Inferring Aspect-Specific Opinion Structure in Product Reviews Using Co-training

Dave Carter[1,2] and Diana Inkpen[1]

[1] University of Ottawa, School of Electrical Engineering and Computer Science
david.carter@cnrc-nrc.gc.ca
[2] National Research Council Canada
diana.inkpen@uottawa.ca

Abstract. Opinions expressed about a particular subject are often nuanced: a person may have both negative and positive opinions about different aspects of the subject of interest, and these aspect-specific opinions can be independent of the overall opinion. Being able to identify, collect, and count these nuanced opinions in a large set of data offers more insight into the strengths and weaknesses of competing products and services than does aggregating overall ratings. We contribute a new confidence-based co-training algorithm that can identify product aspects and sentiments expressed about such aspects. Our algorithm offers better precision than existing methods, and handles previously unseen language well. We show competitive results on a set of opinionated sentences about laptops and restaurants from a SemEval-2014 Task 4 challenge.

1 Introduction

Humans are opinionated beings. Some opinions may be arbitrary, but many are nuanced and explicitly supported. People share their opinions online in great numbers. The deluge of available text makes these opinions accessible but, paradoxically, due to their sheer number, it becomes increasingly difficult to synthesize and generalize these opinions. The goal of this work is to develop usable and useful software that, given a set of casually written product reviews, identifies products' aspects (features) and infers writers' opinions about these aspects. Such aspect-specific sentiments can be aggregated to support decision making.

For example, consider the sentence: *I love my new iPhone because of its amazing screen but the battery is barely sufficient to get me through the day.*

There are three sentiments expressed in this sentence:

- a positive sentiment about the iPhone itself;
- a positive sentiment about the screen; and
- a negative sentiment about the battery or battery life.

The *screen* and the *battery [life]* are two aspects of the product *iPhone.* We seek to automatically annotate these two aspects in such a sentence and correctly infer that the writer has a positive sentiment about the screen and a

A. Gelbukh (Ed.): CICLing 2015, Part II, LNCS 9042, pp. 225–240, 2015.
DOI: 10.1007/978-3-319-18117-2_17

negative sentiment about the battery life, without being sidetracked by the positive sentiment about the phone itself. (Perhaps a very simple natural language processing system might see that *battery* and *love* appear in the same sentence and infer that the writer has a positive opinion of the battery life; avoiding such incorrect inferences is a challenge of doing aspect-based sentiment analysis well.)

This work uses co-training to try to take advantage of unlabelled data to find these aspects, rather than using only human-annotated data; the former is much cheaper and easier to procure, and is more readily available. Co-training has been used for various tasks since 1998, but has never, to our knowledge, been applied to the task of aspect-specific sentiment analysis.

The co-training algorithm we developed for aspect-specific sentiment analysis offers high precision: it is likely to get its predictions correct, at the expense of making fewer predictions (or, put more archaically: it makes sins of omission, but few sins of commission). High precision matches a naïve intuition of "correctness" fairly well; and high-precision, lower-recall systems can be combined in ensemble learning to create powerful voting systems like IBM's Watson [1].

While sentiment classification of text has been attempted computationally for roughly twenty years now, aspect-specific sentiment identification is a newer task in natural language processing that is undergoing active research at present (e.g., as part of the SemEval-2014 competition, wherein there was a shared task called Aspect Based Sentiment Analysis that attracted submissions from 31 teams).

This paper unfolds as follows. Related work is presented in the following section. An overview of our experimental methods ensues, and includes a description of the algorithm developed. The algorithm is evaluated on data from the SemEval-2014 Task 4 challenge and results are compared to the results of the teams that participated in the challenge. Finally, conclusions follow.

2 Related Work

There have been attempts at inferring the sentiment of sentences using computers for twenty years, with some approaches based on manually coded rules derived from observed linguistic phenomena, and some using machine learning and other forms of artificial intelligence. Our work draws on both approaches.

The broad field of sentiment analysis is well-established. Commendable works that survey the state-of-the-art in sentiment analysis include [2], [3], and [4].

2.1 Sentiment Analysis Using Product Reviews

Product reviews are useful data to work on for aspect-based sentiment analysis. One conclusion of [4] is that reviews of restaurants, hotels, and the like have a significant influence on consumers' purchasing decisions, and that, depending on the type of product or service, 20% to 99% of consumers will pay more for an item that is rated five stars out of five than a competing item ranked four stars out of five. In a similar vein, [5] discusses abstractly why product reviews are useful, while [6] describes how reviews impact the pricing power of an item.

Various experiments have been performed on Amazon reviews. One of the first sets of Amazon data used for sentiment analysis was annotated in [7] and then used in an experiment that predicted aspect-specific sentiments expressed in the data. A similar experiment was performed on the same data in [8], using rules based on parse trees. A system working on a subset of the same data set is described in [9]; it tries to classify (using linguistic rules) the polarity of sentiment-bearing words in context (one person might like his or her phone to be *small* and their car *big*, while another might prefer a *big* phablet and a *small* sporty car). Opinions about aspects of products as stated in Amazon reviews were analyzed by [10], using reviews of books, DVDs, electronics, and kitchen appliances; impressive domain adaptation results were achieved.

Further efforts to identify aspect-specific sentiments expressed in text have used other data sets. An effort to identify aspect-opinion pairs at the sentence level is described in [11], using a mix of web pages, camera reviews, and news articles. Experiments in using latent discourse analysis (LDA) are described in [12] and [13], identifing product aspects in reviews and then matching tokens from the reviews' sentences that correspond to each product aspect. A similar LDA-based experiment is described in [14]; while it does not perform as well as supervised models, it also doesn't need sentiment-bearing words to be labelled in the input data. Finding topics and associated opinions, a task not unlike that of aspect-specific sentiment extraction, is pursued by [15] using opinionated text about laptops, movies, universities, airlines, and cities. Product aspect-sentiment pairs are identified in online message board postings in [16] and are used to generate marketing intelligence summaries. Unsupervised methods are used to mine aspects and sentiments for restaurants and netbooks in [17]. Interestingly, they found that the extracted aspects were more representative than a manually-constructed list on the same data, avoiding problems of over-generalization or over-representation (being too granular or too fine-grained in combining similar aspects). The work of [18] tries to be aspect-specific, first mining the product aspects and then the opinion polarities of each aspect using CNet and Amazon reviews; in practice, the only experiment in which they have reasonable results is classifying the polarity of the review (i.e., at the document level). Topics and sentiment orientations are identified in car reviews in [19], using clustering techniques to mine unigrams mentioned in positive and negative contexts for different makes and models of cars; some aspect-specific sentiments are found in this manner, though results are noisy.

2.2 Co-training

Co-training is a semi-supervised learning approach that uses both labelled and unlabelled data. Two classifiers try to classify the same data into the same classes using different and uncorrelated sets of features ("views", in co-training parlance). The algorithm iteratively builds larger and larger sets of training data.

Co-training was introduced by Blum and Mitchell [20]. They present an approach for using a small set of labelled data and a large set of unlabelled data to iteratively build a more complete classifier model. Classification features are divided into two views. The main example they provided was a task to classify web pages by topic, where one view was the textual content of the pages, and the other view was composed of the text in links used to access the pages. Two assumptions are made: that each view is sufficient to classify the data, and that the views are conditionally independent given the class label.

Many saw promise in Blum and Mitchell's proposed co-training algorithm but sought to alleviate some concern about the two assumptions it made about the co-training views. Evidence supporting that the conditional independence requirement can be relaxed is offered in several works ([21], [22], [23], [24], [25]). An alternative to the conditional independence assumption is offered in [26]; an *expansion* assumption of the data is proposed and a proof is offered that data meeting this assumption will derive benefit from a co-training approach (also assuming that the underlying machine learning classifiers are never confident in their classifications in cases when they are incorrect). The authors assume that the views need be at most "weakly dependent", rather than assuming conditional independence; and are, in fact, quite explicit in stating that this assumption is the "right" assumption compared to the earlier assumption. (It is worth noting that Blum is a co-author of this paper). Finally, a very practical analysis of the assumptions underlying co-training is offered by [27].

Confidence-based co-training (where a classifier estimates class probabilities, and only those with high probabilities are added to training data in subsequent iterations) worked well in [28]. Confidence-based co-training was also used in [29]; they sample the data where the two views' classifiers agree the most.

Limitations of co-training have been posited. Co-training improves classifier performance to a certain threshold (as high-confidence data are added to the training models in early iterations), and then as more examples are added, performance declines slightly [30]. The usefulness of co-training depends largely on (and is roughly proportional to) the difference between the two views [31].

Co-training has been used in a few sentiment analysis tasks. Blum and Mitchell's algorithm is used in [32] to do sentiment classification on reviews, using Chinese data as one view and English data as a second view. Sentiment classification of tweets is offered in [33] and [34] using co-training, while [35] uses co-training to identify sentiment in an online healthcare-related community. Co-training has been applied to other natural language processing tasks including email classification [36], sentence parsing [37], word sense disambiguation [38], co-reference resolution [39], and part-of-speech tagging [40].

3 Methods

We work with text from product reviews collected in [41]. This text is written by the general public, and each sentence contains one or more specific aspects of products along with, in most cases, a stated opinion about each aspect.

We aim to identify the aspect-specific opinions expressed in a sentence. We construe this as a classification task, where each word in a sentence can be classified as one of three mutually exclusive cases:

- The word is inside (is part of) a product aspect
- The word is inside (is part of) an expression of a sentiment
- The word is outside of a product aspect or expression of a sentiment

The software we developed takes a sentence as input and returns a list of zero or more tagged stated opinions about aspects of a product. The system is based on machine learning plus a handful of heuristics. Co-training underlies our method, as it offers two advantages over fully supervised learning: the (optional) ability to use unlabelled data to build a larger set of training data, and its use of independent and sufficient views.

Co-training is a semi-supervised classification algorithm that augments a small set of labelled data with a large set of unlabelled data to reduce the error rate in a classification task [20]. A main motivation of such an approach is that labelled data is "expensive" (as it is usually hand-labelled by humans, which incurs time and/or monetary costs), and so any improvement in results that can be gleaned from unlabelled data is essentially "free" [20]. Co-training uses two conditionally independent "views" of the data being classified. Each such view must be (at least theoretically) sufficient to classify the data. Co-training iteratively builds up each classifier's knowledge by adding high-confidence classified cases to the training set; the expertise of one classifier is used to train the other.

We use [surface-level] lexemes (including predicted part-of-speech) and syntactic features as the two views. In English, as in many languages, there is not a one-to-one relation between lexemes and their part of syntax; they may be independent for all but the most basic of functional words (conjunctions, particles, and the simplest adverbs and personal pronouns, for example).

The lexical view is inspired by collocations. For example, a word following the fragment "I like my phone's ..." is fairly likely to be a product aspect, and is unlikely to be a sentiment-bearing word (unless, perhaps, it is a superlative adjective followed by the aspect). Features in the lexical view include the surface form of the token, its lemma, and its (predicted) part-of-speech. In addition, these same features are recorded for the preceding and following three tokens for each given token; this is somewhat inspired by work on extraction patterns, where a pattern like "I like my *some_product_name* despite its rather poor *some_product_attribute*" can be used to extract product names and product attributes with fairly high confidence.

The syntactic view is inspired by the observation that both product aspects and sentiment-bearing words appear in a limited number of grammatical structures. For example, a noun that is the direct object of a verb may be more likely to be a product aspect than the verb itself; in contrast, a verb is more likely to express sentiment than it is to be a product aspect. Features in the syntactic view include the node in the parse tree immediately above the token; the chain of nodes above the token in the parse tree up to the nearest sentential unit; the chain of nodes above the token in the full parse tree; whether the token is referred to by a pronoun elsewhere in the sentence; a list of dependency relations in which the token participates (e.g., whether it is a direct object of another word in the sentence); its predicted semantic role (e.g., whether it is a subject in a sentence); and whether it participates in a negation clause. The intent is to work in a manner similar to extraction patterns but to do so in a way that reflects the complexity of language, particularly long-distance dependencies that might not be accounted for in an n-gram model.

We begin with the Blum & Mitchell algorithm (Algorithm 1, left column) [20], and modify it to classify using a confidence score (Algorithm 2, right column).

Both algorithms begin with set (L) of labelled training examples and a set (U) of unlabelled examples. The Blum and Mitchell algorithm then selects a random pool of unlabelled examples (*U'*) that is much smaller than the full unlabelled set, and enlarges this pool every iteration; whereas our algorithm considers all remaining unlabelled examples (*U'*) at each iteration. The Blum and Mitchell algorithm iterates a fixed number of times (k); whereas our algorithm keeps running as long as the number of unlabelled examples that could be classified in the previous iteration ($n_{i\text{-}1}$) is greater than zero. Each iteration, both methods have a classifier train itself on a single view of all labelled data (including data that have been labelled successfully in previous iterations), then classify the data in (*U'*). At this point, Blum and Mitchell randomly pick one positive and three negative examples to add to the set of labelled data; whereas our algorithm adds to the set of labelled data those data about which it was most confident. This confidence metric is defined as the confidence of the most confident classifier; a more complex scoring function could be used, such as the amount of agreement or the amount of disagreement between the two classifiers, as suggested by [29].

The most notable divergence from the Blum and Mitchell algorithm is the decision to add a large number of examples at each iteration, so long as the classifiers are confident in their classification of such unlabelled examples. We have chosen to implement an upper limit in the algorithm, so that, rather than accepting all new unlabelled examples that can be classified with a confidence of, say, 55%, it will only accept the top m-most cases. The intuition is that it may be desirable, especially in the early iterations, to add only the most confident examples and retrain so as to be able to more confidently label the next set; the expertise added by only accepting the highly confidently-labelled examples may be sufficient to more confidently classify the merely marginal unlabelled examples that may have been classified with confidence at or just above the threshold. In practice, the algorithm tends to use this upper limit in only the first several iterations; after roughly the fifth iteration, the confidence threshold determines the number of unlabelled examples added at each iteration, as the most obvious examples have already been added to the labelled set.

While Blum and Mitchell's algorithm takes as an input the maximum number of iterations k (which would also presumably have to scale proportionally to the size of the data set), our algorithm requires the maximum number of new examples to label in each iteration m, which roughly determines the number of iterations ($i_{\max}$) for a given confidence threshold. In practice, there tend to be roughly $i_{\max} = 8$ iterations required to process the SemEval-2014 task 4 data.

```
Given:
  – a set $L$ of labelled training examples with features $x$
  – a set $U$ of unlabelled examples with features $x$
Create a pool $U'$ of examples by choosing $u$ examples at random from $U$
Result: An enlarged pool $L'$
initialization;
for $i \leftarrow 1$ to $k$ do
    Use $L$ to train a classifier $h_1$ that considers only the $x_1$ portion of $x$;
    Use $L$ to train a classifier $h_2$ that considers only the $x_2$ portion of $x$;
    Allow $h_1$ to label $p$ positive & $n$ negative examples from $U'$;
    Allow $h_2$ to label $p$ positive & $n$ negative examples from $U'$;
    Add these self-labelled examples to $L$;
    Randomly choose $2p + 2n$ examples from $U$ to replenish $U'$;
end
with typical $p = 1$, $n = 3$, $k = 30$, $u = 75$;
```

Algorithm 1. Blum and Mitchell's co-training algorithm [20] (largely verbatim)

```
Given:
  – a set $L$ of labelled training examples with features $x$
  – a set $U$ of unlabelled examples with features $x$
Create a pool $U'$ of all examples from $U$
Result: An enlarged pool $L'$
initialization;
$i = 1$;
while $i = 1$ or $n_{i-1} > 0$ do
    Use $L$ to train a classifier $h_1$ that considers only the $x_1$ portion of $x$;
    Use $L$ to train a classifier $h_2$ that considers only the $x_2$ portion of $x$;
    Allow $h_1$ to label all examples from $U'$;
    Allow $h_2$ to label all examples from $U'$;
    Sort these self-labelled examples in descending order of $max(confidence\ of\ h_1, confidence\ of\ h_2)$;
    Add the top $n$ most confidently labelled examples to $L$ where $n \leq m$ and the confidence of the prediction of every such example is greater than $c$;
    $i \leftarrow i + 1$;
end
with typical $m = 2500$, $c = 0.55$, $i_{\max} \approx 8$;
```

Algorithm 2. Our co-training algorithm using confidence-based classification

The confidence threshold c in our algorithm is tuneable. This parameter serves as classification confidence floor; the algorithm will not include any labelled examples when the confidence in that example's classification is less than this floor. The support vector machine classifier we selected offers fairly good classification performance, so we set this threshold to a relatively low 0.55 for all experiments described herein. (A grid search classifying the development data with confidence thresholds $c \in \{0.00, 0.45, 0.50, 0.55, 0.65, 0.75, 0.85, 0.95\}$ revealed that 0.55 was close to optimal.)

LibSVM [42], a support vector machine classifier, was used for classification; a radial basis function (RBF) kernel was used. SVM tuning parameters are provided in our source code, which we make available.[1] The SVM classifiers were tuned on the first 20% of the sentences in each of the five data sets in [7].

4 Evaluation

4.1 Data

The data set we selected for experimentation was originally developed in [43], containing restaurant review text from Citysearch New York, and was modified and enlarged for an aspect-specific sentiment analysis task at SemEval-2014 [41]. This SemEval-2014 data set contains sentences extracted from reviews of restaurants (3041 training sentences and 800 test sentences) and reviews of laptops of different brands (3045 training sentences and 800 test sentences). Aspects that appear in the sentence are tagged and assigned a sentiment polarity of positive, neutral, negative, or "conflict", the latter referring to cases where both positive and negative sentiments about the aspect appear in the same sentence (along the lines of "the service was friendly but slow"). The data are written by casual writers, but, subjectively, the quality of the writing appears to be rather good; spelling errors and instances of odd formatting (like informally-bulleted lists) that plague some other data sets seem to be rare.

This particular data set offers a good basis for comparison for our approach to sentiment analysis. The competition drew 57 submissions for the first phase of evaluation and 69 for the second phase of evaluation.

The aspects are carefully tagged in the data, including character positions. The sentiment-bearing words themselves are not tagged, so it is up to the software to determine in some other manner how and where the sentiment is expressed in the sentence; we used the lexicon from [7] for this. An example is:

```
<sentence id="337">
    <text>However, the multi-touch gestures and large tracking area make having an
         external mouse unnecessary (unless you're gaming).</text>
    <aspectTerms>
        <aspectTerm term="multi-touch gestures" polarity="positive" from="13" to="33"/>
        <aspectTerm term="tracking area" polarity="positive" from="44" to="57"/>
        <aspectTerm term="external mouse" polarity="neutral" from="73" to="87"/>
        <aspectTerm term="gaming" polarity="neutral" from="115" to="121"/>
    </aspectTerms>
</sentence>
```

[1] https://github.com/davecart/cotraining

We tokenized and processed the sentences using the Stanford CoreNLP tools, which labelled each token with its predicted part-of-speech, its lemma, whether it is believed to be a named entity (e.g., a brand name); and built a parse tree for the sentence, with coreferences labelled.

4.2 Classifying Product Aspects

We compared our system's ability to label product aspects in sentences (independent of any effort to glean associated sentiments) to the results of those who participated in the SemEval-2014 task 4 subtask 1 challenge.

Our system, operating in a supervised manner and allowing only an exact match in cases where aspects were composed of multiple words, offered higher precision than all other systems on the laptop reviews, though perhaps not significantly so. Our system – whether running in a supervised manner or using co-training with only half of the training data being labelled – offered precision on the restaurant reviews that was roughly tied with the top competitor, and much higher than the mean and median.

Table 1. Comparing aspect classification results on SemEval-2014 task 4 (subtask 1) data

Data set	Laptop reviews				Restaurant reviews			
	P	R	F_1	A	P	R	F_1	A
SemEval-2014 task 4 subtask 1 (aspect term extraction)								
mean performance	0.690	0.504	0.562	-	0.767	0.672	0.708	-
lowest performance	0.231	0.148	0.239	-	0.371	0.340	0.383	-
median performance	0.756	0.551	0.605	-	0.818	0.720	0.727	-
highest performance	0.848	0.671	0.746	-	0.909	0.827	0.840	-
Our results								
- fully supervised, using all training data	0.863	0.401	0.547	0.632	0.915	0.681	0.781	0.647
- training with first half, co-training with second half	0.822	0.292	0.430	0.581	0.909	0.587	0.713	0.589
- training with second half, co-training with first half	0.829	0.224	0.353	0.559	0.910	0.616	0.734	0.606

Our system offered weaker performance in recall: somewhat below the mean and median when processing the laptop reviews in a supervised manner, and roughly tied with the mean and well below the median when examining restaurant reviews. Co-training offered much worse recall, though still better than the weakest of the SemEval-2014 task 4 competitors.

With high precision and relatively weak recall, our system achieved F_1 scores that placed mid-pack among SemEval-2014 competitors when considering all test data in a supervised manner. When co-training with the laptop data, our

F_1 was below average; whereas when co-training with the restaurant reviews, our F_1 scores were slightly above the mean and tied with the median.

The product aspect classification performance of our system on the SemEval-2014 data can be described as being roughly average among the 31 teams, and is characterized by very high precision and rather low recall.

Accuracy was not reported in the competition results, but we offer our system's accuracy performance for future comparison and to illustrate that, even in cases where recall is low, accuracy remains at reasonable levels.

4.3 Classifying the Sentiments of Aspects

The second subtask in the SemEval-2014 task 4 challenge was to predict the stated aspect-specific sentiment of sentences where the product aspect(s) were already labelled. We compared our system's performance on this task to that of those who entered the challenge (Table 2).

Table 2. Comparing sentiment orientation classification results (given tagged aspects) on SemEval-2014 task 4 (subtask 2) data

Data set	Laptop reviews	Restaurant reviews
	Accuracy	Accuracy
SemEval-2014 task 4 subtask 2 (determine polarity, given aspects)		
mean performance	0.590	0.691
lowest performance	0.365	0.417
median performance	0.586	0.708
highest performance	0.705	0.810
Our results		
- fully supervised, using all training data)	0.719	0.690
- training with first half, co-training with second half)	0.668	0.643
- training with second half, co-training with first half)	0.662	0.631

Accuracy was the only metric reported by the challenge organizers; accordingly, this is the only metric that we report.

When our system was used in an entirely supervised manner, it (just barely, and probably not significantly) bested all competitors in the laptops portion of the SemEval-2014 task 4 challenge. Even the co-training results are well above both mean and median on the laptop reviews. On the other hand, when trying to classify sentiments in the restaurant reviews, performance of the supervised system was tied with the mean and very slightly lower than the median competitor; and the accuracy when co-training was almost 10% worse than the mean and median competitor, though still much better than the least successful teams that participated in the challenge.

Our system thus appears to offer fairly compelling performance in classifying the sentiments expressed about known product aspects in these data sets, even when co-training with only half of the training data being labelled.

4.4 Performance Finding All Aspect-Sentiment Pairs in a Sentence

It is a more challenging and more interesting task to classify both product aspects and the associated sentiments in a sentence than is classifying aspects in isolation. Sadly, this was not a part of the SemEval-2014 task 4 challenge, although it would be a natural extension thereof. Our system's sentence-level results are listed in Table 3.

Table 3. Aspect-specific opinion inference results on SemEval-2014 task 4 competition data (classifying aspects and sentiments simultaneously, given unlabelled sentences)

Data set	Laptop reviews				Restaurant reviews			
	P	R	F_1	A	P	R	F_1	A
SemEval-2014 Task 4 sentences								
- fully supervised, using all training data, all test data	0.890	0.268	0.412	0.596	0.936	0.507	0.658	0.600
- training with first half, co-training with second half	0.880	0.121	0.213	0.528	0.933	0.321	0.477	0.468
- training with second half, co-training with first half	0.935	0.103	0.185	0.523	0.923	0.354	0.512	0.488
- mean performance loss when using co-training with only half of the labelled training data	-2%	58%	52%	12%	1%	34%	25%	20%

The performance of co-training in a real-world and suitably difficult task can be analyzed here. The co-trained models were trained using only half as much labelled data as the supervised model. Precision remained sufficiently high to conclude that it was tied with the supervised model. Recall dropped quite a bit. In the laptop reviews, the F_1 score roughly halved, whereas in the restaurant reviews it dropped an average of 25%. Accuracy suffered 22% in the worst of the trials. These results are somewhat comforting: using only half as much training data seems to reduce the F_1 by half, at worst, while maintaining high precision. This could be an acceptable trade-off in a particular application domain, since labelled data is both difficult and expensive to produce.

By comparison, [44] offers insight into humans' classification performance. Humans seem to be able to classify polarity at the sentence level with roughly 88% precision and 70% recall. Our system, performing a more nuanced task of classifying aspect-specific sentiments at the sentence level, meets this level of precision, if not exceeding it; though it does nowhere near as well at recall. (Human brains, viewed as a natural language processing machine, are trained on much larger language models than our system, so one could intuitively expect that humans might have better recall than NLP software trained on a mere 3000 sentences.)

5 Conclusions and Further Work

Useful and useable software was developed that can label sentiments expressed about specific aspects of a product. The software developed is characterized by its very high precision and somewhat weak (or, in some cases, very weak) recall. It is better at classifying the sentiments expressed about known attributes in laptop reviews than any of the 31 teams who performed the same task in a recent international NLP challenge (SemEval-2014 task 4).

The software can be trained with only labelled data (supervised learning), or can be trained with fewer labelled data and a set of unlabelled data (co-training); unlabelled data are more readily available and much cheaper to procure or produce. When using co-training to perform this aspect-specific sentiment analysis, precision remains high or improves very slightly, at the expense of some recall. This appears to be the first application of co-training to aspect-based sentiment analysis. The algorithm implemented differs from the commonly accepted co-training algorithm of [20], offering better scalability and taking advantage of the ability of newer machine learning classifiers to estimate the confidence in their own predictions. We believe that the tuneable parameters of the algorithm herein are more intuitive than those in [20].

The co-training algorithm developed in our work could be applied to other tasks, both within the natural processing domain and outside of it. (By comparison, Blum and Mitchell's co-training algorithm has found diverse applications).

In the future, it could be interesting to incorporate work on opinion strength. At present, we lump together all positive and all negative opinions, whereas in natural language, opinions are more nuanced. If a consumer is using comparative ratings of an aspect-specific sentiment classification system to make informed choices, it is probably advantageous that the strength of the opinions be known and aggregated (e.g., a cell phone with many weakly negative opinions about the battery life might be preferable to one with a similar number of very strong negative opinions about its battery life). There is some existing academic work on strength-based sentiment classification, e.g., [45] and [46], so that would seem a natural pairing.

One necessary compromise in trying to learn only aspect-specific sentiments herein was a willful ignorance of sentiments expressed about the products (atomically) or the products' brands. A step forward might be incorporating classifiers designed to label such expressions at the same time as labelling aspects and sentiments; the sentiment-bearing word classifier could likely be used as-is. The architecture of the system developed herein can be extended to any n lexically mutually exclusive classes; this could include named entities, competing brands, or retailers. With additional learning models for products (and synonyms thereof) and brands (perhaps by using a named entity tagger), a better picture of both the broader and more specific opinions expressed in text might be gleaned, for a better overall understanding of the text.

Some semi-supervised algorithms (e.g., that in [47]) run a prediction on all *training* data at each iteration to see if, for example, a borderline example that was added in a previous iteration should now be rejected from the training data because it now falls below a particular threshold due to the new knowledge gained by the classifier in the meantime (termed "escaping from initial misclassifications" in the Yarowsky paper). That could be a compelling addition to our approach.

References

1. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., et al.: Building Watson: An overview of the DeepQA project. AI Magazine 31, 59–79 (2010)
2. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Mining Text Data, pp. 415–463. Springer (2012)
3. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies 5, 1–167 (2012)
4. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2, 1–135 (2008)
5. Ghose, A., Ipeirotis, P.G.: Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. IEEE Transactions on Knowledge and Data Engineering 23, 1498–1512 (2011)
6. Archak, N., Ghose, A., Ipeirotis, P.G.: Show me the money!: Deriving the pricing power of product features by mining consumer reviews. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007, pp. 56–65. ACM, New York (2007)
7. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004, pp. 168–177. ACM, New York (2004)
8. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Natural Language Processing and Text Mining, pp. 9–28. Springer (2007)
9. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM 2008, pp. 231–240. ACM, New York (2008)
10. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: ACL, vol. 7, pp. 440–447 (2007)
11. Nasukawa, T., Yi, J.: Sentiment analysis: Capturing favorability using natural language processing. In: Proceedings of the 2nd International Conference on Knowledge Capture, K-CAP 2003, pp. 70–77. ACM, New York (2003)
12. Titov, I., McDonald, R.: A joint model of text and aspect ratings for sentiment summarization. In: Proc. ACL 2008: HLT, pp. 308–316 (2008)
13. Titov, I., McDonald, R.: Modeling online reviews with multi-grain topic models. In: Proceedings of the 17th International Conference on World Wide Web, WWW 2008, pp. 111–120. ACM, New York (2008)

14. Jo, Y., Oh, A.H.: Aspect and sentiment unification model for online review analysis. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 815–824. ACM, New York (2011)
15. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of the 16th International Conference on World Wide Web, pp. 171–180. ACM (2007)
16. Glance, N., Hurst, M., Nigam, K., Siegler, M., Stockton, R., Tomokiyo, T.: Deriving marketing intelligence from online discussion. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD 2005, pp. 419–428. ACM, New York (2005)
17. Brody, S., Elhadad, N.: An unsupervised aspect-sentiment model for online reviews. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT 2010, pp. 804–812. Association for Computational Linguistics, Stroudsburg (2010)
18. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th International Conference on World Wide Web, WWW 2003, pp. 519–528. ACM, New York (2003)
19. Gamon, M., Aue, A., Corston-oliver, S., Ringger, E.: Pulse: Mining customer opinions from free text. In: Proc. of the 6th International Symposium on Intelligent Data Analysis, pp. 121–132 (2005)
20. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, pp. 92–100. ACM, New York (1998)
21. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 100–110 (1999)
22. Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proceedings of the 17th International Conference on Machine Learning, pp. 327–334. Morgan Kaufmann (2000)
23. Dasgupta, S., Littman, M.L., McAllester, D.: Pac generalization bounds for co-training. Advances in Neural Information Processing Systems 1, 375–382 (2002)
24. Abney, S.: Bootstrapping. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL 2002, pp. 360–367. Association for Computational Linguistics, Stroudsburg (2002)
25. Wang, W., Zhou, Z.H.: Co-training with insufficient views. In: Asian Conference on Machine Learning, pp. 467–482 (2013)
26. Balcan, M.F., Blum, A., Yang, K.: Co-training and expansion: Towards bridging theory and practice. In: Advances in Neural Information Processing Systems, pp. 89–96 (2004)
27. Du, J., Ling, C.X., Zhou, Z.H.: When does cotraining work in real data? IEEE Trans. on Knowl. and Data Eng. 23, 788–799 (2011)
28. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Proceedings of the Ninth International Conference on Information and Knowledge Management, CIKM 2000, pp. 86–93. ACM, New York (2000)
29. Huang, J., Sayyad-Shirabad, J., Matwin, S., Su, J.: Improving multi-view semi-supervised learning with agreement-based sampling. Intell. Data Anal., 745–761 (2012)
30. Pierce, D., Cardie, C.: Limitations of co-training for natural language learning from large datasets. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing, pp. 1–9 (2001)

31. Wang, W., Zhou, Z.-H.: Analyzing co-training style algorithms. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 454–465. Springer, Heidelberg (2007)
32. Wan, X.: Bilingual co-training for sentiment classification of chinese product reviews. Computational Linguistics 37, 587–616 (2011)
33. Liu, S., Li, F., Li, F., Cheng, X., Shen, H.: Adaptive co-training svm for sentiment classification on tweets. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management, CIKM 2013, pp. 2079–2088. ACM, New York (2013)
34. Liu, S., Zhu, W., Xu, N., Li, F., Cheng, X.Q., Liu, Y., Wang, Y.: Co-training and visualizing sentiment evolvement for tweet events. In: Proceedings of the 22nd International Conference on World Wide Web Companion, WWW 2013 Companion, pp. 105–106. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva (2013)
35. Biyani, P., Caragea, C., Mitra, P., Zhou, C., Yen, J., Greer, G.E., Portier, K.: Co-training over domain-independent and domain-dependent features for sentiment analysis of an online cancer support community. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2013, pp. 413–417. ACM, New York (2013)
36. Kiritchenko, S., Matwin, S.: Email classification with co-training. In: Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research, CASCON 2001, p. 8. IBM Press (2001)
37. Sarkar, A.: Applying co-training methods to statistical parsing. In: Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies, NAACL 2001, pp. 1–8. Association for Computational Linguistics, Stroudsburg (2001)
38. Mihalcea, R.: Co-training and self-training for word sense disambiguation. In: Proceedings of the Conference on Computational Natural Language Learning, CoNLL 2004 (2004)
39. Ng, V., Cardie, C.: Bootstrapping coreference classifiers with multiple machine learning algorithms. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, pp. 113–120. Association for Computational Linguistics, Stroudsburg (2003)
40. Clark, S., Curran, J.R., Osborne, M.: Bootstrapping pos taggers using unlabelled data. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, CONLL 2003, vol. 4, pp. 49–55. Association for Computational Linguistics, Stroudsburg (2003)
41. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: Semeval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the International Workshop on Semantic Evaluation (SemEval) (2014)
42. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
43. Ganu, G., Elhadad, N., Marian, A.: Beyond the stars: Improving rating predictions using review text content. In: Proceedings of the 12th International Workshop on the Web and Databases, WebDB 2009 (2009)
44. Nigam, K., Hurst, M.: Towards a robust metric of opinion. In: AAAI Spring Symposium on Exploring Attitude and Affect in Text, pp. 598–603 (2004)
45. Wilson, T., Wiebe, J., Hwa, R.: Just how mad are you? finding strong and weak opinion clauses. In: Proceedings of AAAI, pp. 761–769 (2004)

46. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems 21, 315–346 (2003)
47. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics, ACL 1995, pp. 189–196. Association for Computational Linguistics, Stroudsburg (1995)