude our paper in Section 7.

2. RELATED WORK

Opinion mining and sentiment analysis is a wide research field and can be divided into different areas [3]. In terms of product reviews there has been a focus on aspect-based sentiment analysis [13,6,15]. In our work, we concentrate on aspect-based opinion mining and aim to extract statements for given aspects rather than sentiments. On this line, the work of Sauper et al. applies an LDA [1] model to simultaneously extract aspects and statements. Unlike us they are using rather clean data with one aspect per sentence and by only considering argumentative sentences, thus preemptively eliminating any noise in the data. Xu et al. also used LDA to jointly extract aspects and sentiments, however they also limit the aspects per sentence to one and extract them both at once. In our case the sentences can have more than one aspect as well as more than one statement in a sentence. We also do not assume that our sentences are argumentative. Furthermore, we apply patterns learned from dependency trees instead of LDA.

Dependency parse inspired patterns were used before in order to extract information from general texts [8,4,10] as well as online reviews [11,15,17]. In some of these studies the patterns are manually generated [4,11,15] and others learn them automatically from the data [8,10,17]. Fixed patterns are used both for learning or extracting aspects [11] and link aspects to statements [4]. Qiu et al. [15] apply relation patterns to find new aspects and statements. Their use of relations patterns is quite successful, but unlike ours has the clear restrictions of static patterns. Unlike our study generated patterns from previous studies have not been applied to extract personal statements for given aspects but rather used, for instance, to extract entity or sentiment related information.

Other approaches use the opposite direction, meaning that they search for aspects given certain, ambiguous, statement. Yauris at al. [19] for example apply the methods used in [15] to extract aspect from game reviews, however they statements are limited to adjectives only while our statements can be whole phrases. Hu et al. [20] uses a frequency based approach to extract aspects or features. The sentiment is given by an orientation and not the actual information like done here. The same underlying method was later also used and enhanced by Marrese-Taylor et al. [18] where they conducted a user study with a visual overview over the sentiment for each aspect.

3. DATA

The raw data is taken from Amazon reviews provided by [5]. These consist in total of 142.8 million reviews from which we annotated 400 randomly selected reviews. The reviews come from 4 different categories or representatively 4 different products with a sufficient review count. We annotated aspects and personal statements within the reviews. Statements are defined as certain assertions given by the reviewer. These can also be seen as a stated opinion or sentiment about some part of a product.

The aspect describes what part or characteristic of the product is being discussed. Aspects are also seen as an opinion target, like the ones used in [16]. All the reviews in our data were annotated by one single expert. Altogether we found 1,666 aspects and 1,987 statements within the annotated reviews. Among the reviews there are a few cases where the review contains only the aspect annotation and does not convey any statement. In our application scenario, we filter out such cases and focus only on reviews entailing both aspect and statements. The total number of reviews containing both annotation types is 1,966. In most cases a review contains only a single aspect and one or more statements. In this case all statements are linked to the single aspect. However, there are also cases where a review contains more than one aspect as the following example shows: *The keyboard and trackpad of this notebook is quite sturdy but not well designed.* This example sentence contains the aspects *keyboard* and *trackpad*. The statements are quite sturdy and *not well designed*. Since there are two aspects both statements are regarded as connected to each of the aspects. We use these 400 reviews to learn patterns based on dependency parse information. These patterns are in turn used to automatically extract subjective statements as well as to link them to aspects.

Table 1. Annotated Data

Categories	Claims	Premises	Relations
SD Card	333	399	396
Earphones	456	549	549
Keyboard	427	535	517
E-reader	450	504	504
all	1,666	1,987	1,966

4. METHODS

The task of extracting the complete statements is split into two successive steps. First, we identify the position of a statement within the sentence and afterwards we limit the borders of the statement. This limitation is needed because a statement might not be given in a single word and can consist of a certain part of the containing sentence. Looking at the previous example sentence: *The keyboard and trackpad of this notebook is quite sturdy but not well designed*, the statements are limited to the words *quite sturdy* and *not well designed*. When retrieving only a partial statement its meaning might be drastically altered. By excluding, for instance, the word *not* in the second statement the meaning is inverted and the actual information is lost.

For learning patterns, we use dependency parse trees, which we obtain using the DKPro framework [2], and word types (POS) for each word. Example dependency trees are shown in figures 2 to 4. Note that for POS tags with multiple variations, like nouns, we abstracted them to one general form. For instance, the sentences *The display is bright* and *The displays are bright* have the noun (aspect) *display* described by and adjective, the statement *bright*. When looking at the POS-tag of the aspects we have a tag for a singular and plural noun. Using the specific POS-tags would generate two different patterns. To avoid this, we simply use an abstract NOUN as word type for this node in the aspect.

Note, the information of the quantity is not needed for our purposes, as in the extraction and outcome each of these nouns are connected to the adjective, giving us the information how each aspect is described by its statement. This means that the correct noun, whether plural or singular will be linked to the adjective and so the information of the quantity is still present.

Figures 2 to 4 show basic examples of possible patterns and relations between the aspect and statement. Aspects are shown in blue, statements in green, the relations are marked as red edges. In general patterns are stored as a list of 3-tuples, $\langle wt(w_l), d, p(w_r) \rangle$, where each tuple describes an edge in the tree. The direction d is used to identify if the word on the right-hand side w_r of the tuple is a predecessor or successor of the current word w_l. This is done as each pattern has a leftmost node which is used as a kind of starting point or anchor for this pattern. $wt(w)$ simply gives the word type of the actual word w. $p(w)$ is a function to evaluate the word connected via the described edge and is described as:

$$p(x) = \begin{cases} \langle wt(w), d, \{p(s_1), p(s_2), \dots\} \rangle, & \text{if } w \text{ has sub-nodes} \\ wt(w), & \text{otherwise} \end{cases}$$

with s_x being the x-th succeeding word after word w in the pattern.

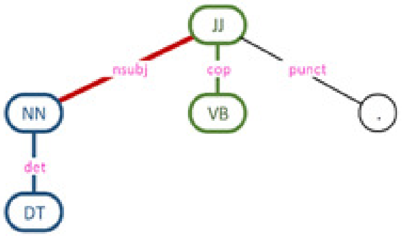

Figure 2. Dependency tree: *The display works great*.
Extracted linking pattern: $\langle NN, +, JJ \rangle$
Extracted limiting pattern: $\langle JJ, -, VB \rangle$

A simple, single pattern like seen in figure 2 contains the word type of the two connected nodes $wt(w_l)$ and $wt(w_r)$ and the direction of the link d. More complex patterns, as those in figure 3 and figure 4, are nested. This indicates which succeeding edges are needed to link the aspect to the statement in these cases.

Besides describing the relationships between an aspect and statement, the patterns can also be used to describe a statement. This allows us to minimize the complexity of our pattern as not only to find a link between an aspect but also to extract a complete statement. The whole process of extracting statements is divided into two steps: head identification of a statement, the linking to an aspect, and boundary detection, the limiting of a statement. Both limiting and linking steps are detailed in the following sections.

4.1 Limiting

The limiting of a statement defines its length and content. In order to extract only the relevant information, we need to distinguish the relevant part of a sentence from the irrelevant ones. To achieve this, we use the underlying dependency within a statement. A statement consists of several words forming a logical and rhetorical structure and have one certain root node. By determining this root node, we can extract a subtree containing all the words from the statement. Based on this subtree we create a pattern describing the word types and dependencies of the words within the statement. In figure 3 we can see the statement with one root node, the noun(NN) *colours*. The adjective *bright* is linked as an adjectival modifier (*amod*) to this noun. We can use these pieces of information to limit our pattern to $\langle NN, -, JJ \rangle$. Apart from determining the boundaries of patterns we can also use these root nodes as a clear target for the preceding step, the linking between an aspect and its statements which we describe in the next section.

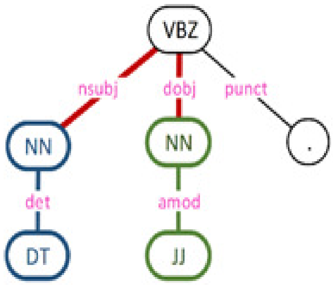

Figure 3. Dependency tree: *The Display has bright colours.*
Extracted linking pattern: $\langle NN, +, \langle VBZ, -, NN \rangle\rangle$
Extracted limiting pattern: $\langle NN, -, JJ \rangle$

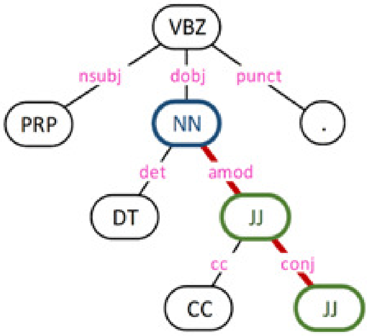

Figure 4. Dependency tree: *It has a bright and colourful display.*
Extracted linking patterns: $\{\langle NN, -, JJ \rangle, \langle NN, -, \langle JJ, -, JJ \rangle\rangle\}$
Extracted limiting patterns: $\{\langle JJ \rangle, \langle JJ \rangle\}$

4.2 Linking

Patterns are also used for linking an aspect to its statements. Similar to the statement extraction we also determine the root nodes of the aspects as a start of our patterns. For each pair of a given aspect and extracted statement we now have two root nodes for which we extract a linking pattern. In most aspects in the review there is only one statement given. For these cases, the path from the aspect to the head of the statement is taken as the pattern. These cases as well as the extracted linking patterns are shown in Figures 2 and 3. However, reviews might have complex structures such as containing more than one subjective phrases (see Figure 4). For these cases, we generate several patterns where each pattern captures only one path between the aspect and the head of each existing subjective phrase.

4.3 Selecting patterns

For both of our steps we have to select the right patterns to apply. This is needed because the extracted patterns can partially overlap each other. When looking again at the example linking patterns from figure 2, *<NN, +, VBZ>,* and figure 3, *<NN, +, <VBZ, -, JJ>>,* we can see that both patterns describe the first edge identical. The first pattern however ends after this edge, while the second pattern continues with another edge. In cases where we can apply the second pattern we could also apply the first one. Therefore, we have to prefer some patterns over others to increase overall performance as well as to have general patterns as much as possible. To achieve this, we use support and accuracy, as well as a combination of both computed over the patterns.

Support The support of a pattern states how often this pattern is observed. Quite common is a linking pattern like seen in figure 3 which is extracted from the sentence: *The display has bright colours*. This pattern is received from every sentence that has a sentence structure: *ASPECT* VERB *STATEMENT*. Instead of adding a new pattern each time we increase the support of the first pattern. The support of a limiting pattern is calculated similarly. Each occurrence of a pattern increases the support of it.

Accuracy The accuracy of a pattern is calculated by evaluating how often a pattern can be correctly applied in our data. When we apply a linking pattern we only know the head node of the aspect. When we look at the patterns from figure 2 and 3 the aspect head node has the same type. Assuming we only have those two linking patterns, we can apply the first pattern not only in the first example but also in the second as we have the same edge from the noun to the verb. This would result in one correct linking and one false linking and would achieve an accuracy of 0.5 For the limiting patterns, we proceed similarly.

Average accuracy and support These support and accuracy values are used to rank the patterns in order to determine the best ones. Additionally, we propose a third ranking by averaging over the normalized accuracy and support. The normalized accuracy a_n and support s_n are calculated by

$$a_n(p_x) = \frac{a(p_x) - \min_{p \in P} a(p)}{\max_{p \in P} a(p) - \min_{p \in P} a(p)} \text{ and}$$

$$s_n(p_x) = \frac{s(p_x) - \min_{p \in P} s(p)}{\max_{p \in P} s(p) - \min_{p \in P} s(p)},$$

with *a*(*p*) as the accuracy, *s*(*n*) as the support and *P* as the set of all patterns.

Threshold For the task of limiting a statement we use the best pattern (most highly ranked pattern) to select a single statement. However, when we want to link the statements to the aspects we have the problem that there can be multiple links per aspect and using the most highly linking pattern does not resolve the problem. Figure 4 for example has two statements *bright* and *colourful*. When we select only one linking pattern we can only retrieve one of the statements. To retrieve both statements we, have to apply more than one linking pattern. We determine the number of patterns that need to be applied using an adaptive threshold t_a This threshold is calculated by $t_a = rank\ (p_h)\ .\ (1-\ r)$, where $rank\ (p_h)$ is the value of the highest matching pattern and r is the percentage of decline which we allow. For our linking patterns, we allow a 10% decline in performance.

Table 2. Results of the predicted links

Ranking	**P**	**R**	**F1**	**P@10**	**P@20**	**P@50**
LinkBaseline	.43	.43	.43	-	-	-
Accuracy	.54	.47	.50	.69	.64	.64
Support	.41	.34	.37	.28	.32	.39
Acc. & Sup.	.48	.44	.46	.29	.33	.38

5. EXPERIMENTAL SETTINGS

As we mentioned in Section 4 we separate our approach for extracting the statements into two elementary steps: linking to the location of a statement and limiting the extracted statement. For each step, we compare our results with a different straightforward and robust baseline. For obtaining patterns, as well as for the evaluation of both steps, we use the gold standard data described in Section 3. To evaluate the significance of our results we use a pairwise McNemar test[12] with Bonferroni correction.

5.1 Evaluation setup

To evaluate the performance of our statement extraction we apply 10-fold cross validation. Note that we keep in each fold only the patterns that occur at least twice. Patterns occurring less frequently in our training set, are not used for statement extraction. This is done to eliminate possible annotation and grammatical errors from our reviews. We compute precision, recall and F1-measure to quantify the performance of our pattern extraction approach. Additionally, as we can rank our retrieved patterns, we calculate the precision at 10, 20 and 50 to evaluate for the quality of the used ranking methods.

5.2 Baseline for linking (LinkBaseline)

As a baseline for finding the statements, we extract the nearest adjective and determine if this adjective is contained in the searched statement. This is a rather simple approach as we do not have any means of limiting a statement based on the adjective, but it will be sufficient enough for detecting the general area where a statement is located. For our previously chosen example from figure 2 we assume the linking is correct if for the aspect *display* the adjective *great* is chosen as the link target.

5.3 Baseline for limiting statement (LimitBaseline)

For the limiting step, we decided to use the dependency subtree of the root node as a baseline. More precisely we extract every word directly or indirectly dependent from the root node as part of the statement. This is again a quite simple baseline and therefore we allow for some noise. We define noise as additional words retrieved in an extracted statement. For instance, for the example sentence: *The display has bright colours* in figure 3 instead of only allowing the statement *bright colours* for the aspect display we also allow *has* leading to *has bright colours* as the statement for this baseline.

6. RESULTS

As previously mentioned we first look at the results of the individual steps and then regard the performance of the whole statement extraction step.

Table 3. Performance of the statement limitation methods

Ranking	P	R	F1	P@10	P@20	P@50
LimitBaseline	.51	.48	.50	-	-	-
Accuracy	.46	.46	.46	.64	.68	.73
Support	.26	.21	.23	.64	.46	.24
Acc. & Sup.	.35	.29	.32	.55	.62	.49

Table 2 shows the performance of our linking step. From the results, we see that best performance is achieved when accuracy alone is used to rank patterns. The support ranking performs overall worse than all the others, including the baseline. When we look at the precision at position 10, 20 and 50 we see that the accuracy ranking has only a small drop in the precision from precision@10 to precision@20. The support and acc. & sup have an increased precision for position 10 to 20, but nevertheless they are still vastly outperformed by the accuracy ranking.

Results for limiting a statement are shown in Table 3. We evaluated only the exact matches between the extracted statements and the gold standard. As we see from the table the baseline performs quite well and is, overall better than our best results. The accuracy ranking outperforms our other rankings by more than 10% in the precision, recall and F1 score. This may be contrary to the intuition, as the support of a pattern indicates its popularity, and therefore should improve the recall. Relying on the most frequent pattern should also receive the most correct results. The data however shows a significantly (p value) worse performance for the support compared to the accuracy ranking.

Table 4 shows the results for the complete extraction process (statement extraction and aspect linking) with different noise levels. Noise, as described in chapter 5.3, is additional words extracted along our statements. In our testing data, we have aspect and statement pairs. In the complete extraction process we aim to determine such pairs too. If the extracted pair is correct according to our evaluation criteria then we have a positive extraction, otherwise the extracted pair is considered as incorrect. From the results, we see again that the accuracy performs best for all the metrics. Contrary to the previous results, overall performance drops noticeably, from a F1 score of about .50 to only .31 for the precise results without the noise. However, when we allow more noise, our results improve by .07 points in the precision, recall and F1 score. When comparing the results from the different noise levels incrementally, we have a significant improvement (p-value < .01) between each noise level step. Furthermore, we can see that our ranked results perform quite stable with a precision of over 50%.

Table 4. Results for extracting statement

Noise	Ranking	P	R	F1	P@10	P@20	P@50
0	Accuracy	.32	.30	.31	.52	.53	.50
	Support	.09	.07	.08	.00	.05	.06
	Acc. & Sup.	.14	.12	.13	.00	.01	.14
1	Accuracy	.36	.34	.35	.52	.53	.51
	Support	.09	.07	.08	.00	.05	.06
	Acc. & Sup.	.22	.20	.21	.00	.01	.20
2	Accuracy	.39	.37	.38	.52	.53	.52
	Support	.10	.08	.09	.00	.05	.06
	Acc. & Sup.	.24	.22	.23	.00	.01	.22

6.1 Discussion

Our results show that ranking of the pattern has an enormous influence on the performance of the extraction methods. The large performance drop between the separate steps and the complete extraction indicates that, although the individual patterns perform rather good, the selection of the correct pair of patterns can be improved. Increasing the noise level in the statements largely improves our results. It shows us however that improving the patterns and their selection could lead to further improvement as either the link is not complete or the patterns are too vague for a better extraction. Either way, this shows that there is room for improvement.

On this line we performed an error analysis. We manually inspected statements which were extracted by our patterns. Table 5 shows some of these statements. Most of the shorter statements, with one or two words, are correct and even the longest and most complex one is extracted completely. Some extracted statements like *great and reasonable* for the aspect *price* were most likely extracted by the wrong pattern. The whole sentence is the following: *Besides that this card is great and very reasonable price of $50*. The statement *great* references to the aspect *card*, but without the knowledge about the first part of the sentence this statement could also be related to the aspect price.

Another area that requires further Sattention is the problem with erroneous reviews. We have seen several reviews that were problematic and yielded wrong dependency parse trees. We aim to implement detection methods for these erroneous cases, so that we can exclude them from processing.

7. CONCLUSION & FUTURE WORK

In this work, we described the extraction of aspect-based statements from product reviews through patterns extracted from dependency parse trees. We introduced methods for identifying the head of a statement and detecting the boundary for the statement given the head. Our evaluation results show that the best method for choosing reliable patterns in both steps separately, as well as at once, is the accuracy of the pattern.

Above, we already discussed some venues for improvement. In addition to these we also want to tackle the automatic extraction of aspects. Finally, we aim to use the aspects as well as all their assigned statements to generate summaries. Such summaries can be used by customers to satisfy their information needs and help them in their decision making purposes.

Table 5. Example extracted subjective phrases

Aspect	Extracted statement	Correct statement
price	fair	fair
	very reasonable	very reasonable
	great and reasonable	*very reasonable*
	low	low
	n't beat the price	*can't beat*
	price matches quality	*Matches the quality well*
battery life	really good	really good
	lasted through the movie and several episodes of a tv show	lasted through the movie and several episodes of a tv show
	awesome	awesome

ACKNOWLEDGEMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant No. GRK 2167, Research Training Group "User-Centred Social Media".

REFERENCES

[1] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research 3(Jan), 993–1022 (2003)

[2] Eckart de Castilho, R., Gurevych, I.: A broad-coverage collection of portable nlp components for building shareable analysis pipelines. In: Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT. pp. 1–11. Association for Computational Linguistics and Dublin City University, Dublin, Ireland (August 2014), http://www.aclweb.org/anthology/W14-5201

[3] Chhabra, S., Bedathurb, S.: Summarizing entities: A survey report

[4] Gindl, S., Weichselbraun, A., Scharl, A.: Rule-based opinion target and aspect extraction to acquire affective knowledge. In: Proceedings of the 22nd International Conference onWorld Wide Web. pp. 557–564. ACM (2013)

[5] He, R., McAuley, J.: Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. CoRR abs/1602.01585 (2016), http://arxiv.org/abs/1602.01585

[6] Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 168–177. KDD '04, ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1014052.1014073

[7] Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 437–442. Association for Computational Linguistics and Dublin City University, Dublin, Ireland (August 2014), http://www.aclweb.org/anthology/S14-2076

[8] Li, P., Jiang, J., Wang, Y.: Generating templates of entity summaries with an entity-aspect model and pattern mining. In: Proceedings of the 48th annual meeting of the Association for Computational Linguistics. pp. 640–649. Association for Computational Linguistics (2010)

[9] Li, P., Wang, Y., Jiang, J.: Automatically building templates for entity summary construction. Information Processing and Management 49(1), 330 – 340 (2013), http://www.sciencedirect.com/science/article/pii/S0306457312000568

[10] Lippi, M., Torroni, P.: Context-independent claim detection for argument mining. In: Proceedings of the 24th International Conference on Artificial Intelligence. pp. 185–191. IJCAI'15, AAAI Press (2015), http://dl.acm.org/citation.cfm?id=2832249.2832275

[11] Maharani,W.,Widyantoro, D.H., Khodra, M.L.: Aspect extraction in customer reviews using syntactic pattern. Procedia Computer Science 59, 244–253 (2015)

[12] McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 12(2), 153–157 (Jun 1947), https://doi.org/10.1007/BF02295996

[13] Moghaddam, S., Ester, M.: Opinion digger: An unsupervised opinion miner from unstructured product reviews. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. pp. 1825–1828. CIKM '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1871437.1871739

[14] Moghaddam, S., Ester, M.: On the design of lda models for aspect-based opinion mining. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 803–812. ACM (2012)

[15] Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Computational linguistics 37(1), 9–27 (2011)

[16] YING, D., Yu, J., Jiang, J.: Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction (2017)

[17] Zhuang, L., Jing, F., Zhu, X.Y.: Movie review mining and summarization. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management. pp. 43–50. CIKM '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1183614.1183625

[18] Edison Marrese-Taylor, Juan D. Velásquez, Felipe Bravo-Marquez, A novel deterministic approach for aspect-based opinion mining in tourism products reviews, In Expert Systems with Applications, Volume 41, Issue 17, 2014, Pages 7764-7775, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2014.05.045.

[19] K. Yauris and M. L. Khodra, "Aspect-based summarization for game review using double propagation," 2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), Denpasar, Indonesia, 2017, pp. 1-6. doi:10.1109/ICAICTA.2017.8090997, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8090997&isnumber=8090950

[20] Hu, M., & Liu, B. (2004, July). Mining opinion features in customer reviews. In AAAI (Vol. 4, No. 4, pp. 755-760).

[21] Xu, X., Cheng, X., Tan, S., Liu, Y., & Shen, H. (2013). Aspect-level opinion mining of online customer reviews. China Communications, 10(3), 25-41.

AUTHORS

M.Sc. Michael Rist,
Research & Teaching assistant,
Workgroup Information Engineering,
Department of Computer Science and Applied Cognitive Science
University Duisburg-Essen

Dr.Ahmet Aker,
Research & Teaching assistant,
Workgroup Information Engineering,
Department of Computer Science and Applied Cognitive Science
University Duisburg-Essen

Prof. Dr. Norbert Fuhr,
Full Professor,
Workgroup Information Engineering,
Department of Computer Science and Applied Cognitive Science
University Duisburg-Essen