Extracting and Aggregating Aspect-Level Sentiment from Product Reviews

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

Abstract

The abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

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 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 Dirchlet Analysis [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 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 [1-2,4-7], except [3], are supervised methods that require labeled aspects/sentiment for training. We propose using unsupervised methods to generate candidate aspects via 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

Describe your problem precisely specifying the dataset to be used, expected results and evaluation

Amazon Reviews [1]

3 Technical Approach and Models

1. Run word2vec to get an embedded representation
2. Start with a small (hand-generated) seed list of attributes (“Entities”).
    - Build Construction
    - Price
    - Battery Life
    - ...
3. Use the word2vec representation to find similar words/phrases. (This process might also generate adjectives or common phrases used to describe this particular attribute.)
    - Build Construction, Build Quality
    - Price, Cost, (Expensive, Cheap, Affordable)
    - Battery Life, (Long-Lasting), Charging
    - ...

    Desmond and Shane’s discussion: it might be counter-intuitive but beneficial to treat Expensive as an attribute (for Price). The NER model will be doing something “wrong” in labeling an adjective as an entity, but you’ll be able to capture the price information when someone says “expensive”.
4. Use this list in an NER model to teach it to learn to recognize attributes as a new class of entities.
5. Next, given that attributes are labelled in the text, we can run sentiment analysis.
    - Most naively, we can just label all the unigrams. Then, we can just take every sentence, find all the attributes listed in that sentence, and then associate the sentiment of the entire sentence to those attributes.
    - More sophisticated models would involve looking at local context and compositionality. Thus, we could run a window (instead of the whole sentence), calculate the sentiment associated with the words in the window (either at the unigram level or at the subtree level, a la Socher et al), and associate that sentiment with the attribute found within the window.
6. Then we would have to sum up the
7. The output would be a (long and sparse) attribute vector where the i-th element is the sentiment associated with the i-th attribute:
    - E.g. attribute$_1$ = build quality, attribute$_2$ = cost
    - If product has good build quality but is very expensive, then outputAttributeVector = [4, 0, . . .] (scale from 0-4)
8. Visualization of output vector.

When one is planning a purchase, one usually has to spend time sifting through multiple reviews across many different websites, distilling the information to get (1) ratings for individual attributes (battery life), and (2) an overall rating / summary of the information, presented in a meta-review. Four separate sub-problems: 1) Doing POS-tagging and named entity recognition for various product attributes (e.g. battery life, weight, portability, etc) (well probably use off-the-shelf (Stanford NLP) tools for this problem) 2) Train word2vec representations of said attributes and clustering them (e.g. build quality and construction can refer to the same attribute) 3) Extract sentiment based on product attributes 4) Aggregate this information across multiple reviews (creating a meta-review) A potential application allows for quick comparisons of various products based on user-generated reviews. Say you are purchasing an Android tablet and are looking at the Samsung Galaxy Tab 4, Nexus 9, Nvidia Shield and other similar products, our system would allow users to identify key attributes of each product (e.g. screen resolution, build quality, battery life, etc) and their associated sentiment, i.e. how positively or negatively do people feel about each product attribute. Given the constraints of a class project, we will be limiting our dataset to reviews of electronic consumer products.

POS and NER: Using off-the-shelf tools Clustering entities: cluster learned word2vec representations (e.g. K-NN or distribution-based clustering methods) Extract entity-level sentiment: modifying and extending Socher et al (2013)s model via calculating the sentiment in a window/neighborhood around the named entity, up to the whole sentence. Aggregation of sentiment (across multiple sentences, and across multiple reviews): this is perhaps where deep-learning might be extremely useful, but as yet we do not know enough about various deep learning architectures to do this optimally. There are several papers that we identified that might give us some insight (although they dont use deep learning)

Describe the methods you intend to apply to solve the given problem

4 Intermediate/Preliminary Experiments & Results

State and evaluate your results up to the milestone

We can evaluate the attribute sentiments against the formal ratings provided by CNET or consumer reports. An aggregation of sentiment for individual reviews can also be compared against a users overall rating of the product. We can evaluate the attribute clustering using standard cluster metrics. A projection to a 2-D space for visualization would also be valuable.

Acknowledgments

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)*