Unsupervised Aspect Extraction from Free-form Conversations

En-Shiun Annie Lee
Verticalscope Inc
111 Peter St
Toronto, Ontario
alee@verticalscope.com

Richie Wenjie Zi
Verticalscope Inc
111 Peter St
Toronto, Ontario
wzi@verticalscope.com

Afsaneh Fazly
University of Toronto
27 King's College Circle
Toronto, Ontario

Brandon Seibel
Verticalscope Inc
111 Peter St
Toronto, Ontario
bseibel@verticalscope.com

Anderson De Andrade
Wattpad
Toronto, Ontario

ABSTRACT

Aspect-based sentiment analysis on forum data can produce a wealth of knowledge due to the massive and free-form nature of the discussions involved. Existing works in aspect extraction for sentiment analysis include: 1) simple frequency count of noun phrases relying on labelled datasets for learning supervised models, and 2) topic modelling trained on large unlabelled datasets requiring tuning of complex parameters. Our goal is to efficiently and effectively extract aspects (features and attributes) of certain *entities* (products or brands) from massive heterogenous collections of user-generated free-form conversations. We construct an aspect dictionary in three steps: 1) first we extract candidate aspects using simple lexico-syntactic patterns that capture the "aspect-of" relation between a noun phrase and a mention of an entity; 2) next, we filter the candidates by drawing on an automatically-compiled commonness blacklist, as well as a neighbourhood-based measure of aspecthood; and 3) lastly, we expand the dictionary to increase coverage using a variety of simple techniques. When compared to state-of-the-art methods for aspect extraction, our method is capable of efficiently constructing an extremely compact aspect dictionary (98% more compact) with comparable performance.

CCS CONCEPTS

•Information Systems → Aspect Extraction; Sentiment Analysis;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference format:
En-Shiun Annie Lee, Richie Wenjie Zi, Afsaneh Fazly, Brandon Seibel, and Anderson De Andrade
. 2017. Unsupervised Aspect Extraction from Free-form Conversations. In *Proceedings of SIGKDD Conference, Halifax, Nova Scotia, Canada, August 2017 (SIGKDD 2017),* 7 pages.
DOI: 10.475/123_4

SIGKDD 2017, Halifax, Nova Scotia, Canada
© 2017 Copyright held by the owner/author(s). 123-4567-24-567/08/06. . . $15.00
DOI: 10.475/123_4

1 INTRODUCTION

Understanding the reasons why consumers like or dislike a certain product or service — a.k.a., aspect-based sentiment analysis — is important for businesses and their customers. It helps businesses who are selling products or services to strategize their marketing campaigns and for their potential customers who are buying those products or services to make purchase decisions. An important step in aspect-based sentiment analysis is to identify terms (words and phrases) that refer to important parts, features, attributes, or properties (a.k.a., *aspects*) of a targeted product or service. Over the past decade, there has been substantial research done on *unsupervised aspect extraction*. Most existing work focuses on extracting aspects of a particular product or service from a *homogeneous* collection of customer reviews already associated with the target product or service.[1] In contrast, we mine aspects from a *heterogenous* collection of forum conversations that contain references to aspects of many different products and services. More specifically, the forum conversations contain discussions that may not be relevant to the products of interest, as well as informal reviews of products scattered throughout a conversation.

Forums are venues in which people with common interests engage in free-form conversations, ask questions, or discuss issues concerning a particular topic (e.g., fishing, pets, health, or cars). Identifying aspect terms in forum conversations is a particularly challenging task because the aspect terms are located in a noisy context. These noisy context includes relevant discussions about the target product aspects that needs to be identified, as well as irrelevant discussions that must be ignored.

Previous work on unsupervised aspect extraction ranges from 1) manually defining classes of features and the words that represent them [21], to 2) simply extracting recurrent noun phrases as aspects [5], to 3) relying on (often manually-built) morpho-syntactic patterns [10, 16]. Several early studies identify aspect terms by first

[1] The only exception is the preliminary study of [9] on blogs and message boards, which involves the manual identification of aspect terms.

grouping similar sentences (often represented as bags of words or noun phrases), and then finding frequent key phrases as potential aspect expressions [3, 7, 8, 11]. Such frequency-based techniques often miss the rare infrequent but still valid and sometimes important aspects. Others use a variation of the Latent Dirichlet Allocation (LDA) algorithm for topic modelling to extract aspect terms from review sentences [1, 18]. These LDA techniques extract many good topics for aspects, but also poor topics such as global concepts and topics that are not necessarily referring to a part, a feature, or an attribute of a product or service. Recently, a group of bootstrapping algorithms have been proposed that draw on manually-identified grammatical relations between opinion expressions and aspect terms to simultaneously identify them [4, 12, 17, 20]. These algorithms (known as variants of Double Propagation) also often include many noisy non-aspect terms in their extractions. In particular, such approaches are not appropriate for heterogeneous data (like ours) with many off-topic conversations with references to many products and services. In addition, these approaches rely on accurately identifying the relevant grammatical dependency relations.

In light of the above shortcomings, we propose a completely unsupervised system for extracting aspects from heterogeneous data. Due to the nature of our heterogeneous data, it is crucial that we rely on a technique that is robust to the inherent noise in the data. Our approach is inspired by the pattern-based methods [10, 16] in that it relies on manually-constructed lexico-syntactic patterns to extract an initial set of candidate aspect terms. However, we perform extensive automatic filtering to purify the candidate terms by 1) a novel automatic algorithm for filtering generic terms that are likely to be non-aspects, and 2) relying on state-of-the-art neural word embeddings.

Our contributions are two fold in the new end-to-end system and novel purification techniques. First, to the best of our knowledge, this is the first system for extracting aspects from massive free-form heterogenous non-review text. Second, we propose two novel methods for purifying an initial list of extracted candidate aspects. Our experiments confirm the superiority of our system to the state-of-the-art Double Propagation algorithm when dealing with large volumes of free-form heterogenous forum conversations: our dictionary is more compact than the existing methods while maintaining comparable quality.

2 METHODOLOGY: UNSUPERVISED ASPECT EXTRACTION

Due to the massive amount of unstructured posts streaming in the forum discussions, having an off-line component (constructing a dictionary of core aspect terms), and an on-line component (identifying aspect terms in context) is crucial. Forum conversations are free form, where people not only write about their experience with (and possibly their opinions towards) aspects of certain products, services, and brands (a.k.a., *mentions*); but also about their general interests, such as locations they have visited over the weekend, movies they have recently watched, or even their stance on a hot political issue. It is thus important to link each aspect to its referent entity mention, in order to accurately learn consumers' sentiments towards an entity of interest. To focus our aspect extraction on the relevant portions of the discussions, we only consider sentences that contain a mention of entities of interest. To do so, we first run a named entity recognizer on our data, trained with forum posts annotated for a set of target entities — i.e., products and brands. Throughout the paper, we use the term *entities* to refer to products and brands. Next, we explain how the steps involved in constructing a dictionary of core aspect terms.

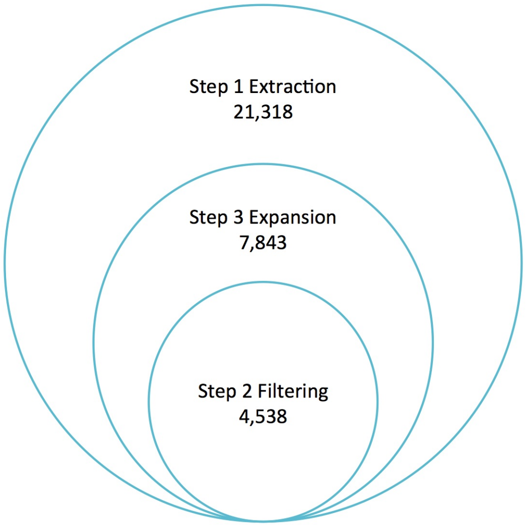

Figure 1: The Venn Diagram demonstrating each step of the Extraction approach.

The purpose of constructing an off-line aspect dictionary is to allow for the fast matching of aspects when there is a high volume of incoming posts, as is the case with online discussion forums. Our approach to construct an aspect dictionary consists of three steps (Figure 1), explained in detail in the following subsections.

Step 1: Candidate Extraction: extracting an initial list of candidate aspect terms.
Step 2: Purification: purifying candidate list by applying several automatic filtering techniques.
Step 3: Expansion: expanding the aspect dictionary by automatically adding missing terms.

2.1 Step 1: Candidate Extraction

This step uses a pattern-based approach to extract noun phrases (**np**s) that could potentially be in an "aspect-of" relationship with an entity mention. [2] The intuition is that important aspects tend to be expressed in the presence of their corresponding entity mentions. Furthermore, we observed that these patterns are simple enough to cover many aspects and are frequently used. Like most previous work on aspect extraction, we also assume that aspects are expressed as noun phrases. We thus focus on sentences that contain entity mentions, and consider noun phrases as aspect candidates. Through analysis of the literature on pattern-based relation extraction [6], we identify two general lexico-syntactic patterns that we believe capture a variety of "aspect-of" relationships between a mention and

[2] We only extract aspects that are nouns and noun phrases, and do not consider implicit aspects.

its potential aspects. Basically, we extract as candidate aspect terms all noun phrases that appear in one of the following two general relations with a mention (explained below in detail).

PATTERN$_1$:

mention `pr:`*with* `[dt]` **np**

SAMPLE MATCH:

> "I would not want an XE$_{\text{MENTION}}$ *with a* **diesel engine**$_{\text{ASPECT}}$ for various reasons."

INTERPRETATION: This pattern extracts as an aspect candidate for every **np** that is inside a prepositional phrase headed by *with* and attached to a **mention**. The intuition is that the **np** and **mention** are grammatically dependent, in that the *mention* has the **np** as a part, attribute, or feature — hence the two have a part-of, attribute-of, or feature-of semantic relation.

Table 1: Description of notation used for patterns.

pr	preposition
dt	determiner
jj	adjective
nn	noun
nu	number term
wh	*wh*-word

PATTERN$_2$:

`~(nu|jj|wh)`**np** `pr:`{*of, on, in, inside*} `[dt]` `[jj|nu]` **mention**`~nn`

SAMPLE MATCH:

> "I have a problem with the **radiator fan**$_{\text{ASPECT}}$ *on my* Accord$_{\text{MENTION}}$ 1991"

INTERPRETATION: This pattern extracts as an aspect candidate every **np** with a prepositional phrase attached that is headed by a relation preposition (i.e., *of*) or a locative preposition (i.e., *on*, *in*, and *inside*) followed by the **mention**. The core of this pattern focuses on extracting **np**s with an aspect-of relation to a **mention**. There is, however, constraints imposed by this pattern with respect to what precedes the **np** and what follows the **mention**. Specifically: (i) **np** may not follow by a numeral, an adjective, or a *wh*-word; and (ii) **mention** may not precede an adjacent noun. The former restriction filters out units of measurement (*11 meters*), relative clauses (*what kind*), and abstract and other non-aspect nouns that tend to be modified by descriptive adjectives (*great challenge*). The latter restriction is to ensure that the mention is actually the head of a noun phrase (hence the focus here), in contrast to the cases where the mention acts as a modifier to another noun (e.g., *personality of BMW guys*, in which *personality* is an aspect of *guys* and not of *BMW*).

To ensure that a given **mention** and the extracted **np** are part of the same phrase, we examine the PoS of the words preceding **np**: if we hit a preposition and then a verb (without encountering a noun), we assume **np** is part of another (verb) phrase and discard it as a candidate. E.g., *mom* in *I talked to my mom in her BMW* should not be extracted as an aspect of *BMW*. In addition, we remove any term whose total frequency of appearing in the above patterns is 1.

2.2 Step 2: Purification

The original list of candidate terms that our two patterns extract still contains many incorrect aspects. For example, the terms *pair* and *something* are incorrectly extracted as aspects from the following two sentences:

"... a set of batteries out of a **pair**$_{\text{ASPECT}}$ *of* Nissan Leafs$_{\text{MENTION}}$."

"Find a ZJ$_{\text{MENTION}}$ *with* **something**$_{\text{ASPECT}}$ irrelevant wrong with it but looks great."

The above examples are common idioms that contains the pattern and thus are frequent. The goal of purification is to remove such erroneous extractions. We do so by applying two filtering techniques: one removes highly common terms that we believe are less likely to be good aspects, and the other one looks at evidence from the neighbourhood of a term (semantically-related terms) to decide its aspecthood likelihood.

(a) Commonness-based Filtering: The initial list of candidate aspects may contain many high-frequency noun phrases that are not true aspects of the target entities, but may still appear in the above patterns. For each candidate term, we measure a *commonness score* that summarizes the occurrence pattern of the term across several domains. We assume a term is *common* (and hence not likely to be a domain-specific aspect), if the term appears with an overall high frequency across many domains. We measure a commonness score for each term by looking at its patterns of occurrence in large collections of forum posts across 10 different domains — including automotive, powersports, outdoors, technology, health, pets, home and garden, collectibles, and others. Specifically, given a matrix of word–domain frequencies compiled from these collections, we calculate commonness(a) for a candidate aspect term a as follows:

$$\text{commonness}(a) = \frac{\text{popularity}(a)}{\text{divergence}(a)} \tag{1}$$

$$\begin{aligned}\text{popularity}(a) &= \log \text{freq}(a,*) \\ &= \log \sum_{j=1}^{N_d} \text{freq}(a,d_j)\end{aligned} \tag{2}$$

$$\text{divergence}(a) = D\Big(P(d|a)||P(d)\Big) \tag{3}$$

where a is the candidate aspect term and d_j is a domain in $\{d_1, d_2, ..., d_{N_d}\}$ such that N_d is the number of domains under study, here 10. Popularity of a term is measured by its overall frequency across domains (in the log scale), and divergence of a term is measured how much the distribution of the term across domains diverges from a "typical" distribution. Divergence is measured as the KL-Divergence between two probability distributions:

$$D\Big(P(d|a)||P(d)\Big) = \sum_{j=1}^{N_d} P(d_j|a) \times \log \frac{P(d_j|a)}{P(d_j)} \tag{4}$$

, where

$$P(d_j|a) = \frac{\text{freq}(a, d_j)}{\text{freq}(a, *)}$$

$$P(d_j) = \frac{\text{freq}(*, d_j)}{\text{freq}(*, *)} = \frac{\sum_{a'} \text{freq}(a', d_j)}{\sum_{a'} \sum_{d'} \text{freq}(a', d')}$$

$P(d|a)$ that is the **posterior** (observed) distribution of domain d given term a, and $P(d)$ that is the **prior** (expected) distribution of domain d. Note that the more divergent the different domains are from one another, the better the commonness score will perform. See Eqn. (4) for details. Basically if a term appears in all domains with frequencies that are proportional to each domain size, the term is considered as having a very small divergence.

The intuition behind our commonness score is that true aspect terms tend to be heavily domain-dependent. Hence, if a term appears in many domains with frequencies that are typical of that domain, then the term is less likely to be a domain-specific aspect term. In other words, if a candidate term is common across several domains, there is a high likelihood that it is not an aspect. We understand that there might be exceptions to such assumption, (e.g., terms such as *price*, *issue* may appear in many domains). Nonetheless, in our experiments we find that most terms with a high commonness are generic non-aspect terms, such as *part*, *stuff*, *everything*, *anyone*, *cons*, and the like.

(b) Neighbourhood-based Filtering: Aspects are often semantically related, e.g., there are many terms that refer to the pricing of a product, including *price*, *fee*, *charge*, *cost*, *base price*, *msrp* (manufacturer's suggested retail price), and more. Thus, we assume that a candidate term is more likely to be a valid aspect if it is more semantically related terms are also in the list of extracted candidates (and have a high frequency of occurrence in our patterns). We thus assign an *aspecthood* score to each candidate term a based on the frequency of occurrence of its *neighbours* (i.e., semantically similar terms) in our patterns. We use this aspecthood score to filter out candidates that are less likely to be good aspects.

Specifically, we measure aspecthood of a term a based on evidence from its top $k(= 10)$ similar terms that also appear in the set of aspect candidates, referred to as neighbours of a or $\mathcal{N}(a)$:

$$\text{aspecthood}(a) = \frac{\sum_{t \in \mathcal{N}(a)} \text{freq}(t) \times \text{sim}(a, t)}{|\mathcal{N}(a)|} \tag{5}$$

where $\text{sim}(a, t)$ is the similarity of a and t, measured as the cosine of the angle between the neural word embeddings of a and t.[3]. The intuition is that, if a candidate term a has many neighbours that are likely to be aspects, then a should be considered as a valid aspect term. We use an experimentally-determined aspecthood threshold to decide which candidates should remain as aspects.

2.3 Step 3: Expansion

The extraction step often extract plural or singular versions of an aspect. Therefore, to ensure our aspect dictionary contains important aspect terms, we automatically add to our dictionary: (i) singular forms of current plural aspect terms, and vice versa, subject to a minimum frequency of 1 in our corpus; and (ii) variations of existing compound terms, e.g., *seatbelt*, *seat belt*, and *seat-belt*, again subject to a minimum frequency of 1 in our corpus (see Section 3 for details on the nature and size of our corpus).

3 EVALUATION

3.1 Corpus

We extract our candidate aspect terms from a large corpus of about 3.2 million forum posts, from the automotive domain, that contain at least one mention. Our corpus contains a total of about 383.7 million tokens after pre-processing (e.g., removing url and html tags). We tokenize and tag the posts using the Stanford tokenizer and the Log-linear PoS-tagger [19], and split each post into a sequence of sentences using the Stanford sentence splitter [13].

We validate our method by extracting aspects of *automotive entities*, including Automobile Manufacturers (e.g., *Holden, Bombardier, Honda Motors*), Automaker Names (e.g., *Volkswagen, GM, Ford*), Automobile Brand Names (e.g., *Chevrolet, Fiat, Lexus*), and Automobile Make-Model Names (e.g., *JX35, Mustang, 300TE*). We use the Stanford Named Entity Recognition [2], re-trained on our forum data, to identify instances of these entities in text; hereafter, we refer to these instances as *entity mentions* or simply as *mentions*.

3.2 Gold-standard Evaluation Data

For evaluation, we take a sample of 395 sentences from a variety of automotive forums, in which mentions of automotive entities are specified (through prior crowdsourcing annotation tasks), and we seek crowdsourcing annotations to identify aspects referring to each specified mention. For both tasks — mention and aspect identification, we used an earlier version of the CrowdFlower platform[4] to collect three judgements per annotation unit (a sentence or a sequence of sentences in which the annotators are to identify certain targets, e.g., mentions or aspects). The final dataset annotated with aspects contains 1157 noun phrase instances (780 unique types) that are potential automotive-related aspects. Out of these, 444 (326 unique types) have been identified as "true" aspects by our annotators, and the remaining 713 (454 unique types) are not aspects. We use these annotated sentences as our *gold-standard* corpus $\mathcal{G}$, which we use to evaluate our various aspect dictionaries resulting after each of the three Steps in our system. We evaluate the three steps of our dictionary construction — namely, Extracting candidates, Purifying candidates, and Expansion — by examining the quality of the three dictionaries resulting after each subsequent step. As is common practise in the field, we evaluate each dictionary by using it for aspect identification on our evaluation data (i.e., for identifying aspect noun phrases in our gold-standard corpus), and measuring type-based P(recision), R(ecall), and F1(-score).

[3] Word embeddings are calculated using the CBOW method of Mikolov and Dean [2013], and were provided to us by [15].

[4] https://www.crowdflower.com

Table 2: Comparison with the Baseline.

Aspect Extraction Method	P%	R%	F1
Baseline (all noun phrases)	40.5	92.2	.56
Step 1: Candidate Extraction	48.0	91.0	.63

Table 3: Comparison to Double Propagation (DP).

Aspect Extraction Method	P%	R%	F1	Size
Double Propagation	50.3	96.9	.66	340505
Step 1: Candidate Extraction	48.0	91.0	.63	**6668**

3.3 Results

Comparison with Baseline. In Table 2, we compare the performance of our dictionary built after Step 1 (pattern-based candidate extraction) against a baseline that considers all noun phrases (NPs) in $\mathcal{G}$ as aspects. We tried two different ways of NP chunking: one that chunks NPs based on sequences of part-of-speech (PoS) tags, and one that performs chunking based on dependency relations among nouns. We found that the simple PoS-based baseline performed better, and so we only report results for that one. We believe our higher Precision (an increase of 7.5 percentage points in Precision) is mainly due to the use of lexico-syntactic patterns that work well at capturing true "aspect-of" relations between aspect noun phrases and mentions, thus decreasing the false positive rate.

Comparison with Double Propagation. We also compare our Step-1 dictionary with one resulting from our implementation of the Double Propagation (DP) algorithm. For a fair comparison, we build both dictionaries from the same corpus of sentences that contain mentions, and perform the same simple filtering of hapaxlegomenon (words appearing only once). Our goal is to compare how well the two methods extract candidate terms. DP iteratively extracts as candidates noun phrases that are associated (through co-occurrence) with a set of known seed opinion words and/or with other candidate aspect terms. Our method extracts (in a single step) noun phrases that appear in a certain lexico-syntactic relation to a mention. Results in Table 3 show that our Step-1 dictionary is 98% smaller than the DP dictionary (size: 6, 668 vs. 340, 505). Interestingly, with a significantly smaller dictionary, our method reaches a performance close to that of DP. Importantly, our pattern-based algorithm performs comparable to a complex algorithm (such as DP) without the need for iteratively examining dependency relations among words. Thus our method is particularly suited for quickly building high-quality aspect dictionaries from free-form streaming data, such as forum conversations.

Role of Purification. The purification step removes unlikely aspect terms from the candidate aspects extracted in Step 1. To assess the quality of the two filters in the purification step, we compared our filters (commonness-based filter and neighbourhood-based filter) against the double propagation filter using different thresholds. We found that our commonness-based filter and our neighbourhood-based filter is a more effective filter than the double propagation filter due to its higher precision-recall trade-off. The precision-recall graph (Figure 2) indicates that the precision-recall curve for both commonness-based filter and neighbourhood-based filter lies above that double propagation curve, indicating a higher precision-recall trade-off when filtering candidate aspects. Furthermore, commonness-based filtering shows a slight improvement over neighbourhood-based filtering in precision-recall trade-off as well as demonstrates more stability in precision values at lower recall values (i.e. between 0.0 to 0.2 on the x-axis).

Table 4: Performance and size of the dictionary resulting from the Commonness-based Filtering technique, applied to Step-1 dictionary. The threshold $n_{\text{commonness}}$ is the number of top candidate aspects sorted by commonness score that are filtered out.

$n_{\text{commonness}}$	P%	R%	F1	Size
0	48.0	**91.0**	62.9	6668
200	50.9	84.2	63.5	6468
350	53.1	81.1	64.2	6318
500	55.6	78.6	65.1	6168
*650	58.0	77.4	**66.3**	6018
800	58.4	73.4	65.0	5868
950	59.0	70.0	64.0	5718
1100	59.3	65.0	62.0	5568
1250	61.0	61.6	61.3	5418
1400	61.9	58.8	60.3	5268
1550	61.7	55.4	58.5	5118

Table 5: Performance and size of the dictionary resulting from the Neighbourhood-based Filtering technique, applied to Step-1 dictionary. The threshold $\theta_{\text{aspecthood}}$ is the cut-off value chosen for deciding what values of aspecthood score are to be filtered out.

$\theta_{\text{aspecthood}}$	P%	R%	F1	Size
0	48.0	**91.0**	62.9	6668
1	48.0	91.08	62.9	6664
2	48.3	88.8	62.6	5684
*3	49.0	86.0	**62.5**	5094
4	49.8	83.2	62.3	4592
5	50.0	81.7	62.0	4226
6	50.9	79.9	62.2	3900
7	51.3	77.1	61.6	3642
8	51.6	75.5	61.3	3403
9	52.6	73.1	61.1	3185
10	53.2	71.2	60.9	3008

The goal of the purification is to remove non-aspects from the candidate aspects. To determine the appropriate threshold for each of the two filters, we applied different thresholds of each filter on the Step-1 dictionary. Table 4 and 5 show that the precision improves as more candidate aspects are removed. However, the improved precision must be balanced with the F1, which also begins to decrease as the filter gets stricter.

The best purification result is from removing 650 common non-aspect terms using the commonness-based filtering (F1 of 66.3 compared to 62.9), which increases precision by 10.0 (from 48.0 to 58.0).

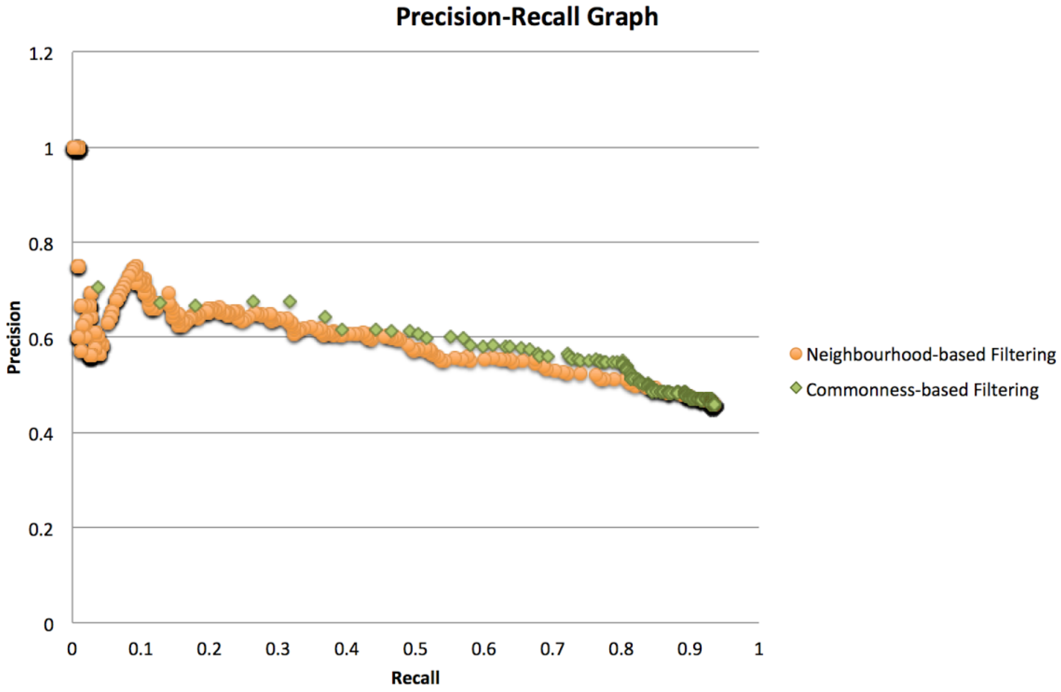

Figure 2: The precision-recall graph for each of the purification methods.

We found that the best threshold for the neighbourhood-based filtering resulted from removing candidate aspects with aspecthood less than 3. With F1 of 62.5 and precision of 49.0, this result yields the best precision while still maintaining high F1.

Overall System Performance. Table 6 shows size of each of our three dictionaries, as well as their performance in aspect identification on the gold-standard corpus $\mathcal{G}$. The thresholds for the filters in Step 2 are selected to improve precision without compromising F1. We observe that the pattern-based candidate extraction (Step 1, without purifying and expansion) has a relatively low precision (48.0) but a high recall (91.0). This high recall is of course expected since the candidate extraction only removes obvious erroneous cases. In addition, while forum data is noisy, it is also massive, and hence by using such general and flexible rules, we can extract many rare aspects to increase recall. Our two filtering techniques (commonness-based and neighbourhood-based) are meant to improve precision by removing candidates that are less likely to be aspects of the target entities (e.g., terms that are common across a variety of domains). The application of both techniques results in a substantial increase in precision (an absolute increase of 12.0 percentage points) while maintaining an acceptable recall (73.4). Looking at the F1 scores, we can see that purification improves the overall performance (62.9 vs. 66.0). Expanding the dictionary (e.g., adding plural and singular forms) boosts recall slightly (an absolute increase of **1.6** percentage points) without dampening precision by much 59.5, and thus results in the best overall performance with a F1 score of 66.3.

3.4 Post-Extraction Evaluation

Recall that our system is composed of an offline dictionary construction — in which we build a dictionary of core aspect terms using the three steps of Candidate Extraction, Purification, and Expansion — as well as an online component that uses this dictionary to extract aspect terms in the forum posts as they arrive. We thus perform a second evaluation of our aspect dictionary by examining a set of the frequent aspect terms it extracts from a large collection of forum posts (one year's worth of data), and verify the correctness of the extractions through post-extraction manual analysis.

Our online component first identifies any terms from our dictionary as a potential core aspect term, and then expands it to the full noun phrase to ensure coverage. For this evaluation, we took a sample of 1938 unique high-frequency aspect phrases extracted in this way. [5] The annotation was completed by one annotator and verified by two experts to result in a final set of consensus annotations. Of the 1938 unique aspect terms that were annotated, 1507 were annotated as true aspects (77.8% true positive) and 431 were considered as non-aspects (22.2% false positive). These results confirm that a majority of high-frequency aspect terms extracted based on our core dictionary are true aspects. However, there is still room for improvement, since the rate of false positives is still undesirably high.

[5] Aspects are chosen by including those that have a frequency higher than 45 in 2015 across all Automotive forums.

Table 6: Performance of each subsequent Step of our Method, as well as Size of the resulting dictionaries; best performances shown in boldface.

Our Method: Subsequent Steps	P%	R%	F1	Size
Step 1: Candidate Extraction	48.0	**91.0**	62.9	6668
Step 2(a): Purification, Commonness-based Filtering	58.0	77.4	66.3	6018
Step 2(b): Purification, Neighbourhood-based Filtering	**60.0**	73.4	66.0	4538
Step-3: Expansion	59.5	75.0	**66.3**	7843

4 CONCLUSIONS

For aspect dictionary construction, our pattern-based algorithm achieves comparable performance to the state-of-the-art while generating a substantially more compact dictionary that is 98% smaller (Table 3) without relying on dependencies. Compared to a baseline aspect identification method (that takes all noun phrases as aspects), identifying aspects based on candidates extracted by the pattern-based algorithm results in a higher precision (an increase of 7.5 percentage points in precision). We believe this increase in precision is mainly due to the patterns capturing true "aspect-of" relations between aspect noun phrases and mentions, thus decreasing the false positive rate. The high recall is due to the use of restrictive lexico-syntactic patterns that are general (contain many function words) and flexible (account for longer or shorter context). Because while forum data is noisy, it is also massive, and hence by using such general and flexible rules, we can extract many rare aspects in addition to common aspects.

The contribution of our unsupervised dictionary construction for free-form heterogeneous data, such as forum conversations, is two folds. First, unlike complex algorithms, our method eliminates the need of expensive dependency relations among words, which is known to be unreliable for free-form heterogeneous data. Second, unlike supervised machine learning algorithms, our method does not require large labelled data sets, which is difficult and expensive to collect high-quality human annotations.

In addition to our simple pattern-based aspect candidate extraction, we propose an effective purification step for filtering "poor" aspect terms, hence increasing the quality of the final aspect dictionary. We proposed an automatic method for quickly compiling a list of commonness non-aspect terms that should be removed from the initial set of candidates, which resulted in an increase in Precision of 10.0 percentage points (Table 6). We also proposed to use evidence from the neighbourhood of a candidate, a method that resulted in a further increase in Precision of 2.0 percentage points (see Table 6).

Nonetheless, our pattern-based method is limited since it requires mentions to be identified prior to aspect extraction. Therefore, the quality of the aspect extraction depends on the performance of the named entity extraction algorithm. Based on the effectiveness and efficiency of the lexico-syntactic patterns for extracting aspects, we are currently devising an unsupervised bootstrapping algorithm that make use of an entity catalogue to extract useful patterns. Future work will focus on examining the possibility of expanding the set of patterns for jointly extracting aspects and named entity mentions.

ACKNOWLEDGMENTS

This work is supported by VerticalScope Inc. The authors would like to thank colleagues Tim Wai, Kyle Noon, Jodi Lai who provided expertise in automotive forums and Kevin Wishart, Li Lin, Ekaphol Petchsri, Justin Huang, and Rochelle Erskine for proofreading. The authors also have to express appreciation to the Paul Cook and SoHyun Park for assistance with word embedding. The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES

[1] Samuel Brody and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *Proceedings of the Annual Conference of the North American Chapter of Association for Computational Linguistics*. 804–812.

[2] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 363–370.

[3] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. 2005. Pulse: Mining Customer Opinions from Free Text. In *Proceedings of the 6th International Symposium on Intelligent Data Analysis (Lecture Notes in Computer Science)*. Springer-Verlag, 121–132.

[4] Qian Liu Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015. Automated Rule Selection for Aspect Extraction in Opinion Mining. In *Proceedings of IJCAI'15*.

[5] Anindya Ghose, Panagiotis G. Ipeirotis, and Arun Sundararajan. 2007. Opinion Mining Using Econometrics: A Case Study on Reputation Systems. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*. 416–423.

[6] Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2006. Automatic Discovery of Part-Whole Relations. *Computational Linguistics* 32, 1 (2006), 83–135.

[7] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 168–177.

[8] Minqing Hu and Bing Liu. 2004. Mining Opinion Features in Customer Reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*. 755–760.

[9] Jason S. Kessler and Nicolas Nicolov. 2009. Targeting Sentiment Expressions through Supervised Ranking of Linguistic Configurations. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*. 90–97.

[10] Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting Aspect-Evaluation and Aspect-of Relations in Opinion Mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1065–1074.

[11] Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of the 14th International Conference on World Wide Web*. 342–351.

[12] Qian Liu, Bing Liu, Yuanlin Zhang, Doo Soon Kim, and Zhiqiang Gao. 2016. Improving Opinion Aspect Extraction Using Semantic Site-Levelmilarity and Aspect Associations. In *Proceedings of AAAI'16*.

[13] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit.. In *ACL (System Demonstrations)*. 55–60.

[14] T. Mikolov and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Processings of the NIPS'13*.

[15] SoHyun Park, Afsaneh Fazly, Annie Lee, Brandon Seibel, Wenjie Zi, and Paul Cook. 2016. Automatically Classifying Out-of-vocabulary Terms in a Domain-Specific Social Media Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

[16] Ana-Maria Popescu and Oren Etzioni. 2005. Extracting Product Features and Opinions from Reviews. In *Proceedings of the Conference on Human Language*

Technology and Empirical Methods in Natural Language Processing. 339–346.

[17] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding Domain Sentiment Lexicon through Double Propagation. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*. 1199–1204.

[18] Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the International World Wide Web Conference (IW3C2)*.

[19] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 173–180.

[20] Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. 2010. Extracting and Ranking Product Features in Opinion Documents. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. 1462–70.

[21] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie Review Mining and Summarization. In *Proceedings of the 15th ACM International Conference on Information and knowledge management*. 43–50.