Collective Intelligence as a Source for Machine Learning Self-Supervision

Saulo D. S. Pedro
Federal University of Sao Carlos, UFSCar
Sao Carlos, Brazil
saulods.pedro@gmail.com

Estevam R. Hruschka Jr.
Federal University of Sao Carlos, UFSCar
Sao Carlos, Brazil
estevam.hruschka@gmail.com

ABSTRACT

The recent growth of virtual communities, social web and information sharing gives to information retrieval and machine learning systems a source of information referred as the "wisdom of crowds". In this work we show that this information could be used not only as a source of knowledge but as a way to bring intelligent systems closer to users by using their opinion as part of the knowledge acquisition/validation allowing self-supervision. For that we have implemented a validation system for the NELL (Never-Ending Language Learner) system using the question answering platform given by the Yahoo!Answers web community. Moreover, we focus in this paper, in the validation of first order rules induced by NELL using its Rule Learning (RL) algorithm. This paper presents the main motivations for using a QA forum instead of other web-based validation sources; describes the proposed approach with a "Macro QA"-based component named SS-Crowd (self-supervisor agent based on the wisdom of crowds) and brings and discusses the obtained results and how they can impact in a never-ending learning system like NELL in which self-supervision plays a crucial role.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*knowledge acquisition*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*text analysis*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Question-answering (fact retrieval) systems*

General Terms

Machine Learning

Keywords

self-supervision, question answering, knowledge validation

1. INTRODUCTION

WI&C'12 April 16, 2012 Lyon, France

Advances on telecommunications and the choice of Internet to share personal information brought lots of users to web communities, social networks, blogs, forums and specific games (called Games With A Purpose - GWAPs). The information contained on these sources hold relevant data for information technology and is an object of research on smart systems, information retrieval, natural language processing and question answering, among others.

The increase in the use of Internet to share information, provided conditions for research works like the one proposed in [8], which introduces the web redundancy over massive information to bring accuracy to question answering (QA) approaches to answer factoid questions. Following along those lines, in this work, we are exploring a way of taking advantage from web communities to solve problems and to introduce new concepts on QA and machine learning systems.

Years ago, computer-based systems were used to solve problems and answer questions about restricted domains. A good example of that are the customer-user websites that gives information about products and services. We are now facing the use of web data as a source of information given as input to machine learning systems to solve problems on most every domain (e.g. Watson Computer from IBM [9], NELL [5], etc.).

Machine learning systems depend on a source of information (input) to learn from. When the input is not perfect, these systems usually have to deal with the problem of optimizing data analysis to prevent the induction of false beliefs. So, the use of the web as a source for knowledge acquisition raises concerns. Most of the reasons are related to the fact that the information available on the web is not always correct neither precise, and if we are working with problems related to a broadly defined domain, it is more unlikely that we will be able to constrain the data to ensure its reliability.

The NELL system, introduced on [5], has potential to gather knowledge from almost any domain since it is learning from web pages in English and it uses acquired knowledge to keep learning continuously. As stated in NELL's website (`http://rtw.ml.cmu.edu`), in a nutshell, the system has as inputs (1) an initial ontology defining hundreds of categories and relations that NELL is expected to read about, (2) 10 to 15 seed examples of each category and relation and (3) a collection of 500 million web pages and access to the remainder of the web through search engine APIs. Based on the given inputs, NELL is running 24 hours per day, continuously (since January, 2010), to perform two ongoing tasks: 1) Extract new instances of categories and relations, and (2) learn to read today better than yesterday. In addition, NELL is

freely available on the web (`http://rtw.ml.cmu.edu`), thus anyone interested in using it or contributing to it, is allowed to get access to all its data and knowledge base (KB).

One key characteristic in a never-ending learning system is its self-supervision capability. In this sense, if a system like NELL could autonomously get access to QA forums to post questions related to its learning results, then it could use the answers (given by human users) in order to help reviewing its beliefs. To show the potential of QA forums on improving a never-ending system (e.g. NELL), in this paper we have focused our attention specifically to one of the algorithms composing NELL, namely the Rule Learner (RL). RL generates rules based on NELL's KB. These rules are logic driven and simply tell the system how different categories should interact with each other (semantic relations discovery). When the induced rules are applied to the system, NELL's beliefs on the categories covered by the rules can be modified.

Here is an example of a rule created by RL (in a Prolog-like syntax):

athletePlaysInLeague(x,NFL):- athletePlaysForTeam(x,Giants)

Applying this rule over NELL's database will raise its belief that every athlete that plays for the Giants, plays in the NFL.

Nevertheless, considering the imperfectly extracted knowledge present in NELL' knowledge base, some of the fact (as well as rules) induced by NELL's learning algorithms might not represent the universe as we know (our common sense) or might even be wrong. Thus, before being introduced into the system as beliefs (true facts), these facts must be validated by a component named *Knowledge Integrator* [6]. To perform such a validation, *Knowledge Integrator* takes into account a number of constraints, as well as, the output of a number of different learning algorithms that are coupled in order to allow higher precision. In addition, NELL also receives periodical shallow human supervision [5] in order to avoid concept drift.

At this point we could see the opportunity to work with the collected intelligence from web communities to help validating NELL's beliefs in an automated way, giving NELL an extra source of information to help self-supervision. Although being a dynamic knowledge-based system that can learn facts and relations on unrestricted domains, and even extend the initial knowledge base structure [16], NELL does not cover the whole universe of knowledge and the common sense that can be obtained from QA forums users. Thus, in our work we intend to show that the combined opinion of users can help to give guidance to a system like NELL, taking advantage of the collaboration of the coupled machine learning algorithms and the knowledge from QA forums users. The obtained results can be used to enhance NELL's performance on validation tasks in a more autonomous way, and that could be a valuable contribution to NELL's self-supervision.

Sources to gather user opinion are available in several systems through several devices such as FAQs, forums and mail lists. We decided to start working with QA systems because it allows us to collect knowledge from any domain and extract simple information. In addition, the popularity of web QA systems offers a variety of systems to harvest data.

Studies in QA usually presents techniques to optimize methods to find an answer for a specifc question by somehow interfacing with users through web QA systems such as Yahoo! Answers (`http://answers.yahoo.com/`), and WikiAnswers(`http://www.answers.com/`) among others. In this work, however, the QA approach is presented having an unusual flow and a "Macro-Reading" approach [15]. Thus, we are taking advantage of web questions and answers to validate the rules generated by NELL's RL algorithm. The proposed approach can be summarized in the following steps:

1) Automatically converting the first order rules (induced by RL) into understandable questions that would prompt users to decide whether the rule is suitable or not to our world.

2) Automatically assessing the validity of each analyzed rule.

3) Automatically discarding the invalid rules and giving the valid rules back to NELL as correct knowledge.

In others words, we are working with a *backwards* (or *reversed*) QA task. Traditionally in QA tasks, computer systems manipulate knowledge bases (automatically created from corpora or manually built by experts) to answer questions made by people. Instead, in our approach, the computational system asks questions to people and interprets the given answers improving its performance on the learning tasks, and consequently, also improving its self-supervision. In addition, instead of adopting a "deep QA" strategy, in this work, the "Macro-Reading" idea taken from [15] is used to define a "Macro-QA" task.

Consider that *"Micro-QA"* can be defined as in *Definition 1*:

Definition 1: "Micro-QA" can refer to the traditional QA task (or even the "deep QA" approach) where a single question is given as input, and the QA system must fully understand every bit of information present in the natural language sentence used as question. The desired output of such a system is the full information content in form of a natural language sentence that can be used as an answer.

In contrast, we define *"Macro-QA"* as follows:

Definition 2: "Macro-QA" is a task where the input is a set of questions (paraphrasing a single target question), and the QA system must try to get the general idea embedded in most of the questions. The desired output is a simple answer (e.g yes or no) that reflects the main idea behind the majority of the given questions.

The **Macro-QA** tends to be much easier than the **Micro-QA** because when analyzing a set of questions (instead of a single question) two main issues are raised. First, **Macro-QA** does not require "understanding" every bit of information embedded in all questions. If a specific question is given in a very complex form the system can simply discard it and focus in the questions that are easier to be automatically interpreted. Second, facts will be stated redundantly, using different wordings. In this sense a **Macro-QA** can benefit from this redundancy by focusing on analyzing only the simple wordings, ignoring hopelessly complex sentences, and by statistically combining evidence from many text fragments in order to determine how strongly to believe a particular candidate hypothesis.

Definition 3: "Reversed QA" is a task where the questions are proposed by the computational system which receives a set of answers (for each question) from human users.

Based on *Definition 3*, the **Reversed Macro-QA** idea can be defined as follows.

Definition 4: A "Reversed Macro QA" task has a set of answers (to a specific question) as input, and the system

must "understand" the main idea in the most simple answers (discarding the long and complex answers) and it should base its "answer understanding" also on the redundancy of the main ideas identified in the set of answers.

Following *Definition 4*, in a "Reversed Macro QA" task, for a question like "Is it true that If an athlete X has coach Z and coach Z coaches team Y, then athlete X plays for team Y?". If the system receives a set of 5 answers like[1]:

1) no.
2) no not always.
3) no it is not always true BYE
4) No, not unless you postulate that coach z coaches team y exclusively
5) athletes run jump etc they dont play for any team

It should discard the last ones (answers 4 and 5), try to get the main idea behind the simplest ones and observe the redundancy present in them.

When using the "Reversed Macro QA" approach to model a self-supervision component to a never-ending learning system (as we are doing in this work) an interesting aspect is worth mentioning: the system should ask questions to human users in a way the answers can be naturally generated in a format that can be easily interpretable by a machine. In this sense, answers given in a format that is *difficult* for machine understanding should be considered bad answers (even if they are correct answers). Therefore, in the same way researchers in the Human-Computer Interaction (HCI) area [2, 12] investigate interaction between people (users) and computers focusing on the human use of the "information", we are herein, investigating the "Reverse" Computer-Human Interaction (RHCI) focusing on the machine use of "information" given by humans.

One of the main goals of HCI is securing user satisfaction, and the main goal in a RHCI approach should be securing machine capability of getting help from humans in an easy and comprehensible way. The RHCI investigation should start being an important research area mainly if the development of new intelligent and autonomous systems (that interact with human users in order to improve their performance on self-supervision and other specific tasks) become more frequent. In Section 3, some initial ideas towards RHCI are given.

Another interesting issue, is that the work here proposed takes similar concepts from studies in active learning (AL) where users often have a role to prepare the dataset to reduce labeling costs and improve inference accuracy by selecting relevant data to a specific problem [19, 17]. In our case, RL extracts rules from NELL's KB and only the rules having a higher "impact" in the KB (as described in Section 3) are to be used as input to *sscrowd* (instead of validating the whole knowledge base). This filtering idea reduces the labeling cost focusing on the most relevant rules only. Afterwards, the knowledge of the crowds is used only on the selected extracted rules.

Since RL generates rules based on NELL's knowledge, we can use them to monitor the system's knowledge health. RL works as a mechanism that express NELL's beliefs but, when interacting with the web community, we are not looking for labels but for answers that carry opinions from users about the persistence of those beliefs on real universe instead.

[1]Here we show a real example of a question generated by *sscrowd* and the answers given by YahooAnswers users. It was extracted from the performed experiments described in more detail in Section 4

2. RELATED WORK

Building questions [20], or natural languages sentences in general, from different sources of information is a research topic related to text generation [7]. The work presented in [11] which shows how ground facts from an information extractor can be validated by a community. In addition, the paper considers several different interfaces, and measures how intrusive this process is considered by users. In spite of both works being closely related to our idea of automatically generating questions to be answer by humans using virtual communities, in our work we focus more on the *never-ending learning* aspect of self-supervision than on the issues discussed in our colleagues' works.

The idea of taking advantage on the redundancy of information from large content available on the web is focused in [8] to resolve QA problems. In the work, the amount of data available on line makes answer extraction easier and this task has a good performance even working on large dataset and simple natural language processing. The work introduces a new way of treating factoid questions on QA systems.

[4] presented the use of frequently asked questions (FAQ) instead of traditional text files as a source to retrieve answers for a QA problem. This work introduces the FAQFinder system and has an approach to reduce the costs of natural language processing to understand complex questions. The system proposed uses FAQs as its knowledge base and match the user's questions with existing questions on FAQ files.

The never-ending learning system is presented on [5]. The system extracts information from the web to grow its knowledge and learns to read better everyday. The work proposes an architecture for a learning agent that runs forever.

The collected intelligence as mentioned in [10], is the data retrieved from the social web and contains high valued information to web semantic development, but it should be taken to a scientific level. Gruber suggests that the real collective intelligence comes from the creation of knowledge impossible to be obtained manually and new ways of learning through the recombination of data from social web. Gruber describes the class of systems that can deliver at this opportunity as *collective knowledge systems* and he suggests four key properties that characterizes them. They are: user generated content, human-machine synergy, increasing results with scale and emergent knowledge. The last one, is some inference over the collected data obtained by computation that would lead to results not found in human collaboration itself.

3. SS-CROWD MODEL AND ANSWER EXTRACTION

When concerning the use of "the knowledge of the crowds" for labeling data or validating machine learning experiments, some options might be taken into consideration. Web-based validation approaches can use Amazon's Mechanical Turk, for instance, which is a convenient and inexpensive tool that enables well-designed experiments and direct answers in minutes [13]. In this work, however, we use a QA forum instead of Mechanical Turk. The main motivation for this choice is to investigate how a system like NELL can autonomously use the web (as we, human beings, do) to find answers to questions it could not answer "reading" web

pages. If we consider the human behavior, it is not so common for a person to design experiments and submit them to Mechanical Turk when wanting to find an answer to a specific question. The use of Mechanical Turk can be seen as a more sophisticated technique used mainly by researchers and not by average web users. One of NELL's principles is to investigate whether a computer can learn as we (human beings) do, thus, we think using a QA web-forum might be a suitable choice in this scenario.

There are a lot of QA forums available on the web community. In this work we use Yahoo! Answers because it provides easy implementation with access to database through API's, it is very popular on the web (which favors user participation from the whole world), has different categories of questions that lead to specific kind of users to answer, also has score and nominations for the best users (which should be a good source of confidence about the information collected). Yahoo! Answers already have three out of four Gruber's key properties to collective knowledge since users creates questions and answers, the database store is friendly available for the users through UI as human-machine synergy and it is more likely to find useful information with the increased participation of the users.

In the SS-Crowd, we have modeled a system combining the RL data extracted from NELL and information extracted from Yahoo! Answers API. We present now more details over the model.

Algorithm 1 retrieves the user's opinion about a question

```
for all answers as a do
  opinion is the user's opinion retrieved from answer ans
  if opinion = approved then
    app ⇐ app + 1
  else if opinion = rejected then
    rej ⇐ rej + 1
  end if
end for
score ⇐ app − rej
if bestAnswer = approved then
  score ⇐ score + (app + rej)/rej
else if bestAnswer = rejected then
  score ⇐ score − (rej + app)/app
end if
if score > 0 then
  return approved
else if score < 0 then
  return rejected
else
  return bestAnswer
end if
```

3.1 Convert rules into understandable questions.

As aforementioned, in this work we based the use of *SS-Crowd* on the First Order rules that NELL learns using its Rule Learner (RL) algorithm. In a nutshell, Rule Learner uses a variant of the FOIL algorithm [18]. The input for RL is a simple set of positive and negative examples of a rule's consequent. To learn the rule *R1*:

***R1**: statelocatedincountry(x, y):- statehascapital(x, z), citylocatedincoutry(z, y)*

RL uses positive and negative examples for the consequent *statelocatedincountry(x, y)* to induce the complete rule based on a separate-and-conquer algorithm. Each rule is initially learned as general rule, and progressively the algorithm specializes it, so that it still covers many positive examples but covers few negative examples. After a clause is learned, the examples covered by that clause are removed from the training set, and the process repeats until no positive examples remain [14].

For each rule, RL calculates an estimated conditional probability P?(conclusion|preconditions) using a Dirichlet prior according to following formulation:

$$P = (N_+ + m * prior)/(N_+ + N_- + m) \quad (1)$$

where N_+ is the number of positive instances matched by the rule in the training data, N_- is the number of negative instances matched, m = 5 and prior = 0.5.

Having a dataset containing RL induced rules, our first step was to run an algorithm to automatically create one question to represent each induced rule. One of the main focus here was to set an algorithm which would generate very direct questions so the user would not be confused about what to answer. Since the rules are logic driven formatted (in a PROLOG style), the conversion is simple as in the following example. Having *R1* rule mentioned above, the created question is:

Is this statement always true? If state X has capital Z and city Z is located in country Y then state X is located in country Y.

Note that the question above is not as natural as we usually speak. The generated questions reproduce the logic form of the rule and force the user to provide a straightforward answer so we can prevent complex answers that may cause difficulties on analysis. To translate the predicate names (e.g. statelocatedincountry(arg1,arg2)) into a more "readable" form we use NELL's metadata (e.g. "arg1 is a sate located in country arg2") available in the Read The Web project website (`http://rtw.ml.cmu.edu/rtw/kbbrowser/predmeta:statelocatedincountry`).

During tests, some users answered our questions with concerns about the question formatting. We took that feedback to improve our question generation algorithm and raise expectations on receiving better and more accurate answers.

With the first observations after test results, we noticed that even with straightforward question format, some opinions could not be resolved due to complex answers. To reduce the noise of unresolved answers, if we work interacting with users, it would be possible to force them to send information that would be easier to resolve. Since we are just harvesting for simple opinions, we performed another set of experiments where we have modified our question generation algorithm to force the system to ask the user to only answer yes or no. The approach is quite simple as well, we just add the phrase *"please just answer yes or no"* before the question. Using the rule of our last example with this approach, the conversion would be as follows:

Question created from rule: (please just answer yes or no) Is this statement always true? If state X has capital Z and city Z is located in country Y then state X is located in country Y.

The adherence of users contribution was very good, so we decided to consider this approach in our final results. Details are described in Section 4.

3.2 Input the question on Yahoo! Answers.

The Yahoo! Answers system as well as other web QA systems has several categories and sub-categories to input and answer questions. These categories help users to collaborate in their area of interest. In this work we used the category "Trivia" where users share information of any area of expertise, thus we could harvest information asking questions where people are more likely to answer.

After posting the generated questions to the QA forum, our component started to monitor the forum waiting for the answers to be retrieved. With the answers retrieved, we evaluate them matching with regular expression patterns defining keywords. These keywords are used as indication of an user opinion and give to the system an impression of the user about the correctness of the question. For us, this means the pertinence of the rule in the real universe. Even using this very simple approach (keywords-based), we had good results on finding the right opinion in most of our answers. We have examples of the extraction keywords in Table 1.

We automatized the creation of the question, defining a sentence for every relation from RL and access the web QA system through an API and web parsing procedures. The bottleneck for an application like the one here proposed was the limitations of the QA system which defines a maximum number of questions that could be asked in one day and the "points per question" policy which defines that each question made costs points to the user asking. These points can be obtained answering questions from other users. This limitation was overcome (in this version of the proposed approach) having a human user answering some questions whenever Yahoo! Answers asked for it.

keywords	
for approved rules	for rejected rules
yes	no
correct	incorrect
true	false
yeah	not
yup	nope

Table 1: Examples of the simple approach for the keywords used to retrieve user opinion from the answer.

3.3 Gather opinion from users and provide final result.

As shown in the Algorithm 1, we summarize all answers to a question and find the overall opinion. A higher weight is assigned for the best answer to reproduce the community confidence about that collaboration. Although the best answer is chosen by the community, there is a chance it does not fully represent the other answers. To address it on our model, the Algorithm assigns to the best answer a bonus as high as its agreement with the other answers. The overall opinion is now ready to feedback NELL and compose its knowledge base.

Notice that we are attempting to use the collected intelligence from the answers to provide NELL a background of human common sense, an inference over collected data that could improve the knowledge of a computer and its self-supervision capability.

4. EXPERIMENTS AND RESULTS

For experimentation, we ran our model with rules extracted from NELL and already evaluated by human inspection. The developers of NELL assigned a flag for each rule induced by RL. The flags indicate if a rule is approved or rejected, that is, if the rule truly represents reality. With these flags, it was possible to confront results from SS-Crowd and the results NELL has used coming from its developers. We ran the questions on US Yahoo! Answers because it uses the same language (English) than NELL's current database, then we would reduce noise due to language translation issues. It would be difficult to run every rule generated by RL because Yahoo! Answers as other web QA system, limits the inputs that a user can do in a period. Therefore, from a total of 639 rules NELL induced from its KB, we picked a dataset containing roughly 10% of the rules that most would affect NELL's belief if applied to the knowledge base. A total of 60 rules were converted into questions and asked with the regular approach and the yes/no question approach described in Section 3, generating 120 questions and 350 answers. The main motivations to perform these specific tests are as follows:

- Observe user's participation on answering logic driven questions.
- Find out user's behavior over questions about random fields of expertise and what the common sense would tell us.
- Compare user's overall opinion with specialists judgment over the rules extracted from NELL.
- Asses how the proposed approach would help NELL's self-supervision ability.

We intend to observe results through different perspectives, driving to achieve to benefit from results as much as possible. We ran our experiments with the whole amount of answers from users and compared these results with the Yes/No question approach that simplifies the question to obtain more understandable answers. We also ran experiments with the best answers only to estimate how it affects the results with the algorithm proposed in Section 3.

4.1 Combining Answers to Extract Final Result

In this experiment, when a rule is converted to a question, and put into the QA system, the retrieved answers can be analyzed either individually or combined. For example, if a question has five answers and one of these five answers was chosen as the best one, then we have five individual results approving or rejecting the rule and another result that is the combination of those five, considering the influence of the best answer as described in Algorithm 1. In the combined result, as answers are compared together, the influence of unresolved opinions are dissolved by approved opinions, rejected opinions and the opinion from the best answer. In Figure 1, we show the progression of the improvements when comparing individual and combined answers.

Although it is curious that we have the same total of false negatives and true negatives in Table 2, this coincidence is not our focus here. Otherwise, the 21 false positives in the same table, represent 7.19% of the total of resolved individual answers. These false positives indicates that it is difficult

for the users to agree with some rules previously approved by the specialists. Users have so very low tolerance about questions acceptance that a single hypothesis that does not match the proposed rule would lead the rule to be rejected. The universe of knowledge of the specialists and the web community are higher NELL's, but the analysis of the community is more restrictive. At the moment this low tolerance of web users is positive to our intentions. We can explain that by showing a rule example.

Rule extracted from RL: teamplayssport(x, hockey) :- teamplaysinleague(x, nhl)

This rule represents the belief that a team that plays in league NHL, plays the sport hockey. Although it might seem obvious, users pointed that NHL could refer to New Hampshire Lacrosse, and the rule would not be true for all values of X. With restrictive judgment like this, we can point to NELL the characteristics of the KB that could be improved in a self-revision task, showing for an example, that NHL could refer to other entities.

We can also use SS-Crowd to identify not applicable rules that are created by aspects of NELL's KB.

Rule extracted from RL: athleteplaysinleague(x, nba) :- (athleteplayssport(x, basketball)

We note that applying the rule, all basketball players should be in the NBA league. The specialists as well as the web community agreed that this is wrong and should not be included as knowledge. A rule like this could be generated because information retrieved from web pages do not keep large data about less known basketball players and most of the known ones plays on NBA. Besides, today NELL works with web pages in English only, which would make even more unlikely to find players from countries where English is not spoken. Therefore, the rules were generated by RL over the same limits of NELL's database, leading to a tendency to prefer relations that makes sense in a universe smaller than the user's common sense.

		Answers from Users	
		+	-
Developers Opinion	+	103	21
	-	84	84

Table 2: The confusion matrix for individual results over 60 rules converted into 120 questions that received 350 answers. The unresolved answers are not considered.

4.2 Analysis of Best Answers

Since the web community chooses the best answer for a question, it is reasonable to think about using only best answers to compose our validation, but for the web users the best answer is the one that explains better the problem described by the question. Most of users explanations are very detailed and contains examples and citations. In our experiments, 16.84% of the total of best answers were unresolved. Complex explanations are hard to evaluate and it is too much risk to rely the whole analysis on a single answer even if it's the better one and specially if we cannot resolve it to be approved or rejected. Furthermore, as described in Section 3, there is a chance that the best answer does not represent the other opinion from users. In Table 3, we show a comparison between the best answers and the other answers from the community.

		Not Best Answers	
		+	-
Best Answers	+	48	11
	-	43	115

Table 3: The confusion matrix comparing the best answers and not best answers. The unresolved results were not considered.

4.3 Yes/No Questions Against Regular Questions

As explained in Section 3, even with straightforward question format, some opinions could not be resolved due to complex answers. The Yes/No questions approach had good acceptance on Yahoo! Answers community and we could dramatically reduce the total amount of unresolved answers when compared to regular question approach. The reduction of unresolved answers are shown in Figure 1. With the good results on Yes/No questions approach to resolve answers, we ran tests to compare the results with the regular questions and look for any difference to estimate how we could take advantage of Yes/No Questions on final results. Our results confirmed a tendency in having yes/no answers with similar accuracy results when compared to regular answers specially between the regular-individual approach and the yes/no-combined approach. These results indicates that we can use the Yes/No question approach without forcing the community to return different opinion. The Yes/No questions are easier to read and interpret, but it is not biased to interfere in the user's common sense. The results are shown in Table 4. In addition, we received 23% more answers from Yes/No questions than the total amount received from regular questions. Thus, apparently, users are more likely to collaborate when prompted to answer simple questions, which are easier to manipulate and extract answer. Considering these facts together, we realized that prompting the user to answer driven questions is a good way to retrieve high valued information from QA systems if we know exactly what kind of answer would best fit our machine learning task. These results contribute to the idea of RHCI (Reverse Human-Computer Interaction).

4.4 Overall Analysis and Discussion

We are concerned to guarantee that our approach does not interfere on NELL's belief by reducing promotions of relevant data to its knowledge but, since its belief grows infinitely, the knowledge base tends to comprehend more information like for example, learning about basketball players from other countries and other leagues.

With detailed analysis on the received answers, we noticed that most users collaborate even if the question apparently has no clear purpose. Users often access QA systems to discuss subjects beyond answering questions. Since our questions are mostly obvious for users, they are not likely to turn into a topic of discussion.

Even with unusual questions, over 90% of the 350 answers were received before the first 3 hours the questions are opened. So it does not seem that working with QA systems would lead to a response time bottleneck. We had to wait for the choice of best answers and any other analysis from the user's side, but as soon as our question is on line, we started harvesting results.

Question Type	TP	TN	FP	FN	Precision	Recall	Accuracy	F-Measure
Regular Individual	41	31	7	39	0.85	0.51	0.61	0.64
Regular Combined	27	2	17	7	0.61	0.79	0.54	0.69
Yes/No Individual	62	53	14	45	0.81	0.57	0.66	0.67
Yes/No Combined	27	7	11	11	0.71	0.71	0.60	0.71
Best Answers	19	28	3	29	0.86	0.39	0.59	0.54

Table 4: Evaluation of effectiveness. To generate these results, the unresolved answers were not considered

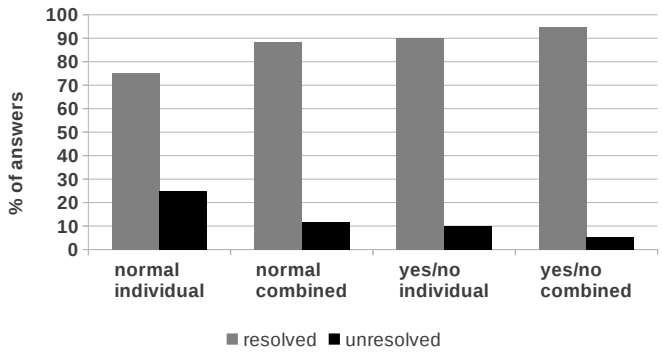

Figure 1: Percentage details of the comparison between regular and yes/no questions through different approaches. The approaches are individual answers and combined answers. This figure shows the progression of the improvements attempting to resolve user's opinion from answers.

Although there is enough data to work with, the amount of answers asked by SS-Crowd is significantly smaller than the amount of answers received by usual questions (in Yahoo! Answers) and it would be difficult to take advantage from high redundancy of web knowledge if we submit the questions to a single QA forum. We are concerned about completeness and accuracy to identify the web community common sense. We are also concerned about known question processing problems [3] that possibly comes from the lack of confidence over information retrieved from the web and new challenges like extracting an opinion from an answer. As we are not focusing on these problems right now, we treated them as simple as we could and explored extra resources (like knowing the best answer) the QA forum could give us.

In a nutshell, the results from our experiments revealed that it is better to drive the user to a specific kind of answer such as Yes/No questions. This way, it is easier to extract opinions and even forcing the user to give a specific answer, our approach seems to not interfere in their common sense. We also found that the "best answers" could represent the opinion of a single user and these answers are vulnerable to not contain any redundant information, thus, not contributing in an effective way for machine learning tasks and self-supervision if taken individually. In Table 4, we show a good match between best answers and other approaches, which means that the opinion chosen as best, in our experiments, represents well the other opinions. The combined results are more accurate to identify user opinion, in Figure 1, our results show a reduction in the unresolved questions from 24.85% to 5.09% and since users are not experts, we are most interested in their common sense than their individual observations. Results also revealed false positives over the comparison between users and specialists in most experiments. The low accuracy values on Table 4, are the results of the rules approved by NELL that users don't agree to fit their common sense. The low values accuracy represents the difference between the universe of NELL's KB and the universe of web users knowledge.

Information that does not fit reality, or false positives could affect NELL more than other learning system since NELL learns infinitely and these false positives could be propagated leading the system to increase its false beliefs. The false negatives instead, is knowledge not acquired yet and although NELL learns forever, it still does not have enough data to cover it. Thus, the tendency to point false positives (revealed in the experiments) is an advantage for learning tasks.

Another interesting issue to be mentioned is that in a never-ending learning system, such as NELL, there are a number of different methods and algorithms for information extraction, learning and validation. Therefore, in a system having these properties, the *Macro-QA* approach becomes even more interesting. Consider, for example, a situation where, based on a rule *R SS-Crowd* generates a question, posts it to YahooAnswers forum and cannot interpret any of the given answers (due to their complexity). The *Macro QA*-approach will induce the algorithm to simply ignore this rule (R), as well as its respective question and the process will continue with the next question to be generated (based on another rule). Considering that NELL has other algorithms (in addition to *Rule Learner*) that are also extracting rules from its knowledge base (such as the ones described in [1][14]) NELL's *Knowledge Integrator*, will have other ways to try to validate that specific rule. In other words, failing to asses the validity of a specific rul R, does not generated an error and is not a big problem to NELL. The intuition behind this example is that *SS-Crowd* should give feedback to NELL only on the rules that could be solved, and the unresolved rules should not have impact in the knowledge base. For that reason, Table 4 does not include unresolved questions.

5. CONCLUSION

The increased use of the Internet to share personal information and opinions from unrestricted domains, allowed researchers from machine learning and information retrieval to gather data from social web and convert it to knowledge.

In this work we proposed SS-Crowd, a component that uses web QA systems to reach the web community and use its collected intelligence as a source to improve self-supervision in machine learning tasks. To show some of the possibilities of our component we implemented a validation task for rules from NELL's RL by converting the rules into user understandable questions and asking the users from Ya-

hoo! Answers about the pertinence of the rule on their common sense. We could also compare the results from NELL's developers in RL rules and the results from the proposed approach using the web community.

The observed results increased our expectations that it is possible to count on data available on social web as collective intelligence to compose validation algorithms that can improve machine learning system's self-supervision capability. However a more detailed study is required over the methods for extraction and interpretation of collected data.

One advantage of collective intelligence for validation is that if users are well directed to a specific kind of collaboration, such as answering for simple questions, it would not be necessary to gather a large dataset of information to improve learning systems because we are prompting the user exactly what we want to know about the learning system. We noticed the progress on retrieving well interpretable information from users in our results on Figure 1 where a simple manipulation of the question format led to an easier and straightforward opinion retrieval from answers.

We could also see that emergent knowledge suggested on [10] as a determinant factor for a collective knowledge system could be acquired from the combination of collected data and learning systems like NELL, therefore, the resulting knowledge could be directed to the system itself.

In the future we intend to gather a larger set of rules and automate this validation to a large scale system, improve the interaction with users , improve the opinion analysis of the answers allowing a more accurate extraction of complex answers, combine more data available on QA systems such as user's scores and reputation to enhance our confidence on the information from web users. We also intend to explore the possibilities provided by other QA systems and web communities. We are most interested to show we can use the collected intelligence from web communities as part of self-supervised systems through automated knowledge validation. A lot of learning systems depends on human supervision and the opportunity raised by crowd computing may take machine learning to a more collaborative level towards self-supervision.

6. REFERENCES

[1] A. P. Appel and E. R. Hruschka Jr., Prophet - a link-predictor to learn new rules on NELL. In the IEEE ICDM 2011 Workshop on Data Mining in Networks, Vancouver, Canada, 2011

[2] R. Baecker. *Readings in Human-Computer Interaction: toward the year 2000*. Morgan Kaufmann, 1995.

[3] J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C. Lin, S. Maiorano, G. Miller, et al. Issues, tasks and program structures to roadmap research in question & answering (q&a). *Document Understanding Conferences Roadmapping Documents*, 2001.

[4] R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. *AI magazine*, 18(2):57, 1997.

[5] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka Jr., and T. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, volume 2, pages 3–3, 2010.

[6] A. Carlson, J. Betteridge, R. Wang, E. Hruschka Jr, and T. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 101–110. ACM, 2010.

[7] N. Dethlefs and H. Cuay. Hierarchical reinforcement learning for adaptive text generation. In Proceedings of the 6th International Natural Language Generation Conference (INLG '10), 2010.

[8] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298. ACM, 2002.

[9] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. Murdock, E. Nyberg, J. Prager, et al. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.

[10] T. Gruber. Collective knowledge systems: Where the social web meets the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):4–13, 2008.

[11] Raphael Hoffmann , Saleema Amershi , Kayur Patel , Fei Wu , James Fogarty , Daniel S. Weld, Amplifying community content creation with mixed initiative information extraction, Proceedings of the 27th international conference on Human factors in computing systems, April 04-09, 2009.

[12] J. Jacko and A. Sears. *The human-computer interaction handbook*. L. Erlbaum Associates.

[13] A. Kittur, E. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 453–456. ACM, 2008.

[14] N. Lao, T.M. Mitchell, W.W. Cohen. Random Walk Inference and Learning in A Large Scale Knowledge Base. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2011.

[15] T. Mitchell, J. Betteridge, A. Carlson, E. Hruschka, and R. Wang. Populating the semantic web by macro-reading internet text. *The Semantic Web-ISWC 2009*, pages 998–1002, 2009.

[16] T. Mohamed, E.R. Hruschka Jr. and T.M. Mitchell. Discovering Relations between Noun Categories. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2011.

[17] F. Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.

[18] J. Ross Quinlan and R. Mike Cameron-Jones. FOIL: A Midterm Report. In Proceedings of the European Conference on Machine Learning (ECML '93), Pavel Brazdil (Ed.). Springer-Verlag, London, UK, 3-20, 1993.

[19] B. Settles. Active learning literature survey. *Science*, 10(3):237–304, 1995..

[20] S. Zhao, H. Wang, C. Li, T. Liu, Y. Guan.

Automatically Generating Questions from Queries for Community-based Question Answering. In Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 929?937, Chiang Mai, Thailand, November 8 ? 13, 2011.