Journal of Network and Innovative Computing.
ISSN 2160-2174 Volume 3 (2015) pp. 078–087
© MIR Labs, www.mirlabs.net/jnic/index.html

# Two different approaches to Ontology Extension Through Machine Reading

**Paulo Henrique Barchi**[1], **Estevam Rafael Hruschka Jr.**[2]

[1]Federal University of São Carlos, DC/UFSCar, Computer Science Department,
São Carlos - São Paulo, Brazil
*paulobarchi@gmail.com*

[2]Federal University of São Carlos, DC/UFSCar, Computer Science Department,
São Carlos - São Paulo, Brazil
*estevam@dc.ufscar.br*

*Abstract*: **NELL (Never Ending Language Learning system) is the first system to practice the Never-Ending Machine Learning paradigm techniques. It has an inactive component to continually extend its KB: OntExt. Its main idea is to identify and add to the KB new relations which are frequently asserted in huge text data. Co-occurrence matrices are used to structure the normalized values of co-occurrence between the contexts for each category pair to identify those context patterns. The clustering of each matrix is done with Weka K-means algorithm: from each cluster, a new possible relation. This work present newOntExt: a new approach with new features to turn the ontology extension task feasible to NELL. This approach has also an alternative task of naming new relations found by another NELL component: Prophet. The relations are classified as valid or invalid by humans; the precision is calculated for each experiment and the results are compared to those relative to OntExt. Initial results show that ontology extension with newOntExt can help Never-Ending Learning systems to expand its volume of beliefs and to keep learning with high precision by acting in auto-supervision and auto-reflection.**

***Keywords***: knowledge aquisition, ontology extension, machine reading, machine learning.

## I. Introduction

The goal of making machines smart enough to help us with post-modern day-to-day tasks has made the Machine Learning area expands to novel paradigms. A continuously growing field of Machine Learning is focused on read information automatically from open domain texts, like the web. The Read The Web project aims to built a never-ending machine learning system which improves constantly its ability to convert non-structured information into structured information. This system is called Never-Ending Language Learning system (NELL)[1, 2]: the first (described in the literature) to implement never-ending learning principles. NELL uses its own learning capability as well as its continuously growing knowledge base to learn better each day. It has as input an initial ontology and initial seeds, it takes advantage of the combination of several strategies and algorithms to continuously induce new knowledge from millions of web pages. To be able to keep learning forever, NELL counts on two important properties: self-supervision and self-reflection. In addition, NELL also counts on social networks (such as Twitter and YahooAnswers!) and some shallow human supervision to ensure it is free from errors, thus avoiding concept drifting. Figure 1 ilustrates an example of the hierarchy in a knowledge representation graph: Company, Person, Sport and City are categories of knowledge which derives from Everything; some examples of relations are presented in the picture: *(Company, LocatedIn, City)*, *(Person, WorksFor, Company)*, *(Person, Plays, Sport)* and *(Sport, PlayedIn, City)*.

NELL has different components to perform this continuous learning task. OntExt is one of the components. This system uses redundant Web information: semantically similar context patterns which are frequently stated in a huge volume of text and unknown to NELL must be found and learned to perform the ontology expansion. The existing NELL's KB is the source of labeled examples as contexts, instances and categories already known. Relations with unknown context patterns and known instances are sought in the corpus with the relational information. These relations are the candidates to be included in NELL's KB. For example, if NELL knows Vioxx and Arthritis as instances of the categories Medicine and Disease, respectively, and ”Vioxx can cure Arthritis” and ”Vioxx is a treatment for Arthritis” (this one many times) are relations stated in the corpus, then ”Vioxx is a treatment for Arthritis” is a good candidate to be included in NELL's KB[3].

This task has a high computational cost to be integrated with NELL, considering the few number of relations which were created by OntExt methodology (more details in subsection III-C). This work presents newOntExt, the system which has as main goal to turn the ontology extension task feasible to NELL. In order to do this, newOntExt proposes to overcome some problems faced by OntExt with new features (details in section IV) as: new implementation of OntExt methodology; better Open Information Extraction with state-of-art systems (ReVerb and R2A2, details about these systems in subsection III-A); a computationally elegant file structure to perform a

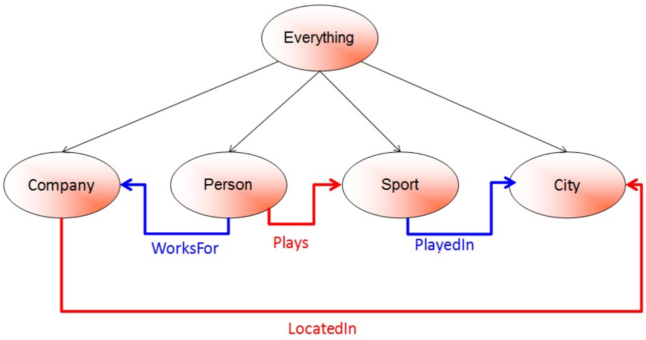


**Figure. 1**: Example of hierarchy of an ontology fragment.

quicker search through the relevant sentences; a divide-and-conquer method to act in reduced category groups of interest; and a collaboration with Prophet to propose names to relations found by this other system component of NELL.

The main experiment described in section V is a collaboration of newOntExt with Prophet, another system component of NELL: Prophet collects unnamed possible new relations by going through the graph which represents NELL's Knowledge Base; newOntExt proposes a name to each relation by performing its ontology extension approach (see section IV).

This paper presents newOntExt system, the background in which it emerges (section III), its methodology to ontology extension (section IV), experiments (section V) and final considerations (section VI).

## II. Problem Definition

The terminology used and the formalization of the problem considered in this work are presented in this section (based on [4]). A Knowledge Base (KB) $B$ is defined as a 4-tuple ($C$, $I_C$, $R$ e $I_R$), where $C$ is a set of categories (for example, a set consisting of categories relationed to sports: athlete, sport, team and sports league), $I_C$ is the set of pairs instance-category (for example, *(Neymar, athlete)*) for categories in $C$, $R$ is the set of relations (in this sports context, *athletePlaysInTeam* is an example of relation), and $I_R$ is the set fo triples instance-relation-instance for relations present in $R$ (for example, *(Neymar, athletePlaysInTeam, Barcelona)*).

Each instance of a relation $r \in R$ is a 3-tuple $(e_1, r, e_2) \in I_R$, where $(e_1, c_1) \in I_C$, and $(e_2, c_2) \in I_C$ for categories $c_1$ e $c_2 \in C$. Each category instance can be referenced by one or more Noun Phrase (NP). For example, the instance *Neymar* can be referenced by the own NP *Neymar* or by *Neymar Jr*. From this reasoning, *N(i)* is defined as the set of NPs which corresponds to the category instance *i*.

In addition to the KB, another input to this methodology is a huge set of triples in the format Subject-Verb-Object (SVO) extracted from text corpus of natural language. Let $D$ be this resource wiht a big set of tuples in the format $(sn_1, v, sn_2, f)$, where $sn_1$ and $sn_2$ are the NPs which corresponds to the subject and object of the sentence, respectively, $v$ is a verb (or a verb phrase), and $f \in \mathbb{R}_+$ is the normalized count of this tuple.

For each triple of $D$, verify if $sn_1$ and $sn_2$ are in the KB $B$: $\{\exists\, ((e_1,\, c_1),\, (e_2,\, c_2) \in I_C;\, c_1,\, c_2 \in C) \mid e_1 \equiv sn_1;\, e_2 \equiv sn_2)\}$. For the positive cases, each triple $(sn_1, v, sn_2, f)$ is stored. In the next step, these triples are considered for building the co-occurrence matrices.

For each category pair *($c_1$, $c_2$)* $\in C$ a co-ocurrence matrix is built up. Let $n_{v(c_1,\, c_2)}$ be the number of verbs (or verb phrases) with which instances of $c_1$ and $c_2$ co-occur. The co-occurence matrix for these categories has the dimensions: $[n_{\mathrm{v(c_1,\, c_2)}}][n_{\mathrm{v(c_1,\, c_2)}}]$ (same amount of lines and columns, which is the amount of verbs (or verbal phrases). The elements of this matrix are filled up which normalized co-occurence values. These values are normalized by the greatest value of co-occurences: all elements (values) are divided by this greatest number.

The *K-means* clustering is applied to every co-occurrence matrix to obtain clusters which represent the new relations. For each co-occurrence matrix, the values are clustered in $k$ clusters ($k \in \mathbb{N}^+$). This implies that for each cateogory pair $(c_1, c_2) \in C$ is possible to generate $k$ relations.

Let $k$ be the number of clusters by co-occurrence matrices, consider $i \in \mathbb{N} \mid 0 \leq i \leq k$. For each cluster, the nearest verb $v_{i(c_1,\, c_2)}$ to the *centroid* is considered the best candidate to create the relation. This new relation is stated as the relation between the categories $c_1$ and $c_2$ and by means of the verb $v_{i(c_1,\, c_2)}$, which can be represented as the triple *($c_1$, $\mathrm{v_{i(c_1,\, c_2)}}$, $c_2$)*.

**Problem Definition**: Given a Knowledge Base (KB) $B$ ($C_1$, $I_{C_1}$, $R_1$ and $I_{R_1}$) and a text corpus $D$ with a set of tuples in SVO format, newOntExt searches for new relations in $D$ still not present in $B$ with category instances already present in $B$. Briefly, the task aims to find new relations for category instances which are already known to expand the KB.

## III. Background

### A. Machine Reading

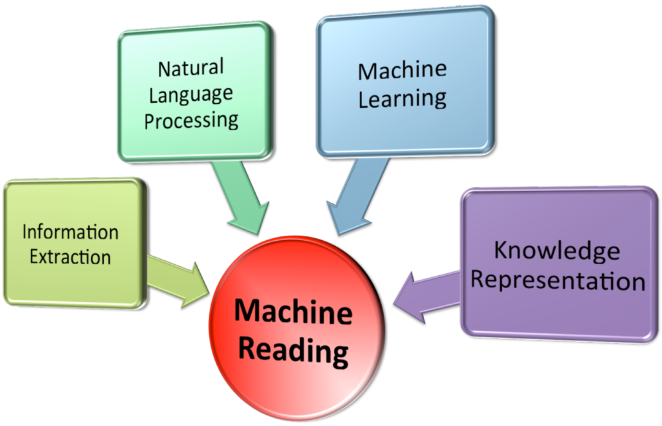
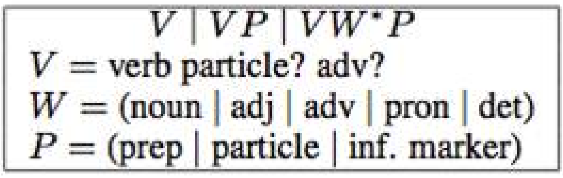


**Figure. 2**: Machine Reading and areas of influece [5].

A computer is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$ [6]. One Machine Learning (ML) approach example is the study of the Machine Reading (MR) problem: read and undestand texts (especifically in english, for example) using MR techniques is the main task $T$; the processing time in relation to the input text size, precision and coverage of the data set are the perfomance measures $P$; and, concept,



$V \mid VP \mid VW^{*}P$

$V =$ verb particle? adv?

$W =$ (noun | adj | adv | pron | det)

$P =$ (prep | particle | inf. marker)

**Figure. 3**: Patterns which drives TextRunner reading of phrases [11].

context and tuple instances in the format *Subject-Verb-Object (SVO)* are the training experience *E*.

MR faces the textual interpretation challenge, to undestand what was implied by the written text [7]. Briefly, MR aims to organize textual information for learning task. There are many distinct techniques which cover this task. As shown in figure 2, some main areas and techniques of influence for Machine Reading are Information Extraction, Natural Language Processing, Machine Learning and Knowlegde Representation.

Natural Language Processing (NPL), which aims that computers perform tasks envolving human language, like human-machine communication viability or simply useful text or speech processing, provides own methods for the relational information extraction (or identification) task. Lexical-syntatic data set is used to perform the relation extraction from a *corpus*, with prespecified relations, in experiment described by [8]. The linguistic view is very important to this task.

Extract factual and manipulable data to the machine level from a textual source, and store them in a knowledge representation structure is the main purpose of Information Extraction (IE). Traditionally, it requires human involvement in form of handmade extraction rules or manual labeling training examples [9]. YAGO (Yet Another Great Ontology) collects relational facts from structured data like Wikipedia infoboxes and categories to build an understandable KB from human knowledge [10].

Open IE (OIE) overcomes human need through relation phrases identification - phrases which denote sentences relations in english [9]. The automatic identification of relation phrases allows to extract random relations; it avoids the prespecified vocabulary restriction. The main advantage of OIE systems is efficient processing, as well as the capability to extract an ilimited number of relations [11].

TextRunner is the first system to consolidate the OIE pratical viability. Compared to KnowItAll (Web IE state-of-the-art system until then), TextRunner reached 33% of error reduction in a comparable extractions set [9]. TextRunner uses the patterns described in figure 3 to identify valid information phrases in open texts. Despite notable improvements, TextRunner shows some problems: relational tuples set full of non-informative and incoherent extractions [12].

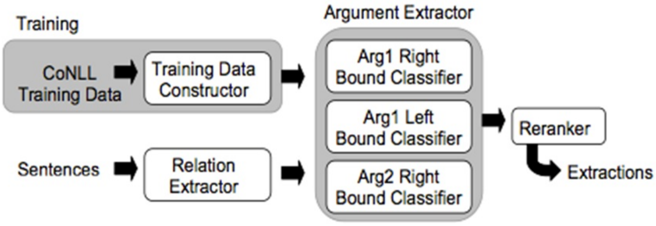

Training
CoNLL Training Data
Training Data Constructor
Argument Extractor
Arg1 Right Bound Classifier
Arg1 Left Bound Classifier
Arg2 Right Bound Classifier
Reranker
Extractions
Sentences
Relation Extractor

**Figure. 4**: ArgLearner architecture outline [11].

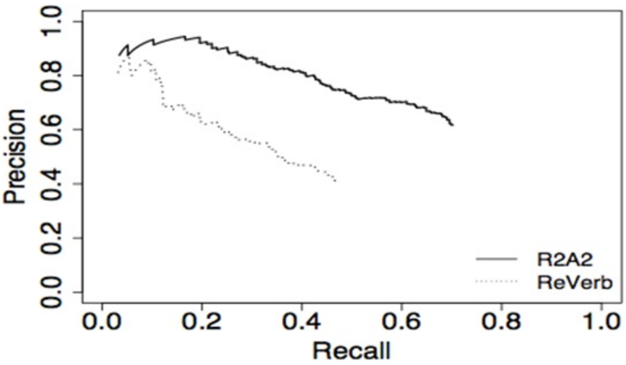


**Figure. 5**: Graphic representation of R2A2 superiority over simply ReVerb [11].

In order to overcome these problems, the OIE second generation system ReVerb has two simples constraints (syntatic and lexical). The syntatic constraint serves two purposes: (1) to eliminate incoherent extractions, and (2) to reduce non-informative extractions capturing relation phrases expressed by a verb-noun combination, including light verbs construction. With this, ReVerb more than doubles the area under the precision-coverage curve compared to TextRunner. In adition, more than 30% of ReVerb extractions are with 0.8 or more of precision, compared to virtually nothing of previous systems [11, 12].

Despite the good results, ReVerb presents invalid or incoherent relations. These errors are mostly related to the argument identification heuristic. R2A2 continues the evolution of KnowItAll systems: adds ArgLearner to ReVerb implementation, an argument identifier to better extract them.

ArgLearner have three limit classifiers: for the first argument, one to the left and one to the right; for the second argument, just one to the right (since its left limit is the relational phrase end) — figure 4. Thus, with ReVerb's relation extraction and ArgLearner's arguments extraction methodologies, R2A2 almost double the precision-coverage curve compared to simply ReVerb [11], as figure 5 outlines.

### B. *Read The Web project - NELL*

The main goal of Read The Web project is to build a never-ending machine learning system which improves constantly its ability to convert non-structured information into structured information. If succeed, it will result in a Knowledge Base with structured information which mirrors the Web content. Read The Web intends to formally define and prove that the newcomer never-ending learning paradigm is efficient and practicable. In this context, NELL emerges: a system which operates 24 hours a day and continuously improves its ability to extract facts from the Web.

NELL takes as input: an ontology which defines hundreds of categories (for example, person, drink, athlete, sport) and typed relations between these categories; a set from 10 to 20 positive examples for each category and relation; a collection

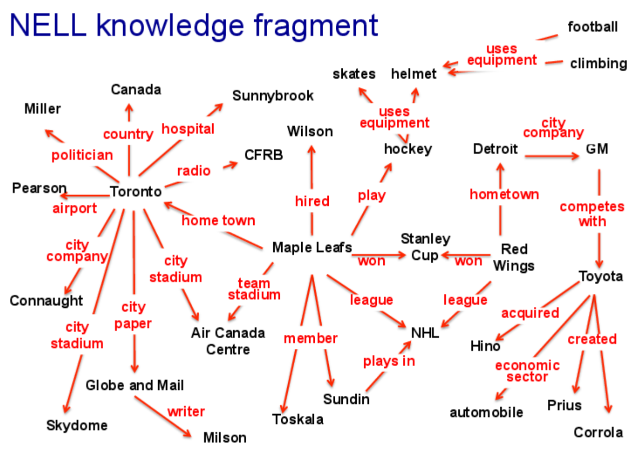


**Figure. 6**: Fragment of the 80 million beliefs NELL has read from the web [2].

of 500 million web pages from ClueWeb09 *corpus* as unlabeled data; and access to 100 thousand Google web searches. NELL has two diary tasks: (1) to extract new beliefs from the Web to populate its growing KB with category and relation instances from this ontology, and (2) learn to perform task 1 better today than yesterday [13]. NELL's working since January 2010. As a result, it has a continuously growing KB with more than 1.400.000 extracted beliefs. The KB and more information about NELL can be found at Read the Web page[1]. A fragment of the Knowlegde Representation of NELL is ilustrated by figure 6.

### C. NELL - Learning Components

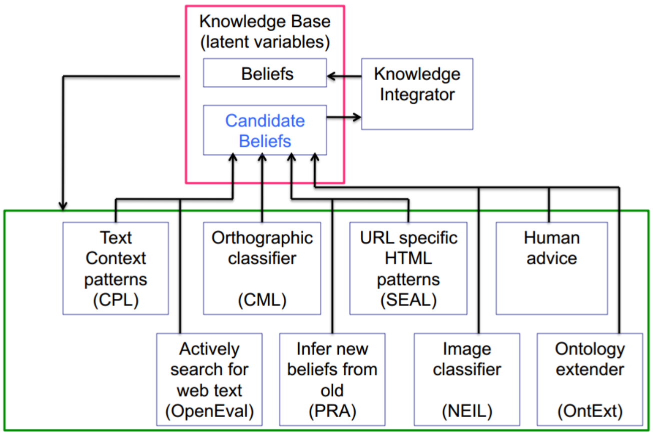

Knowledge Base (latent variables)
Beliefs
Candidate Beliefs
Knowledge Integrator
Text Context patterns (CPL)
Orthographic classifier (CML)
URL specific HTML patterns (SEAL)
Human advice
Actively search for web text (OpenEval)
Infer new beliefs from old (PRA)
Image classifier (NEIL)
Ontology extender (OntExt)

**Figure. 7**: NELL's software architecture [2].

NELL has some components focused especially in expand its Knowledge Base (KB). PRA (Path Ranking Algorithm) aims to populate relations, generate new instances with novel arguments and relations which already exists by going through graph paths and ranking them [13]. Prophet predicts new relations between existing nodes, induces inference rules and identifies incorrect links (wrong facts) from (NELL's KB) graph mining [14].

Figure 7 outlines NELL system architecture: briefly, beliefs and candidate beliefs feed up NELL's components to guide and/or help in their processments for better learning and knowledge base supervision and reflection. The knowledge integrator works on which candidate beliefs may turn out to be really beliefs.

NELL has a component which uses Relational Information Extraction from the Web as source to generate new relations: OntExt. The OntExt system combines features of traditional Relation Extraction with OIE to discover new relations between categories which are already known by NELL, and for which many instances already exist in the Knowledge Base [3].

#### 1) OntExt

uses OIE techniques to generate new relations: the focus of the approach is to use redundant Web information: relational facts which are frequently stated in huge text corpus, with different context patterns. Thus, semantically similar context patterns are clustered together although there is the possibility of lexical dissimilarity between them. To record the number of co-occurrence between the contexts which links two categories, a matrix is used: initially, each cell corresponds to the number of instances pairs of categories in which both contexts co-occur (*Matrix(i, j)* value to contexts *i* and *j* - e.g. the sentences "Vioxx can cure Arthritis" and "Vioxx is a treatment for Arthritis" provide a case where the 2 contexts 'can cure' and 'is a treatment for' co-occur with an instance pair [Vioxx, Arthritis]); then, the matrix is normalized - each cell value is divided by the total count of its line. Higher weight is given to contexts which co-occur with only a few contexts, to promote less generic contexts. Below, OntExt's algorithm to generate new relations [3].

$$Matrix(i,j) = \frac{Matrix(i,j)}{\sum_{j=0}^{N} Matrix(i,j)} \tag{1}$$

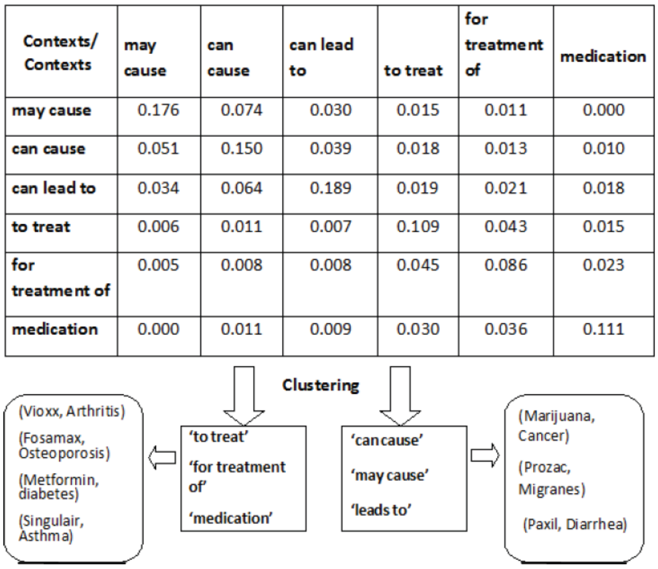
*Algorithm to generate relations*

```
Input: a pair of categories (C1, C2) and
set of sentences, each containing a pair
of known instances which belongs to C1
and C2, respectively.
Output: Relations and their seed
instances.
Steps:
1. From the input sentences, build a
context by context co-occurrence matrix.
The matrix is then normalized.
2.  Apply Weka \cite{weka} K-means
clustering on the matrix to cluster the
related contexts together. Each cluster
corresponds to a possible new relation
between the two input categories.
3.  Rank the known instance pairs
(belonging to C1, C2) fot each cluster
and take the top 50 as seed instances
for the relation.
```

[1] NELL/Read The Web page: http://rtw.ml.cmu.edu.

(Algorithm from [3])

Consider the example presented in [3]: for the category pair <*drug, disease*>, contexts like "*to treat*", "*for treatment of*", "*medication*" have high co-occurrence value, because they have the same intention of relation, ("*drug -to treat- disease*"); thus, these contexts are clustured together. Similarly, "*can cause*", "*may cause*", "*can lead to*" (which denote the relation "*drug -can cause- disease*") have high co-ocurrence values too. Another cluster is built for these last contexts. The clustering of this co-occurrence matrix is ilustrated by figure 8



| Contexts/ Contexts | may cause | can cause | can lead to | to treat | for treatment of | medication |
|---|---|---|---|---|---|---|
| may cause | 0.176 | 0.074 | 0.030 | 0.015 | 0.011 | 0.000 |
| can cause | 0.051 | 0.150 | 0.039 | 0.018 | 0.013 | 0.010 |
| can lead to | 0.034 | 0.064 | 0.189 | 0.019 | 0.021 | 0.018 |
| to treat | 0.006 | 0.011 | 0.007 | 0.109 | 0.043 | 0.015 |
| for treatment of | 0.005 | 0.008 | 0.008 | 0.045 | 0.086 | 0.023 |
| medication | 0.000 | 0.011 | 0.009 | 0.030 | 0.036 | 0.111 |

**Figure. 8**: Example of co-occurrence matrix built by OntExt [3].

Each cluster is used to propose a new relation. The relation name is obtained by the centroid [2] of the cluster, so, the seed instances which co-occur with contexts corresponding to the cluster centroid or close to the centroid will be best representative of the relation. Then, the strength of each seed instance is inversely proportional to the standard deviation of the context from the centroid of the relation contexts cluster, and directly proportional to the number of times it co-occurs with the context [3]. The formula 2 indicates how to calculate the weigth of each seed instance $s$ (subject-object pair):

$$\sum_{c \,\in\, AP} \frac{Occ(c,\ s)}{1\ +\ sd(c)} \tag{2}$$

Where $AP$ is the cluster of context patterns for this relation; $Occ(c,\ s)$ is the number of times that the instance $s$ co-occurs with the context pattern $c$; $sd(c)$ is the standard deviation of the context from the centroid of the pattern cluster. The top 50 are chosen as initial seed instances for this proposed relation.

Even chosing the 50 best instances, more than the half of the generated relations are invalid. The main motives are: errors in instances of categories, semantic ambiguity, semantically incomplete sentences and non-logic relations. As the introduction of these invalid relations could adversily affect NELL's performance, a classifier is used to overcome this issue of classify semantically valid relations. Features and resources compose the classifier: normalized count of frequency of each category instance; distribuition of the extraction patterns; number of context patterns reached through the clustering of patterns for the relation; and, how especific the context pattern is for the relation.

In [3] experiments, OntExt obtained 71.6% of precision and 72,2% of coverage for valid relations; 76,5% o precision and 75,9% of coverage for invalid relations; and, 74,2% of precision and coverage for the weighted average.

Despite satisfactory results of OntExt, it is not integrated with NELL working system nowadays, because it generates just a few valid relations with a high perfomance cost.

## IV. Methodology - newOntExt

This work aims to turn OntExt's task feasible to NELL. In other words, to automatically discover new valid relevant relations with some additional new features described is this section.

The resultant system of this work is called newOntExt. A new implementation for ontology extension (based in OntExt [3]) which will be integrated in NELL's system.

The complete traditional approach used by newOntExt to ontology extension can be divided in the following steps. (The alternative approach of newOntExt is described in subsection IV-E). The first two steps are necessary only if the desired input is an open, unstructured text.

1. Corpus pre-processing: prepare the input *corpus* to be processed by the KnowItAll systems - adapts to expected input format.

2. Open Information Extraction: extract relational information from a given corpus - ReVerb [12] and its evolution R2A2 [11] extract relational facts from any open domain corpus.

3. Identify extractions with known category instances: to merge the extracted information with NELL's KB, extractions with relating category instances known by NELL are sought and recorded.

4. Identify co-occurrence relations between the same category instances: for example, in "A relation1 B" and "A relation2 B", relation1 and relation2 co-occur with A and B.

5. Build co-occurence matrices: build co-occurence matrices with normalized values[3] of co-occurrence between relations.

6. Matrices data clustering: Apply Weka K-means over the co-occurrence matrices data to cluster relations. For each generated cluster, a new possible relation.

7. Generate new relations: generate seed instances for each proposed relation. The seed instances which co-occur with contexts corresponding to the cluster centroid[4] or close to centroid will be best representative

[2] Relation in the center of the cluster, better placed, with higher score.

[3] Normalized values: co-occurrence values divided by the total count of co-occurrence in the matrix line.

[4] Relation in the center of the cluster, better situated, with higher score.

of the relation. So the strength of the seed instance is inversely proportional to the standard deviation of the context from the centroid of the relation contexts cluster; and, directly proportional to the number of times it co-occurs with the context [3].

The experiments described in section V have started at step 3, because they use as input pre-processed datasets (better described at subsection IV-B).

### A. *Open Information Extraction - Second Generation*

OntExt uses TextRunner to extract the relational information from the corpus with textual data. TextRunner is the first system to use Open Information Extraction (OIE) principles, an unsupervised approach of Machine Reading (MR) [9]. As described in section III-A, the evolution of Open Information Extraction (OIE) techniques are consolidated by the systems ReVerb [12] and R2A2 [11]. These systems perform the OIE task as described in step 2 in this section, so, the extractions manipulated by newOntExt are much more reliable, compared to extractions made by TextRunner (as shown in section III-A).

### B. *Computationaly elegant file structure to perform a quicker search through the relevant sentences*

The input corpora which is used as input to perform the ontology extension task is often available in a unique huge file, as ReVerb/R2A2 default output, RCE 1.1[5] and SVO (Subject-Verb-Object dataset)[6]. This corpora often are in tab-separated columns format:

$$Subject < tab > Verb < tab > Object \tag{3}$$

The subject can be called first argument; the verb, (pattern) context; and the object, second argument.
One of the main factors for high computational cost for the task is the sequential search for extractions with category instances already known by NELL, because the possible source copora are enormous. So, to improve this performance the extractions are split in several different files, which are stored in a three-level directory hierarchy. For each category instance present as a subject of at least one extraction is created a file (named by this subject) containing all extractions (each extraction containing verb, object and frequency count) with this subject. Thus, it is possible to seach directly the file with the relevant information (extractions) for each known category instance. Nevertheless, operational systems does not support that a directory had thousands or millions files in it. To solve this problem, the directories are organized as:

- Root directory: *Extractions/*;
- Subdirectories in root directory, each one named with an alphabet letter (for example, *Extractions/a/*, *Extractions/b/*, *Extractions/c/* etc.);
- Subsubdirectories: each subdirectory contains subsubdirectories with the alphabet letter (of the subdirectory which it belongs) with the alphabet letters appended (for example, *Extractions/a/aa*, *Extractions/ba/*, *Extractions/ab/*, *Extractions/ca/* etc.);
- Subsubsubdirectories: the same subdivision (based on the alphabet letters) is done again (for example, *Extractions/a/ab/aba*, *Extractions/c/ca/can*, etc.);
- For each subject of the extractions present in NELL's knowledge base is created a file named by this subject, in the path (three-levels in the root directory) correspondent to its prefix. For example, for the subject "banana", the path and file would be: *Extractions/b/ba/ban/banana.txt*.

Consider that newOntExt is going through every instance which belongs to "fruit" category searching for extractions with fruits as subject. So, for example, the category instance "banana", the system verifies if the file *Extractions/b/ba/ban/banana.txt* exists; if so, these extractions are taken into account to build the co-occurrence matrices; if not, go to the next fruit instance, and so on.

### C. *Divide and conquer method for categories of interest*

In the first experiments of newOntExt (describred in section V), the Knowledge Base used has 240 categories, with 2.240.651 total instances. The corpora used as input for these experiments have millions extractions (RCE 1.1 has 14 million; SVO dataset, 604 million). So, to find all the extractions from one of these corpora containing known instances by NELL (traditional approach) is a great challenge. Equation 4 presents the combination formula for all 4.597.174 instances of 241 categories from NELL's Knowledge Base at iteration 656; Equation 5 shows the total number of comparisons using the SVO corpus.

$$\binom{4597174}{2} = \frac{4597174!}{2! \times 45971722!} = 10567002094551 \approxeq 1{\times}10^{13}. \tag{4}$$

$$10567002094551 \times 604934719 \approxeq 6,4 \times 10^{21} \tag{5}$$

To overcome this challenge, a divide and conquer method is proposed to reduce the knowledge domain of work. With this approach, experiments are carried out with categories group(s) to focus in certain areas of interest. Equation 6 shows the total number of combinations between instances related to the category "animal"; Equation 7 presents the total number of comparisons considering the "animal" subset and the input corpus partitioned by every 26 millions tuples. The experiments are described in section V.

$$\binom{102765}{2} = \frac{102765!}{2! \times 102763!} = 5280271230 \approxeq 5,3 \times 10^9. \tag{6}$$

$$5280271230 \times 26000000 \approxeq 1,37 \times 10^{17}. \tag{7}$$

[5] ReVerb ClueWeb Extractions 1.1: a dataset with about 15 million filtered extractions made by ReVerb system from the english web pages of ClueWeb09 corpus - a collection of about 1 billion web pages in ten languages

[6] About 604 million extractions from ClueWeb09.

### *D. Classification of valid relation and results evaluation*

To classify the generated relations, it is used the same approach described and used by OntExt [3]. More specifically, a generated relation is considered incorrect (by human) if there is:

1. Semantic ambiguity: if the instances belonging to one or both cateogies involved int the relation are ambiguous and don't make sense in the relation context. For example, *insect-such_as-animal* (relation extracted by OntExt).

2. Error in instance classification: if one or both instances involved are wrongly associated to the respective category(ies). For example, *animal-using-animal* (relation extracted by OntExt).

3. Semantically incomplete information: if the relation needs more information to make semantic sense. For example: *arthropod-can_be_use_instead_of-mollusk* (relation extracted by newOntExt).

4. Incorrect logic: if the relation simply don't make logic sense. For example: *animal-be_a_lovely_alternative_to-mollusk*, (relation extracted by newOntExt).

After this classification between valid and invalid relations, the calculation is made to obtain the Precision ($P$) through the formula:

$$P = \frac{True\ Positive\ Results}{(True\ Positive\ Results + False\ Positive\ Results)} \tag{8}$$

which, for this work, is equals to

$$P = \frac{VR}{(VR + IR)} \tag{9}$$

where $RV$ is the number of valid relations and, $RI$, the number of invalid relations.

With this result and with the total count of generated relations (valid and invalid), the comparison with OntExt can be done[7].

### *E. Prophet And newOntExt - collaboration to generate new relations*

Prophet uses the characteristics of the graph which represents NELL's Knowledge Base as unique input to perform some tasks, among them, collect new possible relations between category pairs. These possible new relations are not named yet, in other words, they does not have a pattern context which represents them well. For each relation collected, Prophet supplies the category pair involved in the relation (for example, "sport" as domain category - subjects must belong to "sport", and "sportsleague" as target category - objects must belong to "sportsleague") and the instances pairs found for this relation (subject-object pair).

So, newOntExt searches for sentences in a huge corpus containing these instances or categories from these relations found by Prophet, in order to name the relations found. For each sentence encountered, the pattern context which links subject and object is considered to built the co-occurrence matrices up and then apply the clustering for each relation. Each relation has 1 or 2 clusters, depending on the number of contexts found conecting these two categories. The centroid of the cluster is the name of the relation, such as the traditional approach.

Initially, the methodology to name relations found by Prophet was tested with all possible combinations between subjects and objects (experiment describred in V). This experiment is important because of the first results for new OntExt, although the methodology with instances pairs (not all the possible combinations) is more appropriated for this naming task.

So, for present (running) and future experiments, newOntExt has two possible methodologies to name each relation proposed by Prophet: (a) by category instances pair (subject and object of the relations): newOntExt searches for sentences in huge corpus which have the instances pairs found by Prophet. And (b) by the category pair: newOntExt searches for sentences which has the subject belonging to the domain category, and, the object, to the range category.

## V. Experiments

As mentioned in Section IV, ontology extension strategies based in redundant information require processing a big amount of information. As the time needed to obtain results about the traditional methodology as a whole using the whole Knowledge Base (KB) is set to be prohibited, alternative strategies must be adopted by NELL to obtain practical results which can contribute effectively to the sequence of the never-ending learning. Thus, the experiments described in this section also follow these alternative strategies (which present themselves feasible and appropriate for practical usage of NELL system).

For the described experiments, it is considered NELL's KB till iteration 656. This ontology was chosen because it is robust enough to these experiments. Currently, NELL's KB is available till iteration 909. The more recent is the KB (i.e, the bigger the iteration number), the more reliable and full of examples it is — it has a greater number of beliefs; however, the longer it takes to perform the task.

Descriptions in respect of each performed experiment and respective result analysis are shown in the following subsections.

### *A. Experiments with tradicional methodology and reduced scope*

The experiments described in the Subsections V-A.1, V-A.2 and V-A.3 focus in thematic subsets of knowledge. These experiments have as input the corpus Reverb ClueWeb Extractions (RCE) 1.1. The table 1 presents a general summary of the experiments with this methodology and strategy: more than half of the generated results are candidates to become new knowledge beliefs.

[7] The calculus of the recall is not applied in this work because it involves false negatives — newOntExt does not generate relations with previous intention of them being invalid or negative; the focus is only in positive relations, which can aggregate knowledge to the ontology.

| | | |
|---|---|---|
| Possible beliefs | 22 | 40,74% |
| Incorrect relations | 32 | 59,26% |
| Total of generated relations | 54 | 100% |

*Table 1*: Summary of the generated relations with experiments with subset of categories.

| | | |
|---|---|---|
| Possible beliefs | 15 | 50% |
| Incorrect relations | 15 | 50% |
| Total of generated relations | 30 | 100% |

*Table 2*: Summary of the generated relations with subset of categories related to animal.

$$P_{\text{g}} = \frac{RV}{RV + RI} = \frac{22}{22 + 32} \approxeq 41\%. \qquad (10)$$

### 1) With subset of categories related to animal

In this experiment with focus on the subset of knowledge categories related to animal are considered the following categories: animal, mollusk, insect, reptile, mammal and arthropod. As results, there is a total of 30 generated relations, 13 of them considered correct and 17 incorrect.

One example of correct relation is: “*arthropod-can_be_very_irritating_to-mammal*”), as it is the case of mosquitoes with humans. As real example generated by newOntExt, there is the pair {*“flea”, “dog”*}, which indicates that the flea can be irritating to a dog.

On the other side, an example of incorrect generated knowledge is: “*arthropod-can_be_use_instead_of-mollusk*”) — an arthropod has the possibility to be used instead of a mollusk for determined application, however, this information is missing.

$$P_{\text{a}} = \frac{VR}{VR+IR} = \frac{15}{15 + 15} = 50\%. \qquad (11)$$

### 2) With subset of categories related to civil construction

The considered categories for the subset of interest related to civil construction are: construction resource, construction material, tourist attraction and city. As the Table 3 describes, of a total of 8 generatade results, 2 are possible beliefs and 6 are incorrect.

As an example of correct result, consider the result: “*attraction-be_fall_in-city*”, which can mean that the attraction is located in the city, what is semantically and logically valid.

The majority of the relations are negative or incorrect result, because of wrongly classified instances — instances indicated to belong to a category which they don’t belong indeed. For example, the relation “*city-be_the_city_of-buildingmaterial*”) could be considered as semantically and logically valid, if the generated instances were not wrong: subjects which are not cities and/or objects which are not construction materials. This indicates an alert to supervision and correction of the Knowledge Base (KB) it this graph area.

$$P_{\text{c}} = \frac{VR}{VR+IR} = \frac{2}{2 + 6} = 25\%. \qquad (12)$$

| | | |
|---|---|---|
| Possible beliefs | 2 | 25% |
| Incorrect relations | 6 | 75% |
| Total of generated relations | 8 | 100% |

*Table 3*: Summary of the generated relations with subset of categories related to civil construction.

| | | |
|---|---|---|
| Possible beliefs | 5 | 31,25% |
| Incorrect relations | 11 | 68,75% |
| Total of generated relations | 16 | 100% |

*Table 4*: Summary of the generated relations with subset of categories related to sport.

### 3) With subset of categories related to sport

For this experiment with the subset of categories related to sport, the considered categories are: sports league, sport, athlete and sports team.

An example of positive relation with sports is: (“*athlete-fly_out_to-sportsteamposition*”) which can mean the case of an athlete move fast to another position (from defense to offense, for instance).

As an incorrect result: “*athlete-can_be_play_at-sport*”, which has not a semantically and logically full sense.

$$P_{\text{e}} = \frac{VR}{VR+IR} = \frac{5}{5 + 11} = 31,25\%. \qquad (13)$$

## B. Comparaes com OntExt

For comparative effects, from the total of 781 new relations generated by OntExt [3], filters are applied according to the subsets of categories in focus in this work: a subset of categories related to animal, another related to civil construction and the last related to sport. Thus, in this Subsection, the number of generated relations for this subsets by each system (OntExt and newOntExt) are compared.

For the three subsets of categories in focus, OntExt generated 10 relations, all of them judged as incorrect by OntExt itself. On the other side, newOntExt generated 54 relations, being 22 of them correct. Thus, even with a relatively low precision (41%), newOntExt can generate knowledge in areas of the graph which OntExt could not.

This low precision is completely consistent with the Never-Ending Learning (NEL). Systems based on NEL need reliable beliefs to not have noise and concept drift. So, these systems can evolve in a slow way yet reliable and consistent. Which leads to the conclusion that newOntExt can contribute to the learning of NELL. And, when the precision is very low, the processment indicates a necessary supervision in the subset of categories in focus.

| | OntExt | newOntExt |
|---|---|---|
| Incorrect relations | 10 | 32 |
| Correct relations | 0 | 22 |
| Total of generated relations | 10 | 54 |
| Precision | 0% | 41% |

*Table 5*: Comparative summary of generated relations by OntExt and by newOntExt for the subsets of categories related to animal, civil construction and sport.

### C. Naming Prophet relations

We present here experiments using newOntExt after Prophet discovered relations between categories not named yet. From all relations found by Prophet, we selected the top ranked relations (ranked by Prophet) to choose the first 20 best ranked and valid relations according to this classification (very similar to the classification shown in Subsection IV-D):

1. Invalid due to semantic ambiguity. The instances belonging to one or both predicates in this relation, are ambiguous and do not come in the context of the predicate.

2. Invalid due to error in instance classification - The instances corresponding to either or both predicates in this relation are mistakes;

3. Semantically Information incomplete: Here there is no ambiguity. But the relation needs more information to make semantic sense;

4. Illogical relations;

5. The symmetric is valid but not the original;

6. Relation not found in Prophet output file;

7. At least 1 category does not exist in NELL's KB.

Most of the new relation seeds are incorrect due to ambiguity of category instances and instances which does not really belong to the category assigned. Some examples of incorrect *(instance, category)* pairs are: (water, sport), (strength, convention), (photos, park), (page, arthropod), (lentil, musicfestival), (center, visualartform), (resources, economicsector), (edit, monument), (third_party, politicalparty), (zero, food), (home, athlete), (students, sportsteam).

From these 20 best valid relations, newOntExt generated a relation name to 17 of them. For each of these 17 new relations, 2 clusters were found to put together seed instances with the same sense. Besides incorrect *(instance, category)* seed pairs, 9 relations are logically correct and makes complete sense as (*category1-verb_phrase-category2* format):

*cognitiveactions-can_spill_into-park*,
*cognitiveactions-started_on-visualartform*,
*sportsleague-lodge_has_crowned-sportsteamposition*,
*economicsector-grown_with-musicfestival*,
*politicalparty-makes-musicfestival*,
*athlete-infringes-food*,
*musicalbum-focuses_on-visualartform*,
*musicalbum-has-visualartform*,
*sportsteam-have_charged_on-convention*.

For the 20 worst identified relations, scored and ordered by Prophet itself, newOntExt did not found possible names for any of them, i.e., it did not colect enough data to perform the matrices clustering. Thus, it can be concluded that, for this experiment sample, the strategie did well to invalidate relations which were bad scored by Prophet, as the methodology of validation and naming proposes.

| | |
|---|---|
| Considered relations from Prophet | 20 |
| Total of possible clusters | 40 |
| Relations from Prophet invalidated by newOntExt | 3 |
| Generated clusters | 33 |
| Valid generated relations | 9 |
| Incorrectly name relations | 24 |
| Precision | 27,27% |

*Table 6*: Data about naming task of 20 best relations founded by Prophet.

## VI. Conclusion

The new paradigm of never-ending learning, in which NELL system is based, has many main characteristics which allows the constant improvement in the learning capacity of the system. One of the fundamental characteristics for which the never-ending learning can appropriately happen is the automatic and continuous extension of the ontology.

This paper presents an updated approach to continuous ontology extension, which is a great challenge to overcome. OntExt generated some new relations to the never-ending learning process of NELL, but most of them were invalid and the performance had a high computational cost.

The newOntExt system presents four new main contribuitions (in relation to OntExt). The first is relationed with the preprocessing of the corpus. OntExt has this stage intrinsically in its method, which adds in computacional cost. Furthermore, the preprocessing strategy has its base on principles defined in the first generation of Open Information Extraction systems [9]. Already newOntExt has its preprocessing method independent from the search for new relations, thus, different corpora can be used, besides taking as base the principles defined in the second generation of Open Information Extraction systems [11, 12], which turns the filter of used sentences more precise for the discovery of new relations.

The second contribuition is linked to the new algorithm, which was implemented with many new resources of preprocessing and code optimization. This contribuition is more of software engineering than of machine learning, but it has shown itself fundamental to enable newOntExt feasible as a component to be used by NELL in practice. For the subset of knowledge categories related to animal, for instance, 30 relations were generated, half of them are candidates to be considered as beliefs to the Knowledge Base. By the results obtained, it follows that the category instances of this area of the knowledge graph of NELL are defined well enough to obtain a reliable guide to generate a greater knowledge around this matter. But it can't be discarded the supervision in this ontology area, since 10 out of 15 invalid relations are classified as invalid due to Error in instance classification. For the subset of categories related to civi construction, there is a total of 8 generated relations, 6 incorrect and all due to instances wrongly classified in the respective categories. Thus, an area of the graph is identified as an area of knowledge which needs a supervision over its beliefs.

The alert to supervision is a new contribuition of this work. With some of the performed experiments, it can be stated that newOntExt (as it was with OntExt too) is sensible to noise, i.e., it depends on many examples which are reliable beliefs. Ambiguities and instances with bad association to

the respective categories prejudice the learning process. With base on this observation, the chosen approach was to adopt a strategy of using newOntExt also as an alert to corrections in the ontology, thus, colaborate in a more effective way with the self-supervision and self-reflection of the NELL system.
The last contribuition is linked to the integration with the graph based method: Prophet. Such integration allows the semantic information (name of relations) to be inserted in the graph based model, so, the process of creating new relations for NELL ontology become more robust.
Despite of less than half of the relations be adequate to alter NELL's Knowledge Base, when compared to OntExt (which has not generated any correct relation to the same subset of categories), newOntExt demonstrated (empirically) with these initial results that it can bring gains to the never-ending learning system.
The analysis of the experiments showed in section V indicates that the method is sensible to noise and depends on correct instance-category examples. Thus, this methodology alerts to a manual revision on the Knowledge Base to correct wrong instance-categories. For each experiment done, we must remove the incorrect instance-category pairs pointed by the method. So, this method helps in auto-supervision and auto-reflection of the learning system, in addition to create new relations to the Knowledge Base (KB).

## References

[1] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. H. Jr., and T. Mitchell, "Toward an architecture for never-ending language learning," in *Proceedings of the Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2010, pp. 1306–1313.

[2] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling, "Never-ending learning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.

[3] T. Mohamed, E. R. H. Jr., and T. M. Mitchell, "Discovering relations between noun categories." in *EMNLP*. ACL, 2011, pp. 1447–1455.

[4] D. T. Wijaya, P. P. Talukdar, and T. M. Mitchell, "Pidgin: ontology alignment using web text as interlingua." in *CIKM*, Q. He, A. Iyengar, W. Nejdl, J. Pei, and R. Rastogi, Eds. ACM, 2013, pp. 589–598.

[5] E. R. Hruschka Jr, "Machine Learning, Machine Reading and the Web," in *Tutorial presented at IBERAMIA 2012 - 13th Ibero-American Conference on AI*, Cartagena de Indias, Colombia, 2012.

[6] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[7] P. Norvig, "Inference in text understanding." in *AAAI Spring Symposium: Machine Reading*. AAAI, 2007, pp. 6–10.

[8] L. S. Taba and H. de Medeiros Caseli, "Automatic hyponymy identification from brazilian portuguese texts." in *PROPOR*, ser. Lecture Notes in Computer Science, H. de Medeiros Caseli, A. Villavicencio, A. J. S. Teixeira, and F. Perdigo, Eds., vol. 7243. Springer, 2012, pp. 186–192.

[9] M. Banko and O. Etzioni, "The Tradeoffs Between Open and Traditional Relation Extraction," in *Proceedings of ACL-08: HLT*. Columbus, Ohio, USA: Association for Computational Linguistics, June 2008, pp. 28–36.

[10] G. Weikum and M. Theobald, "From information to knowledge: harvesting entities and relationships from web sources." in *PODS*, J. Paredaens and D. V. Gucht, Eds. ACM, 2010, pp. 65–76.

[11] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam, "Open information extraction: The second generation." in *IJCAI*, T. Walsh, Ed. IJCAI/AAAI, 2011, pp. 3–10.

[12] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1535–1545.

[13] N. Lao, T. Mitchell, and 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*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 529–539.

[14] A. P. Appel and E. R. Hruschka Jr, "Prophet – A Link-Predictor to Learn New Rules on NELL," *2011 IEEE 11th International Conference on Data Mining Workshops*, pp. 917–924, Dec. 2011.

# Author Biographies

**Paulo Henrique Barchi** Born in Campinas - São Paulo, Brazil, on february 3rd, 1988. Bachelor in Computer Science (2010) by Federal University of São Carlos (UFSCar), São Carlos - São Paulo, Brazil. Master in Computer Science (2015) by Federal University of São Carlos (UFSCar), São Carlos - São Paulo, Brazil. Currently interested in Philosophy of Science, Technology and Society aspects.

**Estevam Rafael Hruschka Jr.** Bachelor in Computer Science (1994) by Londrina State University, Londrina - Paraná, Brazil. Master in Computer Science (1997) by University of Brasilia, Brasília - Federal District, Brazil. Ph.D. in Computer Systems (2003) by Federal University of Rio de Janeiro, Rio de Janeiro - Rio de Janeiro, Brazil. Post-Ph.D. in Computer Science (2010) by Carnegie Mellon University - EUA, Pittsburgh - Pennsylvania, USA. Mainly interested in Machine Learning.