ExtRA: A Framework for Extracting Prominent Review Aspects from Customer Feedback

Zhiyi Luo [1], *Bill Y. Lin* [2], *Frank F. Xu* [3], *Shi Feng* [4], *Hanyuan Shi* [5], *Kenny Q. Zhu* [6]

Department of Computer Science & Engineering
Shanghai Jiao Tong University, Shanghai, China
{jessherlock[1],yuchenlin[2],frankxu[3]}@sjtu.edu.cn,
{sjtufs[4],shihanyuan1995[5]}@gmail.com, kzhu@cs.sjtu.edu.cn[6]

***Abstract*—We appreciate the precious comments given by the reviewers and thank them for their hard work. Below, we list the questions of each reviewer and provide our answers accordingly. For the revision requirements, we mark the corresponding revised parts in the paper in bold for easy identification. Below are our responses to the individual reviews.**

I. Response to Reviewer 1

Q1: While I pointed out in S2 that the proposed framework includes all the main steps, one concern I have is that the various steps are not novel.

A1: Some of the NLP techniques included in our framework, such as sentence vector representation, sentence clustering and topic modeling, are not new. However, our framework is not a simple "re-sequence" of these techniques. We propose innovative methods to combine those techniques to better represente the aspects. For example, in Stage 3, instead of solely utilizing the word distributions after topic modeling, we propose to cluster the topics (word distributions) by representing them as "topic vectors." Those topic vectors are constructed by incorporating word embeddings as external information. Another example is our AspVec. We eventually represent each potential aspect as an aspect vector, which can 1) better compute the aspect similarity, 2) keep the representations in original word vector space, Kenny: What do you mean? 3) and thus enables extracting aspect phrases at the same time.

Q2: It is also well known in the NLP domain that extracting keywords as topic labels are generally inadequate. There are various studies in the broader NLP literature that extracts aspects automatically. E.g., the 2014 study by Carenini et al. on summarization of product reviews using discourse structure.

A2: Carenini et al. (2014) leverages the discourse structure and discourse relations in the product reviews to select important aspect words for the further abstractive summarization. We discuss this related work in the revised version.

Q3: Topic phrases are now believed to be more advanced. This paper still operates under the paradigm of keywords; see Table V for instance.

A3: Our framework can extract both aspect terms and phrases. The description of the phrase extraction method is in "AspVec-based Extraction", stage G, of sec. II. We asked the human annotators to provide words as groundtruth in Table V for two reasons: 1) to enable easy comparison between our model and other strong baselines (e.g. MG-LDA, shown in Table VII) that can only extract aspect words; 2) to ensure higher inter-annotator agreement among the five annotators.

Q4: The experimental results are also not convincing. The datasets summarized in Table IV are still in product domains that can be constructed manually. What the paper needs would be datasets where aspects are hard to be constructed manually, which is the original motivation of the paper.

A4: Our experiments were performed on both product and service (e.g. restaurant) reviews. Although such aspects can be extracted by human efforts, it would be very expensive to do this for thousands of product or service types. Our experiment is actually a proof of concept with an evaluation metric purposed designed to compare human and algorithmic performance. Our experiments also show that our framework and algorithm can be applied on larger text corpora and on other domains where aspect terms need to be mined.

Q5: The topic cluster ranking in stage 4 seems very arbitrary. There needs to be stronger justification on why this is better than ones in the literature, of which there are many.

A5: We rank the topic clusters by their distinctiveness scores in stage 4. This score indicates the amount by which a cluster overlaps with others. This can help remove redundant top clusters as shown in Table II.

II. Response to Reviewer 2

Q1: The approach requires tuning a lot of parameters. The authors indicate in the paper how they have selected the values for the parameters. However they do not discuss whether their results would still be valid for different values of these parameters. Also they do not discuss how such parameters would be tuned when the framework is used in a real setting. Would the parameters be set by the end-user?

A1: Most of the parameters are tuned without knowing the ground truth aspect words. For example, we evaluate the parameter N (the number of the sentence clusters in part B, sec. III) is based on an unsupervised loss, which is the sum of euclidean distances from cluster center to each point within the cluster (Figure 5). Such parameters can be tuned in the same way in a real setting. Moreover, end-users usually only care about the desired number of aspect words, and other parameters of our framework can be automatically tuned in such an unsupervised way.

Q2: I appreciate particularly the use of ontology to improve the clustering. One question, though, is how to obtain/generate the ontology in real settings.

A2: Our parameters have been set for real settings. Kenny: What is this??

III. Response to Reviewer 3

Q1: The mapping of words to the vector space is not explained. Several embedding techniques for sentences are mentioned, but they paper does not say, which one has been chosen.

A1: In our framework, we used word2vec (skip-gram) model to embed the discrete words from the vocabulary of the product reviews into a low-dimensional vector space, commonly called word embedding space. The word vectors are trained using the whole corpus of review sentences. Also note that the word vectors, phrase vectors as well as aspect vectors showned in Fig.3 are simple linear combination of word vectors stated above (i.e., weighted average/sum, etc.), thus they are all in the same embedding vector space, where they can be safely compared against each other.

For sentence embedding techniques, we mainly compared two different methods, Paragraph Vector and LSTM-based AutoEncoder (as stated in II.C Stage 1). Actually the experimental results of both techniques are shown in later experiments (TABLE VII and VIII), dubbed ExtRA-PV and ExtRA-LSTM. Since the evaluation of this task is more open ended, we regard both variants as possible approaches.

Q2: The second version of the approach, which includes the extension to phrases, is only sketched. The description of the AspVec variant, which allows for the extraction of aspect phrases is only sketched. For instance, Fig. 3 shows that for this variant, word vectors and and quality phrases are mined, which are processed differently from the sentences in the first variant.

A2: As for the phrase-capable version of AspVec, we first mine quality phrases using SegPhrase/AutoPhrase (Shang et. al.) from review corpus as candidates. Then, by combining the words in a given phrase by averaging their word embeddings, we can easily obtain the phrase embedding while ensuring that they are in the same embedding space. Now the vocabulary has been expanded with newly added candidate quality phrases, along with original words. The rest of the process are the same since the phrases are just tokens with embedding vectors from now on (See Fig 3). We have revised this part of the paper to make it clearer and more detailed.

Q3: The new datasets and the open-source implementation, which are listed as contributions of the paper, are not accessible.

A3: At the time when we submitted the paper initially, we were working on organizing the datasets and the code base (adding comments, readme, dataset introduction, etc.) We believe that reproducibility is highly important and will make the datasets and code publicly available by the time the revised paper is submitted.

Q4: The measure of accuracy used in the experiments is unintuitive. An accuracy of 100% is only reachable if the human annotators agree completely on their terms. It would be better to have a measure that makes it clear how close a method is to what could be achieved in principle. For instance, one could count the total number of occurrences of the five most frequent terms supplied by the human experts and take that as a benchmark. Since that would still have the drawback that missing one of the terms could lead to a significant reduction, the authors may want to think further about a more appropriate measure.

A4: In the revision, we adopt a new measure for accuracy which achieves the reviewer's requirement and allows a method to achieve 100% accuracy, even if the human annotators did not completely agree on their terms. We follow the reviewers advice to select the five most frequent terms provided by the human annotators and take those as the ground truth. Because our ground truth consists of only 5 terms, counting the number of matches makes the accuracy score very discrete and coarse-grained. To solve this problem, we use the semantic similarity between terms as a soft accuracy measure. First, we align each generated aspect with one golden aspect. Then we calculate the semantic similarity of each aspect and its corresponding golden aspect term as the soft matching score (from 0 to 1) by using the standard pretrained word2vecs (i.e., Glove). We use the summation of such soft matching scores to represent the accuracy of our model. Such measure for accuracy can achieve 100% even when the human annotators do not completely agree with each other.

Q5: There is no explanation as to the mapping of words to vectors. Several embedding techniques for sentences are mentioned, but they paper does not say, which one has been chosen. In addition, the description of the AspVec variant, which allows for the extraction of aspect phrases is only sketched. For instance, Fig. 3 shows that for this variant, word vectors and and quality phrases are mined, which are processed differently from the sentences in the first variant.

A5: Please refer to A1 and A2 above.

Q6: The explanation of the word embeddings and topic modeling should also explain why the terms in topics and

clusters are always nouns. Is the approach only applied to nouns? Also, why is it the case that words expression opinions do not appear there? Is the reason just that they are less frequent than objective nouns?

A6: Our framework mainly aims to extract the prominent aspects from different kinds of custom reviews. Such expected aspects are usually nouns or noun phrases (B. Liu et. al.). Instead of explicitly extracting the opinion words from topics for sentiment analysis in the downstream application, we use a LSTM-based neural network model trained on the Stanford Sentiment Treebank to compute the sentiment score for the target aspects. We give an extended discussion on this topic in the revised paper.

Q7: When opinions about aspects are extracted in the downstream analysis, the authors used apparently not only the aspect terms, but also other terms of topic cluster. How were those terms extracted so that interference between clusters was minimized?

A7: In the downstream application, the sentiment or opinion terms (e.g., in Fig. 9) targeting a particular aspect can be automatically extracted from the same sentence containing the aspect term in question. We achieve this in this paper by a LSTM-based neural model trained on Stanford Sentiment Treebanks. This is, however, not the focus of this paper. Kenny: can you sleep on the above a bit more?

Kenny: Is the following really necessary? I thought we have said in the abstract that we will bold the changes in the revision? No point repeating what they asked for?

1) Reviewer 2: R1. The paper should address R1 (if space is needed, the application demo part can be reduced)

R2. The paper should discuss approaches for D1.

2) Reviewer 3: R1: Extend the description of the AspVec variant so that it becomes clear how phrases are generated. Extend experiments to the phrase generation variant. Show examples of phrases.

R2: Provide more details about the word embedding used in the experiments. Which framework did you use and why (e.g., Google's word2vec vectors trained on the Google News corpus)?

R3: Find a measure for accuracy that allows a method to achieve 100% accuracy, even if the annotators did not agree on the terms and express the performance using that measure.

R4 : Explain how the clusters for opinion extraction were created.