Aspect and Opinion Extraction for Amazon Reviews

Achyut Joshi, Andrew Giannotto, Ishika Arora & Sumedha Raman

INTRODUCTION

Do you find youself spending a lot of time on looking at reviews when shopping online? What if we can make your lives easier by showing you a quick summary of people's opinions about a product?

Our project attempts at extracting different aspects from product reviews and categorising them into clusters based on the similarity and frequency of words. We finally merged the results into a user interface which can help easily look through the reviews of a product

We used Amazon customer review dataset available on S3 buckets comprising of review data for 50 different product categories.

★★ The battery life of this phone is awful! I can't go anywhere without my charger. However my pictures are coming out to be great.

★★★ The camera of the phone is incredible

130M+ reviews

8M+ products

>80 % 4-5 stars

582 avg. characters

STEP 1 : ASPECT EXTRACTION

We crafted structured rules to filter out relevant aspect-modifier pairs based on the dependency parser tree of each review.

NOUN VERB ADJ
The sound quality of the speakers is wonderful.
A ←nsubj— * —acomp→ M

NOUN ADV ADJ
The product is not comfortable.
A ←nsubj— * —acomp→ M
M' ←neg—

ADV ADJ NOUN
These are the most comfortable headphones ever.
A —amod→ M —advmod→ M'

NOUN AUX VERB ADJ
The product packaging could have been better
A ←nsubj— * —acomp→ M
M' ←aux—

it this they these
Product

STEP 2 & 3 : CLUSTERING AND SCORING

We used word vectorisation in spaCy package to cluster the aspects into categories of similar nouns. The word vectors were then grouped into four or less clusters using *K-Means clustering* algorithm in Scikit-Learn.

To score the adjectives, we used the VADER Sentiment Analysis from the NLTK toolkit.

The cable is nice and all. Signal is already poor 45-50% signal. I may have picked something too long.

Good extension for extra antenna

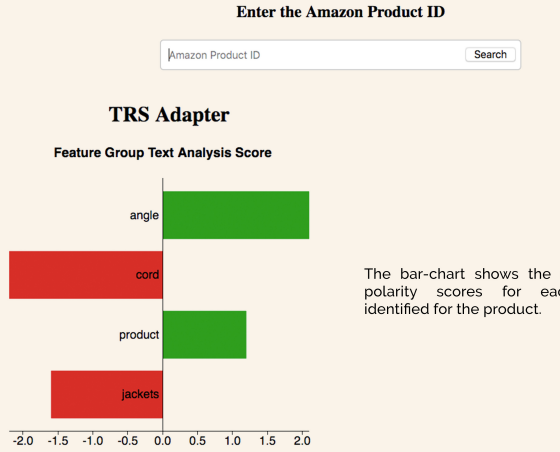
poor signal → Signal ← extra antenna

long something → Product ← good extension

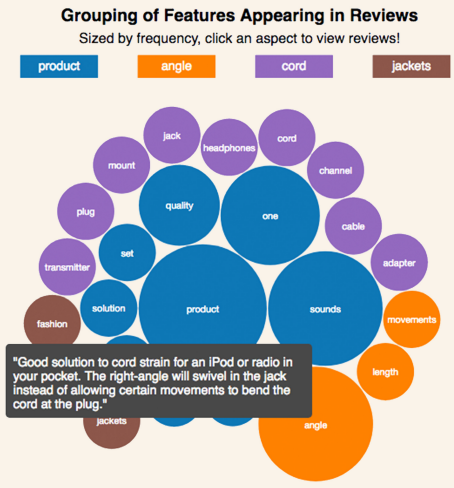
nice cable → Wire

spaCy python

VISUALIZATIONS



The bar-chart shows the aggregated polarity scores for each cluster identified for the product.



EXPERIMENTS & CONCLUSIONS

Evaluation of Aspect Pairs

Our major focus was to enhance aspect extraction process, since this serves as the building block of the whole model.

We manually analysed the aspect results of more than 100 reviews to design and engineer dependency rules. We iteratively added rules to the model, like identifying negative relationships and modal auxilary verbs.

We also replaced common pronouns (it, these, they, this) used to describe a product with the keyword '*product*'

Evaluation of Polarity

With the NLTK polarity score calculation, a lot of aspects were scored as 'Neutral' and given a score of 0.0. Our model scores each word individually, and does not take into account the contextual meaning. Thus, our polarity score results were sub-optimal.

This wire is thin compared to others.
0

This wire is more efficient conductor.
0.42

Conclusions

We used AWS EC2 instance to run different steps of the model which decreased the model run time.

We hosted a SQL database to store the model results for low latency and page load time in the UI.

Visualization delivers key information faster than reading through the reviews.

Future Work

Implement contextual approach for getting polarity scores.

Develop a browser plug-in for easy access directly from the host website

Optimise the model training time which currently is roughly 7-8hrs for 1M reviews

Enhance the aspect extraction model

aws