# Unsupervised Aspect Classification and Sentiment Analysis for Amazon Product Reviews

Jennifer Mahle (Section 3) and Joanna Wang (Section 1)
April 20, 2020

## Abstract

As consumers rely more on online shopping, they use product ratings and user generated reviews to help them make purchasing decisions. Existing online review and rating systems are not usually based on product features or aspects, which could be helpful for both consumers and retailers to categorize product reviews. Additionally, much of the existing aspect-based sentiment analysis uses supervised training methodologies that rely on large corpus of labeled training data, which limits applicability to real world data. In this project we provided a system to predict the sentiment of a product based on the product aspect. Our contribution is applying the Unsupervised Neural Attention Model developed by He, el al. to new unlabeled Amazon review data and generating sentiment for those aspects using the Stanford Sentiment Tool. The system achieves coherent, accurate results and the unsupervised nature of the system increases applicability to a wide variety of data.

## 1. Introduction

For our final project, we use an unsupervised aspect extraction model developed by He et al. and apply it to Amazon product reviews (Ni J., 2018). Then we apply sentiment analysis (Socher R., 2013) to create an aspect-based sentiment classification system using unlabeled product review data. The system categorizes each product review sentence based on the topic of that sentence, then determines whether the review is positive or negative for a given topic (i.e., durability, sound, etc). As a user, star ratings alone might not give enough information about the product, so reading the reviews still is the best way to determine if the product fits the user's needs. The challenge is, sometimes there are hundreds of reviews for a product. So we provide this classification system to reduce the review reading process and help users to find what they need.

More specifically, this system provides online retailers an easy way to categorize product reviews into different aspects and determine the sentiment associated with each aspect. This would likely help to reduce time consumers spend to determine which product is best suited to their needs, increasing the likelihood of a sale.

While aspect-based sentiment analysis is not a new topic of research, unlike much of the existing research, this system is unsupervised. It performs well, generating coherent results without reliance on a large corpus of labeled data. Obtaining a large corpus of labeled data is time-consuming and costly, so by removing that need, the system enables more automated deployment on a wide variety of product reviews.

## 2. Related Work

There are many existing papers on aspect-based sentiment analysis for user generated review content. Much of the research in aspect-based sentiment analysis relies on "large, domain specific datasets" and manually labeled training data (Do et al., 2019). While there has been increased work using attention based neural networks, these tend to require labeled datasets like the SemEval datasets (Hu el al., 2019, Wang et al., 2018a, Tay et al., 2018, Ma et al., 2017). Reliance on topic-specific training using labeled data limits the practical use of applying these models to a wider variety of real-world data. As such, our system uses an unsupervised attention-based approach using

unlabeled Amazon review data, rather than relying on a large corpus of labeled data.

## 3. Approach

To conduct unsupervised aspect extraction, we leverage the paper "An Unsupervised Neural Attention Model for Aspect Extraction" by He et al, the underlying code, as well as an updated version of the code (citations). The aspect extraction uses clustering to determine groups of aspect-related words and an attention later to refine the potential aspect-related words. We then conduct sentiment analysis to determine the sentiment for the sentence/aspect. Our sentiment determination uses the Stanford Sentiment tool, which creates a more granular tree-based sentiment scoring (Socher R., 2013). Similar to the He et al paper, we limit our sentences to those that have one aspect per sentence; however, the tree-based structure allows for expansion to aspect-based sentiment for more than one aspect per sentence, which is why we selected this framework.

### 3.1 Aspect Extraction Methodology

The aspect extraction model and methodology is based on work done by Ruidan He, et al (citation). The goal of the model is to create clusters of words corresponding to different aspects. After cleaning the text and segmenting each review into sentences, we train the Word2Vec on the training data and apply it to get word embeddings, $\mathbf{e}_w$. Out of the ~78,000 unique words in the training set, only the most commonly used words within the data are kept in the vocabulary, creating a word embedding matrix **E**. The word embeddings are then clustered using k-means to generate the initial aspect embedding matrix, **T**.

An attention layer is used to generate weights; words that are not related to an aspect are down weighted. The sentence-level embedding, $z_s$ is generated for each input sentence by summing the weighted word embeddings, such that

$$z_s = \sum_{i=1}^{n} a_i e_{wi} \qquad (1)$$

Where $a_i$ are the positive weights computed from the attention layer, which are based on both the word embedding relevance to an aspect cluster and the context of the sentence as a whole. The context of the sentence is captured as the average of the word embeddings for that sentence, $y_s$ below. The level of relevance to an aspect cluster is a weight matrix **M** that is learned in the attention layer, which is a mapping between the context and the word embedding, capturing the relevance of the word as compared to the aspect clusters. More specifically,

$$a_i = \frac{exp(d_i)}{\sum_{j=1}^{n} exp(d_j)} \qquad (2)$$

$$d_i = e_{wi}^T M y_s \qquad (3)$$

$$y_s = \frac{1}{n} \sum_{i=1}^{n} e_{wi} \qquad (4)$$

The weights can be thought of as the probability a word is the correct aspect-related word within that sentence. The results of the attention layer are the sentence embeddings created using the weights.

In order to determine the probabilities that a sentence aligns with each of the aspect clusters, we create a sentence reconstruction $r_s$ using the aspect embeddings, T, such that

$$r_s = T^T p_t \qquad (5)$$

Where $p_t$ is a weight vector for the K aspect clusters and is obtained by reducing dimension of the sentence embedding to K and applying a softmax. It can be interpreted as the probability the sentence aligns with each of the aspect clusters.

Once the word clusters are created, each cluster is mapped to one of the four speaker/headphone related aspects or to miscellaneous. This mapping is used to determine the predicted aspect label for each sentence based on the most probable cluster alignment. Table 1 shows examples of clustered words and the aspects they map to. For example, clusters 1-3 (C1-C3) all map to design, but each has a specific sub-aspect focus, like cluster 1 (C1) focuses on construction, cluster 2 (C2) finishes and colors, and cluster 3 (C3) is headphone fit. Clusters 4 and 5 (C4-C5) are both sound-related. Cluster 4 (C4) is about the headphone sound quality,

whereas cluster 5 (C5) is more about background noise and sound cancellation.

Table 1: Example of how clustered words map from the model-generated clusters into topic-specific aspects

| **C1: Design** | **C2: Design** | **C3: Design** | **C4: Sound** | **C5: Sound** |
|---|---|---|---|---|
| Crafted | Shiny | Fit | Bass | Jet |
| Constructed | Chrome | Size | Punchy | Eliminated |
| Protects | Matte | Eartip | Midrange | Noise |

### 3.2 Data

We used Amazon product data without aspect labels or aspect-based sentiment labels. We pulled the data from the "Amazon Product Data" dataset published by Jianmo Ni, UCSD[1]. The data is divided into groups based on product type. Some of the key information that the dataset includes are the product ID, star rating, and review text. We used the electronic review data for this project, and extracted speaker and headphone product reviews specifically to limit the scope.

Since product name is not a given column in the original data set, we decided to use key words in the review text to help us extract speaker/headphone product reviews. We selected reviews that contains vocabularies that could potentially associated with speakers/headphones (such as "speaker", "earbud", "earphone", "headphone"), and excluded reviews that contains words that could be associated with speaker/headphone accessories (such as "adapter", "cable", "radio"). We limited the aspects to the four most common aspects for speaker/headphone reviews: Price, Design, Sound and Durability. To further select the data, we broke the review paragraphs into sentences by using NLTK sentence tokenizer, and only kept sentences that contain information that are related to the four aspects we have selected, based on keywords[2].

Once we have a set of review sentences we want to use for our aspect model and sentiment model, we took approximately 1000 sentences as our test set for the aspect model and manually labeled the aspects for each sentence in order to assess the model performance. For the aspect extraction model input data, we pre-processed the sentences by converting to lowercase, removing stop words, extra spaces, html tags, and punctuations[3]. We did not do the same pre-processing for the sentiment analysis model input because Stanford CoreNLP[4] relies on these features.

### 3.3 Aspect Extraction Experiments

While our main objective for the project was to implement the existing model developed by (He et al., 2017) and achieve good performance using the existing model on unlabeled data, we did experiment with implementing enhancements. To improve accuracy and tune the model to our dataset, we conducted many different rounds of parameter tuning. We also attempted to implement improved versions of the word embeddings and enhance the existing word embedding training. We focused on trying to improve the word embeddings because of the difference between the model performance on the restaurant data using the author's pre-trained embedding and training the embedding using the provided code. We attempted to implement GloVe and ELMo word embeddings, but were unsuccessful in getting them to work within the greater model structure.

The original code specified that the word embedding should not be trained within the layers,

[1] https://nijianmo.github.io/amazon/index.html Data provided for 2018 Amazon Review Data

[2] https://github.com/JoannaWangBK/AspectSentimentAnalysisOnAmaonReview/blob/master/data/Speaker_Headphone_InputData.ipynb Code for data selection and cleaning

[3] https://github.com/JoannaWangBK/Attention-Based-Aspect-Extraction/blob/W266Project/code/preprocess.py Code for pre-process data

[4] https://stanfordnlp.github.io/CoreNLP/files/input.txt Input example from Stanford CoreNLP documentation

which we suspected could have contributed to the difference in performance between the pre-trained embedding and the one we generated on the restaurant data (see appendix). We attempted to run the model allowing the word embedding layer to be trained; however, we were unable to get the model to run to completion[5] despite many attempts. Preliminary results produced on a smaller number of epochs appeared to have lower coherence compared to runs on the same number of epochs without training the embedding layer. Given that and the fact that we got good results on our data, it appears that step is not necessary.

Despite these setbacks, we were able to improve the model performance over our baseline on the speaker/headphone data. We conducted many parameter tuning experiments, adjusting the number of clusters, epochs, vocabulary size, and other parameters improved performance over the baseline. Overall, we found that most of the parameters the authors used worked optimally for our data as well. Increasing or decreasing the number of epochs, the vocabulary size and several other metrics resulted in worse model performance. A sample of these results can be found in the appendix. While several of our attempts improved performance over the baseline, we present our final model in the following section.

### 3.4 Aspect Extraction Results

To identify potential aspects, we started with a baseline using the unsupervised neural attention model (He et al., 2017). They provided code which was subsequently updated[6] to use Python 3 and updated packages. For further discussion about code testing and results validation, please see section 1 of the appendix.

We evaluate our model based on two criteria: (1) Does the model improve over baseline performance? (2) Is the model able to learn coherent aspects?

Our initial model run gave the results shown in Table 2. These results have very poor performance for durability, which is likely driven by the fact that there were only 52 durability reviews in our test set. This may be because online product reviews tend to be written soon after receiving a product so there may be fewer reviews overall that discuss durability. To counter this issue for evaluation, we combined durability and design, as durability could be seen as a part of a product's design.

Table 2: Preliminary Results on Speaker/Headphone Data

| Aspect | Precision | Recall | F1 | Count |
|---|---|---|---|---|
| Sound | 0.757 | 0.410 | 0.532 | 402 |
| Design | 0.475 | 0.561 | 0.515 | 189 |
| Price | 0.222 | 0.644 | 0.330 | 87 |
| Durability | 0.077 | 0.019 | 0.031 | 52 |
| Weighted Avg | 0.572 | 0.449 | 0.468 | 730 |

Table 3 shows an updated baseline using He et al.'s model and parameters[7], but with the combined design & durability aspects. We also incorporated some improvements for the cluster mappings for durability in the baseline (Base) presented below. We compare that baseline to the results for our final unsupervised aspect extraction (UAE) in Table 3. We also include some intermediate experimental results. UAE is the result of many experiments and parameter tuning to improve the model, where the best result used 20 clusters and a vocabulary size of 11,000. Additional experimental results are provided in the appendix.

[5] One team member was locked out of Google Cloud for "suspicious activity" after attempting to run this model multiple times over several days, requiring proof of identity and billing confirmation prior to the account being unlocked.

[6] The code can be found at https://github.com/madrugado/Attention-Based-Aspect-Extraction. This code removed reliance on inputting the embedding dimensions. It was also expanded to optionally incorporate ingesting seed words to assist in training the model using Russian language text; however, without that input, the logic is the same as that of the underlying paper.

[7] Using a vocabulary of 9,000 words and 15 clusters

Table 3: Results comparison for speaker & headphone Amazon reviews, bold indicates best metric performance

| Aspect | Model | Precision | Recall | f1 |
|---|---|---|---|---|
| Sound | Base | 0.883 | 0.565 | 0.689 |
| | Cluster=20 | 0.922 | 0.470 | 0.623 |
| | Vocab=11k | 0.813 | **0.669** | **0.734** |
| | UAE | **0.935** | 0.498 | 0.649 |
| Design & Durability | Base | 0.495 | 0.793 | 0.609 |
| | Cluster=20 | 0.492 | 0.838 | 0.619 |
| | Vocab=11k | **0.552** | 0.664 | 0.603 |
| | UAE | 0.512 | **0.888** | **0.649** |
| Price | Base | 0.310 | 0.310 | 0.310 |
| | Cluster=20 | 0.593 | 0.552 | 0.571 |
| | Vocab=11k | 0.326 | 0.322 | 0.324 |
| | UAE | **0.600** | **0.552** | **0.575** |
| Weighted Average | Base | 0.563 | 0.556 | 0.536 |
| | Cluster=20 | 0.740 | 0.601 | 0.615 |
| | Vocab=11k | 0.668 | 0.626 | **0.642** |
| | UAE | **0.755** | **0.633** | 0.641 |

Overall, our final model outperforms the baseline on all metrics. While our model does not outperform the intermediate models on all metrics, our model does a better job overall, as evidenced by the weighted averages and the substantial performance boost for price. Similar to the baseline, the precision for sound aspect outperforms all the other aspects, likely because they are over half of the data; however, the highest recall was for design & durability. That said, precision for the sound aspect is very high, but the recall is not, indicating the number of false negatives is higher than the false positives. For the design & durability aspect, we see the opposite performance, the recall is pretty high, but the precision is lower. Lastly, for price, we achieved pretty solid increases over the baseline; however, it is still the lowest performing aspect overall, looking at the f1 score. The precision and recall are similar but neither is very high. The weighted averages across all aspects are in the 0.64-0.76 range, likely driven by the higher performance of the larger aspect groups.

We also investigated the coherency of the aspects by reviewing the aspect clusters to determine whether the words contained in the clusters were related to each other and to an aspect. For the model with 20 clusters, we determined that 17 of them were coherent, as compared to 10 in the model with 14 clusters.

Now we discuss our intermediate results shown in table 3. We found that reducing the vocabulary size decreased the performance metrics, but increasing the vocabulary size improved performance on some metrics. Although the model with a vocabulary size of 11,000 out performed UAE on some metrics, the performance for price and design & durability was not as good. Additionally, the clusters were less coherent. For example, one cluster contained the following as the top aspect-related “words”: denied, bdescriptortype, v3, ssl, en, tester, behalf, pb, north, kong, seagate, australia. We did not get this level of noise in the clusters when limiting the vocabulary to 9,000 words; nor do we see that when we have the 11,000 word vocab and 20 clusters.

### 3.5 Aspect Extraction Analysis

We conducted additional in-depth review and analysis of the results. We looked into the hand labeled aspect versus predicted aspect and show some examples in Table 4. Since the price aspect had the worst performance, the examples focus on that aspect; however, they are representative of issues that occurred to a lesser degree for the other aspects as well. Example 1 is correctly labeled and predicted. Example 2 was hand labeled as a price-related aspect, but it appears there is some human error occurring in our labeled data, and

perhaps the model has more accurately labeled that review as miscellaneous. We tried to limit our test data to sentences with only one aspect, but example 3 does touch on both design and price. Although a human reader can see that price is not the primary aspect in the sentence, it proves difficult for the model to discern. Example 4 shows that this was labeled correctly, but the model does not label this correctly because it is comparing sound quality to cheap headphones, but price is not the accurate aspect here.

Table 4: Example Reviews with Actual labels and Model Predicted Labels

| Ex | Review Text | Label | Predicted |
|---|---|---|---|
| 1 | If you consider the price you pay for a set of earbuds that don't perform all that well, then these are a real steal. | Price | Price |
| 2 | I had it on full volume in the backyard while I was working for about 4 hours and it drained the battery. | Price | Misc |
| 3 | If you need a solid, premium quality product and are willing to pay extra for it, this product is for you. | Design | Price |
| 4 | It sounded like a cheap, disposable airline headphone using the bluetooth. | Sound | Price |

In comparing the predicted aspect to the hand labeled aspects for all four of the categories, we found that many of the mismatches were driven by sentences that have more than one aspect and those that were mislabeled in our hand labeling process. That said, there are also instances where the model did not capture the correct aspect, and these instances are more highly concentrated in the price aspect than in the others. This could be driven by the fact that more than half of the review sentences are in the sound aspect. In future work it would be interesting to over sample sentences that are not sound related and train the model on that to see if it is able to learn other aspects better. Additionally, it is likely that our model performance would increase if we improved our training data labels, since we encountered instances where the manual label is less correct than the predicted one.

### 3.6 Aspect-Based Sentiment Analysis

To determine the sentiment of a sentence that has a model estimated aspect, we used the Stanford SentimentPipeline tool, which is included in the Stanford CoreNLP[8] package. The model is trained on movie review data that has been manually labeled with fine-grained sentiment for each word and subset of words to allow for the sentiment tree structure. It uses a recursive neural tensor network and predicts five sentiment classes.

To evaluate the accuracy of the sentiment prediction for the review sentences on the four aspects, we originally attempted to use the star rating of the review as a benchmark. The 5 star classes correspond to the 5 sentiment classes: 1 start would be equivalent to “Very Negative”, 3 star would be equivalent to “Neutral” and so on. However, after taking a few samples from the output, it became clear that the star rating does not accurately reflect the aspect-level sentiment, even if we were to average the sentiment for all the sentences in the review. This created too much noise in the evaluation process, so instead we used manual evaluation. We took 100 sentiment prediction outputs and evaluated based on the sentence aspects. Here are the sentiment results:

Table 5:Sentiment Prediction Accuracy on 100 Samples

| | Sound | Design | Price | Durability |
|---|---|---|---|---|
| **Total** | **43** | **43** | **9** | **5** |
| **True Count (Percent)** | 30 (69.8%) | 32 (74.4%) | 7 (77.8%) | 4 (80.0%) |
| **False Count** | 13 | 11 | 2 | 1 |

[8] https://stanfordnlp.github.io/CoreNLP/download.html We used the official published 3.9.2 version of the Stanford CoreNLP package

| | | | | |
|---|---|---|---|---|
| **False within Positive Sentiment** | 5 | 2 | 0 | 0 |
| **False within Negative Sentiment** | 8 | 7 | 1 | 0 |
| **False within Neutral Sentiment** | 0 | 2 | 1 | 1 |

Overall the sentiment performance is pretty good, at 73% overall, and ranging from 69.8% to 80% accuracy at the aspect level. Since there were very few samples for the Price and Durability aspects, we focused our analysis on the Sound and Design aspects. More of the incorrectly labeled sentiments tend to be negative, likely due to the challenges that negations present in determining sentiment, although the original paper claimed one of the strengths of the model is to be able to detect negation (Socher R., 2013). There are only 5 sentences out of the 100 sentences reviewed that are classified as Very Positive or Very Negative, and they are all correct predictions. This is mostly because there is clear sentiment at the node level of the sentence. For example, the sentence with Design aspect: "*Comfortable padding, secure fit, easy, seamless pairing with iPhone, and great (GREAT) sound.*" has Very Positive sentiment. The word "great" contributed to "very positive" sentiment for the right branch of the tree.

There are sentences that have a clear explanation of rating, in terms of which part of the aspect the sentiment is related to. The model does well with these sentences. A good example would be:

> *The 2 star rating is due to the stiff design of this product.*
> *Sentiment: Negative - Predicted Correctly*

We also noticed that punctuation can change the sentiment prediction of a sentence. After looking more into it, we suspect that it is because the punctuation changes the tree structure, resulting in different sentiment classes. Graphs 1 and 2 in Section 3 of the appendix shows an example sentence where the sentiment prediction changes from "Very Positive" with punctuation, but is "Positive" without punctuation.

## 4. Conclusion

In this project we combined the usage of Unsupervised Neural Attention Model and the Stanford SentimentPipeline tool to provide a system that scores the sentiment of product reviews based on the product aspects. We achieved accurate, coherent results on both the aspect extraction and the sentiment analysis. Our aspect extraction results using unlabeled Amazon data out performed our baseline. While our system is currently limited to one aspect per review sentence, we designed it to be able to generalize and accommodate multiple aspects within one sentence, generating even more fine-grained aspect-based sentiment. Despite this, our system provides the advantage of being widely applicable to a variety of different data without the need for a large labeled corpus. The ability to conduct reliable aspect-based sentiment analysis on unlabeled data could prove helpful for both online retailers and shoppers.

## References


Hai Ha Do, PWC Prasad, Angelika Maag, Abeer Alsadoon. 2019. Deep learning for aspect-based sentiment analysis: a comparative review. *Expert Systems with Applications* 118: 272-299.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, Daniel Dahlmeier. 2017. An Unsupervised Neural Attention Model for Aspect Extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.*

Mengting Hu, Shiwan Zhao, Honglei Guo, Renhong Cheng, Zhong Su. 2019. Learning to Detect Opinion Snippet for Aspect-Based Sentiment Analysis. arXiv preprint arXiv:1909.11297.

Dehong Ma, Sujian Li, Xiaodong Zhang, HoufengWang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4068–4074.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *2013 Empirical Methods in Natural Language Processing (EMNLP)*

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Learning to attend via word-aspect associative fu-sion for aspect-based sentiment analysis. In *The Thirty-Second Conference on the Association for theAdvance of Artificial Intelligence (AAAI).*

Jingjing Wang, Jie Li, Shoushan Li, Yangyang Kang, Min Zhang, Luo Si, and Guodong Zhou. 2018a. Aspect sentiment classification with both word-level and clause-level attention networks. In *27th International Joint Conference on Artificial Intelligence (IJCAI)*.

# Appendix

## Section 1: Initial Code Testing Results using the Author's Data

To test the code works as expected, we started by running it on the author's provided datasets and pre-trained embeddings, model inputs and specifications, from which we achieved the following performance. These results, shown in the left-hand columns of Table 1, pretty closely match those presented in the original paper, indicating the code appears to be functioning as expected for the pre-trained model.

Appendix Table 1: Results on Restaurant Domain

| | **Pre-Trained Weights** | | | **Our Trained Weights** | | | |
|---|---|---|---|---|---|---|---|
| **Aspect** | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** | **Count** |
| Food | 0.943 | 0.638 | 0.761 | 0.635 | 0.349 | 0.451 | 887 |
| Staff | 0.868 | 0.634 | 0.732 | 0.263 | 0.116 | 0.161 | 352 |
| Ambience | 0.767 | 0.629 | 0.691 | 0.215 | 0.155 | 0.181 | 251 |
| Weighted Average | 0.896 | 0.636 | 0.743 | 0.477 | 0.262 | 0.337 | 1490 |

Using the code to process the data and create the embeddings, we got the results shown in the right-hand set of columns in Table 1, which are substantially worse than using their pre-trained inputs. We suspect there is something missing or different from the published code that they used to train the embedding. The only other inputs are the reviews themselves, which upon further investigation appear to match very closely with what we got using the pre-processing code.

## Section 2: Additional Experimental Results

This section provides a sample of experimental results gathered to test model performance using a variety of different parameters. This sample represents some of the more promising results we obtained. Some of the experiments we ran resulted in clusters that were not particularly coherent, which could be driven by over or under fitting due to different parameter settings. The metrics in bold represent the highest value for that metric and aspect. Overall, the model that performed the best was the one using 20 clusters and an 11,000 word vocabulary with the remaining parameters set to the same parameters as the original model. Another model with good performance metrics was the model that used 11,000 words in the vocabulary; however, aspects contained some nonsense and unrelated words that contributed to a lower overall coherency. These issues were resolved in the model that uses 20 clusters with the 11,000 word vocabulary; therefore, we selected the model that performed well across aspects and had the higher aspect coherence.

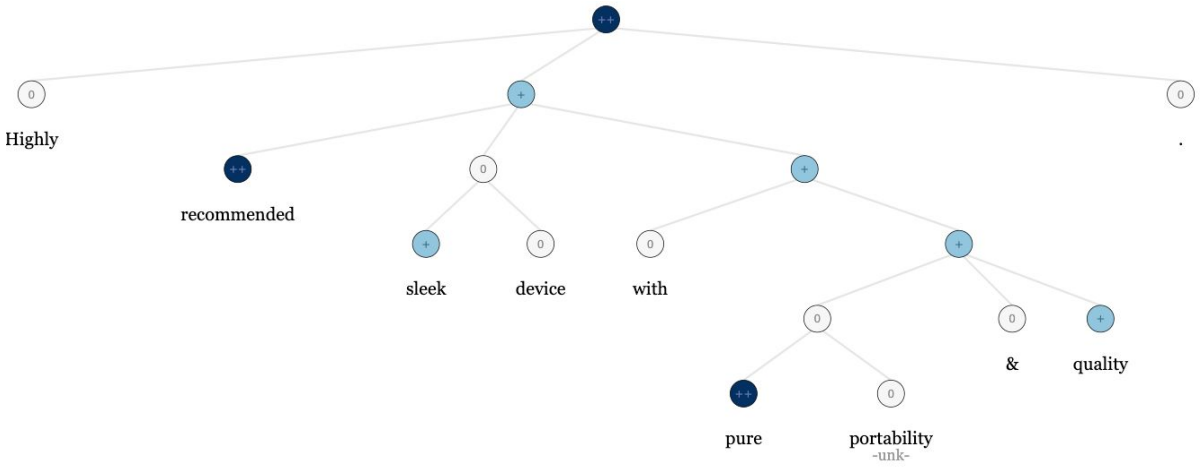
| **Aspect** | **Parameter Values** | **Precision** | **Recall** | **f1-Score** |
|---|---|---|---|---|
| Sound | 20 Clusters | **0.922** | 0.470 | 0.623 |
| | 10 Epochs | 0.919 | 0.338 | 0.495 |
| | 20 Epochs | 0.870 | 0.532 | **0.660** |
| | 20 Epochs, 20 Clusters | 0.921 | 0.408 | 0.566 |
| | 200 Embedding Dimension | 0.087 | 0.005 | 0.009 |
| | 7000 Word Vocab | 0.840 | 0.470 | 0.603 |
| | 11000 Word Vocab | 0.813 | **0.669** | **0.734** |
| Design & Durability | 20 Clusters | 0.492 | 0.838 | **0.619** |
| | 10 Epochs | 0.438 | **0.871** | 0.583 |
| | 20 Epochs | 0.480 | 0.780 | 0.594 |
| | 20 Epochs, 20 Clusters | 0.468 | 0.797 | 0.590 |
| | 200 Embedding Dimension | 0.321 | 0.627 | 0.424 |
| | 7000 Word Vocab | 0.486 | 0.734 | 0.585 |
| | 11000 Word Vocab | **0.552** | 0.664 | 0.603 |
| Price | 20 Clusters | **0.593** | 0.552 | **0.571** |
| | 10 Epochs | 0.330 | 0.379 | 0.353 |
| | 20 Epochs | 0.293 | 0.310 | 0.302 |
| | 20 Epochs, 20 Clusters | 0.563 | **0.563** | 0.563 |
| | 200 Embedding Dimension | 0.062 | 0.011 | 0.019 |

| | | | | |
|---|---|---|---|---|
| | 7000 Word Vocab | 0.257 | 0.414 | 0.317 |
| | 11000 Word Vocab | 0.326 | 0.322 | 0.324 |
| Weighted Average | 20 Clusters | **0.740** | 0.601 | 0.615 |
| | 10 Epochs | 0.690 | 0.519 | 0.507 |
| | 20 Epochs | 0.672 | 0.588 | 0.596 |
| | 20 Epochs, 20 Clusters | 0.729 | 0.555 | 0.573 |
| | 200 Embedding Dimension | 0.161 | 0.211 | 0.148 |
| | 7000 Word Vocab | 0.654 | 0.551 | 0.563 |
| | 11000 Word Vocab | 0.668 | **0.626** | **0.642** |

## Section 3: Additional Graphs

This section displays graphs that demonstrate how punctuation can change the sentiment prediction of a sentence within the Stanford Sentiment Tool. It seems this is due to the fact that the punctuation changes the tree structure, resulting in different sentiment classes. Graphs 1 and 2 below show the sentiment prediction for the same sentence with and without punctuation; however, with punctuation, the prediction is “Very Positive” and without punctuation, the prediction is “positive”. Even though both tree nodes with the words “recommended” and “pure” are marked as “Very Positive”, due to the different tree structure, the root results are different.

Graph 1: Sentiment with punctuations




Graph 2: Sentiment without punctuations