Evaluating Commonsense Knowledge with a Computer Game

Juan F. Mancilla-Caceres and Eyal Amir

Computer Science Department
University of Illinois at Urbana-Champaign
201 N. Goodwin Avenue, Urbana, IL 61801, USA
{mancill1, eyal}@illinois.edu

Abstract. Collecting commonsense knowledge from freely available text can reduce the cost and effort of creating large knowledge bases. For the acquired knowledge to be useful, we must ensure that it is correct, and that it carries information about its relevance and about the context in which it can be considered commonsense. In this paper, we design, and evaluate an online game that classifies, using the input from players, text extracted from the web as either commonsense knowledge, domain-specific knowledge, or nonsense. A continuous scale is defined to classify the knowledge as nonsense or commonsense and it is later used during the evaluation of the data to identify which knowledge is reliable and which one needs further qualification. When comparing our results to other similar knowledge acquisition systems, our game performs better with respect to coverage, redundancy, and reliability of the commonsense acquired.

1 Introduction

A vast amount of information about the world is needed to create an artificial commonsensical agent [7]. This kind of knowledge, which includes facts about events, and objects, is what we call commonsense knowledge. Collecting commonsense knowledge is difficult because it is dynamic and dependent on context. This makes it impossible to generate it randomly and to verify it automatically, which implies that humans are needed to either collect the commonsense knowledge or to verify it.

The main difficulty of needing humans is that they need to be encouraged to participate. As previous studies show [2], contributors tend to produce noisy data even when they are paid for their services because they are trying to maximize the reward, whether they are producing the right data or not.

In this paper we introduce a game that takes text extracted automatically from the Web and uses input from players to verify it as commonsense. The main difficulty of this approach lies in the fact that the game needs to encourage players to supply the correct information. With the help of the players, our game classifies the knowledge as commonsense, domain-specific, or meaningless. It also reports if more information about a given fact is needed in order to classify it correctly. Correctness of the data is ensured through the design of several stages within the game, and through restricting communication among players. Also, we create a continuous scale that ranges from nonsensical (or unknown) sentences to commonly known facts. A discretization of such scale can be used to classify the sentences as commonsense or not. This scale also allows us to clearly identify which knowledge needs revision.

The focus of this paper is to present a method that provides guarantees on the correctness of the data collected through a game. Although the design of the game does not constrain the source of the input data, we are currently obtaining it from *Simple Wikipedia* [13] because some of its policies regarding the content of the articles are appropriate for commonsense extraction. For example, unverified research is not allowed in any article. Within a period of five weeks, more than 150 people played the game and more than 3,000 sentences were evaluated.

2 Related Work

The best known approaches to gather commonsense are Cyc [5] and OpenMind [11]. Cyc uses experts to input the knowledge, whereas OpenMind used volunteers. Some issues with these approaches are that Cyc requires the user to be familiar with their language, and OpenMind lacks a way to motivate volunteers to participate.

Another effort to collect common knowledge from contributors is LEARNER2 [1]. It collected data about *part-of* relations. Unlike our game, LEARNER2 lacks the capability of redirecting the user's effort to improve the reliability of data already collected. There are also several games that encourage participation of users to enter data into knowledge bases. Among these, Cyc released the game FACTory which is similar in format to the first stage of our game but does not include any way to guarantee the correct behavior of players. *Verbosity* [12], uses two players to fill in templates, and *Common Consensus* [6] asks two players questions about achieving a given goal.

Combining the computational power of machines and humans has been addressed in [10]. The use of a continuous scale to reason about commonsense knowledge was explored before in [3].

3 Game Design

The initial input information for the game comes from an off-the-shelf parser [4] that extracts a sentence from an article in *Simple Wikipedia* and, together with the action of the user, produces an update to the knowledge base as the output.

The simplest implementation of the game would have a single user classifying sentences as either commonsense or not. This is not enough because it would be impossible to evaluate the answers of the player as correct or incorrect. Also having several players evaluating the same sentence and accepting the input only if they agree amongst one another is not appropriate because it would be easy for a group of players to act in collusion and agree on entering the same answer, regardless of the question.

The basic problem is that human players can always agree on a fixed strategy, and yes/no questions are not enough to correctly classify the knowledge. To solve this, we add a non-human player to the set of players and classify the input text in four different categories: Nonsense, Unknown, True, False.[1]

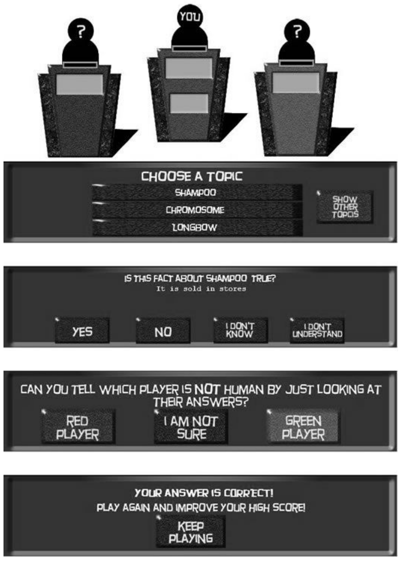
We devised a three player game in which the purpose of the player is to distinguish between another human and a machine. Both humans will give the same answer on the sentence while the machine will guess its answer. The design of the game solves the problem outlined before: The two humans can no longer agree on any strategy because the identity of the players is unknown. Also, if the answer of one player does not follow commonsense, the other human might erroneously identify the player as a machine, which results in a penalization on the player's score.

Because commonsense depends on the context, it is necessary to consider context explicitly. In [8], the author proposes a formula *Holds(p,c)* to assert that the proposition p holds in context c. Using this idea, the appropriate task for the player is to answer a question based on that formula. In our case, the context is handled by the name of the *Simple Wikipedia* article used as source for the sentence. The context can be used by the player to answer correctly, while addressing the problem of uninstantiated sentences that may be produced by the parser.

Figure 1 shows snapshots of the game in all its stages. The game works as follows:

- In the first stage, the player chooses a topic (which matches the title of the Wikipedia article from which a sentence is to be retrieved).
- A sentence is randomly selected from the article. The system chooses either a new sentence or a sentence that has been verified before. This balances the coverage and reliability of the data by increasing the times a sentence has been verified. Then, the player indicates whether the fact expressed by the sentence is true in the context of the article using the four options previously described.
- In the second stage, the player sees the answer of the other two players and identifies which of the two is the machine that is answering randomly. In the case of a single player playing the game, the other answer comes from recorded games. If it is impossible to distinguish between the two players, there is an option to pass and avoid making a decision. If the player identifies the human as the machine, points are deducted; otherwise, points are awarded.
- After this, the player gets the opportunity to play again.

[1] If the parser extracted an incomplete sentence or any other nonsensical data, the sentence is nonsense, if the sentence was extracted correctly the content may either be known (true or false), or unknown.



IS THIS FACT ABOUT SHAMPOO TRUE?
It is sold in stores

CAN YOU TELL WHICH PLAYER IS NOT HUMAN BY JUST LOOKING AT THEIR ANSWERS?

YOUR ANSWER IS CORRECT!
PLAY AGAIN AND IMPROVE YOUR HIGH SCORE!

Fig. 1. Snapshot of the game. All the stages of the game are shown one below the other.

4 Identifying Commonsense Knowledge

A majority vote is not enough to identify a fact as commonsense, because we have more confidence if a sentence was evaluated by a large amount of players rather than by a few. Thus, we create a scale of commonsense that describes how common a specific fact is. The scale needs to be proportional to the ratio of people who know the given fact, and also contain information about the confidence of such ratio. We first define four quantities, t_{count},f_{count}, u_{count}, n_{count}, that hold the number of times a sentence s has been classified as true, false, unknown, and nonsense, respectively.

Definition 1. Let $P_\sigma(s)$ be the ratio of people that have answered true, false, and unknown over the total number of instances the sentence s has been verified. Let $m(s)=t_{count}(s)+f_{count}(s)+u_{count}(s)+n_{count}(s)$.

$$P_\sigma(s) = \frac{t_{count}(s) + f_{count}(s) + u_{count}(s)}{m(s)} . \quad \textbf{(1)}$$

Under the assumption of independence, each instance of the game can be considered a Bernoulli trial. $P_\sigma(s)$ is then an estimator of the real proportion of people that understand the sentence. Our null hypothesis is that the ratio of people classifying the sentence as nonsense should be 0.5. If we fail to reject the null hypothesis, we must conclude that we don't have enough information to identify the sentence as meaningful or nonsense.

Definition 2. Let $e_n(s)$ be the *effect size*, the difference between the actual and expected number of times the sentences have been marked as nonsense.

$$e_n(s) = \left| n_{count}(s) - \frac{m(s)}{2} \right| . \tag{2}$$

Definition 3. Let $p_n(s)$ be the p-value of the Binomial Hypothesis Test, the probability of observing a difference in the value of a random variable of at least the size of the effect size $e_n(s)$.

$$p_n(s) = P\left(X < \frac{m(s)}{2} - e_n(s) \right) + P\left(X > \frac{m(s)}{2} + e_n(s) \right). \tag{3}$$

The p-value $p_n(s)$ is the probability of observing the current counters given the null hypothesis. The lower its value, the more confident we are about their values.

Table 1 shows some sentences with their corresponding $P_\sigma(s)$ and $p_n(s)$. Notice that $P_\sigma(s)$ and the p-value cannot distinguish amongst all sentences because one only considers the ratio of people that agree on the sentence, whereas the other only considers the amount of people that has evaluated the sentence. With this in mind we define $\pi_s(s)$, which allows us to easily classify sentences as meaningful or nonsense.

Table 1. The Id is used to refer to each sentence in the paper. Eval refers to the number of times that the sentence has been evaluated. $P_\sigma(s)$ and $P_\gamma(s)$ are the proportion of people who didn't answer *nonsense*, or who answer *true* or *false*, respectively. The value of $\pi_s(s)$ and $\pi_c(s)$ represents the confidence that we have when classifying the sentence as meaningful or known, respectively. The last column is the decision made with a significance of 0.1.

Id	Sentence	Article	Eval	$P_\sigma(s)$	p-value	$\pi_s(s)$	Meaningful
1	*People are known acting in comedies are comedians*	Comedy	1	1	1	0.5	Unknown
2	*Computer can use many bits*	Computer	6	1	0.03	0.98	**Yes**
3	*For example some languages (e.g.Chinese,Indonesian)*	Verb	6	0.17	0.03	0.02	**No**
Id	Sentence	Article	Eval	$P_\gamma(s)$	p-value	$\pi_c(s)$	Known
4	*It is a county in the U.S. state of North Carolina*	Anson County	9	0	0.004	0.002	**No**
5	*The level experience is needed to level*	Diablo II	1	1	1	0.5	Unable
6	*Chess is a very complex*	Chess	9	1	0.004	0.99	**Yes**

Definition 4. Let $\pi_s(s)$ be the value that represents how much confidence we have on a sentence *s* being meaningful.

$$\pi_s(s) = \begin{cases} 1 - p_n(s)/2 & \textit{if } P_\sigma(s) > 0.5 \\ p_n(s)/2 & \textit{if } P_\sigma(s) \leq 0.5 \end{cases} . \tag{4}$$

To classify the sentence as meaningful, we only need to define a threshold α against which we can compare $\pi_s(s)$. If $\pi_s(s) < \alpha$ we have a confidence of 1-α that the sentence is nonsense and if $\pi_s(s) >$ 1-α, we have a confidence of 1-α that the sentence is meaningful. Otherwise, we can only conclude that we need more players to evaluate the sentence. We perform a similar analysis to the one described previously to define a scale $\pi_c(s)$ that represents the fact that a given sentence *s* is commonly known. In order to classify a sentence as commonsense we combine both $\pi_s(s)$ and $\pi_c(s)$.

Definition 5. Let $\pi(s)$ represent the confidence about a sentence being commonsense.

$$\pi(s) = \pi_s(s)\pi_c(s) . \tag{5}$$

Table 2 shows the corresponding value of $\pi(s)$ of the sentences from Table 1. Notice that to classify a sentence as commonsense it requires both $\pi_s(s)$ and $\pi_c(s)$ to be high.

Table 2. Id, Eval, $\pi_s(s)$, and $\pi_c(s)$ *are defined as* in Table 1. $\pi(s)$ represents the confidence that we have on identifying each sentence as commonsense.

Id	Eval.	π_s(s)	π_c(s)	Π(s)	**Commonsense**
1	1	0.5	0.5	0.25	Unknown
2	6	0.98	0.98	0.97	**Yes**
3	6	0.02	0.5	0.01	**No**
4	9	0.99	0.002	0.002	Domain-specific
5	8	0.004	0.5	0.002	**No**
6	9	0.99	0.99	0.99	**Yes**

5 Evaluation

Coverage, reliability, and identifying the presence of knowledge that needs further classification are of primary interest to knowledge acquisition systems, especially when the knowledge comes from volunteer contributors. In contrast to other systems, our game offers an explicit way to detect knowledge that should be discarded due to errors or noise in the input of contributors. Also, all other previous games do not provide any way to distinguish the data that need further qualification. These features are achieved by the use of our scale $\pi(s)$. If a sentence is not nonsense, commonsense or domain-specific, then the game can be directed to present it to players more often until enough data has been collected to make a decision regarding such sentence.

Among the reviewed systems, only LEARNER2 reports data about redundancy. Out of 6658 entries, only 2088 are different statements and 4416 entries yielded only 350 distinct statements. This means that they collected 1.29 entries per statement. These few entries per statement produce unreliable data, which means that only 350 statements can actually be trusted. In contrast, our game collected 6763 entries and generated 3011 evaluated sentences, with an average of 3.46 entries per statement. Therefore, our data is more reliable than that of LEARNER2. Figure 2 shows the comparison of coverage and reliability between LEARNER2 and our game.

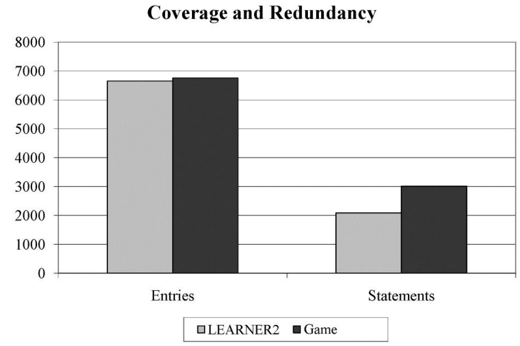


Fig. 2. Comparison between LEARNER2 and our game

For the evaluation, we asked 4 judges to classify a random sample of 50 sentences from our knowledge base. The judges evaluated the knowledge by classifying it in these categories: "*Generally/Definitively True*", "*Sometimes/Probably True*", "*Unknown*" and "*Nonsense/Incomplete*", which correspond to Commonsense, Domain-Specific, Unknown, and Nonsense, respectively. This categories are similar to the ones used by [9]. When comparing the answers of the judges to the ones from the game, the average agreement between players and judges was 94% (α=0.1).

In comparison to the other systems, *Verbosity* asked the judges to rate each input as correct or incorrect; the judges reported 0.85 of the data to be correct. LEARNER2 used a scale similar to ours and reported that 89.8% of the data that was entered by at least 2 people was correctly common knowledge. Our game outperforms the previous systems.

6 Conclusions and Future Work

We presented the design of a game that evaluates and classifies sentences extracted automatically from the Web. The main advantage of our design is that it classifies commonsense knowledge in a continuous scale, which allows us to talk about how common a commonsense fact is. Our analysis gives us confidence about the results even when some of the players disregard the rules and create noisy data. Also, we distinguish between data that needs to be evaluated further and data that has been classified with certainty. Although the game has already provided data that shows that our approach is viable, improvements in the design of the game are possible. One feature that will be further explored in future work is the demographics of the players. Each answer given by the player is stored according to their age group and location. This is useful because we will not only be able to classify commonsense knowledge, but we will also be able to cluster commonsense knowledge according to demographic information.

7 References

1. Chlovski, T., Gil, Y.: An analysis of knowledge collected from volunteer contributors. In AAAI'05: Proceedings of the 20th national conference on Artificial intelligence, pages 564–570. AAAI Press. (2005)
2. Downs, J.S., Holbrook, M.B., Sheng, S., Cranor, L.F.: Are your participants gaming the system?: screening mechanical turk workers. In Proceedings of the 28th international conference on Human factors in computing systems, CHI '10, pages 2399–2402, New York, NY, USA, 2010. ACM.
3. Druzdzel, M.J.: Probabilistic reasoning in decision support systems: from computation to common sense. PhD thesis, Pittsburgh, PA, USA. (1993)
4. Hadidi, B., Johri, N., Pantley, D., Pradham, A., Wang, F.: Automated knowledge extraction from wikipedia. Available upon request to authors. (2010)
5. Lenat, D.B., Guha, R.V., Pittman, K., Pratt, D., Sheperd, M.: Cyc: toward programs with common sense. Commun. ACM, 33(8):30–49. (1990)
6. Lieberman, H., Smith, D.A., Teeters, A.: Common Consensus: a web-based game for collecting commonsense goals. In In Proceedings of the Workshop on Common Sense and Intelligent User Interfaces held in conjunction with the 2007 International Conference on Intelligent User Interfaces (IUI 2007). (2007)
7. McCarthy, J.: Programs with common sense. In Semantic Information Processing, pages 403–418. MIT Press. (1968)
8. McCarthy, J.: Artificial intelligence, logic and formalizing common sense. In Philosophical Logic and Artificial Intelligence, pages 161–190. Kluwer Academic. (1990)
9. Schubert, L., Tong, M.: Extracting and evaluating general world knowledge from the brown corpus. In Proceedings of the HLT-NAACL 2003 workshop on Text meaning, pages 7–13, Morristown, NJ, USA, 2003. Association for Computational Linguistics.
10. Shahaf, D., Amir, E.: Towards a theory of AI completeness. In 8th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'07). (2007)
11. Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L.: Open mind common sense: Knowledge acquisition from the general public. pages 1223–1237. Springer-Verlag/ (2002)
12. von Ahn, L., Kedia, M., Blum, M.: Verbosity: a game for collecting commonsense facts. In In Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems, volume 1 of Games, pages 75–78. ACM Press. (2006)
13. Simple Wikipedia, `http://simple.wikipedia.org`