Opinion Miner

Project Preparation Report

July 22, 2025

1 Introduction

This report outlines data analysis, system design and data pre-processing project preparation for an opinion miner system that is to be evaluated on a series of customer technology product reviews. The proposed system will use the RS-DP$^+$ algorithm established by Q. Liu et al. (2015) for feature extraction, and the OrientationPrediction and SentenceOrientation algorithms from Hu and B. Liu (2004a) for sentiment analysis. It will output a mapping from product features to per-review sentiment.

2 Data Analysis

The datasets are of mixed provenance, with Customer_review_data and CustomerReviews-3_domains containing Amazon product reviews annotated as part of opinion mining studies (Hu and B. Liu, 2004b; Q. Liu et al., 2015), but Reviews-9-products is of unknown origin.

Each file contains reviews for one product, and some sentences have been annotated by humans with the identified product features and the corresponding sentiment score. Some of the files have additional annotations, such as indicating comparative sentiment, but they are not considered by this system. Of the $10,295$ sentences only 44% are annotated (Figure 1), meaning more than half of the corpus cannot be used to test feature extraction. Some of the annotations are of suspect quality. For example, in “Apex AD2600 Progressive-scan DVD player.txt”, there is the review: “play[-2], disney movie[-2]##many of our disney movies do n't play on this dvd player”. “Disney movie” is not a feature of the DVD player and a better annotator would instead extract an implicit feature like “compatibility”. However, manual inspection reveals many of the annotations are of high quality; given the historical context of these datasets, they are fit for purpose.

The files are parsed review-wise and each sentence is associated with zero or more features and sentiment scores. A class imbalance is observed with an approximately 2:1 ratio of positive to negative sentiments expressed in the corpus (Figure 2). Because the corpus is imbalanced with respect to both feature extraction and sentiment analysis, this opinion miner and its subsystems will be evaluated with precision, recall and F1-score measures.

The corpus was partitioned into an 80% training set and a 20% hold out set, stratified on the number of annotations per sentence to preserve class balance. Since sentences contain on average 1.25 annotated features, the stratification ensures that the distribution of annotation counts is similar in both sets (Figure 3). The split mitigates the risk of overfitting; for example, bootstrapping lexicons on training reviews during the double propagation (DP) algorithm (Qiu et al., 2011) may inflate recall. The split also allows the opinion miner to be tested on unseen reviews, albeit of the same domain, to give an indication of production performance.

The English language follows a Zipf distribution (Zipf, 1949). Plotting whole corpus token log frequency against log rank reveals a Zipf-like distribution (Figure 4), meaning if the system performs well on this corpus, provided the lexicons are sufficiently large, it should generalise well to other domains. Part-of-speech, sentence length and sentiment count distributions are also calculated; the distributions motivate and validate subsystem decisions throughout this report.

3 Feature and Opinion Identification

The system uses the RS-DP$^+$ algorithm established in Q. Liu et al. (2015). Since then, large language models (LLMs) have dominated research and commercial applications in NLP. However, recent benchmarking shows that LLMs struggle to extract structured features and opinions, motivating the exploration of more fundamental techniques (Heo et al., 2025). Furthermore, RS-DP$^+$ was shown to be more performant than other state-of-the-art syntactical and statistical approaches to feature extraction (Q. Liu et al., 2015).

In RS-DP$^+$, sentences are converted to a directed graph, where each node is a word and each edge is a Stanford typed-dependency relationship. It works by combining 18 typed-dependency rules with 8 feature extraction patterns to produce 144 total rules, each of which is applied to the sentence dependency graph to define relationships between features and opinions (Q. Liu et al. (2015); Equation 1).

RS-DP$^+$ distinguishes itself from the DP algorithm by its inclusion of more typed-dependency relations and the implementation of automated rule selection. Automated rule selection consists of rule evaluation, ranking and selection, and is used to maximise precision and recall on feature extraction (Qiu et al., 2011).

The algorithm extracts features using language syntax, which is effective because there exist natural relationships between opinion words and features, as opinion words are used to modify features (Qiu et al., 2011). Furthermore, it is implicitly domain independent and a suitable foundation for applicability beyond the training corpus.

4 Opinion Direction Identification

A word polarity lexicon is constructed using the OrientationPrediction algorithm (Hu and B. Liu, 2004a). The algorithm works by iteratively searching WordNet (Princeton, 2010) for the synonyms and antonyms of a pre-supplied seed list, for which the sentiment direction is known. This works because adjectives generally possess the same opinion direction as their synonyms and the opposite direction as their antonyms (Hu and B. Liu, 2004a). The corpus alone contains approximately $2,500$ unique adjectives (Figure 7), so a large lexicon of polarity-aligned adjectives is required.

To calculate an opinion score, this opinion miner will use the SentenceOrientation algorithm established in Hu and B. Liu (2004a). The feature and opinion pairs output from RS-DP$^+$ are combined with the opinion direction lexicon to identify opinion words and their sentiment (Figure 6). The algorithm calculates sentence sentiment by summing the direction scores of the opinion words, accounting for nearby negation words like "but" and "however". If the direction for sentence with index i cannot be determined, the algorithm uses the direction of sentence $i-1$. Sentence-level opinion direction evaluation is justified by the data analysis; Figures 8 and 9 show that the average sentence length is 19 tokens and each has an average of 1 adjective. A review's sentence-level, per-feature sentiments are summed to arrive at a per-review, per-feature score.

5 Data Pre-Processing

The datasets are ingested by custom parsers, which were validated with unit tests to reduce the chance of bugs and possible corruption of the ground truth. The differences between the datasets are detailed in Table 2. Sentences with invalid annotations are ignored, and punctuation whitespace separators are not necessary for dependency parsers and are left in place. All of CustomerReviews-3_domains and 2 files in Reviews-9-products are missing review separators; in these cases, each sentence is treated as a separate review to increase test data quantity. However, this opens the door to review spamming, and review delineation must be enforced in a production system.

The system works on a per-review basis, and it assumes each review is tagged with the product it describes and is grouped with the other sentences. Each sentence's dependency tree is calculated on ingestion, and the lexicon required by the OrientationPrediction algorithm is generated using WordNet and the seed list detailed in Table 1.

With respect to feature extraction, the ground truth is the lemmas of the annotated features per sentence. Lemmatisation is employed to ensure tested feature word variations with the same base form as the ground truth resolve to a match. For example, this would ensure the annotation "battery" matches with an extracted feature "batteries" by resolving both to the lemma "battery" (Table 3). With respect to sentiment analysis, the ground truth for all features in a sentence is the sum of the annotated sentiment scores, normalised to 1, -1 or 0 (Equation 2). If the sum equals 0, take the normalised sentiment score of the "effective opinion" as defined in (Hu and B. Liu, 2004a). This approach incorporates and transforms the ground truth into a form enabling system evaluation.

Some annotated features are not nouns, for example, in "Overall[+1]## Comments: Overall this is still a great value", "overall" is an adverb and not a valid feature. Re-annotating the corpus is labour intensive, prone to error and prevents comparison to historical studies using this data. Therefore, for this and other cases of poor annotation, this system will attempt to extract features and sentiment based on sound linguistic principles and will not attempt to match erroneous ground truth values. This will reduce recall on the test set but will increase precision in a production environment.

The corpus contains over $30,000$ unique noun phrases (Figure 7). Many of the annotations are noun phrases such as "battery life", and the output from RS-DP$^+$ inference includes only singular nouns. Therefore, the system uses each sentence's dependency tree to calculate the possible noun phrases, backing the noun into the noun phrase if there's a match and removing the noun phrase from the sentence's pool.

Pre-processing significantly reduces the inference time complexity of both subsystems by pre-computing required lexicons in parallel. RS-DP$^+$ and SentenceOrientation drive inference complexity and both run in $O(n)$, where n is the number of tokens (Jurafsky and Martin, 2009).

6 Limitations

The proposed system has several limitations. The lexicons created during bootstrapping are fixed and online model updates are potentially needed to update them for new contexts. This is risky for production applications as it may change the precision and recall performance profile from that tested.

The SentenceOrientation algorithm excludes verbs when evaluating sentiment, thereby excluding informative sentences like "I recommend the bike" and limiting the applicability of this system in real-world scenarios by reducing recall. This algorithm also outputs uniform sentiment for each sentence and loses granular detail if sentences contain both positive and negative comments on multiple features, although this is unlikely (Hu and B. Liu, 2004a).

Finally, the system does not attempt to fix misspellings such as "batery" in "great batery life". This harms the system's ability to account for otherwise meaningful user reviews.

7 Conclusion

The system blends the sentiment analysis devised by Hu and B. Liu (2004a) with the feature extraction methodology created by Q. Liu et al. (2015). Based on established literature and a syntactic parsing, rules-based approach, this system has the foundation to be a performant, explainable and generalisable opinion miner.

References

Heo, R., Seo, Y., Lee, J., and Lee, D., 2025. *Can large language models be effective online opinion miners?* Unpublished. arXiv: `2505.15695`.

Hu, M. and Liu, B., 2004a. Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.168–177.

Hu, M. and Liu, B., 2004b. Mining opinion features in customer reviews. *AAAI*. Vol. 4, 4, pp.755–760.

Jurafsky, D. and Martin, J.H., 2009. *Speech and language processing (2nd edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Liu, Q., Gao, Z., Liu, B., and Zhang, Y., 2015. Automated rule selection for aspect extraction in opinion mining. *Ijcai*. Vol. 15, pp.1291–1297.

Princeton, 2010. *About WordNet* [Online]. Available from: `https://wordnet.princeton.edu`.

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

Zipf, G.K., 1949. *Human Behavior and the Principle of Least Effort*. Cambridge, MA: Addison–Wesley.

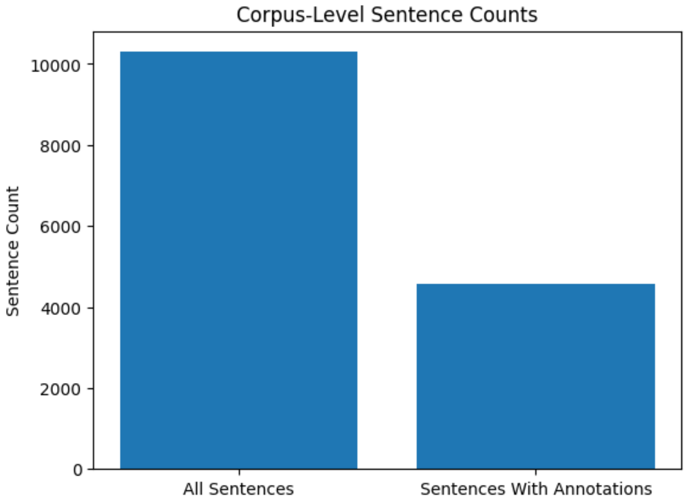

Figure 1: The proportion of annotated sentences relative to all sentences; out of 10, 295 sentences, 4, 580 are annotated.

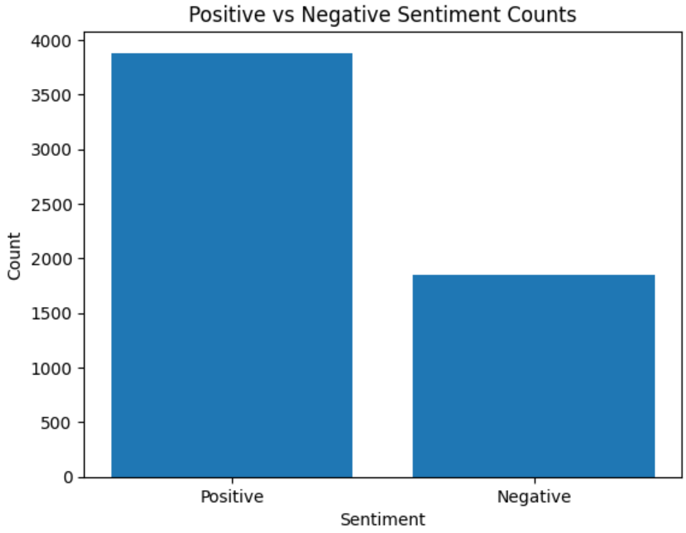

Figure 2: A class imbalance is observed with an approximately 2:1 ratio of positive to negative sentiments expressed in the corpus.

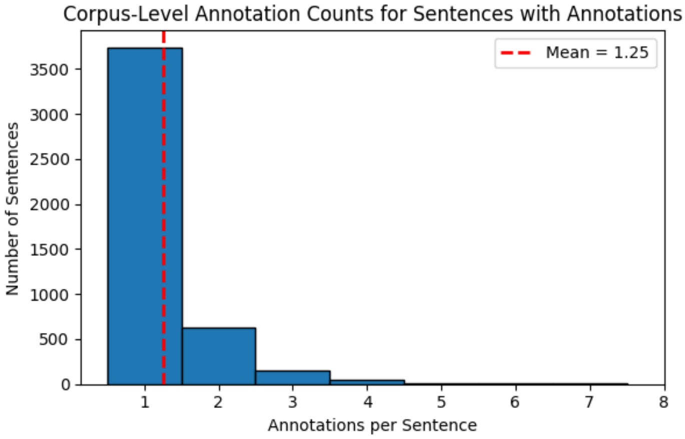

Figure 3: Annotation counts for sentences with annotations. The vast majority of annotated sentences are annotated with a single extracted feature and sentiment.

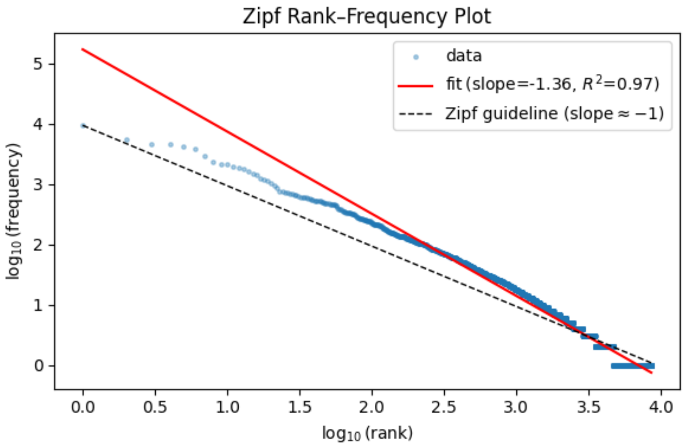

Figure 4: Plotting log frequency against rank reveals the corpus resembles a Zipf-like distribution, confirming that the frequency of a word is inversely proportional to its rank.

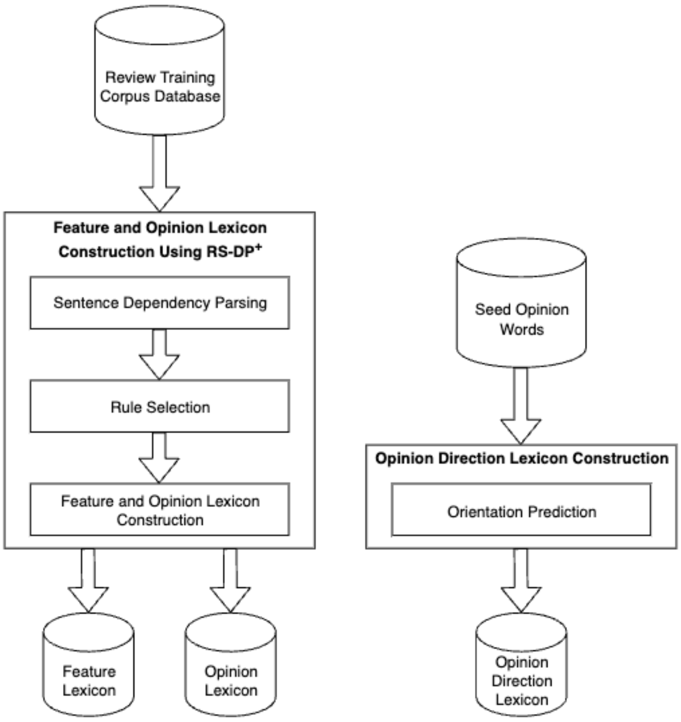
Review Training Corpus Database
Feature and Opinion Lexicon Construction Using RS-DP+
Sentence Dependency Parsing
Rule Selection
Feature and Opinion Lexicon Construction
Feature Lexicon
Opinion Lexicon
Seed Opinion Words
Opinion Direction Lexicon Construction
Orientation Prediction
Opinion Direction Lexicon

Figure 5: System diagrams for the training phase of the two subsystems. The subsystems are independent at this phase and can run in parallel. The output of the feature extraction subsystem is a feature and opinion lexicon, containing the set of features and opinions with defined relationships encountered in the test set. The output of the opinion direction subsystem is an opinion direction lexicon, containing two sets of words associated with either a positive or a negative sentiment.

$$O \xrightarrow{\text{O-Dep}} T \quad \text{s.t.} \quad O \in \mathcal{O},\ \text{O-Dep} \in \mathcal{MR},\ \text{POS}(T) = \text{NN}$$

Equation 1: Is equivalent to: “If a known opinion word is connected to a noun by one of the modifier typed-dependency relations, promote that noun to a target feature”. Adapted from Qiu et al. (2011).

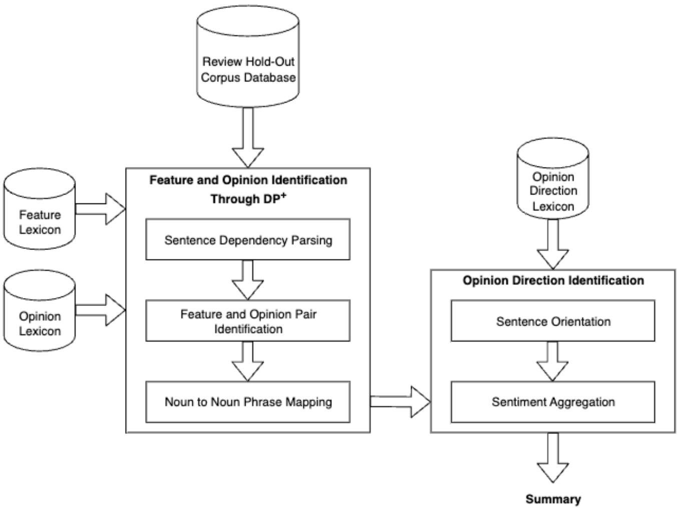
Review Hold-Out Corpus Database
Feature and Opinion Identification Through DP⁺
Feature Lexicon
Opinion Lexicon
Sentence Dependency Parsing
Feature and Opinion Pair Identification
Noun to Noun Phrase Mapping
Opinion Direction Lexicon
Opinion Direction Identification
Sentence Orientation
Sentiment Aggregation
Summary

Figure 6: Opinion miner inference time system architecture. Inputs for both subsystems are processed sentence-wise, and all reviews are considered to be equally important. The feature and opinion identification subsystem uses pre-computed lexicons, dependency graph construction and noun to noun phrase mapping to build per-sentence mappings of features to opinions. The opinion direction identification subsystem uses the pre-computed opinion direction lexicon and the sentence orientation algorithm to produce a summary for the sentences, aggregating sentiment to the review level.

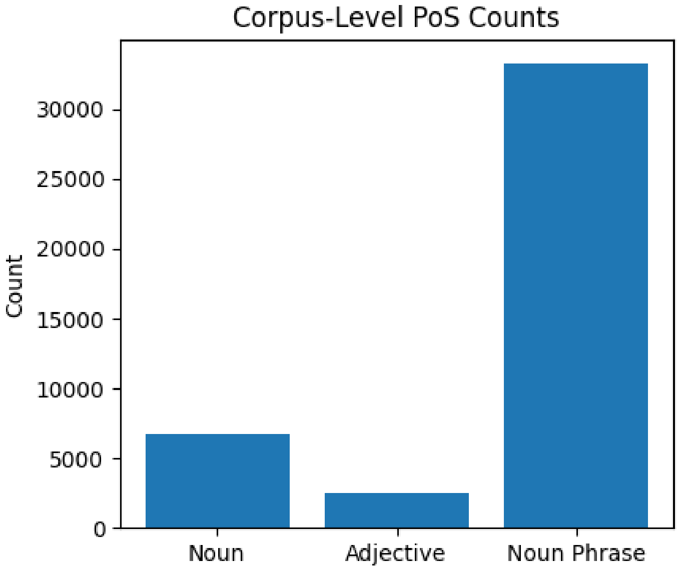

Figure 7: The counts of unique nouns, noun phrases and adjectives in the corpus.

$$\mathrm{sgn}(x) \;=\; \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0. \end{cases}$$

Equation 2: The sign function.

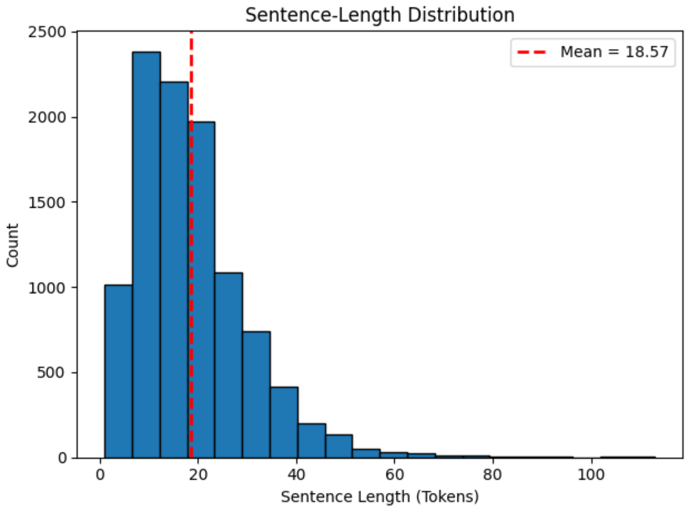

Figure 8: Tokens per sentence follows a log-normal distribution, and the average sentence length in this corpus is approximately 19.

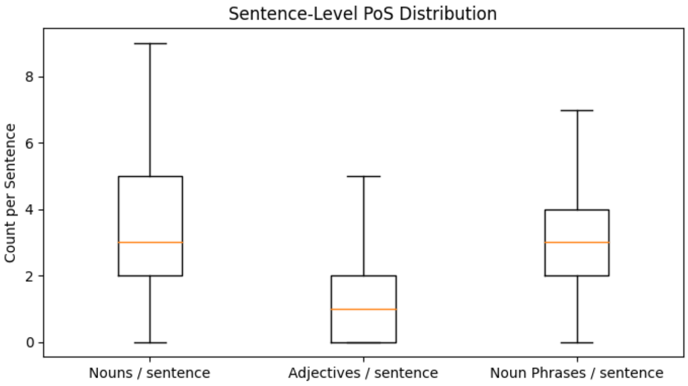

Figure 9: The counts per sentence from the review corpus of nouns, adjectives, verbs, noun phrases and verb phrases, respectively.

Positive seeds	**Negative seeds**
pleasant	dismal
brilliant	poor
gorgeous	awful
impressive	unpleasant
fantastic	lousy
stellar	faulty
great	unacceptable
adorable	ghastly
awesome	nasty
terrific	terrible
favourable	miserable
amazing	ugly
good	inferior
lovely	horrible
admirable	bad
delightful	problematic
fabulous	disagreeable
excellent	wretched
marvellous	inadequate
beneficial	grim
perfect	negative
satisfactory	dreadful
valuable	subpar
outstanding	worthless
enjoyable	troubling
remarkable	sad
exceptional	mediocre
positive	defective
superb	pathetic
wonderful	unsatisfactory

Table 1: Seed lexicon of 30 positive and 30 negative words used to initialise sentiment orientation.

Dataset	**Inline README**	**Review Separator**	**Punctuation Separator**
`Customer_review_data`	Yes	Yes	Yes
`CustomerReviews-3_domains`	No	No	Yes
`Reviews-9-products`	No	Mixed	No

Table 2: Comparison of dataset formats: presence of an inline README and review separators.

Original Sentence	**GT Feature**	**GT Sentiment**	**SGT Feature**	**SGT Sentiment**
`battery[+2]##( batteries last longer too !  )`	Battery	+2	Battery	+1
`batteries[-3]## This was obviously ...  replaceable batteries!!`	Batteries	−3	Battery	−1
`look[+1], speed[-3]## Looks good, but the speed is terrible`	Look, Speed	+1, −3	Look, Speed	−1, −1

Table 3: Standardised ground truth features and their sentiment orientation for example sentences, where GT is Ground Truth and SGT is Standardised Ground Truth.