Constructing Context-Aware Sentiment Lexicons with an Asynchronous Game with a Purpose

Marina Boia, Claudiu Cristian Musat, and Boi Faltings

École Polytechnique Fédérale de Lausanne
Switzerland
firstname.lastname@epfl.ch

Abstract. One of the reasons sentiment lexicons do not reach human-level performance is that they lack the contexts that define the polarities of words. While obtaining this knowledge through machine learning would require huge amounts of data, context is commonsense knowledge for people, so human computation is a better choice. We identify context using a game with a purpose that increases the workers' engagement in this complex task. With the contextual knowledge we obtain from only a small set of answers, we already halve the sentiment lexicons' performance gap relative to human performance.

1 Introduction

Sentiment analysis identifies expressions of subjectivity in texts, such as sentiments or emotional states. We consider the sentiment classification task, which determines whether the sentiments expressed in a text are positive or negative. This task requires commonsense knowledge about the polarities of sentiment words.

The relative ease of construction led early researchers in the field toward corpus-based sentiment classification [1–3]. These methods aggregate statistical, syntactic, and semantic relations between words. A significant downside is that the classifiers that result are efficient only on narrow domains. This may be the reason why the competing, lexicon-based approach is currently the backbone of sentiment classification. Several sentiment lexicons [4–6] have been available for a significant period of time. However, multiple lexicons continue to appear [7, 8], showing that a satisfying solution has not yet been found.

The most successful methods perform syntactic preprocessing to extract relevant words, and then consider the resulting set of independent words as features of the text. Sentiment classification is performed on these features, by adding word polarity scores compiled in sentiment lexicons or learned with statistical methods. These models obtain from 60% to 80% accuracy [2, 1]. Better results can sometimes be achieved by training domain-specific classifiers, but only at the expense of narrow coverage. This performance is lower than that of people, who can extract sentiment with 80% to 90% agreement [9], depending on the domain of the texts.

A reason why these classifiers cannot reach human-level performance is that the words' polarities are influenced by context: a *small hotel room* is negative, while a *small digital camera* is positive. By representing texts as independent words, context

A. Gelbukh (Ed.): CICLing 2014, Part II, LNCS 8404, pp. 32–44, 2014.

is ignored. In narrow domains, words mostly occur in a single context, thus high accuracy can be achieved. For broad domains, it is necessary to enrich the feature set with contexts, by including word combinations. However, the complexity of the resulting models would explode, and it would no longer be feasible to acquire them from data. Nevertheless, the polarity of most words has only a few exceptions, so the size of these models could be manageable if these exceptions are identified. This is very difficult to do by statistical methods, but easy for people. We thus investigate how we can obtain contextual knowledge through human computation.

Most human computation tasks consist of simple tasks, such as object labeling [10]. We ask workers to find the contexts that influence the polarities of words. Because our task is more difficult, workers can become demotivated and give sloppy answers. Moreover, there is considerable freedom in indicating context, thus quality assurance by agreement with peers is unfeasible.

To increase engagement, we develop a novel task design that combines the entertaining aspect of games with the large worker pool available on a paid crowdsourcing platform. Workers play a game in rounds, where in each round they increase their score by submitting answers that contain sentiment words, contexts, and polarities. At the end, workers receive a payment proportional to their score.

To solve the quality assurance issue, we develop a scoring mechanism that elicits useful answers. The scoring rewards answers using a model that we derive from existing sentiment knowledge and the workers' input. We thus create the illusion of a game played with others, which ensures quality and also makes the task fun.

To further boost quality, we improve task understanding with tutorials. We develop a static tutorial with textual information, as well as an interactive one with quizzes. We show that the latter greatly increases worker performance.

The output of the game is a context-dependent sentiment lexicon. Our contextual knowledge refines the polarities of sentiment words with contexts, so it naturally complements the standard, context-independent lexicons. We thus assess how our lexicon improves several standard ones. Even with a small number of answers, we considerably improve the performance of two context-independent lexicons. For the lexicon that performs best on our corpus, we increase the accuracy from 68% to almost 75%, halving its gap relative to human performance.

This paper thus makes several contributions:

- To the best of our knowledge, we are the first that use human computation to acquire contextual knowledge for sentiment classification.
- We obtain a context-dependent lexicon that significantly improves sentiment classification accuracy.
- We develop a game that increases the workers' motivation in a complex task.
- We create the illusion of synchronous worker interaction using a model that rewards useful answers, thus ensuring quality.
- We analyze the effect of tutorials on worker performance.

Section 2 overviews the related work, Section 3 defines the context, and Section 4 presents our game. Subsequently, Section 5 discusses quality assurance, Section 6 presents our experiments and results, and Section 7 draws conclusions.

2 Related Work

There are two main directions in sentiment analysis. Lexicon-based methods use sentiment lexicons, which are out-of-the-box lists of sentiment words paired with polarities. Several lexicons [4–6, 11] have been available for a significant period of time. However, multiple ones continue to appear. [7] combined ANEW [12] and WordNet [13] to create a superior lexicon. [8] developed a domain-specific lexicon using a random walk algorithm, while [14] extended a lexicon with additional knowledge, such as parts of speech and word senses. The continual emergence of similar methods for a known problem shows a satisfactory solution has still not been found. A key problem of these lexicons is that they do not consider the influence of context on the polarities of words.

Corpus-based methods learn sentiment words and their polarities from text corpora. This is typically done with machine learning algorithms applied to annotated datasets [1], or by aggregating syntactic and semantic relations between words [2, 3]. These methods do not explicitly take context into account. However, when the texts target narrow enough domains, the resources learned become domain-specific, with sentiment words that occur in only a single context. This is why the corpus-based methods typically perform much better than their lexicon-based counterparts, but only in the domain of the corpus.

Context is important in sentiment analysis: [15] noted that words may have different polarities, depending on the features they describe. [16] learned a taxonomic organization of the entities, features, and sentiment words in a domain. They used this taxonomy to represent the contextual variations of sentiment words. Similarly, [17] generated context-dependent lexicons of sentiment words paired with features. However, expanding these approaches beyond a limited set of features is hard without a priori domain knowledge. We believe context can be more effectively identified through human computation.

Previous human computation applications to sentiment analysis have been limited to eliciting simple polarity annotations. Training data have been obtained through tasks that required polarity or emotion annotation of texts [18]. Several lexicons have been produced through polarity or emotion annotation of some predefined vocabularies [19]. The tasks in [20, 21] were more complex and required humans to select both sentiment words and polarities. Most of these annotation tasks had a simple design that motivated workers with payment, whereas others were designed as games [19–21]. We no longer focus on basic annotations and selections. We require workers to complete a more complex task, by characterizing the contexts in which sentiment words have certain polarities.

The outcome of a task is highly sensitive to the workers' motivation. One way to inspire motivation is through payment, as in online labor markets, such as Amazon Mechanical Turk [1]. Another way is through enjoyment - tasks can be packaged as games with a purpose, when players submit answers and are rewarded with points, reputation, badges etc. 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

[1] www.mturk.com

crowdsourcing platforms. Instead, we combine enjoyment and payment, and obtain a game played for money, like poker or like the games on Swagbucks [2].

A further consideration is quality assurance, which can be done before, during, or after workers complete the task. During the task, output agreement setups place workers in teams and require them to agree on their answers [10]. Posterior measures are applied after the work is done, by filtering or aggregating data to remove answers that are irrelevant or malicious [22]. Preemptive measures are applied before collecting answers, by making workers aware of the desired level of performance. Tutorials aim to induce a basic understanding of the task and to explain what kind of answers are required [23]. We employ all three types of quality control.

3 Context Definition

We use human computation to obtain contextual knowledge for sentiment classification. We structure this 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 the sentiment conveyed by a word construct.

Depending on whether a context is needed to define its polarity, a phrase can be unambiguous or ambiguous. The unambiguous phrases have the same polarity in every context: the word *amazing* is always positive. The ambiguous phrases have different polarities depending on the context: the word *high* is positive in the context of *salary*, but negative in the context of *debt*.

The most common phrases are typically compiled in sentiment lexicons. Given a phrase vocabulary P, these lexicons pair phrases with their default polarities: $L = \{(phr, pol) | phr \in P, pol \in \{pos, neg\}\}$. This context-independent representation either includes ambiguous phrases with polarities that do not make sense, or excludes them altogether. Instead, we consider a more articulate representation that is sensitive to context. Given an additional context vocabulary C, context-dependent lexicons include the contexts that disambiguate the polarities of ambiguous phrases: $CL = \{(phr, con, pol) | phr \in P, con \in C, pol \in \{pos, neg\}\}$.

4 Human Computation Task

To build context-dependent lexicons, we ask workers to find the contexts that disambiguate the ambiguous phrases. Our task thus requires cognitive engagement, so workers might quickly lose interest in it. It is unclear if extrinsic motivators alone can keep workers interested. In previous experiments, we obtained poor results for a simple polarity annotation task where we incentivized colleagues with prizes. To motivate workers, we make our task fun by designing it as a game.

[2] `www.swagbucks.com`

4.1 Task Design

In our task, workers see text fragments from which they submit answers that contain a phrase, a context, and a polarity. Workers construct an answer by selecting a phrase and a context from a text fragment, and annotating the resulting word combination with a polarity. For instance, from the text *I don't like this camera. It has tiny buttons*, workers could submit the answer $(tiny, buttons, neg)$.

We motivate workers through enjoyment and payment. Enjoyment impacts intrinsic motivation. We entertain the workers with point rewards that reflect the usefulness of their answers. Payment targets extrinsic motivation. Once they finish playing, workers receive monetary rewards proportional to their scores. By combining enjoyment and payment, we obtain a game played for money.

We elicit useful answers with an intelligent scoring mechanism. An answer is useful if it has common sense and is novel. We consider an answer commonsensical if it agrees with the contextual knowledge acquired in the game, which means that it complies with the opinion of many workers. We consider an answer novel if it greatly impacts the contextual knowledge, which means that it is given early in the game or that it contains ambiguous phrases and disambiguating contexts. We reward answers with scores that are the sum of an agreement score and a novelty score (Section 4.3).

4.2 Gameplay

Guesstiment (Figure 1) is a round-based game. In each round, a worker sees a text fragment, from which she submits an answer. She then receives score and payment rewards, and starts a new round. The worker constructs an answer in three steps:

- *Step 1: phrase selection*, when she selects a phrase from the text fragment.
- *Step 2: context selection*, when she optionally selects a context.
- *Step 3: polarity annotation*, when she annotates the resulting phrase and context combination with a polarity.

The worker has the option to skip to the next round without submitting an answer, if the text fragment does not contain one.

4.3 Scoring Mechanism

To compute score rewards, we use a model that contains our beliefs on the polarities of phrase and context combinations. To a phrase phr and a context con (possibly empty: $con = nil$), we associate a Beta distribution [24] from which we estimate the probabilities that, in the context con, the phrase phr has positive and negative polarities respectively: Pr$(pos|phr, con)$ and Pr$(neg|phr, con)$. We assign a Beta distribution to every word and word combination that appear in the sentences of our corpus $Train$ (Section 6.1). We initialize these distributions using word frequencies in positive and negative documents, as well as word polarities from context-independent lexicons. We incorporate incoming answers by modifying the distributions through a Bayesian update process. Therefore, the probabilities Pr$(pos|phr, con)$ and Pr$(neg|phr, con)$ assimilate the fractions of positive and negative answers respectively: (phr, con, pos) and (phr, con, neg).

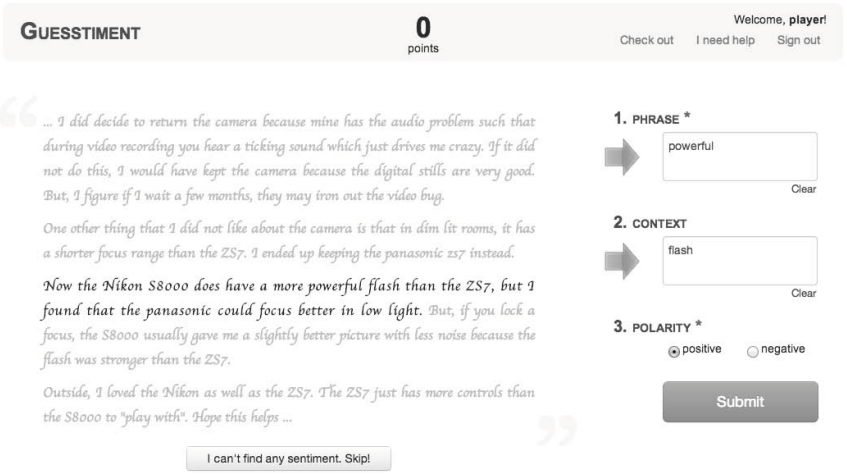

Fig. 1. The game interface

For an answer (phr, con, pol), we compute an agreement score $ag \in [0, ag_{max}]$ that reflects if it is commonsensical. We reward the answer with a high score if it agrees with the scoring model early in the game, because it substantially improves the model's confidence. We compute a low score if the answer disagrees with the model early in the game, because it substantially harms the model's confidence. We give a medium score if the answer comes late in the game, because it has less impact on the model. We use the entropy over $\Pr(pol|phr, con)$ to measure the model's uncertainty in the polarity of the phrase and context combination. The answer either decreases or increases entropy, depending on whether it agrees or disagrees with the model. Moreover, the change in entropy is bigger when the answer is submitted early in the game. We thus compute ag by linearly mapping the entropy update to $[0, ag_{max}]$.

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

We reward an answer with a score that is the sum of the agreement score and the context novelty score. Because we do not want to gather only unique answers, we give a bigger importance to agreement. We control this by making $ag_{max} > cn_{max}$, and we use $ag_{max} := 40$ and $cn_{max} := 10$.

5 Quality Assurance

We control quality before, during, and after the workers' activity. The scoring mechanism ensures answer quality during the game. Before the game, we introduce tutorials. After the game, we filter out the bad answers, and we aggregate the remaining ones.

5.1 Tutorials

With tutorials, we ensure workers understand the task. We create a static tutorial with text instructions only. We divide the instructions into sections that explain: the concepts of phrase, context, and polarity; how to construct an answer in a game round; and how the scoring mechanism works. We illustrate the scoring with examples of simple game rounds, along with potential answers and score updates.

We used the static tutorial when we first launched the game, but noticed that the quality of answers was not high. Therefore, we created another tutorial that teaches workers with text instructions and interactive quizzes. The first quiz presents several word constructs and asks which words are phrases and which words are contexts. This quiz also asks which phrases are ambiguous and which ones are unambiguous. The second quiz emulates the game interface (Figure 1) and asks workers to construct a simple answer. The final quizzes also emulate the game interface and ask workers to give specific types of answers: with an ambiguous or unambiguous phrase, with or without a context.

5.2 Answer Filtering

From the answers $A = \{(phr, con, pol) | phr \in P, con \in C, pol \in \{pos, neg\}\}$, we remove the ones that are not commonsensical. As filtering takes place only when all answers have been gathered, we can use a more precise evaluation than for the scoring mechanism (Section 4.3), based on actual agreement between workers:

Heuristic *wa*. An answer (phr, con, pol) is commonsensical if its phrase phr and context con have been included in answers by at least two workers. These answers form the *worker agreement* set: $wa(A)$.

We initially used the worker agreement heuristic, but noticed that a lot of answers were removed because workers could not agree on the same interpretation of context. Therefore, we introduced other heuristics in which we check whether answers contain proper phrases and contexts (Section 3), as workers are more likely to agree on well-constructed answers. We thus consider an answer (phr, con, pol) commonsensical if:

Heuristic *slr*. its phrase phr is a word from context-independent lexicons, and its context con is a noun or a verb. These answers form the *sentiment lexicon restrictive* set: $slr(A)$.

Heuristic *slp*. its phrase phr is at most two words with at least one from context-independent lexicons, and its context con is at most two words with at least one a noun or a verb. These answers form the *sentiment lexicon permissive* set: $slp(A)$.

Heuristic *psr*. its phrase phr is an adjective or an adverb, and its context con is a noun or a verb. These answers form the *part of speech restrictive* set: $psr(A)$.

Heuristic *psp*. its phrase phr is at most two words with at least one an adjective or an adverb, and its context con is at most two words with at least one a noun or a verb. These answers form the *part of speech permissive* set: $psp(A)$.

We combine these heuristics to obtain two higher-order ones, which consider an answer commonsensical if:

Heuristic *res*. it complies with the worker agreement heuristic or with the two restrictive ones. These answers form the *restrictive* set: $res(A)$.

Heuristic *per*. it complies with the worker agreement heuristic or with the two permissive ones. These answers form the *permissive* set: $per(A)$.

5.3 Answer Aggregation

After we filter out the bad answers, we aggregate the remaining ones to form the set: $AA = \{(phr, con, wkr^{pos}, wkr^{neg}) | phr \in P, con \in C, wkr^{pos}, wkr^{neg} \in \mathbb{N}\}$. An aggregate answer $(phr, con, wkr^{pos}, wkr^{neg})$ contains a phrase and a context that appear together in answers, along with the number of workers that included them in positive and negative answers respectively. We use the aggregate answers to obtain a context-dependent lexicon: $CL = \{(phr, con, pol)\}$. For each aggregate answer $(phr, con, wkr^{pos}, wkr^{neg}) \in AA$, we add (phr, con, pos) to CL if $wkr^{pos} > wkr^{neg}$. Conversely, we add (phr, con, neg) to CL if $wkr^{pos} < wkr^{neg}$.

6 Experiments and Results

6.1 Dataset

To construct the game rounds, we used Amazon [3] product reviews of four categories of vacuum cleaners and digital cameras respectively. We used the reviews' numerical ratings as a gold standard for their polarities, and we ensured each product category had equal class representation. We randomly split the positive and negative reviews from each product category into training and test data, in a ratio of $2:1$. We used the training data $Train$ to obtain two sentiment classifiers: a sentiment lexicon and a support vector machine. We applied these classifiers on the test data $Test$, and identified the reviews $Test_{game}$ that both misclassified. We constructed the game rounds using 2000 sentences that we extracted from $Test_{game}$. To test our context-dependent lexicons, we removed $Test_{game}$ from $Test$ and obtained a new test set $Test_{\overline{game}}$.

6.2 Worker Participation

We deployed the game on Amazon Mechanical Turk. We ran a first HIT, with the static tutorial, for a week. We ran a second HIT, with the quiz tutorial, for another week. Both were visible to only United States residents and had a base payment of $0.1, with

[3] `www.amazon.com`

bonuses of $0.1 for every 100 and 500 points milestones reached. In the first HIT, the game was accessed by 71 workers, 75% of which completed the tutorial and played at least one round. In the second HIT, 279 workers accessed the game, 27% of which completed the tutorial and played at least one round.

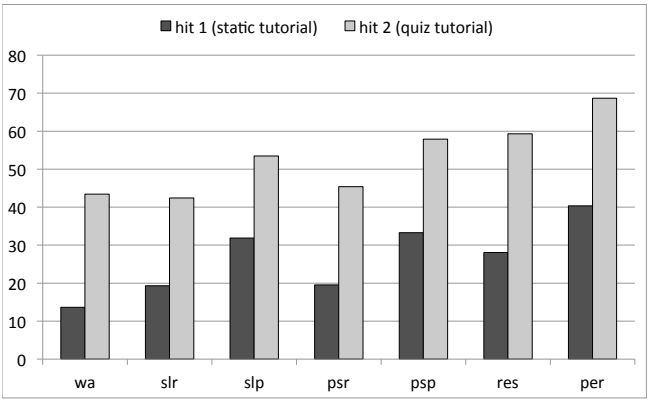

Fig. 2. The *number* of commonsensical answers per worker (according to the heuristics for common sense in Section 5.2)

On average, the workers in the first HIT played 95.51 rounds and gave 73.60 answers, while the ones in the second HIT played 105.52 rounds and gave 85.84 answers. Moreover, the workers in the first and second HITs earned 19.81 and 23.06 points per answer respectively. Two-tailed t-tests showed that these differences are significantly relevant at the 1% confidence level. The increase in performance shows that the workers that passed the quiz tutorial were more engaged, playing more and better.

6.3 Answer Usefulness Evaluation

We analyzed the frequency of useful answers. Figures 2 and 3 show that the workers in the second HIT submitted more commonsensical answers than the ones in the first HIT, according to the heuristics in Section 5.2. Two-tailed t-tests showed that these differences are significantly relevant at the 1% confidence level. Moreover, in the second HIT, workers submitted answers of which, on average, 4% paired phrases from context-independent lexicons with contexts that inverted their polarities; and 42% paired new phrases with well-defined contexts. Therefore, the workers who completed the quiz tutorial were more consistent about submitting answers with common sense, and they also identified new knowledge.

6.4 Context-Dependent Lexicon Evaluation

Context-Aware Classification. We classify documents by complementing a context-independent lexicon $L = \{(phr, pol^L)\}$ with a context-dependent one $CL = \{(phr\ ,$

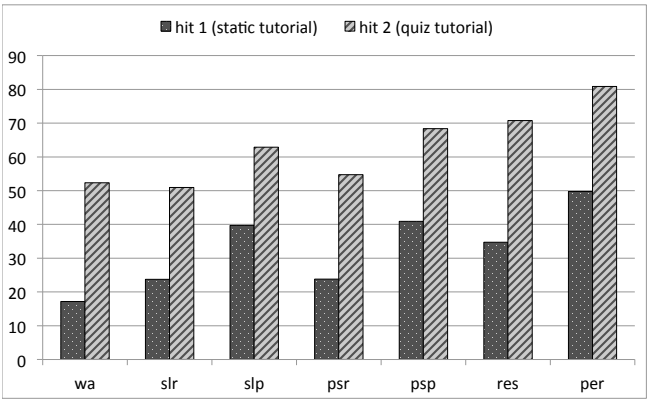

Fig. 3. The *percentage* of commonsensical answers per worker (according to the heuristics for common sense in Section 5.2)

$con, pol^{CL})\}$, thus assessing the impact of context. For each document d, we compute a sentiment score $\sigma(d)$, then we obtain a sentiment label by taking the sign of $\sigma(d)$. To compute the score $\sigma(d)$, we split d into sentences. For each sentence $s \in d$, we identify the phrases that are in CL or in L. For every phrase phr we find:

- We scan a window of t words centered around it to identify the contexts $con \neq nil$ for which we have $(phr, con, pol^{CL}) \in CL$. We use $t = 7$.
- For every $(phr, con, pol^{CL}) \in CL$ that we find, we update $\sigma(d)$ with one unit using the sign of pol^{CL}.
- If we cannot find any $(phr, con, pol^{CL}) \in CL$, we determine if $(phr, nil, pol^{CL}) \in CL$ and if $(phr, pol^{L}) \in L$:
  - If $(phr, pol^{L}) \in L$, we update $\sigma(d)$ with one unit using the sign of pol^{L}.
  - If $(phr, pol^{L}) \notin L$ but $(phr, nil, pol^{CL}) \in CL$, we update $\sigma(d)$ with one unit using the sign of pol^{CL}.

Context Impact. We analyzed the performance of our context-dependent lexicons. From the HITs with static and quiz tutorials, we obtained two lexicons[4]: CL^{static} and CL^{quiz}. We compared them with three context-independent lexicons: General Inquirer L^{gi} [4], OpinionFinder L^{of} [6], and the lexicon of Bing Liu L^{bl} [25]. We first used the context-independent lexicons individually, then we complemented them with our context-dependent ones. We tested on the review corpus $Test_{\overline{game}}$, averaging accuracies over all product categories.

Figure 4 shows the lexicons' performance. CL^{static} increased the accuracy of L^{gi} and L^{bl} by 6% and 3.5% respectively. Moreover, CL^{quiz} improved L^{gi} and L^{bl} by 10% and 6% respectively. Two-tailed t-tests showed that these improvements are significantly relevant at the 1% confidence level. Our contextual knowledge thus successfully complemented the context-independent lexicons.

[4] Available at `http://liawww.epfl.ch/data/lexicons.zip`

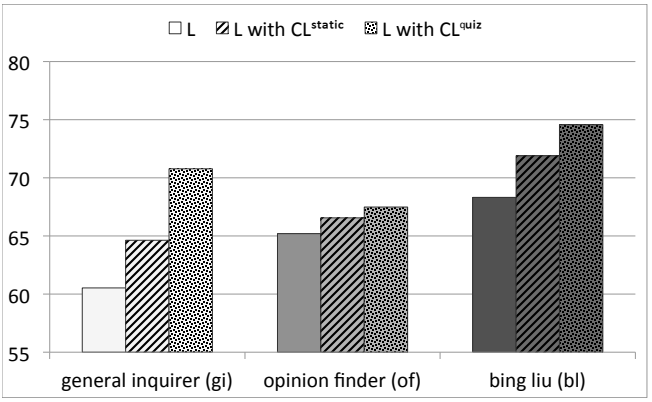

Fig. 4. Lexicon performance

Figure 5 illustrates how the lexicons' performance relates to the human performance of 83.50% [9]. For L^{gi}, CL^{quiz} reduced the gap relative to human performance by 44.52%. For L^{of}, CL^{quiz} reduced the gap by 12.51%. Finally, for L^{bl}, we decreased the gap by 41.23%. Our contextual knowledge thus halved the context-independent lexicons' deficit relative to humans.

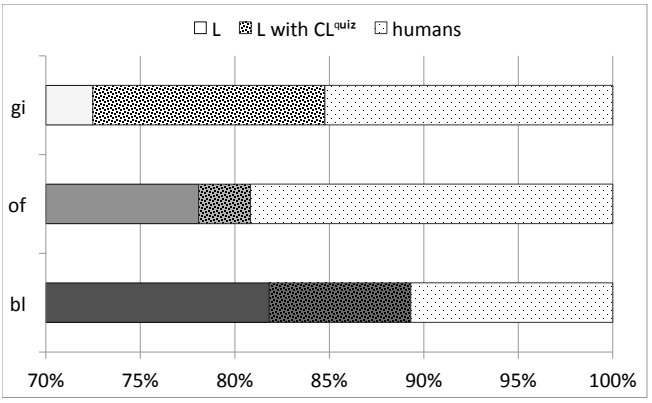

Fig. 5. Lexicon performance gap relative to human performance (83.50%)

7 Conclusions

Sentiment lexicons need commonsense knowledge about the contexts that impact the polarities of words. Because automatically extracting context would require a huge amount of data, human computation is a better alternative.

We are the first to acquire contextual knowledge with human computation. A first challenge was motivating workers to do a task that required cognitive engagement. We

solved this by designing the task as a game that we launched on Amazon Mechanical Turk. A second challenge was controlling answer quality when workers were not simultaneously present and thus we could not use worker agreement. We overcame this with a scoring mechanism that assessed answer common sense and novelty using a model derived from the workers' activity.

We improved the workers' understanding of the task with two tutorials: one that used textual instructions and one that with interactive quizzes. We showed that the latter substantially improved worker performance.

We obtained contextual knowledge of good quality. Even with a small set of answers, our context-dependent lexicons substantially improved two established context-independent ones, halving their deficit relative to humans. We believe that a significantly larger number of answers could further improve performance.

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing, pp. 79–86 (2002)
2. Turney, P.D.: 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, pp. 417–424 (2002)
3. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pp. 174–181 (1997)
4. Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie, D.M.: The General Inquirer: A Computer Approach to Content Analysis. MIT Press (1966)
5. Whissell, C.M.: The dictionary of affect in language, vol. 4, pp. 113–131. Academic Press (1989)
6. Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: OpinionFinder: A system for subjectivity analysis. In: Proceedings of HLT/EMNLP on Interactive Demonstrations, pp. 34–35 (2005)
7. Loureiro, D., Marreiros, G., Neves, J.: Sentiment analysis of news titles the role of entities and a new affective lexicon. In: Antunes, L., Pinto, H.S. (eds.) EPIA 2011. LNCS (LNAI), vol. 7026, pp. 1–14. Springer, Heidelberg (2011)
8. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: User-level sentiment analysis incorporating social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, pp. 1397–1405. ACM (2011)
9. Musat, C.C., Faltings, B.: A novel human computation game for critique aggregation. In: AAAI (2013)
10. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 319–326 (2004)
11. Esuli, A., Sebastiani, F.: SentiWordNet: A publicly available lexical resource for opinion mining. In: Proceedings of the 5th Conference on Language Resources and Evaluation, pp. 417–422 (2006)
12. Bradley, M.M., Lang, P.J.: Affective norms for english words (ANEW): Instruction manual and affective ratings. Technical Report C1 The Center for Research in Psychophysiology (1999)
13. Miller, G.A.: WordNet: a lexical database for english. Commun. ACM 38, 39–41 (1995)

14. Maks, I., Vossen, P.: A lexicon model for deep sentiment analysis and opinion mining applications. Decision Support Systems 53, 680–688 (2012)
15. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of the International Conference on Web Search and Web Data Mining, pp. 231–240 (2008)
16. Lau, R.Y.K., Lai, C.C.L., Ma, J., Li, Y.: Automatic domain ontology extraction for context-sensitive opinion mining. In: Proceedings of the 30th International Conference on Information Systems, pp. 35–53 (2009)
17. Lu, Y., Castellanos, M., Dayal, U., Zhai, C.: Automatic construction of a context-aware sentiment lexicon: An optimization approach. In: Proceedings of the 20th International Conference on World Wide Web, pp. 347–356 (2011)
18. Brew, A., Greene, D., Cunningham, P.: Using crowdsourcing and active learning to track sentiment in online media. In: Proceedings of the 19th European Conference on Artificial Intelligence, pp. 145–150 (2010)
19. Hong, Y., Kwak, H., Baek, Y., Moon, S.: 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, pp. 549–556 (2013)
20. Al-Subaihin, A., Al-Khalifa, H., Al-Salman, A.: 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, pp. 543–546 (2011)
21. Musat, C.C., Ghasemi, A., Faltings, B.: Sentiment analysis using a novel human computation game. In: Proceedings of the 3rd Workshop on the People's Web Meets Natural Language Processing, pp. 1–9 (2012)
22. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on Amazon Mechanical Turk. In: Proceedings of the ACM SIGKDD Workshop on Human Computation, pp. 64–67 (2010)
23. Sintsova, V., Musat, C.C., Pu, P.: Fine-grained emotion rrecognition in olympic tweets based on human computation. In: Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 12–20 (2013)
24. Gupta, A.K.: Beta distribution. In: Lovric, M. (ed.) International Encyclopedia of Statistical Science, pp. 144–145. Springer, Heidelberg (2011)
25. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)