Aspect-Based Sentiment Analysis: A Comparative Study of BornClassifier vs. Large Language Models

Ali Tabaraei[1*] and Alfio Ferrara[1]

[1]Department of Computer Science, University of Milan, Milan, Italy.

*Corresponding author(s). E-mail(s): ali.tabaraei@studenti.unimi.it;
Contributing authors: alfio.ferrara@unimi.it;

Abstract

This paper presents an approach to aspect-based sentiment analysis (ABSA) using the `BornClassifier`, comparing its performance with NLTK and RoBERTa large language models. The methodology involves sentiment classification, feature extraction, and aspect detection for sentiment-related text. We utilize the `SemEval2014Dataset` and `AmazonReviewsDataset`, analyzing their effectiveness through experiments. Results demonstrate that the `BornClassifier` outperforms both NLTK and RoBERTa in ABSA tasks. We also discuss potential future improvements, including the incorporation of specialized datasets and overcoming the problem of out-of-vocabulary (OOV) tokens.

Keywords: NLP, Aspect-Based Sentiment Analysis, Born Classifier

1 Introduction

Sentiment analysis (SA) has emerged as a highly active and increasingly popular field in information retrieval and text mining, driven by the rapid growth and widespread use of the internet, which involves extracting sentiments from textual data [1] and can be performed at the document, sentence, or aspect level [2].

Aspect-based sentiment analysis (ABSA) offers a fine-grained approach by identifying sentiments tied to specific aspects of an entity [3, 4]. Some methods rely on a predefined list of aspects, while others dynamically identify aspects directly from the text. Common methods in identifying aspects include frequency-based approaches that identify frequent noun phrases [4–6], syntax-based techniques leveraging syntactical relations like adjectival modifiers [7–9], supervised learning [10], and unsupervised models relying on topic models like LDA [11, 12].

This study adopts a frequency-based approach for aspect detection, utilizing an explainable classification algorithm inspired by quantum physics called `BornClassifier` [13]. This classifier models text as a quantum system, representing words as quantum states and documents as their superpositions, where transition probability of a document collapsing into a target class is computed using Born's rule. The methodology and results are discussed in detail in subsequent sections.

2 Research question and methodology

The objective of this project is to leverage the Born explanation for Aspect-Based Sentiment Analysis. In short, after performing sentiment classification on documents using `BornClassifier`, the explanatory features for documents are extracted, which are further analyzed to be grouped into candidate aspects. Then, each aspect is assigned to a text portion, for which the corresponding sentiment is predicted and evaluated at last. The sub-sections below precisely describe the adopted methodology.

2.1 Text Transformation

Below, we provide a brief overview of the choices made for tokenization, normalization, and vectorization during text preprocessing:

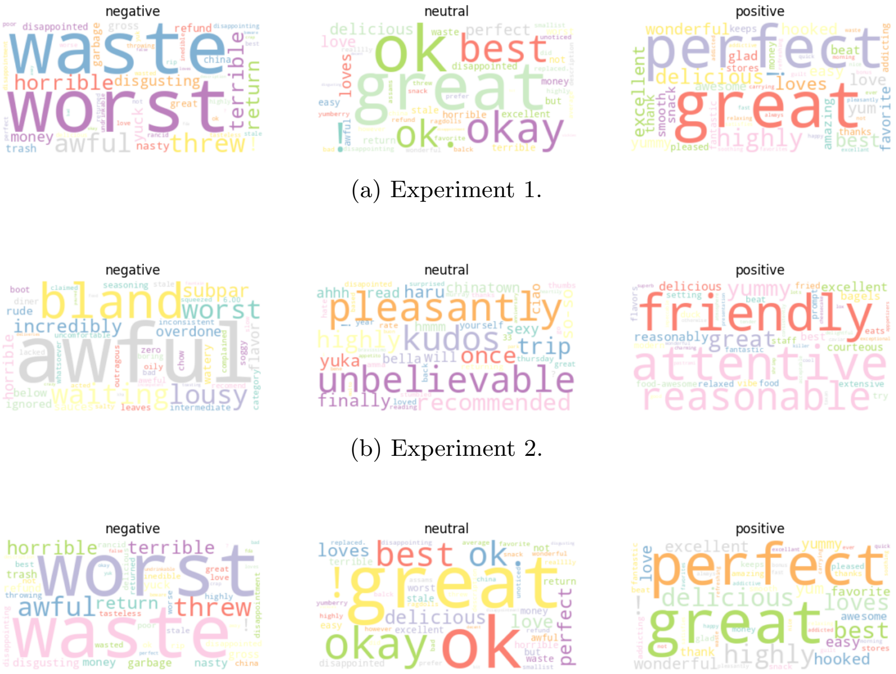
- **Tokenization:** NLTK's `word_tokenize` was used for tokenization due to its simplicity and efficiency in splitting text into individual tokens.
- **Normalization:** The tokens were normalized by converting them to lowercase. While lemmatization and stemming were initially considered, these transformations were not implemented due to challenges with out-of-vocabulary (OOV) tokens. Such transformations could alter tokens, creating inconsistencies that impact aspect detection, as discussed in detail in a later section.
- **Vectorization:** For vectorization, scikit-learn's `CountVectorizer` was employed, which transforms a collection of text documents into a sparse matrix of token counts. Additionally, `TfidfVectorizer` was tested as an alternative, but it did not yield significant improvements over the simple bag-of-words approach.

2.2 Sentiment Analysis

The `SentimentAnalysis` class has been implemented to provide functionalities for vectorizing the documents, training and testing the `BornClassifier` on the given datasets, and accessing the global explanations provided by Born. These global explanations can give us a good intuition of the most informative tokens for each given class among `['negative', 'neutral', 'positive']`.

In the following, three experiments are conducted to evaluate the `BornClassifier`'s performance in document-level sentiment classification. The primary objective is to determine if training on the `AmazonReviewsDataset` improves test performance on the `SemEval2014Dataset`:

1. **Train and test on `AmazonReviewsDataset`:** This experiment has no practical use in our aspect-based sentiment analysis, as the dataset does not contain labeled sentiment for aspects.
2. **Train and test on `SemEval2014Dataset`:** Given the training set and the overall sentiment for each document, we train the `BornClassifier` and report the performance on test set accordingly.
3. **Train on `AmazonReviewsDataset`, test on `SemEval2014Dataset`:** As told earlier, the intuition behind this experiment was to benefit from the large data available by `AmazonReviewsDataset` to see if it improves the test performance on `SemEval2014Dataset`.



(a) Experiment 1.

(b) Experiment 2.

(c) Experiment 3.

Fig. 1: Word clouds of global explanatory features extracted from `BornClassifier`

An interesting outcome of the BornClassifier is the global explanatory features, which can be extracted for each of the sentiments. To analyze better, Fig 1 represents a word cloud for each of the experiments, visualizing the words according to their weight probability in the global explanations.

2.3 Aspect-Based Sentiment Analysis

This section of the project encompasses the aspect detection and aspect-to-sentence association, each of which is elaborated upon in the following sub-sections:

2.3.1 Aspect Detection

As using the AmazonReviewsDataset for the training set of the sentiment classification was not shown to be effective in detecting the overall sentiment of documents in SemEval2014Dataset, we will proceed to use the model trained on SemEval2014Dataset for our ABSA task.

The AspectDetection class has been designed to encompass all the essential features for the aspect-based sentiment analysis. The strategy for detecting the aspects can be summarized as below:

1. **Acquiring the candidate aspects:** For a given vectorized document, first we predict its overall sentiment, based on which we retrieve the corresponding local explanatory features from the BornClassifier. Then, we keep the non-zero features in the local explanation, which convey the importance of each token of that document in predicting the overall sentiment.
2. **Detecting the aspects:** To identify aspects, we filtered the candidate aspect token set, retaining only noun groups as potential aspects. Initially, part-of-speech tagging was performed using the pos_tag module from NLTK, but its results were unsatisfactory. Therefore, spaCy was employed to obtain more accurate tags.
3. **Finalizing the aspect set:** Not all nouns in a document form aspects, so weights from Born's local explanatory features must be analyzed to identify the most informative candidates. Since documents may lack aspects and the exact number of aspects is unknown, selecting the top-k features by importance is impractical. Instead, a more sophisticated aspect_threshold is defined to filter candidates, retaining only those exceeding the specified importance level.
4. **Deciding the best_aspect_threshold:** Rather than randomly assigning the threshold, NumPy's np.linspace was used to generate 1200 values within the range [0.0001, 0.002], each evaluated separately on the training set to determine the optimal threshold for aspect detection, which is then applied to the test set for further evaluation. This optimization alone took 06:25:11 to complete, giving the corresponding best_aspect_threshold of 0.000395 with F1-score of 0.568 on the training set. Figure 2 illustrates this selection process.

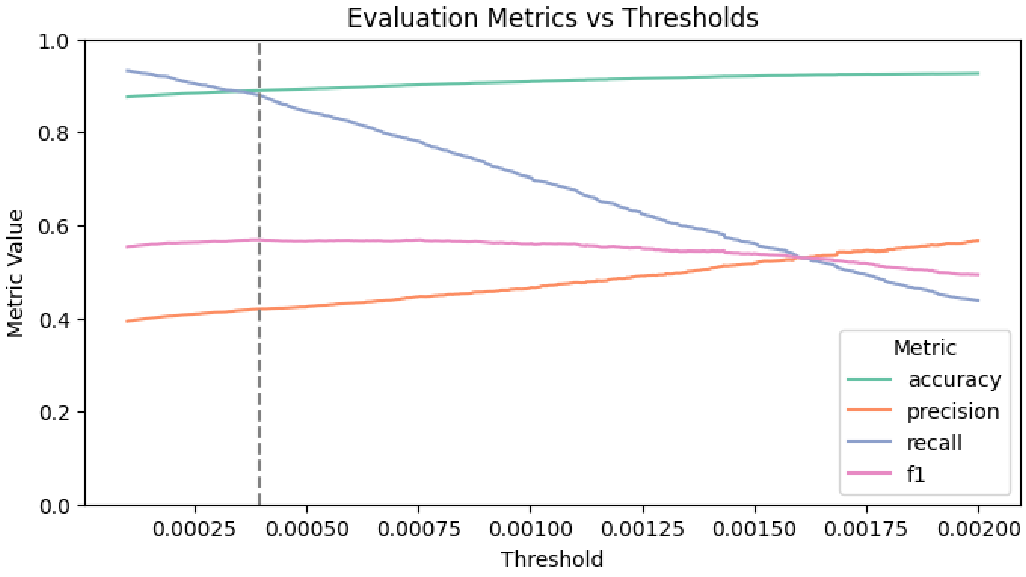


Fig. 2: Evaluation metrics for threshold optimization.

2.3.2 Aspect-to-Sentence Association

The `AspectDetection` class was extended with additional functions to locate the corresponding text portion for each of the aspects. This process can be divided into the following sub-tasks:

1. **Document Segmentation:** The first step involves splitting each document into its composing sentences. Initially, NLTK's `sent_tokenize` was used but proved ineffective since most documents in the dataset are short, and sentences are only split when a period (`.`) is encountered, leaving many unaffected. To improve segmentation, a custom regular expression (`(?<=[:;.!?()])`) was implemented, yielding significantly better results for the dataset.
2. **Aspect Lookup:** To identify the text segment containing a given aspect from those segmented in the previous step, two approaches were evaluated, with the second chosen as the final method:
   - **Vectorized Approach:** The candidate aspect tokens derived from Born's explanatory features may have undergone transformations (lowercasing, lemmatization, n-grams), altering their form, hence making it infeasible to perform simple matching. Instead, the token indices for both the aspect and the sentence were identified and checked to see if the aspect's indices were a subset of those in the sentence. However, out-of-vocabulary tokens in the test set, lacking corresponding indices, caused disruptions, making the method unreliable for consistent use.
   - **Normal Approach:** Since the only transformation applied to the tokens was lowercasing, it was reasonable to directly search for them within each sentence. Therefore, the first occurrence of a given aspect in any of the segmented text portions was returned as a match.

As an example, given the document *"You must try Odessa stew or Rabbit stew; salads-all good; and kompot is soo refreshing during the hot summer day (they make it the way my mom does, reminds me of home a lot)."* as input with `['Odessa stew', 'Rabbit stew', 'salads', 'kompot']` as true aspects, Table 1 illustrates the corresponding detected sentence for each of the aspects.

Table 1: Association of Aspects to Text Portions (sentences)

Aspect	Sentence	Aspect Sentiment
Odessa stew	You must try Odessa stew or Rabbit stew	Positive
Rabbit stew	You must try Odessa stew or Rabbit stew	Positive
Salads	salads-all good	Positive
Kompot	and kompot is soo refreshing during the hot summer	Positive

3 Experimental results

This section offers a comprehensive overview of the dataset employed in the experiments, the evaluation metrics used to assess performance, and the experimental methodology adopted. The results of the experiments are presented through a combination of detailed plots and tables.

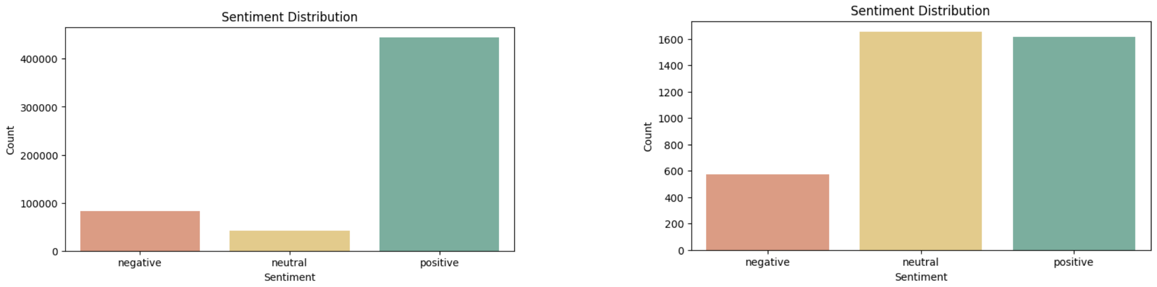
3.1 Datasets

Although numerous datasets are available for ABSA [14], the experiments in this notebook utilize the `SemEval2014Dataset` and `AmazonReviewsDataset`, which are described in more detail in the following. The sentiment distributions for these datasets are shown in Figure 3.

3.1.1 `AmazonReviewsDataset`

This dataset, available on Kaggle [15], was used solely for experimenting with sentiment classification and is not specifically designed for aspect-based sentiment analysis. It contains approximately 500,000 reviews of fine foods from Amazon, each rated from 1 to 5. As the dataset only includes user-provided scores and our focus is on overall sentiment, the ratings were converted into sentiment categories: scores of 4 and 5 as `positive`, 3 as `neutral`, and 1 and 2 as `negative`.

The intuition behind selecting this dataset was that, as the `BornClassifier` is a statistical model, training on a larger corpus would likely improve its performance. The dataset was expected to expose the model to a broader vocabulary, with the `SemEval2014Dataset` serving as the test set, a topic which will be addressed subsequently.



(a) Amazon Reviews Dataset

(b) SemEval2014 Dataset

Fig. 3: Distribution of negative, neutral, and positive sentiments in the datasets.

3.1.2 `SemEval2014Dataset`

This dataset is a benchmark for aspect-based sentiment analysis tasks, and is accessible on HuggingFace [16]. While this repository provides two dataset categories, "restaurants" and "laptops" (both following the same format), our experiments were conducted solely on the "restaurants" dataset. This dataset contains a total of 7,886 rows, with each entry consisting of a given text, a list of aspect terms, and the corresponding sentiment associated with each of those aspects.

`SemEval2014Dataset` is primarily designed for aspect-based sentiment analysis. However, since it lacks overall sentiment labels for each document—necessary for the initial phase of the project involving sentiment classification with the `BornClassifier`—these labels were inferred based on the sentiments assigned to the aspects within each document. The labeling approach is as follows:

- `positive`, if the number of positive aspects exceeds the number of negative aspects.
- `negative`, if the number of negative aspects exceeds the number of positive aspects.
- `neutral`, otherwise.

3.2 Sentiment Analysis Evaluation

Table 2 compares the classification reports for all these experiments together. As our hypothesis regarding the effectiveness of training on `AmazonReviewsDataset` was rejected, we will proceed with the `BornClassifier` model which was trained on the training set of `SemEval2014Dataset`.

Table 2: Comparison of Classification Reports Across Experiments

Class	Precision	Recall	F1-Score	Support
Negative	0.61 / 0.51 / 0.32	0.71 / 0.58 / 0.51	0.65 / 0.55 / 0.39	16407 / 125 / 575
Neutral	0.29 / 0.72 / 0.47	0.46 / 0.57 / 0.16	0.36 / 0.63 / 0.24	8528 / 279 / 1653
Positive	0.94 / 0.74 / 0.55	0.86 / 0.82 / 0.80	0.90 / 0.78 / 0.65	88756 / 396 / 1613
Accuracy			0.81 / 0.70 / 0.48	113691 / 800 / 3841
Macro Avg	0.61 / 0.66 / 0.45	0.67 / 0.66 / 0.49	0.64 / 0.65 / 0.43	113691 / 800 / 3841
Weighted Avg	0.84 / 0.70 / 0.48	0.81 / 0.70 / 0.48	0.82 / 0.69 / 0.44	113691 / 800 / 3841

This table compares classification metrics across Experiment 1, Experiment 2, and Experiment 3.

3.3 Aspect Detection Evaluation

While evaluating predicted sentiments in both sentiment analysis and ABSA is relatively straightforward, the evaluation of aspect detection requires careful consideration. The key points to be taken into account are outlined below:

- **Ground-truth aspect token length:** True aspects in the dataset are represented as noun phrases, some spanning up to six tokens, making even bigram or trigram tokenization insufficient for capturing all true aspects. To address this, a specialized function was employed to align candidate aspects with true aspects by shortening them to their minimal matching tokens, facilitating evaluation. For example, in the document *"The pizza is the best if you like thin crusted pizza."*, the true aspects `['pizza', 'thin crusted pizza']` and the predicted aspect `['pizza']` from our unigram-based detection algorithm were both reduced to `{'pizza'}`, resulting in an accurate interpretation of the document.

- **Evaluation metrics:** Each document contains tokens, some of which are *true* aspects and others are not. According to the definitions provided below, aspect detection is evaluated by computing TP, FP, FN, and TN for each of documents, accumulating over all the given set of documents.
  - **True Positive (TP):** Tokens in both `candidate_set` and `true_set`.
  - **False Positive (FP):** Tokens in `candidate_set` but not in `true_set`.
  - **False Negative (FN):** Tokens in `true_set` but not in `candidate_set`.
  - **True Negative (TN):** Tokens except in `candidate_set` and `true_set`.

  The formulas for the evaluation metrics are defined below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

$$\text{Precision} = \begin{cases} \frac{TP}{TP+FP}, & \text{if } TP + FP > 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{Recall} = \begin{cases} \frac{TP}{TP+FN}, & \text{if } TP + FN > 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$F_1 = \begin{cases} \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, & \text{if Precision} + \text{Recall} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

On the test set, with the same optimal `best_aspect_threshold` found in Section 2.3.1, we achieved an accuracy of `0.918509`, precision `0.548289`, recall `0.800893`, and F1-score of `0.650943` in detecting the aspects.

3.4 Aspect-Based Sentiment Analysis Evaluation

It is evident that our `AspectDetection` algorithm is not optimal, meaning it does not accurately identify all aspects. As a result, there may be detected aspects without corresponding ground-truth sentiments, or true aspects that our algorithm fails to detect.

To address this limitation, we evaluated sentiment prediction for aspects independently of the aspect detection process. In other words, we did not rely on the candidate aspects detected by our algorithm for aspect-based sentiment analysis. Instead, we assumed the availability of a perfect algorithm capable of identifying all true aspects correctly. Using this ground-truth set of aspects, we proceeded to evaluate sentiment predictions for the text portions where each aspect appeared, which were identified according to Section 2.3.2.

Table 3 summarizes the performance in predicting the sentiment for each text portion, reporting the results over the whole dataset (train and test combined).

Table 3: Performance of `BornClassifier` in Aspect-based Sentiment Analysis

Class	Precision	Recall	F1-Score	Support
Negative	0.68	0.80	0.73	1001
Neutral	0.55	0.31	0.40	828
Positive	0.85	0.91	0.88	2874
Accuracy			0.78	4703
Macro Avg	0.70	0.67	0.67	4703
Weighted Avg	0.76	0.78	0.76	4703

3.5 Experiments with Large Language Models (LLMs)

To provide a more thorough analysis in this study, our final attempt is to evaluate the sentiment prediction leveraging LLMs. This includes assessing both overall sentiment prediction for entire documents and aspect-based sentiment analysis within specific text portions. Finally, we compare the obtained results with those of `BornClassifer`.

The large language models used for comparison are listed as below:

- **NLTK's `SentimentIntensityAnalyzer` model:** This model assigns a sentiment intensity score to sentences, returning a float for sentiment strength based on the input text [17].
- **HuggingFace's `RoBERTa` model:** This RoBERTa-base model has been trained on nearly 58M tweets and fine-tuned for sentiment analysis with the TweetEval benchmark. It returns a sentiment score for each of the negative, neutral, and positive cases [18].

Tables 4 and 5 compare the performance of these LLMs in both sentiment analysis and aspect-based sentiment analysis, respectively.

Table 4: NLTK vs. RoBERTa in Sentiment Analysis

Class	Precision	Recall	F1-Score	Support
Negative	0.45 / 0.51	0.41 / 0.67	0.43 / 0.58	575
Neutral	0.59 / 0.64	0.36 / 0.37	0.45 / 0.47	1653
Positive	0.57 / 0.65	0.82 / 0.85	0.67 / 0.74	1613
Accuracy			0.56 / 0.62	3841
Macro Avg	0.54 / 0.60	0.53 / 0.63	0.52 / 0.59	3841
Weighted Avg	0.56 / 0.62	0.56 / 0.62	0.54 / 0.60	3841

Table 5: NLTK vs. RoBERTa in Aspect-based Sentiment Analysis

Class	Precision	Recall	F1-Score	Support
Negative	0.65 / 0.72	0.39 / 0.63	0.49 / 0.67	1001
Neutral	0.33 / 0.38	0.45 / 0.52	0.38 / 0.44	828
Positive	0.78 / 0.87	0.80 / 0.82	0.79 / 0.84	2874
Accuracy			0.65 / 0.73	4703
Macro Avg	0.58 / 0.66	0.55 / 0.66	0.55 / 0.65	4703
Weighted Avg	0.67 / 0.75	0.65 / 0.73	0.65 / 0.74	4703

In both scenarios of sentiment analysis and aspect-based sentiment analysis, it is clear that the `BornClassifier` has demonstrated superior performance compared to both the NLTK and RoBERTa models.

4 Concluding remarks

The following remarks can be concluded from this study:

- Using the `SemEval2014Dataset` for training the `BornClassifier` yielded better results compared to the `AmazonReviewsDataset` in sentiment analysis, suggesting the importance of domain-specific training data for better generalization.
- Aspect detection, although effective for simple aspects, struggled with complex noun phrases due to reliance on unigram tokenization. In addition, the `spaCy` model itself was not sufficient to find all the candidate aspects according to POS tags.
- The lack of advanced techniques such as lemmatization or n-grams due to OOV concerns may have hindered performance.
- In both sentiment analysis and aspect-based sentiment analysis (ABSA), the `BornClassifier` has demonstrated superior performance compared to the NLTK and RoBERTa models.

For future work, a key priority would be to develop specialized datasets that do not rely on sentiment mapping, allowing for a more accurate ground-truth set. Moreover, addressing the issue of OOV tokens is crucial, as this would enable the use of advanced normalization techniques and the application of n-grams for improved performance.

References

[1] Rana, T.A., Cheah, Y.-N.: Aspect extraction in sentiment analysis: comparative analysis and survey. Artificial Intelligence Review **46**, 459–483 (2016)

[2] Liu, B.: Sentiment Analysis and Opinion Mining. Springer, Switzerland (2022)

[3] Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE transactions on knowledge and data engineering **28**(3), 813–830 (2015)

[4] 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 (2004)

[5] Hu, M., Liu, B.: Mining opinion features in customer reviews. In: AAAI, vol. 4, pp. 755–760 (2004)

[6] Scaffidi, C.: Red opal: product-feature scoring from reviews. In: Proceedings of the 8th ACM Conference on Electronic Commerce (2007)

[7] Zhao, Y., Qin, B., Hu, S., Liu, T.: Generalizing syntactic structures for product attribute candidate extraction. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 377–380 (2010)

[8] Qiu, G., Liu, B., Bu, J., Chen, C.: Expanding domain sentiment lexicon through double propagation. In: Twenty-first International Joint Conference on Artificial Intelligence (2009). Citeseer

[9] Zhang, L., Liu, B., Lim, S.H., O'Brien-Strain, E.: Extracting and ranking product features in opinion documents. In: Coling 2010: Posters, pp. 1462–1470 (2010)

[10] 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 (2010)

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

[12] Hofmann, T.: Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. Advances in neural information processing systems **12** (1999)

[13] Guidotti, E., Ferrara, A.: Text classification with born's rule. In: Advances in Neural Information Processing Systems, vol. 35, pp. 30990–31001 (2022)

[14] PapersWithCode: Aspect-based Sentiment Analysis Datasets. https://paperswithcode.com/datasets?task=aspect-based-sentiment-analysis&page=1

[15] Project, S.N.A.: Amazon Fine Food Reviews. https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews

[16] Cadillon, A.: SemEval2014 Task 4 Dataset. https://huggingface.co/datasets/alexcadillon/SemEval2014Task4

[17] Project, N.: NLTK Sentiment Intensity Analyzer. https://www.nltk.org/api/nltk.sentiment.SentimentIntensityAnalyzer.html

[18] Snap: Twitter-roBERTa-base for Sentiment Analysis. https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment