Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Acquiring Commonsense Knowledge for Sentiment Analysis through Human Computation

Marina Boia and **Claudiu Cristian Musat** and **Boi Faltings**
École Polytechnique Fédérale de Lausanne
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
{marina.boia, claudiu-cristian.musat, boi.faltings}@epfl.ch

Abstract

Many Artificial Intelligence tasks need large amounts of commonsense knowledge. Because obtaining this knowledge through machine learning would require a huge amount of data, a better alternative is to elicit it from people through human computation. We consider the sentiment classification task, where knowledge about the contexts that impact word polarities is crucial, but hard to acquire from data. We describe a novel task design that allows us to crowdsource this knowledge through Amazon Mechanical Turk with high quality. We show that the commonsense knowledge acquired in this way dramatically improves the performance of established sentiment classification methods.

1 Introduction

Many tasks in Artificial Intelligence (AI) require commonsense knowledge about the real world. Examples include image understanding, question answering, or processing language text. We consider sentiment classification, which determines whether a text passage expresses a positive or a negative sentiment. This task requires commonsense knowledge about the polarities of sentiment words.

Humans can identify the sentiment of texts with 80% to 90% agreement (Musat and Faltings 2013), depending on the text domain. The most successful AI methods syntactically preprocess the texts to extract the relevant words, then use the individual words as features. Sentiment classification is performed by summing up word polarity scores compiled in sentiment lexicons or automatically extracted with machine learning algorithms. These methods obtain from 60% to 80% accuracy (Turney 2002; Pang, Lee, and Vaithyanathan 2002), well below that of humans.

A reason why these classifiers do not achieve human-level performance is that word polarities are context-dependent: a *cold pizza* is negative, but a *cold beer* is positive. When texts are treated as sets of independent words, context is lost. For very specific domains, the context of sentiment words is constant and thus does not need to be modeled. Also, the context itself can be considered a sentiment word and thus taken into account without an explicit model. For example, on a corpus of hotel reviews, a machine learning algorithm may identify the term *carpet* as a negative sentiment word since the corpus contains it only in negative documents. However, approximating contextual dependencies through very specific models is unlikely to reach human-level performance on broad document collections, since classifiers trained and evaluated on different source and target domains typically perform very poorly (Blitzer, McDonald, and Pereira 2006; Pan et al. 2010).

Nonetheless, even without knowing the topic of conversation, people are able to identify the correct polarity by using only the context within a sentence, producing a much more reliable classification. To reach a similar performance, sentiment classifiers should also model context at the sentence-level, for example by considering word combinations. However, because of their high complexity, it is not feasible to acquire word combination models from data. Still, contextual dependencies are not very complex to characterize. The polarities of most words have only a few exceptions, so the size of these models could be manageable if these exceptions are identified. This is very difficult to do with statistical methods, but easy for people. Therefore, we investigate how contextual knowledge can be obtained through human computation.

We ask people to describe the contexts that impact the polarities of sentiment words. Compared to traditional polarity annotation tasks (Hong et al. 2013), this knowledge acquisition task is more challenging because:

- It requires cognitive engagement, so humans can quickly lose motivation, giving sloppy answers.
- Context can be indicated in many ways, thus quality assurance by agreement with peers is generally not feasible.

To increase motivation, we introduce the entertaining aspect of games to the large worker pool available on Amazon Mechanical Turk[1]. We make our task fun by packaging it as a game. Workers play in rounds and increase their score by submitting answers that contain a sentiment word, a context, and a polarity. At the end, we reward workers with a payment proportional to their score.

To ensure quality, we use an intelligent scoring mechanism which drives workers to give useful answers that have common sense and are novel. We facilitate this by maintain-

[1] www.mturk.com

ing a model of word polarities in context, which we initialize from existing sentiment knowledge and refine with the workers' input. We reward the answers that agree with and strengthen this model, which is what we consider commonsensical and novel, respectively. With this interpretation of useful answers, we successfully create the illusion of a game played with others.

The output of the game is a context-dependent sentiment lexicon, which we obtain from the workers' activity on reviews of vacuum cleaners and digital cameras. On these source domains, our knowledge dramatically outperforms several established context-independent lexicons. For the best performing lexicon, we increase accuracy with 16.8%. On four target domains, we also substantially surpass a machine learning model and an automatically generated context-dependent lexicon. On book reviews, we top the former with 12.8% accuracy. On hotel reviews, we surpass the latter with 22.5% recall.

This paper thus makes three main contributions:

- We are the first ones to obtain high-quality contextual sentiment knowledge using human computation. The resulting knowledge dramatically improves several established lexicons and surpasses both a machine learning model and an automatically generated context-dependent lexicon.
- We convert a complex commonsense knowledge acquisition task into a fun game for Amazon Mechanical Turk, thus motivating workers.
- We develop a scoring mechanism that maintains the illusion of synchronous worker interaction, thus driving workers to give useful answers.

This paper is structured as follows. Section 2 presents related work. In Section 3 we explain how we represent contextual sentiment knowledge. Next, Sections 4 and 5 describe the game and quality assurance. In Section 6 we explain how we use the context in sentiment classification, and in Section 7 presents our experiments. Finally, Section 8 draws conclusions.

2 Related Work

2.1 Context in Sentiment Analysis

Sentiment analysis knows two main directions. Lexicon-based methods rely on sentiment lexicons, which enumerate the most common sentiment words and their polarities (Stone et al. 1966; Wilson et al. 2005; Esuli and Sebastiani 2006; Hu and Liu 2004). A main source of errors is that sentiment lexicons contain the most common polarities of words, overlooking the fact that these may be context-dependent.

Corpus-based methods automatically extract sentiment words and polarities from text corpora. This is typically done by applying supervised machine learning algorithms to annotated datasets (Pang, Lee, and Vaithyanathan 2002), or by aggregating syntactic and semantic relations between words (Turney 2002; Hatzivassiloglou and McKeown 1997). These methods do not explicitly model context, but when the texts target narrow enough domains, the resources learned become domain-specific, with sentiment words that occur in only a single context. However, when texts come from a wide range of domains, sentiment words appear in multiple contexts, and accuracy suffers. For instance, classifiers trained and evaluated on different source and target domains normally have serious drops in accuracy (Blitzer, McDonald, and Pereira 2006; Pan et al. 2010).

The methods that do model context at the sentence-level typically extract it from data. (Wilson, Wiebe, and Hoffmann 2005) predicted the polarities of phrases in context with a machine learning model that used as features the context-independent polarities of words and the nearby presence of negations or intensifiers. (Pang, Lee, and Vaithyanathan 2002; Kennedy and Inkpen 2005) trained a sentiment classified model by extending the feature set from individual words only to also include pairs of words. (Lu et al. 2011) automatically generated context-dependent lexicons of sentiment words paired with feature nouns. In feature-level sentiment analysis (Hu and Liu 2004), feature nouns serve as context for sentiment words, but not all methods model how the polarities of words vary depending on what feature is involved (Lau et al. 2009). Nevertheless, automatically learning contextual variations of word polarities is difficult when annotated data is limited. We believe context can be more efficiently identified through human computation, and we ask workers to identify the contexts that influence the polarities of sentiment words.

The deep learning approach of (Socher et al. 2013) has a goal similar to ours. They created a sentiment treebank by asking workers to annotate with polarities all the phrases that occur in the parsee trees of movie review sentences. They used this treebank to train a recursive neural tensor network that learns sentiment compositionality in a semantic vector space for words. However, this approach does not scale, as it requires the polarities of 215,154 phrases (each annotated by three workers) to handle a corpus of only 11,855 sentences. Moreover, the model that results is completely specific to the movie reviews and performs poorly on other domains, like our corpus of vacuum cleaners and digital camera reviews. By comparison, we ask workers to perform feature selection, and we identify only the relevant word combinations. Our context-dependent lexicon is smaller and simpler, and performs well on various domains.

2.2 Human Computation for Sentiment Analysis

The human computation tasks that created resources for sentiment analysis asked workers to provide only simple annotations. (Brew, Greene, and Cunningham 2010) obtained training data through a text polarity annotation task, whereas (Hong et al. 2013) generated a sentiment lexicon through a vocabulary annotation task. (Al-Subaihin, Al-Khalifa, and Al-Salman 2011; Musat, Ghasemi, and Faltings 2012) asked workers to both select sentiment words and annotate them with polarities. By comparison, we require workers to do a more complex task, by characterizing the contexts that impact the polarities sentiment words.

The quality of the resources created depend on the workers' motivation. There are several ways to inspire motivation. Payment is one choice. In online labor markets like Amazon Mechanical Turk, workers are payed for solving

tasks. Another way is through enjoyment. Tasks can be designed as games that reward players with points, reputation, badges etc (von Ahn and Dabbish 2004; Musat, Ghasemi, and Faltings 2012). In human computation, the two have been mutually exclusive so far, and games have not yet taken advantage of the large worker pools available on crowdsourcing platforms. However, we do not see a conflict between payment and enjoyment. We combine the two and create a game played for money, like poker or the games on Swagbucks [2].

Quality can also be explicitly controlled before, during, and after workers solve a task. Before a task, workers can be taught about the desired level of performance with tutorials (Sintsova, Musat, and Pu 2013). During a task, workers can be grouped in teams and required to agree on the answers they provide (von Ahn and Dabbish 2004), or guided through an intelligent scoring mechanism. After a task, it is common to filter or aggregate answers in order to cope with the irrelevant or malicious ones (Ipeirotis, Provost, and Wang 2010). We ensure quality using all three measures

3 Contextual Knowledge Representation

We represent contextual sentiment knowledge using the following concepts:

- A phrase phr is a word construct that can carry sentiment.
- A context con is a word construct in the presence of which a phrase carries sentiment.
- A polarity pol is the positive pos or negative neg orientation of a word construct's sentiment.

We distinguish two kinds of phrases. Unambiguous phrases have the same polarity in every context: *excellent* is always positive. Ambiguous phrases have context-dependent polarities: *low* is positive in the context *price*, but negative in the context *salary*.

Sentiment lexicons are traditionally context-independent. Given a phrase vocabulary P, they enumerate phrases with their polarities: $L = \{(phr, pol) \mid phr \in P, pol \in \{pos, neg\}\}$. This representation either includes ambiguous phrases without their disambiguating contexts, or omits them altogether. Instead, we consider a more expressive, context-dependent representation. Given the phrase and context vocabularies P and C, a context-dependent lexicon adds contexts that clarify the polarities of ambiguous phrases: $CL = \{(phr, con, pol) \mid phr \in P, con \in C, pol \in \{pos, neg\}\}$.

4 Human Computation Task

To build context-dependent lexicons, we ask workers to find the ambiguous phrases and their disambiguating contexts. This task requires cognitive engagement, as not all phrases are ambiguous and not all contexts disambiguate. Workers can thus quickly lose interest, giving sloppy answers. Is is unclear whether extrinsic motivators alone are enough to keep workers motivated. In previous experiments, we obtained poor results for a simple review polarity annotation task where we incentivized colleagues with prizes. Therefore, to ensure participants stay motivated, we make the task fun by designing it as a game.

4.1 Game Design

In our task, workers see fragments of text, from which they construct answers that contain a phrase, a context, and a polarity. To build an answer, workers select phrases and contexts from these texts, then annotate the resulting word constructs with their polarities. For instance, from the text *I like this small camera, it fits in every pocket*, workers could construct the answer $(small, camera, pos)$.

To increase motivation, we rely on payment and enjoyment. Through payment, we touch on extrinsic motivation. Workers receive monetary rewards once they finish the task. Through enjoyment, we target intrinsic motivation. We entertain workers with point rewards that reflect the value of their answers. We also introduce riddles that workers gradually solve by submitting answers. By combining payment and enjoyment, we create a game played for money.

To ensure quality, we use a scoring mechanism that encourages workers to submit useful answers. We judge usefulness with two criteria: whether answers have common sense and whether they provide new knowledge. We consider an answer commonsensical if it is consistent with the contextual knowledge acquired in the game up to that point, meaning that it agrees with the common opinion of many workers. We consider an answer novel if it greatly impacts the contextual knowledge in the game, meaning that it is submitted early on and that it contains an ambiguous phrase along with a disambiguating context. We compute point rewards that are the sum of an agreement score and a novelty score (Section 4.3).

4.2 Gameplay

Guesstiment is a round-based game (Figure 1). In a round, a worker sees a text fragment and constructs an answer in three steps:

- *Step 1: phrase selection*, in which she selects a phrase from the text.
- *Step 2: context selection*, in which she optionally selects a context.
- *Step 3: polarity annotation*, in which she annotates the phrase and context pair with a polarity.

The worker then submits her answer, receives a point reward, and starts a new round. Through this activity, she solves riddles that point to animals with interesting behavior, like: *These animals taste with their feel* with *butterfly* as solution. The player's answers unlock hints for these riddles, shown by gradually uncovering a picture that portrays the solution (Figure 1).

4.3 Scoring Mechanism

We reward answers using a scoring model that contains beliefs about the polarities of phrase and context pairs. To each combination of a phrase phr and context con (possibly empty: $con = nil$), this model associates a Beta distribution

[2] www.swagbucks.com

Figure 1: The interface of Guesstiment

(Gupta 2011). From this distribution we estimate the probabilities that, in the context con, the phrase phr has positive and negative polarities, respectively: Pr$(pos \mid phr, con)$ and Pr$(neg \mid phr, con)$. For the word pairs that co-occur in the sentences of our review corpus $Train^{vac\text{-}cam}$ (Section 7.1), we initialize the corresponding Beta distributions using corpus statistics and several context-independent sentiment lexicons. We use the difference between a word pairs' frequencies in positive and negative documents, respectively. When one of the words in the pair appears in a sentiment lexicon, we complement the corpus frequencies with that word's polarity score. We incorporate incoming answers by modifying these distributions through a Bayesian update process. Therefore, the probabilities Pr$(pos \mid phr, con)$ and Pr$(neg \mid phr, con)$ assimilate the fractions of positive and negative answers, respectively: (phr, con, pos) and (phr, con, neg).

For an answer (phr, con, pol), we compute an agreement score $ag \in [0, ag_{max}]$ that reflects whether it is commonsensical. We set ag highest if the answer agrees with the model early in the game, because by doing so it considerably improves the model's confidence. We set ag lowest if the answer contradicts the model early in the game, because this way it greatly damages the model's confidence. Finally, we assign a medium value to ag when the answer comes late in the game, because then it has a smaller impact on the model's confidence. We use the entropy over Pr$(pol \mid phr, con)$ to quantify the model's uncertainty in the polarity of the phrase and context pair. The answer decreases entropy if it agrees with the model and vice versa. Moreover, the answer produces bigger changes in entropy early in the game. We thus obtain ag by linearly mapping the updates in entropy to $[0, ag_{max}]$.

For an answer (phr, con, pol), we also compute a context novelty score $cn \in [0, cn_{max}]$ that reflects whether the answer contains an ambiguous phrase and a well-defined context. As an indicator for the phrases's ambiguity, we use the model's uncertainty in the phrase's out of context polarity. If $con = nil$, we set $cn := 0$. If, however, $con \neq nil$, we use the entropy over Pr$(pos \mid phr, nil)$ to quantify the ambiguity of phr out of context. We obtain cn by linearly mapping this entropy to $[0, cn_{max}]$.

We reward the answer with a total score $su := ag + cn$. Because we do not want to encourage only unique answers, we give a bigger importance to agreement by setting $ag_{max} > cn_{max}$. We use $ag_{max} := 40$ and $cn_{max} := 10$.

5 Quality Assurance

We ensure answer quality in three steps: with a tutorial before the game, through the scoring mechanism during the game, by filtering lazy workers and bad answers after the game.

The tutorial teaches the rules of the game through text instructions and interactive quizzes. We start by explaining the concepts of phrase, context, and polarity. Then, workers solve quizzes in which they learn to construct answers from simple text snippets. Finally, we explain the round-based nature of the game, the scoring, and the riddles.

5.1 Worker Filtering

We filter out the lazy workers by measuring their performance using a gold-standard lexicon that we obtain by intersecting several context-independent lexicons. For each worker, we establish: the number $gold$ of answers that contain a phrase from our gold standard lexicon; and the number $gold^{corr}$ of answers that contain a phrase from the gold standard lexicon along with its correct polarity. We consider the worker's performance is good if $gold \geq 5$ and if $\frac{gold^{corr}}{gold} \geq 0.8$.

We initially used the method above, but noticed the quality of the remaining answers was still quite low. We thus assess performance with an alternate method. We define several gold-standard game rounds that we interleave with the regular rounds. The gold rounds show simple text fragments with only a few acceptable, gold standard answers. For each worker, we establish: the number $gold$ of answers she submitted in gold rounds; and the number $gold^{corr}$ of those answers that belong to our set of acceptable, gold standard answers. We then judge the worker using the same criteria as for the previous method.

5.2 Answer Filtering

After we eliminate the bad workers, we have the remaining answers: $A = \{(phr, con, pol) \mid phr \in P, con \in C, pol \in \{pos, neg\}\}$. From these, we derive the aggregate answers: $AA = \{(phr, con, wkr^{pos}, wkr^{neg}) \mid phr \in P, con \in C, wkr^{pos}, wkr^{neg} \in \mathbb{N}\}$. An aggregate answer $(phr, con, wkr^{pos}, wkr^{neg})$ contains a phrase and context pair, along with the number of workers that included them in positive and negative answers respectively. We use these aggregate answers to obtain a context-dependent lexicon: $CL^{game} = \{(phr, con, pol)\}$. For each $(phr, con, wkr^{pos}, wkr^{neg}) \in AA$, we add (phr, con, pol) to CL^{game}: we set $pol := pos$ if $wkr^{pos} > wkr^{neg}$, and $pol := neg$ if $wkr^{pos} < wkr^{neg}$.

From CL^{game}, we remove the elements that damage sentiment classification performance. We use a classification method that combines a context-independent lexicon L with a context-dependent one CL (Section 6). When CL is empty, the method relies on the lexicon L alone. When CL is well-defined, the method uses it to enhance L. We apply the method in these two modes: we first use a lexicon L^{bl} alone (Section 7.2), then we combine L^{bl} with CL^{game}. We apply the classification method on the corpus $Train^{vac\text{-}cam}$. For each document, three scenarios can happen: CL^{game} fixes an error of L^{bl}, CL^{game} damages a correct classification of L^{bl}, or CL^{game} neither helps nor harms the output of L^{bl}. Thus, for each element in CL^{game}, we maintain improvement and error counts that we increment when the first or second scenarios occur, respectively. We use these counts to remove the elements that damage L^{bl}.

We prune the bad elements in four iterations. In each iteration, we classify documents and compute the improvement and error counts. We then choose a criterion for pruning CL^{game}. In the first two iterations, we eliminate the elements with high error counts. We first remove the elements that act as stop words and produce errors in a considerable fraction of the corpus. These elements pollute the error counts of the other lexicon items they co-occur with. We use $error \geq 0.00007 \times |Train^{vac\text{-}cam}|$. We then remove all elements with error counts above a fixed threshold. We use $error \geq 20$. In the last two iterations, we remove the elements that harm more than they improve.

6 Context Impact

To classify documents, we combine a context-independent sentiment lexicon $L = \{(phr, pol^L)\}$ with a context-dependent one $CL = \{(phr, con, pol^{CL})\}$, thus evaluating the impact of context. For each document, we compute a sentiment score scr. We label the document as positive if $scr > 0$ and as negative if $scr < 0$. If $scr = 0$, we do not assign a label. To compute the score, we split the text into sentences. For each sentence, we identify the phrases that are in L or in CL. For every phrase phr that we find:

- We scan a window of seven words centered around it to identify the contexts $con \neq nil$ for which we have $(phr, con, pol^{CL}) \in CL$.
- For every $(phr, con, pol^{CL}) \in CL$ that we find, we update the score with one unit based on the sign of pol^{CL}.
- If we cannot find any $(phr, con, pol^{CL}) \in CL$, we determine if $(phr, pol^L) \in L$ and if $(phr, nil, pol^{CL}) \in CL$:
  - If $(phr, pol^L) \in L$, we update the score with one unit using the context-independent polarity.
  - If $(phr, pol^L) \notin L$, but $(phr, nil, pol^{CL}) \in CL$, we update the score with one unit using the context-dependent polarity

7 Experiments and Results

7.1 Task Setup

To create the game rounds, we used Amazon[3] product reviews for four categories of vacuum cleaners and digital cameras, respectively. We used the reviews' numeric ratings as a gold standard for their sentiment: the ones with high ratings as positive, and the ones with low ratings as negative. We ensured each product category had equal numbers of positive and negative reviews, and randomly split the reviews from each category into train and test data, using a ratio of roughly 2:1. We used the train data $Train^{vac\text{-}cam}$ of 56,000 reviews to obtain two classifiers: a sentiment lexicon and a support vector machine. We applied the two models on the test data $Test^{vac\text{-}cam}$, and identified roughly 2,000 reviews $Test^{vac\text{-}cam}_{game}$ misclassified by both. We used $Test^{vac\text{-}cam}_{game}$ to define text fragments for the game. We then removed $Test^{vac\text{-}cam}_{game}$ from $Test^{vac\text{-}cam}$ to obtain a new test set $Test^{vac\text{-}cam}_{\overline{game}}$ of 31,700 texts.

We deployed the game on Amazon Mechanical Turk. We launched several HITs, mainly using a base payment of \$0.25, approved with the first 100 points earned, and bonuses established based on game activity: \$0.05 for every additional 100 points earned, and \$0.05 for every 500 points milestone reached. In total, 980 workers played the game. In a post-game survey, 88% of them indicated they liked the game. We kept the activity of 640 workers, amounting to 59,400 answers from a total of 78,000. We aggregated these answers to a lexicon CL^{game} of 36,300 items, which we pruned down to 16,600 elements.

7.2 Context Evaluation

We first assessed the performance of our context-dependent lexicon on the source vacuum and camera domains, using the corpus $Test^{vac\text{-}cam}_{\overline{game}}$. As baselines, we took several context-independent lexicons and a machine learning model. We used the lexicons: General Inquirer L^{gi} (Stone et al.

[3] www.amazon.com

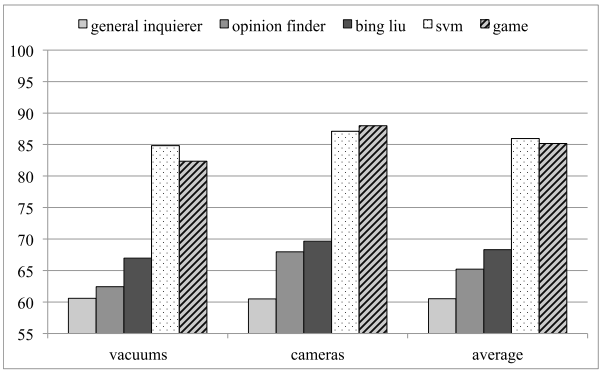

Figure 2: Performance on source domains

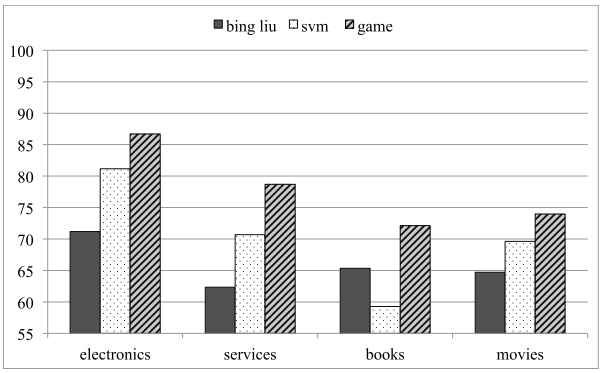

Figure 3: Cross-domain performance

1966), OpinionFinder L^{of} (Wilson et al. 2005), and the lexicon of Bing Liu L^{bl}. We obtained the statistical model L^{svm} by training a support vector machine on a subset of 2,000 reviews from $Train^{vac\text{-}cam}$, thus comparable in size with the corpus in the game (we used a linear kernel and 1,000 unigram presence features). Finally, we evaluated our contextual knowledge CL^{game} by combining it with L^{bl}, the best among the context-independent lexicons. We recorded the average performance on vacuums, cameras, and overall.

Our lexicon CL^{game} gave an accuracy of 85.17% (Figure 2). The increase in accuracy relative to L^{gi}, L^{of}, and L^{bl} was of 24.63%$^{+}$[4], 19.96%$^{+}$, and 16.85%$^{+}$, respectively. Moreover, the difference in performance relative to L^{svm} was of -0.78%$^{-}$[5]. Our contextual knowledge thus dramatically improved the standard lexicons and performed as well as the statistical model trained on these specific domains.

We then assessed the cross-domain performance on four target domains. We used corpora of: 1,200 Epinions[6] electronics reviews (cellular phones and blenders), 12,600 Epinions services reviews (restaurants and hotels), 3,600 Epinions book reviews, and 11,900 Amazon movie reviews. As baselines, we took L^{bl} and L^{svm}. We evaluated our contextual knowledge CL^{game} by combining it with L^{bl}.

On the electronics, services, book, and movie domains, our lexicon had accuracies of 86.68%, 78.71%, 72.13%, and 73.96%, respectively (Figure 3). The increase in accuracy relative L^{bl} was of 15.49%$^{+}$, 16.38%$^{+}$, 6.78%$^{+}$, and 9.24%$^{+}$, respectively. Moreover, the accuracy boost with respect to L^{svm} was of 5.53%*[7], 8.03%$^{+}$, 12.85%$^{+}$ and 4.34%$^{+}$, respectively. Our contextual knowledge thus greatly surpassed both the best standard lexicon and the machine learning model we considered.

We also compared our performance on the hotel reviews with that of (Lu et al. 2011), who automatically generated a context-dependent lexicon for feature-level sentiment classification on this domain. CL^{game} gave a precision of 73.98% and a recall of 100%. Relative to (Lu et al. 2011), who reported a precision and recall of 72.83% and 77.56% respectively, CL^{game} increased recall by 22.44%. Our contextual knowledge thus substantially outperformed an automatically generated context-dependent lexicon in terms of recall.

8 Conclusions

For many Artificial Intelligence tasks, the major difficulty is that they require large amounts of commonsense knowledge. Since people share this knowledge, it could be obtained through human computation.

We are the first ones to consider the acquisition of contextual knowledge for sentiment classification. A big challenge was keeping workers interested in a complex task that required cognitive engagement. To overcome this, we made the task enjoyable by designing it as a game. Another obstacle was ensuring answer quality when workers were not simultaneously present and thus agreement with peers was not feasible. Instead, we simulated agreement with other workers by rewarding novel commonsensical answers using a model derived from the workers' activity. We thus showed that human computation makes knowledge acquisition feasible, even when complex actions are required from workers.

From the workers' activity, we obtained powerful contextual knowledge. On the two source domains for which we acquired the knowledge, our context-dependent lexicon greatly improved several established sentiment lexicons. On four target domains, we also substantially surpassed a machine learning model and an automatically generated context-dependent lexicon. We thus proved that using contextual knowledge obtained through human computation dramatically improves sentiment classification.

In future work, we plan to also improve corpus-based methods by extending their feature set from individual words only to also include the contexts that we acquired through human computation.

References

Al-Subaihin, A.; Al-Khalifa, H.; and Al-Salman, A. 2011. A proposed sentiment analysis tool for modern arabic using human-based computing. In *Proceedings of the 13th International Conference on Information Integration and Web-Based Applications and Services*, 543–546.

[4] $+$ statistically significant at the 99% confidence level

[5] $-$ not statistically significant on six out of eight categories

[6] www.epinions.com

[7] $*$ statistically significant at the 95% confidence level

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*, 120–128.

Brew, A.; Greene, D.; and Cunningham, P. 2010. Using crowdsourcing and active learning to track sentiment in online media. In *Proceedings of the 19th European Conference on Artificial Intelligence*, 145–150.

Esuli, A., and Sebastiani, F. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation*, 417–422.

Gupta, A. K. 2011. Beta distribution. In Lovric, M., ed., *International Encyclopedia of Statistical Science*. Springer Berlin Heidelberg. 144–145.

Hatzivassiloglou, V., and McKeown, K. R. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 174–181.

Hong, Y.; Kwak, H.; Baek, Y.; and Moon, S. 2013. Tower of Babel: A crowdsourcing game building sentiment lexicons for resource-scarce languages. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, 549–556.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168–177.

Ipeirotis, P. G.; Provost, F.; and Wang, J. 2010. Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 64–67.

Kennedy, A., and Inkpen, D. 2005. Sentiment classification of movie and product reviews using contextual valence shifters. *Computational Intelligence* 110–125.

Lau, R. Y. K.; Lai, C. C. L.; Ma, J.; and Li, Y. 2009. Automatic domain ontology extraction for context-sensitive opinion mining. In *Proceedings of the 30th International Conference on Information Systems*, 35–53.

Lu, Y.; Castellanos, M.; Dayal, U.; and Zhai, C. 2011. Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, 347–356.

Musat, C. C., and Faltings, B. 2013. A novel human computation game for critique aggregation. In *AAAI (Late-Breaking Developments)*.

Musat, C. C.; Ghasemi, A.; and Faltings, B. 2012. Sentiment analysis using a novel human computation game. In *Proceedings of the 3rd Workshop on the People's Web Meets Natural Language Processing*, 1–9.

Pan, S. J.; Ni, X.; Sun, J.-T.; Yang, Q.; and Chen, Z. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, 751–760.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*, 79–86.

Sintsova, V.; Musat, C. C.; and Pu, P. 2013. Fine-grained emotion recognition in olympic tweets based on human computation. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 12–20.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*, 1631–1642.

Stone, P. J.; Dunphy, D. C.; Smith, M. S.; and Ogilvie, D. M. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

Turney, P. D. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 417–424.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 319–326.

Wilson, T.; Hoffmann, P.; Somasundaran, S.; Kessler, J.; Wiebe, J.; Choi, Y.; Cardie, C.; Riloff, E.; and Patwardhan, S. 2005. OpinionFinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, 34–35.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 347–354.