Autonomously Reviewing and Validating the Knowledge Base of a Never-Ending Learning System

Saulo D. S. Pedro
Federal University of São Carlos - UFSCar
São Carlos - SP - Brazil
saulods.pedro@gmail.com

Ana Paula Appel
IBM Research - Brazil
São Paulo, SP, Brazil
apappel@br.ibm.com

Estevam R. Hruschka Jr.
Federal University of São Carlos - UFSCar
São Carlos - SP - Brazil
estevam@dc.ufscar.br

ABSTRACT

The amount of information available on the Web has been increasing daily. However, how one might know what is right or wrong? Does the Web itself can be used as a source for verification of information? NELL (Never-Ending Language Learner) is a computer system that gathers knowledge from Web. Prophet is a link prediction component on NELL that has been successfully used to help populate its knowledge database. However, during link prediction task performance *Prophet* classify some edges as misplaced edges, that is, edges that we can not assure if they are right or not. In this paper we use the Web itself, using question answer (QA) systems, as a Prophet extension to validate these edges. This is an important issue when working with a self-supervised system where inserted errors might be propagate and generate dangerous concept drifting.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

anomaly link detection, question answering, active learning, graph mining, never-ending-learning

1. INTRODUCTION

Over the past years the amount of information available in the Web has been substantially increased. Thus, the Web became a powerful source of data available to be used for computational systems to acquire knowledge. Taking advantage of that fact, a number of information extraction systems are using the web to build knowledge bases by extracting facts from free text from webpages. NELL [4], KnowItAll [11], TextRunner (Banko et al., 2007), Yago [30], WOE (Wu and Weld, 2010), Snowball [1], PROSPERA [22] and ReVerb [12] are examples of systems designed for harvesting facts from Web sources and storing them in a structured way.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

Another important research area benefited from this amount of data available on the Web is the study of complex networks that have revealed useful properties related to data represented by graphs. Such properties reveal relevant common characteristics of different complex networks [23]. Some interesting and relevant properties are: the power-law degree distributions [13], the shrink diameter on evolving networks [19], the Small World phenomenon [21], triad closure [33], among others. These patterns can be used to help understanding not only the interaction among human beings and social networks [18], but also the dissemination of information and diseases [7], intrusion detection [14] and so on.

Another interesting idea, related to knowledge extraction and data available on the Web is the NELL (Never-Ending Language Learner) system, which is a computer system that runs 24 hours per day, 7 days per week [5] focusing on the main goal of gathering knowledge from web pages in English and using its acquired knowledge to become a *better learner* each day. It was started up on January, 12th, 2010 and should be running forever, gathering more and more knowledge from the web, thus, learning to read better each day.

The content available on the web is not, however, always reliable. Therefore, the use of the web as a source for knowledge acquisition raises concerns, since it can lead a machine learning system to propagate false beliefs extracted from noisy data. Using NELL as an example, to avoid propagating mistakes, part of the knowledge extracted by the never-ending learning system should be supervised by humans before it can be incorporated definitely in the knowledge base (KB). This situation shows the importance of defining an efficient strategy to allow NELL reviewing its own KB identifying possible mistakes.

If we naively apply traditional statistical inference procedures in NELL's KB, which assume that instances are independent, we may be led to inappropriate conclusions about the data. Thus, the potential correlations due to links should be handled appropriately to exploit the knowledge hidden in the data. The link information can be used to improve the predictive accuracy of the learned models: attributes of linked objects are often correlated, and links are more likely to exist between objects that have some commonality [15].

Since the relations and categories extracted by NELL are mapped as an ontology, converting NELL's knowledge base to a complex network is a natural process [2, 17], and this transformation allows us to use graph properties to investigate if the knowledge learned by NELL is correct or not.

Link prediction can be defined as follows: *Given a snapshot of a social network at time t, we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t'* [20]. *Prophet* [2] implements link prediction on NELL to infer new rules and instances, which can help increasing the size of knowledge base (KB) inferring new knowledge from facts already stored in the KB. However, when *Prophet* predict new links on NELL's graph-based KB, a few previously known facts might represent negative evidence of the existence of the new edge, and in such cases those previously learned facts will be flagged.

As an example, consider the situation where *Prophet* predicts a new link that connects all `sports` with a `sportsLeague` and, consider also, the presence of two previously learned facts (in NELL's KB): i) ***Cristiano Ronaldo*** *plays* ***soccer*** (represented by the link connecting both ***Cristiano Ronaldo*** and ***soccer***); ii) ***Cristiano Ronaldo*** *plays in league* ***NBA*** (represented by the link connecting ***Cristiano Ronaldo*** and ***NBA***). In this specific situation (graphically represented in Figure 1), human supervision can correctly identify that the link `(soccer, Cristiano Ronaldo)` is correct, but `(Cristiano Ronaldo, NBA)` is a misplaced connection.

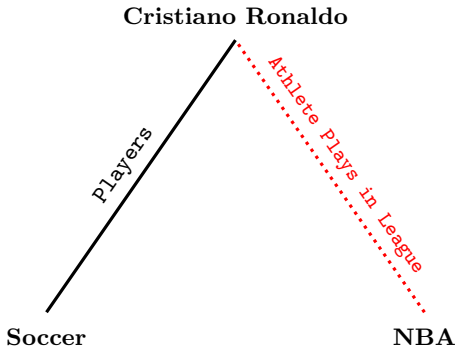

Figure 1: An example of misplaced edge - the connection between "Cristiano Ronaldo" and "NBA"

Considering the complexity of the problem of identifying misplaced edges, in cases like the one represented in Figure 1, both edges are flagged, by *Prophet*, for future human supervision. Currently, NELL's developers (see http://rtw.ml.cmu.edu/people) are responsible for human supervision in NELL's KB and they limit this verification task in 5 minutes a day. Time restriction is imposed in order to assure that NELL is autonomous enough to self-supervise its own KB even when its developers are not available to review and validate new learned facts. Thus, in this paper, our main goal is to show how *Conversing Learning* techniques [25] can be used to help reviewing and validating facts that were learned by NELL and were flagged as possible mistakes by *Prophet*. The main motivation for using *Conversing Learning* is based on fact that it allows the never-ending learning system to collect knowledge from any domain and extract simple and straightforward information given by humans. In addition, the popularity of web QA systems offers a variety of systems to harvest from. Thus, *Conversing Learning* can allow flagged links (that were originally sent to NELL's developers) to be human supervised based on the common sense of people in web communities.

The proposed approach can be summarized by the following steps: i) automatically converting KB's facts into human understandable sentences; ii) using the previously created sentences to automatically generate questions that will prompt users to decide whether the facts are correct or not; based on the web community answers, automatically give feedback to *Prophet* that will use it as a parameter to create or not a new link in NELL's KB. In others words, when approaching the problem based on *Conversing Learning*, we are working with QA idea with the flow system reversed. We are now asking people and gathering their observation to improve learning tasks and self-supervision (instead of learning from corpora to give people an answer, as in tradition QA systems).

The sequence of the paper is organized as follows. Section 2 presents the mathematical definitions used in this paper. Section 3 presents the main related work with our paper. Section 4 presents NELL and Section 5 presents *Prophet*. Section 6 introduces Conversing Learning and the QA component SS-Crowd (Self-Supervisor Agent Based on the Wisdom of Crowds) and how it works. Section 7 the motivation of this paper. Section 8 presents the proposed work. Section 9 shows the results of this work. Finally, Section 10 concludes the work.

2. DEFINITIONS AND PROPERTIES

Let $G = (V, E)$ be an undirected graph with a set of nodes V and a set of edges E. We use the symbol n for the number of nodes and the symbol m for the number of edges. $\Gamma(u) := \{u \in V : \exists\{v, u\} \in E\}$ of node u is defined to be the set of nodes in V that are adjacent to u. A path in a graph is a sequence of nodes such that from each of its nodes there is an edge to the next node in the sequence. Two paths are independent if they do not have any internal node in common.

A triangle Δ of a graph $G = (V, E)$ is a three node subgraph with $V_\Delta = \{u, v, w\} \in V$ and $E_\Delta = \{(u, v), (v, w), (w, u)\} \in E$. An open triangle $\Lambda(u, w)$ of a graph $G = (V, E)$ is a three node subgraph where $\Lambda(u, w) = \{(u, v), (v, w)\} \in E \wedge \{u, w\} \notin E$. The $\aleph(u, w)$ is the number of common neighbors that u and w share.

3. BACKGROUND

There is a significant amount of research related to our problem, which we categorize as anomaly detection, question answering and active learning.

Anomaly Detection: At an abstract level, an anomaly is defined as a pattern that does not conform to expected normal behavior. However, it remains difficult to give a general, formal definition of what an anomaly is. Detecting anomalies in various data sets is an important task in data mining. Although research has been done in this area, little of it has focused on graph-based data. Anomaly detection finds extensive use in a wide variety of applications such as fraud detection for credit cards, insurance or health care, intrusion detection for cyber-security, fault detection in safety critical systems, and military surveillance for enemy activities. Anomalies and outliers are two terms used most commonly in the context of anomaly detection; sometimes interchangeably [8]. In [24, 6, 10] the authors use the Minimum Description Length (MDL) principle to detect anomalies. A more specific task is anomalous link discovery (ALD): given a static or dynamic graph representing

objects and their relationships, identify those links that are anomalous [28].

Question Answering: The increase of the use of Internet to share information allowed research using the web redundancy over massive information to bring accuracy to question answering (QA) [31], [9] [16]. We are now using information from the web to solve problems and introduce new concepts on QA and machine learning systems. The usual task of QA systems is to browse a database looking for answers to a specific question. In this paper we will use a Web QA system as an interface to ask users about the validity of a learning system, thus configuring a supervision task. To reach the users knowledge, in this work we use the SS-Crowd component, an implementation of Conversing Learning (CL) that allows communication between machine and the wisdom of crowds.

Active Learning: On machine learning we are interested to extract knowledge from a database. The learning activities are often performed over a set of labeled instances and the labeling tasks could be complex and time consuming. To resolve this matter, Active Learning (AL) methods aims to reduce the labeling bottleneck by prompting an oracle to provide label for data and reduce the whole database to a more relevant database [29]. The oracle can be any source with enough background to support the evaluation of the relevance of an instance. Usually this source is a human that will select a subset of the database. Alternatively in this work we have an automatic process to perform this task. The *Prophet* algorithm could send all predicted rules to be analyzed by through the *SS-Crowd* component, but it would be a waste of the whole system's performance. Since *Prophet* is interested only on the outliers instead of the whole set of new rules, then it's necessary to select the instances that will be prompted to users through *SS-Crowd*. In our work we apply concepts of AL when we separate the outliers from the total amount of instances analyzed by *Prophet*. We use the *stream based selective sampling* with a heuristic implied on *Prophet* algorithm to find outliers.

4. NELL

NELL (Never-Ending Language Learner) is a computer system that runs 24 hours per day, 7 days per week. It was started up on January, 12th, 2010 and should be running forever, gathering more and more knowledge from the web. In a nutshell, NELL's knowledge base (KB) is an ontology defining hundreds of categories (e.g., `person, sportsTeam, fruit, emotion`) and rules (e.g., `PlaysFor(athlete,sportsTeam), playsInstrument(musician,instrument)`).

NELL's knowledge base is one of the results of the *Read the Web* project[1] (RTW). RTW aims to develop a probabilistic, symbolic knowledge base that mirrors the content of the web. If successful, this will make text information on the web available in computer-understandable form, enabling much more sophisticated information retrieval, natural language understanding, and general problem solving.

The main components of NELL are: CPL, which is described in more details in [3] and works as a free-text knowledge extractor which learns and uses contextual patterns like "mayor of X" and "X plays for Y" to extract instances of categories and relations. CSEAL is based on [32] and implements a semi-structured extractor which queries the Internet with sets of beliefs from each category or relation, and then mines lists and tables to extract novel instances of the corresponding predicate. CMC is a simple set of binary L2-regularized logistic regression models which classify noun phrases based on various morphological features (words, capitalization, affixes, parts-of-speech, etc.). Finally, the Rule Learner (RL) is a first-order relational learning algorithm similar to FOIL [27], which learns probabilistic Horn clauses from the ontology.

[1]http://rtw.ml.cmu.edu/rtw/

5. PROPHET

Prophet [2] is a component that allows NELL to use link prediction to infer new rules. The NELL's KB is composed by rules in the form `SportTeam(Basketball,Milwaukee Bucks)`. In short, *Prophet* converts all rules in NELL's KB to a complex network, *rtwgraph*, where relations are edges and predicates are nodes, and find open triangles. An open triangle is two relations (instanced rules) connected by a predicate. For example, the relations `SportTeam(Basketball,Milwaukee Bucks)` and `TeamPlaysInLeague(Milwaukee Bucks, NBA)` are connect by `Milwaukee Bucks` as presented in Figure 2.

In *rtwgraph* might exist more than one independent path of size two connecting two nodes that are not connected by a path of size one. Thus, *Prophet* combines the common neighbors measure with the number of independent paths of size two to decide when a path between the nodes should be created or not. Figure 2 shows that Basketball may be connected with NBA for more than one independent path.

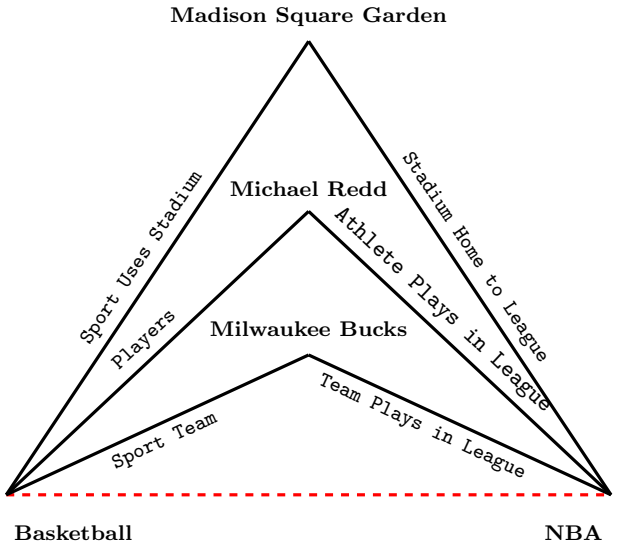

Figure 2: Predicates (nodes) and Relations (edges) extracted by NELL mapped into a complex network (*rtwgraph*). Two relations that share a predicate is viewed as an open triangle.

Since the *rtwgraph* is formed by the instances of the rules in the NELL, one new rule can be predicted more than once. Thus *Prophet* use all the relations of same rule to create a new rule. However, some relations might not have a number of neighbors great or equal than the number required by *Prophet* or the number of independent paths is less than

the number of independent paths defined by the new rules. For these relations a new edge connecting the other two is not created and the edges that compose the open triangle are flagged as misplaced, which means that these instances need to go through some process to be validated as correct or not. Figure 3 presents this situation. Suppose that *Prophet* create the rule `R(Sport, SportLeague)` as present as horn clauses bellow:

- `R12a(sport, sportsleague):- players(sport, athlete), athleteplaysinleague(athlete, sportsleague), numberof(athlete)` $\geq$ `10;`
- `R12b(sport, sportsleague):- sportteam(sport, sportsteam), teamplaysinleague(sportsteam, sportsleague), numberof(sportsteam)` $\geq$ `10;`
- `R12c(sport, sportsleague):- sportusesstadium(sport, stadiumoreventvenue), stadiumhometoleague(stadiumoreventvenue, sportsleague), numberof(stadiumoreventvenue)` $\geq$ `10`
- `R12d(sport, sportsleague):- players(sport, athlete), athleteplaysinleague(athlete, sportsleague),sportteam(sport, sportsteam), teamplaysinleague(sportsteam, sportsleague), sportusesstadium(sport, stadiumoreventvenue), stadiumhometoleague(stadiumoreventvenue, sportsleague);`

Thus, baseball and NFL will only be connected if there are at least 10 common neighbors or all the independent paths between them. Figure 3 shows that not all independent paths exist, the dot line connecting the relations, `Sport Uses Stadium` and `Stadium Home to League` indicates that the connection is absent. Also, the number of common neighbors is less than 10, so the edge will not be created.

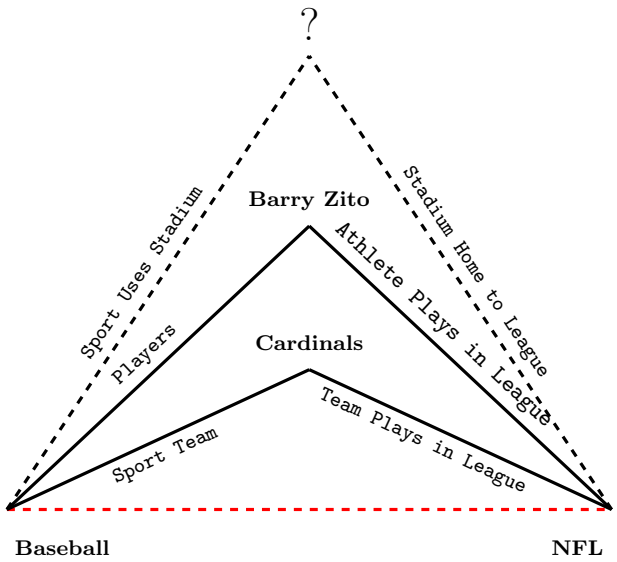

Figure 3: An open triangle that was flagged as uncertain because Baseball and NFL do not have the minimum number of neighbors (10) neither all the independent paths. The dotted lines represent the relations that are absent in this open triangle.

When relations are flagged as uncertain, *Prophet* assumes that at least one of the adjacent edges are false, which means that there might be wrong information on NELL or the adjacent edges are correct, but the edges predicted does not make sense. For example, looking at Cardinals that connect Baseball to NFL, both edges are correct but the connection does not make sense. This happens because Cardinals is a SportsTeam that plays both Baseball and Football, so the connection between Football and NFL is correct but Baseball with NFL is not. In the aforementioned example about Christiano Ronaldo in Figure 1, the relation (edge) AthletePlaysInLeague is a wrong knowledge, since Cristiano Ronaldo does not play in NBA.

However, it's impossible for a link prediction algorithm decide based only in the graph structure when the edges combination make sense or not. Thus, although *Prophet* method seems pessimistic, it only adds relations of which it has complete certainty. This is an important issue when working with a self-supervised system where inserted errors can be propagated and generate dangerous concept drifting. Thus, when a relation is flagged as uncertain, NELL should review (self-reflection) if this relation is a wrong knowledge or not. To do this is necessary a human inspection or an automatic processes.

6. CONVERSING LEARNING AND SS-CROWD

Conversing Learning (CL) is a model of machine learning that allows a system to take advantage of the collective intelligence to improve learning tasks. This model was first presented in [25] and can be seen as an extension of Interactive Learning (IL) that instead of prompting for new information from an oracle and wait for a response, it actively looks for assistance when needed. As well as Active Learning (AL) and IL, this oracle could be automatic systems or humans assistents. One of the advantages of CL is that the system can actively look for collaboration from multiple sources, which favors redundancy and increased recall.

In this work we apply CL in the misplaced edges validation of *Prophet* to infer whether an edge (instanced rule) is valid or not. Activities like this configures a self-supervision task, which was already a target of working with the wisdom of crowds [26]. We have already seen implementation of CL using web communities such as Yahoo! Answers and Twitter users as oracles, but its model allows the use of forums and more known collaborative channels such as Mechanical Turk.

Here, we take advantage of the Yahoo! Answers web community as the oracle of our CL implementation because of its popularity and unrestricted domain which favors redundancy. We are also using the Self-Supervisor Agent Based on the Wisdom of Crowds, so called, SS-Crowd algorithm proposed in [26] to reach this web community, acting like a regular user and gathering information. Usually learning systems require methods to ensure that the knowledge acquired from data is well represented given the problem semantics. In other words, learning systems need a method to validate the knowledge gathered. In many situations, the validation is made by humans manually. However, CL provide an automatic way to perform it through web communities collaboration thus configuring a self-supervision task.

The main motivations to use crowd computing on the web instead of mechanical turk is that the misplaced edges pro-

vided by *Prophet* are already a result biased by NELL's knowledge base and *Prophet* itself. This is a supervision activity, and our collaborators are not aware of the identity of the machine. Doing so, we have a chance to go deep into the users common sense to extract more natural answers. SS-Crowd is also a good component to work with systems that mimic the human behavior. Since NELL aims to learn as humans do, it would be an interesting work to see if we can clarify our questions (validate knowledge) as human do.

The main activities of SS-Crowd and its basic workflow are as follows:

- Convert the information from knowledge base into understandable questions.
- Input the question on Yahoo! Answers
- Gather opinion from users and provide final result

Each answer received from the web community contains an opinion of the user about the data which is being validated. The answer can either approve or reject that knowledge. After receiving all the answers for an specific question, the *SS-Crowd* algorithm combines the answers to produce a single result. Yahoo! Answers has a special feature called "the best answer". This special answer is the one picked by the community as the best one. The *SS-Crowd* algorithm weight the answer and combines it with others answers received to provide a final result, which is returned to the learning system for further evaluation.

Although we use SS-Crowd, other implementation of Conversing Learning could be applied. Since this is a system that requires validation, the greater requirement would be to have a learning system that provides *SS-Crowd* knowledge pure or parsed that could be converted into any understandable human format, such as questions, that would be sent to a web collaborative environment.

7. MOTIVATION

Validation methods are often required in order to ensure that the knowledge acquired from a machine learning system is a good representation of the domain that we intend to learn from. These methods could be performed with the intervention of the system's developers and external sources like mechanical turk. Learning systems are widely raised with human supervision. AL tasks are a good example of this interaction, specially when relevant data is picked from whole dataset by a human oracle to reduce labeling costs.

With a machine like NELL, that runs forever and learning more and better everyday from any domain, as the KB is extended over time, the flow of knowledge to be validated by human inspectors increases. *Prophet* is a component that helps NELL to improve its knowledge acquisition by creating new rules. With the NELL's KB growing forever, *Prophet* will be able to identify more rules over time as well. Although *Prophet* has already a system to identify misplaced edges itself, the addition of validation methods over it could lead us to a new level of supervision with significant details of the knowledge acquired so far. We want this supervision to be automatized using the *SS-Crowd*.

Anomalous link detection is one of the most difficult task in graph mining and has the goal of finding links that are suspicious in some way[34, 28]. It depends only from graph struture is hard and time consuming. Not only link prediction but also anomalous link detection could be improved if we have more information about the network and not only the graph structure.

Thus, we are motivated to benefit from the web community specially when working with a system that learns forever from any domain. The web community suits our intentions very well because the information we use as the source of the validation (the opinion from web community users) is widely available and, instead of the work with human inspection only, this approach would not become a bottleneck of a system with a knowledge base that grows infinitely like NELL. In addition, Web QA systems gather unrestricted knowledge from any domain as does NELL. Thus, we intend to discover more knowledge from the outliers with the help of the web communities before processing the misplaced edges with developers or the learning system itself. We want to maximize the amount of information we can use to learn and raise the accuracy of machine learning tasks.

8. PROPOSED WORK

Prophet is capable of finding new relations in the NELL's KB and also able to identify the anomalies referred as misplaced edges. Machine Learning systems usually need some kind of validation to keep the knowledge acquired in good direction. The outliers identified by *Prophet* are valuable information to be used as guidance for NELL's health-check.

The information gathered by *Prophet* could be just sent to human supervision, however, we are interested to take advantage of the web communities and it's large content of the human common-sense to learn more and take the best profit from these anomalies. Our work proposes a method to combine the knowledge gathered from web communities through the *SS-Crowd* component with the outliers identified by *Prophet*. Our intentions are to use web QA users opinion to validate the anomalies.

When *Prophet* identifies an outliers, it means that the its algorithm was able to determine a new rule but there are a few instances that do not match all the requirements of rule found by *Prophet* and these are called misplaced edges.

Although we know that these outliers are the result of mismatched instances, there is variety of reasons for these results. Thus, the analysis of these misplaced connections may drive us to architect specific solutions if needed.

The *SS-Crowd* component provides an automatic interaction with *Prophet* and the outliers verification will become a self-supervised task, which would be great advantage to NELL because it learns forever which would lead *Prophet* to grow its edges prediction and need for more anomalies verification.

Outliers are composed by two relations (edges) connected by a common predicate. There are two possible scenarios for the anomalies. The first one, is that at least one relation (edge) in the anomaly should be wrong. The second, is that the two rules are right but because of combination made by *Prophet* the relation predicted is wrong.

Our work proposes a way to identify the root of the anomalies. Thereby, we want to ask the web community about each misplaced edge. We want them to answer whether or not the two relations that composes the outliers fits the real world.

Prophet can provide a set of details about the rules (edges) that compose the anomalies. We use these details with the

SS-Crowd component to ask questions to the web community. The same component gathers its answers and returns the overall opinion of the users. We intend to use this result for further evaluation on *Prophet* and NELL's KB.

9. EXPERIMENTS

Our intentions with these experiments are focused on exploring possibilities of combining the abilities of *Prophet* and *SS-Crowd*. We want to empirically assess a system that can create new rules in NELL's KB and, has also the power of a component that gathers knowledge from human opinion available on the Internet through web communities. We drove our experiments to achieve high valuable information that could be used for components and developers of learning systems. In the proposed experiments, we counted on web community common sense to investigate a set of outliers identified by *Prophet* and we expected to discover novel knowledge about NELL's KB and *Prophet* prediction ability.

With the web community opinions about misplaced edges in hands, it will be possible to put to the test the *Prophet*'s ability to identify anomalies. In addition, if we get a good confirmation of *Prophet*'s link prediction accuracy, we may focus our attention to NELL in order to explore the components of the outliers and try to understand why a rule that matches most of the knowledge does not work for a few instances.

To run the experiments, we took a set of previously generated outliers to create an *anomalies* data set. The anomalies are displayed as structured information. Each instance of this anomalies dataset represents a misplaced link containing the two relations and three entities that composes each outlier, as shown in Figure 1 (shown in section 1). Looking at Figure 1, it is possible to notice that NELL's KB already support relations `Players` and `AthletePlaysInLeague`. We want to know about a third relation connecting entities `Soccer` and `NBA`, which would be a transitive relation through entity `Cristiano Ronaldo`.

For the experiments presented in this paper, we used NELL's KB at the 100th iteration to create the undirected graph rtwgraph with 9,419 nodes and 24,132 edges. Then, we ran *Prophet* that found new rules and instances and misplace edges. In the next step, all misplaced edges were sent to *SS-Crowd* to start the human assessment process that will give us the common-sense opinion about *Prophet* accuracy when identifying misplaced concepts. Table 1 shows three pairs of valid rules identified by *Prophet* and chosen to be submitted to *SS-Crowd*, the number of instances flagged as misplaced edges and the number of answers obtained for each pair of rule.

Table 1: Distribution of the relations considered in our tests

Relations	# of outliers	# of answers
`AthletePlaysInLeague & Players`	9	72
`TeamPlaysSport & TeamPlaysInLeague`	20	144
`TeamPlaysSport & TeamWonTrophy`	53	386

9.1 Converting outliers into questions for Yahoo! Answers

To illustrate the process of converting detected outliers into questions for Yahoo!Answers, consider for example that, *Prophet* inferred from NELL's KB that *Barry Zito* is a Baseball player and also that *Barry Zito* is an athlete that plays for NFL league. If an athlete plays an sport and also plays in a league then it's very likely that other players of the same sport would also play in the same league. *Prophet* uses this information amongst others to instantiate a new rule, which will be a relation between players and leagues. Without the anomaly identification, *Prophet* might imply that baseball players plays for NFL league. However the algorithm finds that the singular instance (Barry Zito) of our example does not fit the rule just created.

We want to explore this misplaced edge, measure *Prophet* accuracy when identifying anomalies and also understand why the new rule does not work for a minor set of instances.

In our example, we are dealing with two relations (`Players` and `AthletePlaysInLeague`) and these relations are exactly what we want to explore. We want the web community to answer whether or not these relations suit well the real world. As explained in section 8, we expect that at least one of these relations are wrong, which would indicate a good accuracy of the algorithm to identify misplaced connections. We want *SS-Crowd* to individually examine these relations, so that we have one question for each relation. The *SS-Crowd* algorithm converts relations into questions as follows:

- Is *Barry Zito* a baseball player?
- Is *Barry Zito* an athlete that plays for league NFL?

SS-Crowd sends these questions to Yahoo! Answers and since the method depends on human interaction (guidelines from the web community) we have to wait for the answers. Most answers came in the first 3 hours after the input. Of course if we want to take advantage from the best answer as described in section 8 we need to wait longer (it takes at least 2 days to community pick a best answer). After the waiting period, *SS-Crowd* will gather the answers from Yahoo! Answers and apply its scoring algorithm to determine the overall opinion from the users about the edges (rules) of our anomaly.

In our example, *SS-Crowd* found that, through the eyes of the community about this instance, the relation `Players` is applicable and the relation `AthletePlaysInLeague` is not applicable, as expected. If both relations were classified as right then this unexpected result would be an advice to check on the predicted edge, because the unexpected result might be the consequence of another problem than the anomalies identification.

9.2 Extracting information from outliers

To show that our approach can significantly enhance the benefits of the misplaced connections identification in a learning system like NELL, we took a set of 82 outliers from *Prophet* and ran *SS-Crowd* expecting to find at least one edge classified as wrong for each anomaly detected.

As seen in Table 2, 48% of the verified outliers have two edges classified as right by the web community. Since, in these experiments, we are dealing only with outliers instances previously know to be real outliers, it's expected

Table 2: Numbers for edges evaluated as suitable or not to the real world through the web community eyes.

Outliers	
at least one wrong edge	39 (47.56%)
both edges correct	40 (48.19%)
unresolved edges	3 (03.65%)

that from the total amount of outliers, at least one edge per outlier instance should be flagged as wrong by the Web community. The rate of outliers with at least one wrong edge indicates the health of the anomalies detection algorithm. We don't know exactly what caused the creation of the misplaced edges, but we know which of the edges from the outlier is wrong. This is valuable information for the learning system because it tells developers to focus attention on the knowledge that generated the wrong edge and reduce the costs to harvest the database looking for less relevant bit of knowledge. From our previous analysis, as explained in section 8, the amount of outliers with two correct edges indicates that we might have an inaccurate detection of anomalies. If both edges are correct, then this particular instance should not have been identified as an anomaly.

9.3 Identifying ambiguities in the KB

Since we expected lower rates of outliers with both edges right, at first, we focused our attention in the predicting algorithm expecting to find problems in the anomalous detecting algorithm. With deeper look, the surprisingly high total of misplaced connections with both edges classified as right drew our attentions to these specific subset of outlier instances. Results like these might indicate, for example, that the *SS-Crowd* algorithm is not accurately identifying the opinion from the web community or even that the web community sent flawed information. With the closer look on these edges, we found out that most of them have a particular feature, as shown in the example below involving the relations `TeamPlaysSport` and `TeamWonTrophy`.

1. Manchester United is a team that plays sport basketball.

2. Manchester United is a team that won trophy UEFA Champions League.

The users answered that both relations are right, so instead of looking for noise on *Prophet* prediction we noticed that Manchester United is a basketball team and also a soccer team. In this case the *Prophet* prediction is right as well as the validation from users. From the example above, *Prophet* could imply that UEFA champions league teams winners plays basketball. However, *Prophet* has weak evidence about that since there are few UEFA winners teams that also plays basketball and the instance was identified as a misplaced connection.

This kind of problem have a tendency to happen every time we have an entity used to describe more than one meaning in the knowledge base. In our example, NELL was not successful to decide whether Manchester United is a basketball team or a soccer team. Since NELL learns forever, eventually it will learn about more basketball teams, find more evidence about Manchester United team, and problems like this have a tendency to decrease. However, NELL intends to learn better every day, and new possibilities of misunderstandings like our example could be expected. The *SS-Crowd* component combined with *Prophet* can drive NELL to focus on a specific problem and enhance its knowledge supervision and verification.

In a nutshell, *SS-Crowd* combines the benefits of both *Prophet* and NELL. It could be used by *Prophet* as an indicator of it's accuracy and it could be used by NELL as an indicator of the beliefs that needs attention. These algorithms have a way to provide self-supervision for a system like NELL that depends on human supervision. This approach also opens doors for new experiences with Never Ending Learning systems such as self-revision through the refactoring of knowledge flagged as wrong.

10. CONCLUSION

The increase of Internet and user generated content puts learning systems in a good position to autonomously harvest information and create continuously-expanding knowledge bases. The use of the Web as a source of knowledge, however, raises concerns since the information available online is not always reliable and and accurate. In this work, we showed how can we manage web content to use it as a source of verification and supervision for learning systems through a Conversing Learning approach.

NELL is a system that gather information from web and uses acquired knowledge to keep learning better each day, forever. Link prediction has been successfully used (in previous works) with *Prophet*, a component to infer new rules for NELL with graph mining techniques. *Prophet* has a well-defined process to create new rules for NELL based on link prediction. Some of these predicted links, however, do not attend all the requirements to have enough confidence and to be promoted as knowledge. Therefore, *Prophet* flags these links as misplace edges which need to be revised and validated. This validation was previously done by human inspection, but in this work we proposed a methodology that allows performing it automatically through a Conversing Learning approach. The proposed methodology allows *Prophet* to assess human opinion through Web communities thus configuring a self-supervision approach. The *SS-Crowd* algorithm is based on CL model and gather knowledge from Yahoo! Answers, thus, allowing supervision by using the common-sense of web users to validate learning systems.

The results obtained in the performed experiments have shown that the combination of *Prophet* and *SS-Crowd* allows a never-ending learning system (such as NELL) to identify which edges are really wrong and which edges needs more time (NELL iterations) to fill the gaps on information to be considered valid. Thus, the experiments show that *Prophet* has a great accuracy. Most of the combination of edges that produce a misplaced connection are related to a coreference problem restricted to NELL and is not a misbehavior of *Prophet* itself. After our analysis we may follow up to system developers, not only a set of anomalies but also a good indication of what knowledge should be verified and what could be improved in the learning system. Thus, the validation of a learning machine with *SS-Crowd* is a useful approach to help self-supervision and self-revision in NELL.

11. ACKNOWLEDGMENTS

Authors would like to thank the support of Carnegie Mellon University, FAPESP, CNPQ and CAPES to the project.

12. REFERENCES

[1] Eugene Agichtein and Luis Gravano. Snowball: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, DL '00, pages 85–94, New York, NY, USA, 2000. ACM.

[2] Ana Paula Appel and Estevam Rafael Hruschka Jr. Prophet - a link-predictor to learn new rules on nell. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011*, pages 917–924, 2011.

[3] Andrew Carlson. *Coupled Semi-Supervised Learning*. PhD thesis, Carnegie Mellon University, 2010.

[4] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, 2010.

[5] Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*, 2010.

[6] D. Chakrabarti. AutoPart: Parameter-free graph partitioning and outlier detection. In *PKDD*, 2004.

[7] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur.*, 10(4):1–26, 2008.

[8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58, July 2009.

[9] 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.

[10] William Eberle and Lawrence Holder. Discovering structural anomalies in graph-based data. In *ICDMW '07: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, pages 393–398, Washington, DC, USA, 2007. IEEE Computer Society.

[11] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165:91–134, June 2005.

[12] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[13] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM 1999*, volume 1, pages 251–262, Cambridge, Massachusetts, 1999. ACM Press.

[14] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, February 2010.

[15] Lise Getoor and Christopher P. Diehl. Link mining: a survey. *ACM SIGKDD Explorations*, 7(2):3–12, 2005.

[16] C. Kwok, O. Etzioni, and D.S. Weld. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262, 2001.

[17] Ni Lao, Tom Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 529–539, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[18] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470, New York, NY, USA, 2008. ACM.

[19] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *eleventh ACM SIGKDD*, pages 177–187, New York, NY, USA, 2005. ACM Press.

[20] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM.

[21] Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[22] Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 227–236, New York, NY, USA, 2011. ACM.

[23] Mark Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[24] Caleb C. Noble and Diane J. Cook. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 631–636, New York, NY, USA, 2003. ACM.

[25] Saulo Pedro and Estevam Hruschka. Conversing learning: Active learning and active social interaction for human supervision in never-ending learning systems. pages 231–240. Springer, 2012.

[26] Saulo DS Pedro and Estevam R Hruschka Jr. Collective intelligence as a source for machine learning self-supervision. In *Proceedings of the 4th International Workshop on Web Intelligence & Communities*, page 5. ACM, 2012.

[27] J. R. Quinlan and R. M. Cameron-Jones. Foil: A midterm report. In *Proc. of ECML*, 1993.

[28] Matthew J. Rattigan and David Jensen. The case for anomalous link detection. In *Proceedings of the 4th*

international workshop on Multi-relational mining, MRDM '05, pages 69–74, New York, NY, USA, 2005. ACM.

[29] B. Settles. Active learning literature survey. *Machine Learning*, 15(2):201–221, 1994.

[30] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.

[31] N. Tomuro and S. Lytinen. Retrieval Models and Q&A Learning with FAQ Files. *New Directions in Question Answering*, pages 183–194, 2004.

[32] Richard C. Wang and William W. Cohen. Iterative set expansion of named entities using the web. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 1091–1096, Washington, DC, USA, 2008. IEEE Computer Society.

[33] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[34] Daniel Zeng and Zan Huang. Link analysis-based detection of anomalous communication patterns. In Christopher Yang, Daniel Zeng, Michael Chau, Kuiyu Chang, Qing Yang, Xueqi Cheng, Jue Wang, Fei-Yue Wang, and Hsinchun Chen, editors, *Intelligence and Security Informatics*, volume 4430 of *Lecture Notes in Computer Science*, pages 318–320. Springer Berlin / Heidelberg, 2007.