Is The Grass Greener? Mining Electric Vehicle Opinions

Tommy Carpenter
University of Waterloo
tcarpent@uwaterloo.ca

Lukasz Golab
University of Waterloo
lgolab@uwaterloo.ca

Sohail Javaad Syed
University of Waterloo
sjavaad@uwaterloo.ca

ABSTRACT

Electric Vehicles (EVs) are envisioned to play a large role in the transition from fossil fuel to renewables based transportation. However, their sales thus far are still nominal compared to traditional car sales. It has been difficult for manufacturers to measure owners' initial perceptions in order to build improved vehicles more drivers are likely to adopt. Sentiments regarding EVs have mostly been determined using either field trials, which are expensive to conduct, or large surveys of traditional drivers, which measures public perceptions towards EVs but not the perceptions and experiences of EV owners. We build a system that solves these problems by mining EV owners' sentiments from online forums. Our system has three main uses. First, it graphs the percentage of positive and negative opinions for each vehicle feature of interest, e.g., *battery capacity*, giving the user a high level product overview. There is currently no easily-consumable review system for EVs. Second, it allows the user to read opinions about the specific features they are most interested in without searching though irrelevant text. In our case study, we find only 3% of the comments on EV ownership forums express opinions on the features. The system therefore reduces the space of text the user must read by 97%, even assuming they wish to read all opinions about all features. Finally, in addition to mining the same perceptions found during expensive field trials, our system finds perceptions that were only realized after the owners possessed their EVs for an extended prior of time, i.e., perceptions not available during shorter trials. The system extracts and classifies opinions with a precision and recall of ≈60%, which is on par or better than previous opinion mining systems. We have open sourced our system online.

Keywords

Electric vehicles; sentiment analysis; opinion mining

e-Energy'14, June 11–13, 2014, Cambridge, UK.

ACM 978-1-4503-2819-7/14/067 ...$15.00.
http://dx.doi.org/10.1145/2602044.2602050.

1. INTRODUCTION

As concerns over climate change and oil availability rise, most utilities and auto manufacturers are preparing for the introduction of electric vehicles (EVs) into the electrical and transportation systems.

Major auto manufacturers such as Chevrolet and Nissan have introduced EVs into their product line, and there are now competing manufacturers that specialize in building EVs, e.g., Tesla Motors. However, the sales of these vehicles are still insignificant compared to traditional car sales[1].

manufacturers and researchers have traditionally employed two methods to understand drivers' mobility preferences and requirements, hoping to build improved models more drivers are likely to adopt. First, several organizations have held EV field trials where EVs were loaned to participants in exchange for their feedback. However, because vehicles are expensive and must be shared amongst participants, they are usually limited in both size and duration, thus conclusions are drawn from a small number of (still) inexperienced drivers. Second, researchers have conducted large online surveys to measure drivers' general perceptions towards EVs, but these surveys do not measure EV owners' perceptions (targeting surveys specifically to owners is difficult). These owner perceptions are vital for manufacturers to build improved models more aligned with drivers' mobility preferences and requirements.

We build a system that freely and automatically mines EV ownership forums (e.g., [4, 5]) for these opinions, which are buried in mostly irrelevant text. In our case study using a comprehensive list of EV features, adoption barriers, and a large corpus built from online forums, we find ≈97% of the corpus contains no opinions about any product features. Thus, it is laborious[2] to extract these opinions valuable to prospective buyers, marketers (for determining what features should be advertised) and manufacturers (for determining what features should be improved). With our system, users define the set of features they are interested in, and are presented with a list of positive and negative statements about only those features and several visualizations of this data. Our main contributions are:

[1]In 2013 in North America, Tesla sold 22,450 Model S models, Nissan sold 22,610 Leaf models, and Chevrolet sold 23,094 Volt models [1, 2]. While these sales are largely increased from prior years ([3] gives EV sales per month and cumulative sales), they still represents $< 2\%$ of car sales.

[2]compared to, for example, the ease of buying a digital camera with hundreds of online numerical or star-based reviews

1. We extend previous review mining systems with several new optimizations and EV domain knowledge to build a powerful EV opinion mining system (Section 4).
2. We evaluate our system using a corpus of 330,000 sentences and a manually labeled corpus of 8,000 sentences containing product features. Using these corpora, we demonstrate the system's text reduction capability and its precision and recall (Section 5).

We have open-sourced our system for use [6] because many prior sentiment mining systems are unavailable.

2. TERMINOLOGY

In this paper we use the following notation:

- $\mathcal{P} = \{P_1, P_2, ...\}$ represents a *product space*.
- $\mathcal{F}^p = \{\vec{\mathbf{F}}_1^{\,p}, \vec{\mathbf{F}}_2^{\,p}, ...\}$ represents the *feature space* of $p \in \mathcal{P}$, where $\vec{\mathbf{F}}_i^{\,p}$ is a vector of synonyms describing feature i. For example, for the feature *fuel economy* this vector may be
  $<$ *fuel economy, efficiency, gas mileage, fuel efficiency, mpg..* $>$
- $\mathcal{O}$ represents the *opinion phrase space*, the set of all opinion phrases recognized.
- We refer to an opinion o about feature f as a (f, o) pair.
- Where appropriate, we abbreviate "neutral" with N, positive with $+$, and negative with $-$.

3. OVERVIEW AND RELATED WORK

Customers usually desire certain specifications or features when shopping for expensive products, especially vehicles. One buyer may seek performance while another may look for top safety ratings. For these reasons, we build a *feature-based opinion mining* (FBOM) system. In FBOM, the term "feature" refers to a product feature or attribute. Such a system is concerned with extracting and classifying individual opinionative statements about specific product features, rather than classifying text at the document level. There are five main phases in FBOM:

1. Building a text corpus to be mined
2. Defining or mining the product and product features of interest
3. Extracting sentence fragments from the corpus containing opinions about those features; these fragments denoted as (f, o) pairs.
4. Classifying each (f, o) pair as $\{+, N, -\}$
5. Aggregating results and computing various statistics

In this section, we first discuss previous field trials and surveys conducted to elicit EV opinions, and the problems with these approaches (Section 3.1). We then present work on adjective polarity classification (Section 3.2) and existing FBOM systems (Section 3.3).

3.1 Eliciting Electric Vehicle Opinions

Related work in determining drivers' opinions of EVs is split into two categories: field trials and surveys of non-EV owners. We discuss these works and their limitations here.

Various EV field trials that have taken place [7–15]. In these trials, participants were supplied with EVs and monitored. Monitoring consisted of drivers recording their trip information in travel diaries, surveys and interviews throughout the trials. In some cases, vehicles were also fitted with GPS data loggers that recorded location and charging information. While field trials are useful for drawing conclusions about drivers' experiences with and perceptions towards EVs, they are subject to at least two limitations. First, field trials are expensive because multiple EVs must be purchased or leased for the trial. It is therefore expensive, especially for academic researchers, to conduct field trials with a large number of participants for significant durations. During short trials, drivers may not have time to adjust to driving BEVs or have time to derive well-informed conclusions. Conclusions from field trials are consequently drawn from a small number of still-inexperienced drivers. Large (in terms of number of participants) and long (temporally) field trials are needed to resolve such issues. Second, some drivers stated they changed their normal driving habits during the trials to fully explore and "push" the vehicles' capabilities, thus the results may not indicate whether EVs are suitable for their "normal" driving behavior. This behavior is similar to the *Hawthorne effect*, which states subjects in an experiment often alter their behavior for the duration of an experiment [16].

Researchers and manufacturers have also conducted large online surveys [17–25]. In these surveys, drivers were asked about their perceptions towards EVs and their perceived advantages and disadvantages. Because these were not targeted specifically to EV owners, the respondents were mostly drivers with little or no experience with EVs. A benefit of conducting these surveys is that thousands of drivers can be interviewed at little or no cost. However, they only gauge drivers' general interest in adopting EVs and do not measure owners' perceptions.

3.2 Word and Sentence Polarity Classification

There are four main approaches to classifying opinion phrases. Work discussed here is not specific to FBOM but to sentiment classification in general.

1. *Lexicon methods* [26–34] start with a small set of classified *seed words*. These sets are then grown using synonyms derived from WordNet [35] or other glosses—for each word, the word's synonyms are added to the classified set, and this process is repeated as desired.
2. *Semantic methods* [36–39] classify the sentiments of words and sentences based not only on lexicons, but also on the semantic rules of the English language. For example, two adjectives joined by *and* are likely to share the same polarity, e.g., *sunny* and *beautiful*.
3. *Distance methods* [40–44] measure the polarity of a given word based on the *distance* of that word from a set of positive and negative seed words. Distance is normally computed via WordNet or by analyzing the co-occurence of words in a large corpus, with an example function being $d(\text{word}, \textit{good}) - d(\text{word}, \textit{bad})$ where $d(x, y)$ gives the distance (computed via WordNet) from word x to word y. Another common distance measure is *pointwise mutual information* [45].
4. *Classification methods* [46–50] treat the problem of determining the polarity of opinion phrases as a machine learning problem. Rather than learning the sentiment of individual words, a classifier is trained to classify sentences directly. These authors manually label sentences, train a polarity classifier based on this labeled training data, and then classify sentences in the unlabeled data using the trained classifier.

Some opinion phrases are context-dependent and pose a challenge for opinion mining systems. Ding et al. [26] give an algorithm for querying the sentiment of such phrases. They first attempt to query all opinion phrases in a sentence using a lexicon approach. They then use an algorithm which considers syntactical constructs and the sentiments in neighboring sentences to classify unclassified phrases per these lexicons. We explain how our context-dependent handling differs in Section 4.5. Two other works have also studied classifying context-dependent adjectives [51, 52]. While we use a simpler approach then their methods, they provide avenues for extending our system in future work. Our system combines context-dependent adjective handling with a lexicon approach.

3.3 Feature-Based Review Mining

Hu and Lui define the concept of feature-based opinion mining [30] and introduce the first FBOM system, Opinion Observer [31]. Their system first builds $\mathcal{F}^p$, then finds and summarizes positive and negative opinions corresponding to each feature. The authors use a lexicon to determine the sentiment polarity of adjectives in sentences containing product features, then classify sentences based on the number of positive and negative words in a sentence. While we later show this approach is insufficient, the Opinion Observer system was the precursor to other FBOM systems.

In subsequent work, these authors improve their system. Hu and Lui [29] expand on the product feature identification phase. The authors present an association rule mining process to build $\mathcal{F}^p$. The mining process finds noun phrases (e.g., *digital camera*) that are likely to be product features. Pruning rules are used to trim the set of mined product features. Lui, Hu, and Cheng [31] further update their system to use supervised learning for detecting *implicit features*, e.g., *fast* refers to the feature *performance*. Finally, Ding, Lui, and Yu [26] update their system with a better sentiment classifier. For each feature f in a sentence, the authors compute a scoring function based on all adjectives in the sentence and their distance from f. Hence, if there are two features, the adjectives closest to each will influence their scores the most, but all adjectives have a non-zero contribution to the score of all features. This improvement better classifies (f, o) pairs than simply averaging the classification of all adjectives.

Scaffidi et al. [53] build a system called "Red Opal" which allows users to search for products based on the ratings of specific product features. Products are ranked feature-wise based on numerical review ratings, like those found on Amazon, rather than opinion words in the reviews. While the system achieves good results when numerical reviews are available (which they are typically on online retailers), it is not applicable to our problem as the system cannot mine forums, article comments, or other text.

Popescu and Etzioni [49] present a competing system to Hu's. Their system OPINE uses different algorithms for building $\mathcal{F}^p$ and mining/classifying (f, o) pairs. To mine (f, o) pairs, the authors use *syntactical templates* such as <*feature*> *is* <*value*>—if a sentence matches this pattern, (*feature, value*) is mined as an (f, o) pair. These syntactical templates motivated our use of *chunking* to parse (f, o) pairs, as further explained in Section 4.3. This resolves problems with Hu's system because not all opinions in a sentence are associated with every feature in the sentence. They use statistics and classifier-based methods for classifying word sentiments, as opposed to Hu's lexicon based approach.

Jin et al. [50, 54] build a novel system called Opinion Miner. Their system trains a hidden Markov model (HMM) to find (f, o) pairs *and* classify them; the only work we know of to merge these two steps. The HMM is trained using linguistic constructs, syntactical templates, and word sentiments. The HMM learns to mine constructs such as "negative opinion about [feature]", instead of first finding (f, o) pairs and then separately classifying each pair. The authors manually tag certain constructs, then use synonyms, antonyms, linguistic constructs, and other bootstrapping techniques to grow the set of training examples for the HMM. Our system does not combine the mining and classification phases, but we plan to evaluate this merged approach in future work (see Section 7).

Zhang et al [55] use a graph mining approach to rank several products according to various product features. The authors divide opinionative sentences into two sets, those that express opinions on just one product (subjective), and those that compare two or more products (comparative). Products are treated as nodes in a "feature graph". Subjective sentences and their classification are used to weigh nodes, while comparative sentences and their classification are used to weigh edges between the two products being compared. Then a pageRank algorithm is used to rank the set of nodes according to the feature. In the future, when many EV models are sold and EV sales increase, this work may help compare several EV models.

Other researchers have improved upon these systems. Kobayashi et al. [56] suggest a domain-knowledge-driven feature and opinion phrase selection process, instead of the general association mining techniques offered by Hu et al. They introduce an iterative algorithm that generates candidate features and opinion phrases, and manually select those that are valid. In each iteration, more candidates are selected based on the prior iteration, and the process is repeated until an iteration goes by where the human selects no candidates. Zhuang et al. [38] present a case study of Hu's system using movie reviews. The authors incorporate domain knowledge using supervised learning into feature and opinion keyword mining. For example, they have several movie fans manually tag reviews for features, feature opinions, and cast members. Several other systems similar to these [57–59] have also been built, but the systems described above are representative of the various approaches taken in building FBOMs.

4. SYSTEM ARCHITECTURE

We now describe our opinion mining system.

4.1 Mining Overview

Our mining system is depicted in Figure 1. Forums are first crawled using Scrapy [60], a python web crawling framework, then subsequently cleaned (see Section 4.2) and split into individual sentences. Sentences are mined for (f, o) pairs using a process known as *chunking* (see Section 4.3). Sentence fragments containing (f, o) pairs, known as chunks, are then classified for sentiments (see Section 4.4, 4.5). Finally, the results are output. We note our system has several limitations; we propose improvements in Section 7.

This process builds upon previous work. Like Zhuang et al. [38] and Kobayashi et al. [56], we incorporate domain knowledge (DK) into our mining process, specifically in the

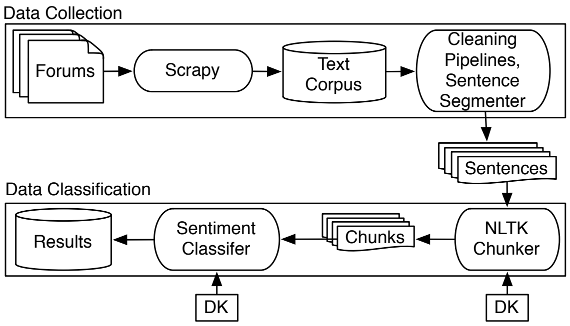

Figure 1: System Architecture

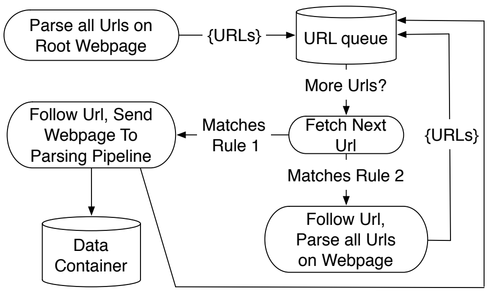

Figure 2: Recursive Scrapy Webcrawling

chunking and querying phases. For feature mining, a part of the chunking phase, we use Hu's association mining technique [29] and then manually prune and collapse the feature set like Kobayashi et al. [56]. We note this is feasible because there are fewer than one hundred common features one may talk about within the context of a car. For parsing sentences, our use of chunking is similar to using syntactical templates like Kobayashi, Popescu, and Zhuang [38, 49, 56], but more powerful (see Section 4.3.3). We use our own methods for the classification stage but partially rely on an open-source sentiment dictionary, the MPQA Opinion Corpus [61, 62]. Like Ding et al. [26], we handle context opinions, but we introduce two new constructs to handle context-dependent opinions (see Section 4.4).

4.2 Data Collection and Preprocessing

We use the Python Scrapy package [60] to collect EV reviews from the Web. Scrapy is a system in which the user writes *spiders* containing two sets of regular expressions (regex). Crawled URLs matching any expression in the first set are *content pages*, and are sent to a *parsing pipeline*. Crawled URLs matching any expression in the second set are *linking* pages which hold links to other (content & linking) pages. The recursive crawling process is depicted in Figure 2.

We preprocess the data before mining it. First, we remove all HTML tags and links. We then convert all characters to lowercase. Next, we iterate through a large list of common typos [63] and fix common misspellings. Finally, we iterate through a list of contractions [64] and expand them. This is done because words such as *not* are *valence shifters* which change the sentiment of opinion phrases as discussed in the following section.

4.3 Parsing Via Chunking

English is not a regular language [65], thus arbitrary English sentences cannot be parsed using regular expressions. Fortunately, most of the sentence constructs people use can be. In the following two sections, we describe our sentence parsing methodology, *chunking*, which works by grouping part of speech (pos) tags with regex. We also describe its advantages over prior work. To the best of our knowledge, we are the first work to use this parsing method.

4.3.1 NLTK Chunking

We use the Python NLTK (*Natural Language Toolkit* [66]) package to parse sentences using *chunking* [67], which makes use of regex to group word sequences with particular parts of speech (pos) together.

A context free grammar (CFG) is a set of production rules of the form $A \rightarrow B$, where this denotes A can be replaced with B in any "string" in the language. In the context of parsing natural language, CFGs state "replace instances of B with the higher-level notion of A". For instance, the rule <verb-phrase> → <subject><verb> replaces the tags <subject><verb> with <verb-phrase>. Chunking is simply an *extended CFG* (E-CFG), a CFG in which the right hand side of production rules can be regex. While E-CFGs provide no functional benefit over traditional CFGs—they describe exactly the same set of languages [68]—an infinite number of CFG production rules may be needed to express the same rule of an E-CFG [68]. The regex operators $\{+, *\}$ provide *compactness*—a way to specify an infinite number of patterns that greatly condense the set of needed rules.

To parse sentences using chunking, we first tag the sentences for pos (we use NLTK, but several tagging tools are available). This produces a list of tuples of the form $[(word_1, pos), (word_2, pos)...]$ for each sentence. We then define a NLTK *chunking grammar*, a series of regex executed on these tagged sentences that combines tuples into *chunks*. The expressions in the chunking grammar are executed in-order and are non-overlapping; that is, words consumed during one chunking will not be part of another chunk. Each expression attempts to match a sequence of tags. The standard regex tokens $\{*, ., +, ?\}$ can be used to capture arbitrarily long groups of tags, and allow for optional parts of speech. For example, the rule X:

X: {<det>?<noun><verb>+<adverb><adjective>+}*[3]

chunks both *the (product-name) is really superb* and *my (product-name) has been reliable*.

The chunking grammar can include as many rules as desired. The most specific rules should be defined first in the grammar since rules are executed in order, and rules with the most flexibility (achieved through the use of the $\{?, +, *\}$ regex tags) should come last in the grammar. With a well-crafted grammar only a few rules are needed; all results presented in Section 5 come from a grammar with only six (albeit complex) rules.

4.3.2 Mining Features Using Domain Knowledge

The goal of the chunking phase is to mine (f, o) pairs. To find chunks containing features, during the tagging phase described above, sentences are searched for all synonyms for all features. Matches are tagged with a special <*feature*> tag which is included somewhere in every chunking rule.

[3] <det> refers to a *determiner* such as *this* or *my*

To build the feature space $\mathcal{F}^p$, we first manually create a *seed* set of features and define a few synonyms for each. We then use *word frequencies, collocations*, and *concordance*, to build and expand the sets above. We use basic word frequencies to generate candidate features missing from the seed feature set—if a noun has a high frequency, it is likely a feature or a feature synonym. We then manually review these results, because not all common nouns are features; e.g., *road* is quite common in our EV review database. Next, we use NLTK's collocation functionality, which produces bigrams and trigrams with high-scoring mutual information—sets of two and three words that often occur together. Multiple-noun features like *battery capacity* and non-adjective opinion phrases like *warranty issues* are found this way. Finally, we use NLTK's concordance functionality. Concordance shows the words surrounding each usage of the target word, e.g., *concordance("battery", k)* prints each occurrence of *battery* with the $k/2$ surrounding words on both sides. Manually reviewing concordance helps identify multi-word features.

Tagging and chunking also allows us to easily handle *implicit features*, words that are both features and opinion phrases. Before chunking a sentence, we replace the part-of-speech tags of implicit features with a special tag *<IF>*. Upon finding a tuple (w,<IF>), we look up the the feature using w in an inverted dictionary that maps feature synonyms to features, and also use w as the opinion phrase. For example, we define *noisy* as a synonym for the feature *sound*, and also as a negative opinion for that feature.

4.3.3 Advantages Of Chunking

In this section, we provide intuition as to why chunking works better than methods used in prior work for mining (f, o) pairs. We briefly define the notion of *valence shifters* since it is integral to the following discussion—valence shifters are words that invert the semantic meaning of a sentence, such as *not*, as in *do not buy this*. Parsing and handling valence shifters is essential to any review mining system [36].

Prior work in FBOM uses one of two methods to classify (f, o) pairs. The first method is to compute a scoring function using the sentiment of adjectives in the sentence and their proximity to features or products. These methods are insufficient if used on a sentence-wide basis (as opposed to using the scoring function within one particular chunk, which we have not seen in prior work), because sentence structure plays a vital role in the meaning of sentences. Consider two simple sentences:

s_0 : {*it does not have good [feature]*}

s_1 : {*the [feature1] is not good, but its [feature2] is excellent*}

Sentences like s_0 are problematic for these methods because the feature is close to the positive opinion *good*, but the opinion is negative. Including rules such as "invert the sentiment if a valence shifter like *not* is found in the sentence" would incorrectly classify s_1 as negative for product2. The solution is to apply the valence inversion rule to only the first part of s_1. We iterate over tagged sentences and replace the part-of-speech tags of valence shifters, products, and features (as described above) with unique tags. These tags are then included in our chunking grammar rules; for example:

r_0 : {*<verb><valenceshft><"have/has"><opinion><feature>*}

r_1 : {*<feature><verb><valenceshft>?<opinion>*}

Using the discussed VS inversion rule, s_0 triggers r_0 which correctly classifies it as negative. Moreover, s_1, chunked as *[the* (r_1)*, but its* (r_1)] fails to trigger r_0 because the sentence structure does not match, triggers r_1 which correctly classifies it as negative for [product1], and triggers r_1 again which correctly classifies it as positive for [product2].

The second method used to mine (f, o) pairs in prior work is to parse sentences according to syntactical templates [38, 49, 56]. Chunking is a more expressive form of these templates. The chunking grammar allows templates to contain optional tags and repeated tags, which allows for a much more concise grammar. Theoretically, an infinite number of fixed syntactical templates may be needed to specify a single chunking rule, due to the power of the regex operators $\{+, *\}$. In practice, one of the most useful features of chunking is the allowance of optional tags. Rules with well-placed optional valence shifters and "filler" words can capture many sentence constructs in a single rule, e.g,

{*<valenceshft>?<opinion><"with"><det>?<article>?<feature>*}

matches all of "*no problems with the <feat>*", "*problems with my <feat>*", "*no issues with my <feat>*", etc.

Thus, chunking has advantages over scoring based methods and template based methods. Moreover, a scoring function can be used within each chunk; even though each chunk contains only one feature, it may contain multiple, sometimes conflicting opinion phrases. A scoring function can classify the chunk with respect to the feature by weighing each opinion phrase in the chunk.

Finally, we do not compare chunking to the HMM approach used by Jin et al. [50], as their system parses and classifies simultaneously while we do not merge these two steps.

4.4 Handling Context-Dependent Opinions

Before discussing our sentiment querying algorithm, we introduce two concepts to handle context-dependent opinions phrases (CDOPs)—phrases that change their sentiment given their context.

For some features, more or less is always better, e.g., *performance* and *price*. We refer to such features as *positively and negatively oriented features*. *Intensity modifiers* like *low* and *high* change their context when referring to such features. For example, note the orientations of the following (f, o) pairs:

$$\begin{aligned}
\text{(Range, low)} &\rightarrow - \\
\text{(Range, high)} &\rightarrow + \\
\text{(Maintenance, low)} &\rightarrow + \\
\text{(Gear, low/high)} &\rightarrow N \\
&\ldots
\end{aligned}$$

For each feature of interest, we specify whether the feature is positively, negatively, or non oriented. We also maintain a list of intensity modifiers (many can be found in Paradis [69]) for querying the oriented features. We empirically find that most CDOPs are of this type, so correctly specifying the orientation of features correctly classifies most CDOPs.

Some opinion phrases which are not intensity modifiers can also change their sentiment given their context, for example *cheap quality* vs. *cheap price*. Given this, we define a *product-feature sentiment dictionary*, $\mathrm{S}^{\mathrm{p}\in\mathcal{P}}_{\mathrm{f}\in\mathcal{F}^p}[o \in \mathcal{O}]$, for each {product, feature} pair (p, f). These dictionaries are small and only contain phrases that have a sentiment when referring to (p, f) that is different than the sentiment it holds when used in other contexts. They produce a label when queried with an opinion phrase o if o is defined to be context-dependent for that feature, or "unknown" otherwise:

$$S[Car,\ Quality](cheap) \rightarrow -$$
$$S[Car,\ Price](cheap) \rightarrow +$$
$$S[Car,\ Performance](awesome) \rightarrow \text{unknown}$$

We perform a laborious but worthwhile domain knowledge (DK) input process to build these dictionaries. In addition to manually adding CDOPs into these dictionaries, we label a portion of the training data and use our system to classify these sentences. We print opinion phrases that are found in both correct and incorrect sentences, because if a phrase is missing from a feature-specific dictionary, it is likely classified correctly for some features and incorrectly for others. We then manually review these results and iterate this process several times, each time adding phrases into the appropriate dictionaries.

4.5 Sentiment Querying

When querying the sentiment of a (f, o) pair, several rules are checked in order. Whenever a rule triggers, a sentiment label is returned and the rest of the rules are not checked.

1. If o is an intensity modifier and f is an oriented feature, we use the following subrules, one of which must trigger:

$$(o:+, f:+) \rightarrow \text{ return } +$$
$$(o:+, f:-) \rightarrow \text{ return } -$$
$$(o:-, f:+) \rightarrow \text{ return } -$$
$$(o:-, f:-) \rightarrow \text{ return } +$$

2. If querying the product-feature sentiment dictionary $\mathrm{S}^{\mathrm{p}}_{\mathrm{f}}$ returns a label, the label is returned.
3. If querying the *default sentiment dictionary* (see below) returns a label, the label is returned.
4. Return neutral (N).

This process always returns a label, because if rules 1-3 fail to trigger, N is returned. If the 4th rule triggers, it is likely that the opinion phrase is seldom used or misspelled—after modifying the freely available MPQA Opinion Corpus [61, 62], our default sentiment dictionary contains over 6,800 opinion phrases.

We now compare our handling of context-dependent opinions with Ding et al.'s. Their algorithm sacrifices classification accuracy for improved recall, because they first query opinion phrases using lexicons, and then treat *all* unknown phrases (per these lexicons) as context-dependent and attempt to classify them as such. However, many adjectives are simply neutral, and using their algorithm to classify all neutral phrases leads to *over-classification*, since some of their rules will trigger even when the phrase was neutral. We take the opposite approach and under-classify, because we would rather mine fewer sentences with high accuracy (there are no shortage of opinions online) than many sentences inaccurately. Since all phrases in the feature-specific dictionaries are manually added context-dependent phrases, if line 2 returns a label, it can only be wrong if there is a parsing or tagging error. Moreover, if line 2 does not return a label, one of two cases must hold—either the phrase is not context-dependent, or the phrase is missing from the dictionary. In the former case, the phrase is classified as normal in the default dictionary. In the latter, a classification error may occur, but this is uncommon because the space of CDOPs that are *not* of the intensity-modifier type is small.

4.6 Other Optimizations

Here we detail various optimizations designed to improve the accuracy of our opinion mining system.

1. We find many sentences (which we were incorrectly classifying) implicitly ask a question or talk about a hypothetical scenario. Words and phrases such as *wondering*, *curious*, and *as long as* were common in sentences that containing an opinion phrase but do not express an opinion, such as *I am curious as to whether the battery lasts a long time*. We use 16 such phrases we manually found and classify all sentences containing these as neutral.
2. Other words nullify opinions only within a chunk; we classify chunks containing such words as neutral, but process other chunks in the sentence as normal. Words like *can*, *may*, and *will* state something will hold in a particular situation, or offer a suggestion, such as "driving too fast may decrease your battery efficiency"; this sentence is not expressing a sentiment, but rather offers a suggestion.
3. Words such as *costs* make classification difficult because they are used in different contexts. Adding *costs* as a synonym for the feature *price* leads to poor accuracy, because too many sentences refer to the cost of something other than the vehicle, e.g., *electricity is cheap*. However, if it is used near the feature *car*, the sentence is likely referring to the *price* of the vehicle. We implement dictionaries of *feature changers*; words that when used near one feature indicate the sentence is referring to another feature. As another example, the word *handles* near the feature *car* indicates that the sentence is not a general sentiment about the car, but rather referring to the car's *performance*.
4. We use a list of "nullifying-synonyms", words near features that indicate the sentence is not actually referring to any feature of interest (in contrast to those just discussed that indicate the sentence is referring to a different feature). For example, if the phrase *12 volt* occurs near the feature *battery*, the sentence is probably referring to the smaller battery in the vehicle, and not the main EV battery.
5. Our mining system handles non-adjective opinions (NAOs) such as *disgrace* and *problem*, whereas most previous work does not. We maintain a list of NAOs, and define the sentiment of these words/phrases. Before chunking a sentence, we replace the tags of words contained in this list with a special tag, and include this special tag in our chunking grammar in the same places we include the tag for an adjective. As an example, *complaint*, while not an adjective, often expresses a negative opinion.

5. EVALUATION METHODOLOGY

Here we discuss our system evaluation methodology. We first define our evaluation metrics (Section 5.1), then our corpora generation (Section 5.2) and definitions used in our results (Section 5.3).

5.1 Evaluation Metrics

Let

- c_f^+, c_f^- represent the number of sentences about feature f classified as $+,-$.
- $\star\left(c_f^+\right)$,$\star\left(c_f^-\right)$ represent the number of sentences classified as $+,-$ for feature f that we also classify as $+,-$ for feature f. We stress "for feature f" because it is possible, and common among classification errors, that the label is correct but the opinion phrase is referring to a different feature.
- t_f^+,t_f^- represent the number of sentences about feature f in the corpus that we classify as $+,-$.

Our two evaluation metrics, *opinion precision* and *opinion recall*, compute for each feature f are defined as:

$$\text{precision}(f) = \frac{\star\left(c_f^+\right) + \star\left(c_f^-\right)}{c_f^+ + c_f^-}$$

$$\text{recall}(f) = \frac{\star\left(c_f^+\right) + \star\left(c_f^-\right)}{t_f^+ + t_f^-}$$

Precision penalizes for incorrect classifications while recall penalizes for failing to mine (f, o) pairs. Chunks misclassified as $(+/-)$ are reflected in precision, and $(+/-)$ chunks classified as neutral are reflected in recall.

5.2 Experimental and Ground Truth Corpus Generation

For our experimental evaluation, we crawled the owner discussion forums for the two best selling EVs—the Nissan Leaf [5] and Chevrolet Volt [4] ownership forums. We crawled every forum post existing on both sites as of February 1st 2013. This led our *experimental corpus* containing 107,293 Volt sentences and 220,906 Leaf sentences. We then classified this corpus using our system which filtered out all sentences containing no features. This left 10,519 Volt sentences and 19,799 Leaf sentences. Next, we sampled a random ≈25% of these and manually labeled them. This led to a *ground truth corpus* containing 2,566 Volt sentences and 5,514 Leaf sentences that contain at least one feature.

We note that due to our labeling methodology, our measure of recall is not "true recall", because we first filter out all sentences which contain no synonyms in our feature set. However, some sentences may refer to a feature implicitly or using a rare synonym. While we include some implicit features and many feature synonyms, we cannot exhaustively include all. To measure true recall, we would randomly read a portion of the experimental corpus without first filtering. However, due to the large percentage of sentences that contain no features ($>$ 90% as shown), we hypothesize this may lead to very few classified sentences and hence be a wasteful effort. Given this, we believe first filtering out sentences which do not contain any feature synonyms is reasonable.

5.3 Definitions

Here we define the non-obvious features of electric vehicles used in our graphs:

- *General* refers to any opinion referring to the *car* itself and not a specific feature, such as *this car is amazing*.
- *Range anxiety* is the term given to EV drivers' fear of being stranded en route to their destination due to a lack of range and charging.
- Current (Lithium Ion) EV batteries lose capacity over time as they are repeatedly charged and discharged, and if they are subjected to extreme temperatures [70]. *Degradation* refers to the effect of charging and climate on a batteries capacity and life.
- We denote anything related to heating and cooling, including features such as heated seats and pre-warming (warming the EV while it is still plugged in at home), as *HVAC*.
- *Carwings* [71] and *Onstar* [72] are products included with the Leaf and Volt respectively that provide various feedback, charging, and safety services to drivers.
- *MiscFeats* refers to a mix of other features including regenerative breaking and navigation systems.

6. RESULTS

We now describe our evaluation results. We claim our system has three main benefits:

- The system graphs the percentage of positive and negative opinions for each feature, giving the user a high-level product overview. This is not currently available for prospective EV buyers.
- The system significantly reduces the space of text the user must read if he or she is interested in determining what other drivers thought about various features.
- The system measures EV owners' perceptions towards EVs. These perceptions include those drawn from expensive field trials in addition to perceptions that were only realized after owning the vehicle for an extended period of time.

In the following three sections, we support these claims. We then discuss our system's accuracy and recall in Section 6.4. We note the results shown are for the ≈25% of the corpus we manually labeled (as discussed in §5.2) as the ground truth corpus—we do not present any unverified results from the unlabeled segment of the corpus.

6.1 High Level Polarity Breakdown

The polarity of sentences in the ground truth corpus is shown in Figures 3 and 4. There are three bars for each feature. The first shows the polarity distribution of sentences we classified (c^+, c^-, c^N), the second shows the distribution of those correctly classified $\left(\star\left(c^+\right), \star\left(c^-\right), \star\left(c^N\right)\right)$, and the last shows the distribution of ground truth sentences (t^+, t^-, t^N). The numbers above the bars show the number of sentences containing that feature. Examining these figures quickly gives the user a high-level view of opinions about the various product features.

6.2 Text Reduction

Figures 3 and 4 also demonstrate the text reduction capability of our system. As discussed in Section 5.2, our experimental corpus contains ≈ 330,000 sentences but only ≈10% contain at least one feature. Moreover, we see in Figures 3 and 4 that ≈ 70% of sentences containing a feature are neutral (based on our random sampling). Extrapolating from these metrics, we hypothesize only $330,000 * .1 * .3 = 9,900$ of the original sentences contain positive or negative opinions about a feature. This results in a ≈97% reduction of text compared to searching through the forums, even if the user wishes to read every opinion about every feature.

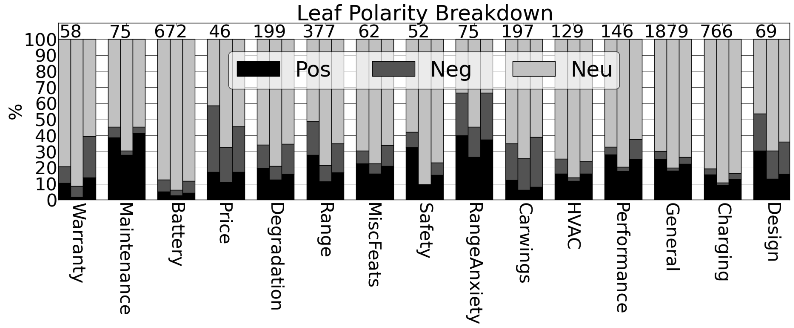

Figure 3: Leaf polarity breakdown

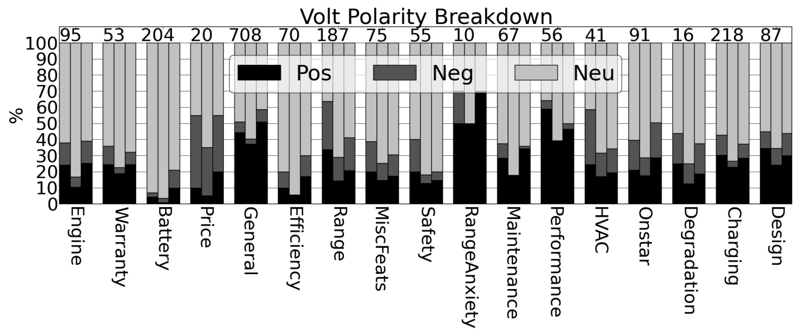

Figure 4: Volt polarity breakdown

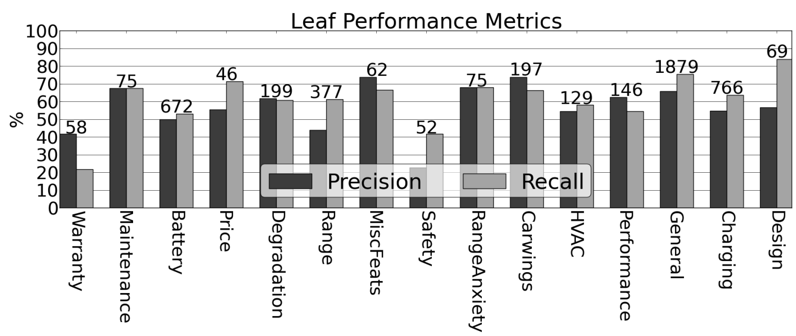

Figure 5: Leaf performance metrics

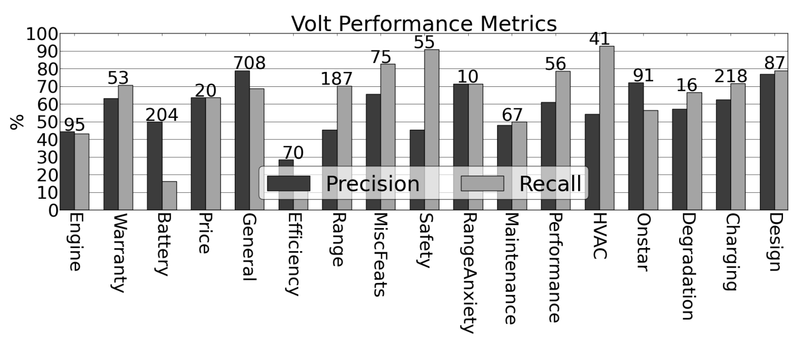

Figure 6: Volt performance metrics

6.3 Insights

We derive insights by examining Figures 3 and 4 and reading the classified sentences. These insights include those derived from field trials in addition to perceptions that were only realized after the owners had their vehicles for a longer period of time.

We give several examples of insights derived that support conclusions from field trials and driver surveys. *Price* and *range anxiety* are commonly cited as the two largest EV adoption barriers [70]; Figures 3 and 4 support this hypothesis. For the Leaf, a BEV that contains no gasoline engine like the Volt, the majority of sentiments about range and range anxiety are negative. Sentiments on range anxiety for the Volt are positive because the Volt is designed to eliminate range anxiety (it has a gasoline engine in addition to the battery). For price, the majority of sentiments for both products are negative (note that there are few sentiments included for price; this problem is discussed in §6.5). *Maintenance* is commonly cited as a major selling point of BEVs—the absence of an engine means fewer moving parts that can fail and less fluids to change. From Figure 3, we see that sentiments towards maintenance for the Leaf are overwhelmingly positive. Field trials often conclude that participants enjoy the EV *charging* process compared to the refueling process—we also find that sentiments towards charging are positive for both vehicles. Reading the classified sentences reveals that many drivers receive free charging at work, have little need for public charging and are thus not concerned about the lack of public charging stations, and live in areas with time-of-day electricity pricing so they are able to charge overnight cheaply. As a final example, the majorly positive sentiments for the *general* category reveals some expected early adopter bias. While many drivers may express concern about the set of features they dislike, most end their discussions with comments like "*... but I love my Leaf*". This is expected, as current EV adopters likely paid more for their vehicles than future EV adopters, and thus want to support their decision.

We are also able to derive insights that were only perceived after the owners had their vehicles for a significant duration of time. Such insights are not possible to elicit from field trials. The most important such example is owners' experience with *battery degradation*, the effect of repeated charging and climate on battery capacity over time. While it is known that climate and charging cycles affect battery life, it is unknown to what extent this is the case [70]. We find that some owners are experiencing non-trivial battery degradation and about 50% of sentiments towards degradation are negative. Manufacturers can use these sentiments in unison with the owners' location (if available on the forum) to derive conclusions about the effect of climate on battery capacity, e.g., we find that owners in hot regions such as Arizona and southern California post more often about degradation. Closely related are sentiments on *warranty*, which are mostly negative, because some owners have filed for battery replacements through battery warranties against capacity loss. Reading classified statements from warranty and maintenance reveals other vehicle problems of interest to manufacturers, such as the replacement and maintenance rate of various parts.

We note we are currently implementing the tracking of owners' sentiments over time as further discussed in §7. Many field trials have attempted to determine how perceptions change over time, for example, by interviewing participants both before and after the trial. However, because individual participants in trials are often only given a vehicle for a few weeks or months, their opinions may not change as much as they would over a period of several years. Thus, tracking owners' opinions over time online can reveal insights about how perceptions change with vehicle experience on a longer scale than is possible in field trials.

6.4 Performance Metrics

Figures 5 and 6 show the precision and recall of our system on the ground truth corpus. The sum over both products equals the size of the ground truth corpus (≈8,000 sentences). The performance is on par with or better than prior FBOM systems discussed in Section 3.3. In the next section, we show several examples of both solvable and unsolvable classification errors; the English language is simply too ambiguous and complex to mine without error.

While our results are on par with prior opinion mining systems, we emphasize two of our contributions. First, prior FBOM systems have focused on products for which easily consumable reviews *are* available. For example, Hu and Lui [30] focus on digital camera reviews, Kobayashi et al. [56] focus on movie reviews, and Scaffidi et al. [53] focus on any product with numerical reviews. While these authors contribute to mining opinions on specific features, a useful addition for consumers, an easily-consumable review system of some form exists for these products (e.g., Amazon and IMDB). We present the first work on mining opinions for a new product for which text has thus far been the main review medium. Moreover, if EVs are not improved and customer adoption barriers are not solved, there may never be a comprehensive database of EV reviews. Second our system is open source [6], and the package includes our domain knowledge input and all manually built lists and dictionaries. Future researchers or users that wish to extend the system have a comprehensive starting point. In contrast, the three closest prior systems are either closed source or unavailable—Opinion Miner is proprietary [54], the Opinion Observer system turned into the proprietary OpinionEQ [73,74], and the OPINE system was never released by the authors.

6.5 Insightful Classification Errors

We present here a few problematic sentences because they give insight into the complexities of opinion mining.

Some features are hard to mine or classify opinions for. For example, our system performs poorly on classifying opinions related to *safety*. We find the word *issue* is heavily overloaded but used often—in some instances it is used synonymously with *hazard*, such as *that is a safety issue!*, and in other cases it is used synonymously with *feature*, e.g., *grounded charging is a safety issue*. Our system also performs poorly on classifying *warranty* opinions for similar reasons. For example, it is difficult to tell programmatically when the phrase *not covered* is used as a negative or neutral sentiment. Sometimes posters state facts with this phrase, e.g., *the windshield is not covered in your warranty*, and other times to express frustration, such as *the repair was not covered by my warranty*. Moreover, notice that *price*, even though it is cited as a major adoption barrier, has very few comments. We find too many posts comment on the price of something other than the price of the vehicle, such as the price of charging and electricity. After experiencing a high rate of classification errors regarding this feature, we tuned our system to only classify a chunk as referring to *price* if the chunk contained both a synonym of *price and* a synonym of *General* such as *car, Volt, Leaf*, etc. This led to a tradeoff: from Figures 5 and 6 we see our system performs well with respect to precision and recall for *price*, but the number of classified sentences to draw conclusions from is small. Conversely, our system classifies *General* opinions well for both products, for which there are many. This is because sentences expressing general sentiments are often clear and brief, e.g., *I love my Leaf!* and *this is an excellent car*.

Table 1 shows examples of errors fixable by updating the feature-specific sentiment dictionaries, updating the default sentiment dictionary, or other adding other domain knowledge (DK) as shown. Others errors are "better left unsolved"—tuning the system to correct these creates larger

Sentence	Class	Tr	Problem	Solution
The standard warranty is more than enough.	N(*Warranty*)	$+$	*more than enough* not recognized as an opinion	add it to the DSD
My Carwings sometimes hangs.	N(*Carwings*)	$-$	*hangs* not recognized	add it as $-$ to the Carwings FSD
This car is a blast to drive!	$-$(*General*)	$+$	*blast* is context dependent	add it as $+$ to the General FSD
The dealer showed me how to perform the recommended maintenance	$+$(*Maintenance*)	N	*recommended* is context dependent	add it as N to the Maintenance FSD
Capacity loss is greatest at the beginning of the battery's life.	$+$(*Degradation*)	N	*greatest* is context dependent	add it as N to the Degradation FSD
I love my Audi, it is a great car.	$+$(*General*)	N	Subject error	classify chunks w/ other popular car names as N; see Sec. 4.6 number 2
I had a low battery warning.	$-$(*Battery*)	N	*warning* is context dependent	classify chunks w/ this phrase as N
It has a 5 star safety rating.	N(*Safety*)	$+$	*5 star* not recognized	add it as $+$ to the *Safety* FSD
The hot weather kills my range.	N(*Range*)	$-$	*kills*, a verb, not recognized as an opinion	add it as a non-adjective opinion to the *Range* FSD.

Table 1: Solvable errors. "FSD" denotes "feature-specific dictionary", and "DSD" refers to the default sentiment dictionary

Sentence	Class	Tr	Problem
"It was charge free."	$+$(*Charging*)	N	poster is not referring to charging but rather the price of something. However, sometimes posters talk about free charging.
The Volt is the best car ever and I will never go back to a crude ICE	$-$(*General*)	$+$	ICE most often refers to the Volt's ICE, but sometimes to non hybrid vehicles in general.
My new radio has far less degradation near mountains.	$+$(*Degradation*)	N	here the poster is referring to radio signal degradation; static near mountains, power lines, or tunnels.
It felt like the engine was on.	$+$(*Engine*)	N	*like* is a tough word. Even within the context of one feature, it can be used as a comparator or as a positive sentiment (more common).
I love everything about this car, with the exception of the exterior.	N(*Exterior*)	$-$	Chunking error. We find most opinions do not "cross over" commas, hence commas are not included in chunking rules. Here the valence shifter "exception" loses the opinion to negate—*love*, since it is not in the chunk.

Table 2: "Better left unsolved" errors

Sentence	Class	Tr	Problem
The Leaf handles 99% of my annual driving."	N(*General*)	?	Should this be +? What's the +/− Cutoff?
I get around 80 mpg."	N(*Efficiency*)	?	Should this be + for *Efficiency*? Cutoff?
The Leaf is very easy to push.	$+$(*General*)	?	This might mean the poster's car died, or something related to performance/handling?
Level 2 charging is very efficient.	$+$(*Charging*)	?	Should this be +? The poster may either be happy with their charging experience or just stating a fact.
I never worry about my mileage.	$+$(*Efficiency*)	?	Is the poster stating a fact or that they have plenty of range?
I am sad my Volt is in for maintenance.	$-$(*Maintanance*)	?	is the poster disappointed the car requires maintenance, or are they stating they miss driving their car?

Table 3: Subjective ambiguity errors

problems elsewhere. Sometimes a feature synonym is used in two different contexts, e.g, "ice" can refer to an engine (Internal Combustion Engine) or the weather condition. We set the system according to which usage is most common, but errors will occur when the word is used in the less common context. Others represent chunking errors where changing the chunking grammar to fix the error caused more problems in other sentences because the offending sentence structure is uncommon. Finally, some words are context-dependent even within the context of one feature. We accept such errors, exampled in Table 2, as necessary due to the ambiguity and complex structure of English. Finally, some ambiguous sentences can be classified differently by different human readers. Some correspond to a parameter which sounds positive to some, such as "100 miles per gallon", but for which it is hard to impose a strict cutoff for which all human readers agree, e.g., "all mileages over X are positive". Others are ambiguous sentences that could be classified in either direction. Examples of such errors are shown in Table 3.

7. CONCLUSIONS & FUTURE WORK

Understanding EV owners' experiences with and perceptions towards EVs is helpful for manufacturers to build later-generation models more aligned with drivers' mobility preferences and requirements. Unfortunately, it is expensive to conduct field trials or to target surveys specifically to EV owners. We build an opinion mining system that classifies opinions on EV ownership forums. Our system helps the user obtain a high-level overview of opinions on various product features, and greatly reduces the space of text the user is required to read to extract opinions. These opinions are useful to prospective buyers, marketers (what features should be advertised?) and manufacturers (what features should be improved?). To build this system, we combine prior opinion mining systems with several new optimizations and our EV domain knowledge. We furthermore open sourced our system for extension by other researchers, as we find most prior opinion mining systems are unavailable.

There are several avenues for extending and improving our work:

1. At the time of system implementation, Tesla's sales were still well nominal compared to Nissan's and Chevrolet's EV sales. However, Tesla's Model S sales are now nearly equivalent (see footnote 1). We thus plan to add Tesla support to our system.
2. We are working to see how sentiments change over time by analyzing sentiments periodically, e.g., monthly. This is interesting to see how perceptions change with vehicle experience and whether sentiments fluctuate with gas, electricity, and vehicle prices. It is possible to see the temporal change for individual owners if they post using a user name, and for owners that post anonymously, we can examine how collective sentiments change.
3. We do not currently perform pronoun resolution. A poster may explicitly mention a feature in one sentence and then write several more opinionative sentences about the same feature using pronouns easily resolved by a human reader. Our system only categorizes opinionative sentences where an explicit or implicit feature is found. Pronoun resolution, while difficult, may be used to infer the features being discussed.
4. We aggregate data for each individual product in a separate database. When mining for product p, we assume all (f, o) pairs found in the database for p refer to p. This produces erroneous results when a poster discusses another product in their post. Future work should detect the product being discussed from the context.
5. We do not perform spam or malicious text detection. We treat all sentences equally even though some may contain sentences injected by malicious sources, such as drivers who oppose a particular brand, or advertisements posted by spamming bots.
6. Our methods for building the feature-specific dictionaries and lists of oriented adjectives are simple; we use pre-built lists and manually add others. This part of our system can be improved using recent work in the field of classifying context-dependent adjectives [51, 52].
7. We do not distinguish between chunks referring to one product and chunks comparing two products like Zhang et al [55]. Modifying the chunking grammar to include comparative templates may reduce classification errors.
8. In all but one prior FBOM systems, (f, o) mining and classification are distinct phases, hence we adopt this approach. However, Jin et al. [50], merge these two phases. In future work we will evaluate this combined approach.

8. REFERENCES

[1] A. Ohnsman, "Tesla rises after model s sales in 2013 exceed forecast," 2014. Accessed March 29th 2014. http://www.bloomberg.com/news/2014-01-14/tesla-delivered-6-900-cars-in-fourth-quarter-executive-says.html.

[2] J. Voelcker, "Plug-in electric car sales for 2013 near double previous year's," 2014. Accessed March 29th 2014. http://www.greencarreports.com/news/1089443_plug-in-electric-car-sales-for-2013-almost-double-last-years.

[3] Electric Drive Transportation Association, "Electric drive sales dashboard," 2014. Accessed March 29th 2014. http://electricdrive.org/index.php?ht=d/sp/i/20952/pid/20952.

[4] "Forum: Volt ownership forum." http://gm-volt.com/forum/forumdisplay.php?16-Volt-Ownership-Forum.

[5] "My Nissan Leaf Discussion Forums." http://www.mynissanleaf.com/.

[6] T. Carpenter, "A Python Electric Vehicle Sentiment Analysis System." http://www.tommyjcarpenter.com/code/evsentiment.html.

[7] L. Bunce, M. Harris, and M. Burgess, "Charge up then charge out? Drivers' perceptions and experiences of electric vehicles in the UK," *Transportation Research Part A*, vol. 59, Jan. 2014.

[8] A. Everett, M. Burgess, M. Harris, S. Mansbridge, E. Lewis, C. Walsh, and S. Carrol, "Initial Findings from the Ultra Low Carbon Vehicle Demonstrator Programme," *Technology Strategy Board Report*, 2011.

[9] S. Pichelmann, T. Franke, and J. F. Krems, "The Timeframe of Adaptation to Electric Vehicle Range," *Lecture Notes in Computer Science*, vol. 8005, no. Chapter 69, 2013.

[10] T. Franke and J. F. Krems, "What drives range preferences in electric vehicle users?," *Transport Policy*, vol. 30, Nov. 2013.

[11] T. Franke, F. Bühler, P. Cocron, I. Neumann, and J. F. Krems, "Enhancing sustainability of electric vehicles: A field study approach to understanding user acceptance and behavior," *Advances in Traffic Psychology*, 2012.

[12] T. Turrentine, D. Garas, A. Lentz, and J. Woodjack, "The UC Davis MINI E Consumer Study," *University of California Davis Institute of Transportation Studies Report*, vol. UCD-ITS-RR-11-05, 2011.

[13] E. Graham-Rowe, B. Gardner, C. Abraham, S. Skippon, H. Dittmar, R. Hutchins, and J. Stannard, "Mainstream consumers driving plug-in battery-electric and plug-in hybrid electric cars: A qualitative analysis of responses and evaluations," *Transportation Research Part A*, vol. 46, no. 1, 2012.

[14] S. Carroll, "The Smart Move Trial: Description and Initial Results," *Centre of Excellence for Low Carbon and Fuel Cell Technologies Report*, 2010.

[15] K. Kurani, "Plug-In Hybrid Electric Vehicle (PHEV) Demonstration and Consumer Education, Outread, and Market Research Program," *University of California Davis Institute of Transportation Studies Report*, vol. UCD-ITS-RR-09-21, 2009.

[16] R. Macefield, "Usability Studies and the Hawthorne Effect," *Journal Of Usability Studies*, vol. 2, May 2007.

[17] K. Lebeau, J. V. Mierlo, P. Lebeau, O. Mairesse, and C. Macharis, "Consumer attitudes towards battery electric vehicles: a large-scale survey," *International Journal of Electric and Hybrid Vehicles*, vol. 5, no. 1, 2013.

[18] R. M. Krause, S. R. Carley, B. W. Lane, and J. D. Graham, "Perception and reality: Public knowledge of plug-in electric vehicles in 21 U.S. cities," *Energy Policy*, vol. 63, Dec. 2013.

[19] O. Egbue and S. Long, "Barriers to widespread adoption of electric vehicles: An analysis of consumer attitudes and perceptions," *Energy Policy*, vol. 48, Sept. 2012.

[20] Deloitte, "Unplugged: Electric vehicle realities versus consumer expectations," *Deloitte Report*, 2011.

[21] S. Musti and K. M. Kockelman, "Evolution of the household vehicle fleet: Anticipating fleet composition, PHEV adoption and GHG emissions in Austin, Texas," *Transportation Research Part A*, 2011.

[22] A. Peters and E. Dütschke, "How do Consumers Perceive Electric Vehicles? A Comparison of German Consumer Groups," *Journal of Environmental Policy & Planning*, Jan. 2014.

[23] EPRI, "Characterizing Consumers' Interest in and Infrastructure Expectations for Electric Vehicles:Research Design and Survey Results," *EPRI Report*, 2010.

[24] Deloitte, "Gaining traction: A customer view of electric vehicle mass adoption in the U.S. automotive market," *Deloitte Report*, 2010.

[25] Ernst & Young, "Gauging interest for plug-in hybrid and electric vehicles in select markets," *Ernst & Young Report*, 2010.

[26] X. Ding, B. Liu, and P. S. Yu, "A Holistic Lexicon-Based Approach to Opinion Mining," *WSDM*, 2009.

[27] A. Esuli and F. Sebastiani, "Determining Term Subjectivity and Term Orientation for Opinion Mining," *EACL*, 2006.

[28] S.-M. Kim and E. Hovy, "Determining the Sentiment of Opinions," *COLING*, 2004.

[29] M. H. Liu and Bing, "Mining Opinion Features in Customer Reviews," *AAAI*, 2004.

[30] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," *KDD*, 2004.

[31] B. Liu, M. Hu, and J. Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web," *WWW*, May 2005.

[32] A. Esuli and F. Sebastiani, "PageRanking WordNet Synsets: An Application to Opinion Mining," *ACL*, 2007.

[33] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," *LREC*, 2006.

[34] J. Kamps, M. Marx, R. J. Mokken, and M. de Rijke, "Using WordNet to Measure Semantic Orientations of Adjectives," *LREC*, 2004.

[35] "WordNet: a lexical database for English." http://wordnet.princeton.edu/.

[36] A. KENNEDY and D. INKPEN, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters ," *Computational Intelligence*, 2006.

[37] V. Hatzivassiloglou and K. R. McKeown, "Predicting the Semantic Orientation of Adjectives," *ACL*, 2012.

[38] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie Review Mining and Summarization," *CIKM*, 2006.

[39] A. Andreevskaia and S. Bergler, "Mining WordNet for Fuzzy Sentiment: Sentiment Tag Extraction fromWordNet Glosses," *Annual Meeting of The European Chapter of The Association of Computational Linguistics*, 2006.

[40] H. Takamura, T. Inui, and M. Okumura, "Extracting Semantic Orientations of Words using Spin Model," *ACL*, 2005.

[41] M. Taboada, C. Anthony, and K. Voll, "Methods for Creating Semantic Orientation Dictionaries," *LREC*, 2006.

[42] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," *ACL*, 2002.

[43] P. D. Turney and M. L. Littman, "Measuring Praise and Criticism: Inference of Semantic Orientation from Association," *ACM Transactions on Information Systems*, 2003.

[44] M. Gamon and A. Aue, "Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms," *The ACL workshop on feature engineering for machine learning in NLP*, 2005.

[45] E. Terra and C. L. A. Clarke, "Frequency estimates for statistical word similarity measures," in *NAACL*, vol. 1, 2003.

[46] T. Wilson, J. Wiebe, and R. Hwa, "Just how mad are you? Finding strong and weak opinion clauses," *AAAI*, 2004.

[47] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger, "Pulse: Mining Customer Opinions from Free Text," *Intelligent Data Analysis*, 2005.

[48] A. Esuli and F. Sebastiani, "Determining the Semantic Orientation of Terms through Gloss Classification," *CIKM*, 2005.

[49] A.-M. Popescu and O. Etzioni, "Extracting Product Features and Opinions from Reviews," *HLT*, 2005.

[50] W. Jin, H. H. Ho, and R. K. Srihari, "OpinionMiner: A Novel Machine Learning System for Web Opinion Mining and Extraction," *KDD*, 2009.

[51] M. Tsytsarau and T. Palpanas, "Disambiguating Dynamic Sentiment Ambiguous Adjectives," *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.

[52] M. Wen and Y. Wu, "Mining the Sentiment Expectation of Nouns Using Bootstrapping Method," *IJCNLP*, 2011.

[53] C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin, "Red Opal: Product-Feature Scoring from Reviews," *EC*, 2007.

[54] OpenDover, "Opinion Miner." http://www.opendover.nl/opinion-miner-0.

[55] K. Zhang, R. Narayanan, and A. Choudhary, "Voice of the Customers: Mining Online Customer Reviews for Product Feature-based Ranking," *WOSN*, 2010.

[56] N. Kobayashi, K. Inui, Y. Matsumoto, K. Tateishi, and T. Fukushima, "Collecting Evaluative Expressions for Opinion Extraction," *IJCNLP*, 2004.

[57] M. Eirinaki, S. Pisal, and J. Singh, "Feature-based opinion mining and ranking," *Journal of Computer and System Sciences*, vol. 78, no. 4, 2012.

[58] B. A. Ojokoh and O. Kayode, "A feature-opinion extraction approach to opinion mining," *Journal of Web Engineering*, vol. 11, no. 1, 2012.

[59] S. Mukherjee and P. Bhattacharyya, "Feature Specific Sentiment Analysis for Product Reviews," *Lecture Notes in Computer Science*, vol. 7181, 2012.

[60] "Scrapy." http://scrapy.org/.

[61] T. Wilson, "Mpqa opinion corpus." http://www.cs.pitt.edu/mpqa/.

[62] T. A. Wilson, *Fine-grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of Private States*. PhD thesis, University of Pittsburgh, 2008.

[63] "Wikipedia: Lists of common misspellings/for machines." http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines.

[64] "Wikipedia:list of english contractions." http://en.wikipedia.org/wiki/Wikipedia:List_of_English_contractions.

[65] A. Clark, C. Fox, and S. Lappin, eds., *The Handbook of Computational Linguistics and Natural Language Processing*. Blackwell Handbooks in Linguistics, John Wiley & Sons, 2010.

[66] "NLTK 2.0 documentation." http://nltk.org/.

[67] "Nltk: Extracting information from text." http://nltk.googlecode.com/svn/trunk/doc/book/ch07.html.

[68] J. Albert, D. Giammarresi, and D. Wood, "Normal form algorithms for extended context-free grammars," *Theoretical Computer Science*, vol. 267, 2000.

[69] C. Paradis, *Degree modifiers of adjectives in spoken British English*. Lund studies in English, Lund University Press, 1997.

[70] A. G. Boulanger, A. C. Chu, S. Maxx, and D. L. Waltz, "Vehicle Electrification: Status and Issues," *Proceedings of the IEEE*, vol. 99, June 2011.

[71] Nissan Carwings. http://www.nissanusa.com/innovations/carwings.article.html.

[72] OnStar. https://www.onstar.com/web/portal/home.

[73] OpinionEQ, "Sentiment Analysis with Unmatched Accuracy." http://www.opinioneq.com/.

[74] B. Liu and M. Hu, "Opinion Mining, Sentiment Analysis, and Opinion Spam Detection." http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.