Extracting and Aggregating Aspect-Level Sentiment from Product Reviews

Matthew Long, Desmond C. Ong, Shane Soh
Stanford University
{mlong14, dco, shanesoh}@stanford.edu

Abstract

Previous work in *aspect-level sentiment analysis*—identifying the sentiment associated with products and their individual attributes, like battery life—have mostly been formulated as supervised learning problems, requiring known labels of both the relevant aspects and their sentiment. Here we propose a hybrid method where we first generate aspects from natural language text via unsupervised clustering of word vector representations, and secondly extract aspect-level sentiment. We further propose a deep learning architecture for aggregating and summarizing these aspect-level sentiment within and across reviews.

1 Introduction

In today's e-commercialized society, the consumer not only has access to online stores through which she might purchase anything she desires, but she also has unprecedented access to a deluge of information—most notably, product reviews written by other consumers—with which she can make her decision. Unfortunately, sifting through hundreds of reviews across tens of different websites to acquire specific information about the product and its important attributes (e.g. *battery life* for electronics) is a time-consuming chore. Because of this, there has been much recent research tackling the two separate but connected components of this problem: (1) entity-level or aspect-specific sentiment analysis, and (2) summarization and aggregation of reviews.

Within the context of a product review, the first component, *aspect-specific sentiment analysis*, involves identifying individual "aspects" (which could be the product itself, or features/attributes of the product), and subsequently identifying the sentiment—positive, neutral, or negative judgments—associated with those aspects [1-5]. Previous methods have relied on graphical models (e.g. Latent Dirichlet Allocation [1-3], Conditional Random Fields [4]), or directly modeling sentiment compositionality [5]. In this paper, we propose extending recent successful advances in deep learning [6-7] to address this problem. In particular, we seek to improve and extend the work in [7], who propose using several deep learning architectures like the Recursive Neural Tensor Network (RNTN) to extract aspect and sentiment in a single step. Most of the previous work mentioned [1-2,4-7] (except [3]) are supervised methods that require labeled aspects-sentiment pairs for training. This is unscalable, and a better approach would be to automatically identify relevant aspects for each product. This is made more difficult by the fact that relevant aspects might differ across similar products. For example, for a given electronic item such as a mobile phone, relevant aspects could be the battery life, screen size, weight, cost, and more. Not all these aspects are relevant across other electronic items: for a computer, weight might not be an issue, and screen size might be moot for headphones. We propose using unsupervised methods to generate candidate aspects via semantic word vector representations, followed by simultaneous extraction of aspect and sentiment.

The second problem involves aggregation of reviews, or constructing a *meta-review*. Previous work [8-10] has shown that simple averaging of the "stars" of reviews for an individual product is suboptimal. Here we propose a deep learning architecture to aggregate the previously identified aspect-

specific sentiment across multiple sentences within a review, and furthermore, to aggregate these sentiment across multiple reviews.

2 Problem Statement

Our proposed workflow can be divided into three main parts: **Aspect Identification**, **Aspect-Specific Sentiment Extraction**, and **Sentiment Aggregation**. See Fig. **??** for an illustration.

2.1 Aspect Identification

Problem: Given an **unlabeled** set of product reviews (for a single product), identify clusters $\{C_1, \ldots, C_k\}$, where the (weighted) centroid of cluster i would represent aspect i, and member words of cluster i would represent synonyms that map onto the same aspect i.

Dataset: Amazon Reviews of Electronics [11] **(fill in how many reviews)**.

Evaluation: **(fill in)**.

2.2 Aspect-Specific Sentiment Extraction

Input: From above, we can generate aspect labels for product reviews. We will also use an existing trained sentiment analysis model to generate sentiment labels for each token / unigram.

Problem: Given a set of product reviews (for a single product), identify (aspect, aspect-related sentiment) tuples. This is a similar problem statement as [7]: we intend to expand upon their model.

Evaluation: **(fill in)**.

2.3 Aspect-Specific Sentiment Aggregation

Input: From above, we have (aspect, aspect-related sentiment) tuples which come from multiple sentences within a review, and across multiple reviews, for a given product.

Dataset: We propose scraping some product reviews off Consumer Reports (or similar sites like CNET.com, or DPReview for Cameras) to collect a small, labeled dataset. These websites provide professionally produced aspect-related sentiment ratings, which we take to be the gold standard.

Problem: Given a set of (aspect, aspect-related sentiment) tuples (for a single product), optimally summarize them into a "meta-review". The desired output is a sentiment-aspect vector where the i-th element is the sentiment for aspect i. We plan to scrape a small set of labeled summaries, and use a semi-supervised recursive neural network architecture (or a variant) to learn how to compositionally combine individual (aspect, aspect-related sentiment) tuples.

Evaluation: We will set aside some of the labeled gold standard meta-reviews into dev and test sets. We will report accuracy on the dev sets (e.g. L2 distance between the predicted and actual sentiment vectors) as a function of training set-size (increased in a semi-supervised manner) and other hyper parameters. Finally, we will report accuracy on the test set.

3 Technical Approach and Models

3.1 Aspect Identification

The first part of the project involves an unsupervised approach to generating aspects. First, we run word2vec to obtain a word vector representation of the data. Then, we would perform k-means clustering (or some other clustering algorithm) on the word2vec representations. This would generate clusters of similar words, such as: $\{$build construction, build quality, durability, etc$\}$. We can then define the weighted centroid of these clusters as **Aspects**, and the other words in the cluster as relevant synonyms. This allows us to label all the aspect-words in the dataset.

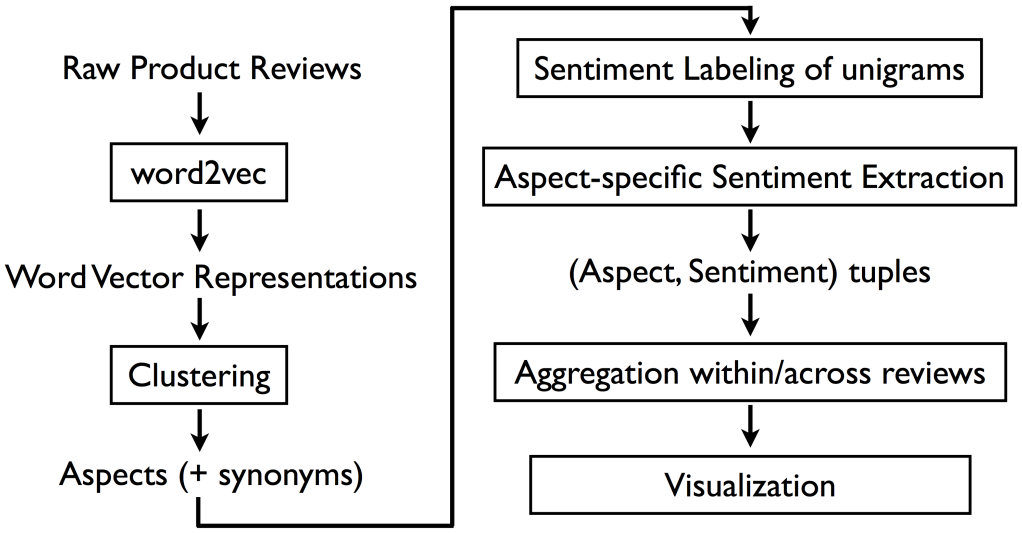

Figure 1: Proposed Workflow. Boxed items are processes, while non-boxed items are outputs. The left half of the diagram details unsupervised aspect identification, while the right half of the diagram involves aspect-specific sentiment extraction, summarization and visualization. Except for the Sentiment Labeling and Visualization processes, every other process involves deep learning.

3.2 Aspect-Specific Sentiment Extraction

An additional pre-processing step would be to label all unigrams with a trained sentiment analysis model. With the aspect-labels from the previous part, we would have a dataset labeled with aspect and (unigram-) sentiment.

Next, we build off and improve on the model in [7] to use a RNTN or improved architecture to extract (aspect, aspect-specific sentiment) tuples from the data.

3.3 Aspect-Specific Sentiment Aggregation

The final part of our project involves aspect-specific sentiment aggregation across sentences within a review, and at a further step, aggregating across reviews. We are still currently reading relevant literature (e.g. [8-10]) to come up with ideas on how to optimally construct this. A non-deep learning approach might be to hand-specify features (review length; timing – more recent reviews might be more important; etc). In contrast, a deep learning proposal would be to have a deep learning network (such as a recursive neural network with individual reviews as tokens) to try to come up with an optimal way of aggregating this information. We plan to scrape professional review websites like Consumer Reports, CNET and/or DPReview, to provide gold standard summary reviews that our model will predict.

An application-related goal would be to have visualization of our output, in terms of a word cloud of important aspects (e.g., sized by word frequency and colored by sentiment), or a bar chart that shows the relevant aspects and their associated sentiment.

4 Intermediate/Preliminary Experiments & Results

4.1 Preprocessing

We tokenized our corpus using NLTK's punkt tokenizer [12] for sentence splitting. We then removed all non-alphanumerical characters and replaced all digits with `DG`. Finally, we performed collocation detection to detect common bigrams.

4.2 Model Training

We trained three different word2vec models. The most successful model was trained using the CBOW (continuous bag-of-words) model with a window size of 10 and feature dimension size of 300. We also ignored all words with total frequency count below 40 (this helped to remove many typos). We determined the performance of each model by querying the model with various aspects common to electronic products (e.g. "portability", "screen quality", etc).

4.3 word2vec Results

We queried our word2vec model and returned the top-10 results based on cosine similarity of the word vectors.

Query	Results
`portability`	(u'portability,', 0.72859996557235718), (u'compactness', 0.64743077754974365), (u'mobility', 0.60842603445053101), (u'versatility', 0.5763777494430542), (u'simplicity', 0.53962129354476929), (u'lightness', 0.53950369358062744), (u'convenience', 0.53897607326507568), (u'ruggedness', 0.5272858738899231), (u'versatility,', 0.5055851936340332), (u'thinness', 0.49253776669502258)
`contrast`	(u'contrast,', 0.65686732530593872), (u'sharpness', 0.62712550163269043), (u'color_saturation', 0.60933655500411987), (u'saturation', 0.57076853513717651), (u'brightness', 0.5553707480430603), (u'gamma', 0.53090476989746094), (u'shadow_detail', 0.52805298566818237), (u'color_accuracy', 0.52408510446548462), (u'dynamic_range', 0.52167940139770508), (u'black_levels', 0.51741272211074829)
`tripod`	(u'monopod', 0.74430769681930542), (u'tripod,', 0.71975594758987427), (u'ball_head', 0.70861411094665527), (u'tripods', 0.68399727344512939), (u'ballhead', 0.60356354713439941), (u'manfrotto', 0.59598124027252197), (u'monopod,', 0.58229464292526245), (u'pole', 0.56997144222259521), (u'quick_release', 0.549965500831604), (u'cold_shoe', 0.5460544228553772)

As shown in the table above, words like `portability` returned many synonyms as well as product aspects that are related to it (e.g. `ruggedness` and `simplicity`). We also find that our model is capable of returning aspects that are specific and unique to the product category it is trained on. In this case of electronic products, a query like `contrast` returned words like `shadow_detail`, `dynamic_range`, `black_levels` and `gamma`, which are aspects specific to devices like monitor displays and cameras.

A query like `tripod` returned various aspects camera tripods, many of them being features that are non-obvious to the layperson. For instance, `ball_head` refers to a ball-type tripod heads and `quick_release` refers to quick-release tripod mounts. Many of these queries would otherwise perform poorly if we were to use lexical databases like WordNet.

Acknowledgments

We would like to acknowledge funding for computing resources provided by the Deep Social Learning Lab at Stanford.

References

[1] Titov, I., & McDonald, R. T. (2008). A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *ACL* (Vol. 8, pp. 308-316).

[2] Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In Proceedings of the fourth ACM international conference on Web search and data mining (pp. 815-824). ACM.

[3] Brody, S., & Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 804-812). Association for Computational Linguistics.

[4] Engonopoulos, N., Lazaridou, A., Paliouras, G., & Chandrinos, K. (2011). ELS: a word-level method for entity-level sentiment analysis. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*

[5] Moilanen, K., & Pulman, S. (2009). Multi-entity Sentiment Scoring. In *Recent Advances in NLP* (pp. 258-263).

[6] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)* (Vol. 1631, p. 1642).

[7] Lakkaraju, H., Socher, R, & Manning, C. (2014). Aspect Specific Sentiment Analysis using Hierarchical Deep Learning. *NIPS Workshop on Deep Learning and Representation Learning*

[8] Ghose, A., & Ipeirotis, P. G. (2007). Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *Proceedings of the Ninth international conference on Electronic commerce* (pp. 303-310). ACM.

[9] Chen, P. Y., Dhanasobhon, S., & Smith, M. D. (2008). All reviews are not created equal: The disaggregate impact of reviews and reviewers at Amazon.com. Available at SSRN: http://ssrn.com/abstract=918083

[10] Dai, W., Jin, G. Z., Lee, J., & Luca, M. (2012). Optimal aggregation of consumer ratings: an application to Yelp.com (No. w18567). National Bureau of Economic Research.

[11] McAuley, J., Targett, C., Shi, J., & van den Hengel, A. (2015). Image-based recommendations on styles and substitutes. *ACM Special Interest Group on Information Retrieval (SIGIR)*

[12] Bird, Steven, Loper, E. and Klein, E. (2009). Natural Language Processing with Python. OReilly Media Inc.