Ontology supported system for searching evidence of wild animals trafficking in social network posts

Rafael da Silva Carrasco [1]
Alcione de Paiva Oliveira [2]
Jugurta Lisboa Filho [3]
Alexandra Moreira [4]

Resumo: O comércio ilegal de animais silvestres é uma das atividades criminais mais lucrativas da atualidade. No Brasil, a grande variedade de fauna nativa tem alimentado o mercado ilegal, o que gera sérias implicações ambientais e sociais. A luta contra o comércio ilegal de animais silvestres é crucial para ajudar a proteger os recursos naturais e evitar a disseminação de outras formas de crime. Esse tipo de comércio ilegal usa cada vez mais, a internet para realizar suas atividades. A fim de combater tal crime, um sistema automático de monitorização é essencial. No entanto, para realizar essa tarefa de forma eficaz, o sistema deve ser capaz de analisar as mensagens trocadas durante essa prática. Para isso, é necessário o conhecimento dos conceitos e relações que ocorrem nesse domínio. Este artigo apresenta um sistema multiagente apoiado por ontologia de domínio e frames semânticos para buscar evidências de comércio ilegal de animais silvestres. No artigo, é mostrado como o sistema pode ser usado na tarefa de rastreamento do comércio ilegal de animais silvestres, além de apresentar os resultados da aplicação do sistema em um pequeno *corpus*.

***Palavras-chave*:** Ontologia. Redes sociais. Semântica de frames. Tráfico de animal silvestre.

Abstract: *The illegal trade of wild animals is one of the most lucrative criminal activities. In Brazil, the wide variety of native wildlife has fed the illegal market, which has serious environmental and social implications. The fight against illegal trade of wild animals is crucial to help protect natural resources and preventing the spread of other forms of crime. This type of illegal trade has been making increasingly use of the Internet to carry out their activities. In order to fight against this criminal activity, an automatic monitoring system is essential. However, for the effective execution of this task, the monitoring system should be able to analyze the dialogues that are carried out during this activity. For this to occur one need to know the concepts and relationships that occur in this domain. This paper presents a multiagent system supported by domain ontology and semantic frames to seek evidence of illegal trade of wild animals. In the article it is shown how the system can be used in the task of tracking illegal wildlife trade also presents the results of applying the system in a small corpus.*

***Keywords*:** *Frame semantics. Ontology. Social network. Wild animal trafficking.*

1 Introduction

The wildlife traffic is amongst the most successful and lucrative criminal activities in the world [20]. In a 2008 report to the U.S. Congress [24], stated that *"Global trade in illegal wildlife is a growing illicit economy, estimated to be worth at least $5 billion and potentially in excess of $20 billion annually"*. Because of being generally labeled as a "minor crime", it runs rampant in several countries, especially in those which have a rich biodiversity,

[1]Curso de Mestrado em Ciência da Computação,Universidade Federal de Viçosa UFV - Brasil
{mestrecarrasco@gmail.com}
[2]{alcione@gmail.com}
[3]{jugurta@ufv.br}
[4]Universidade Federal de Juiz de Fora - Brasil
{xandramoreira@yahoo.com.br}

http://dx.doi.org/10.5335/rbca.2014.3140

as is the case of Brazil. The Brazilian Institute of the Environment and Renewable Natural Resources (IBAMA) stated that just in 2003 at least 12 million specimens have been illegally taken from Brazilian ecosystems [11]. Still according to them, it could well be 38 million. In fact, these criminal activities go along with other forms of traffic, such as valuable stones, weapons and drugs. Being highly regarded as one of the most pernicious activities in almost every modern city, it is quite impressive that drug trafficking is in fact being helped by wildlife traffic, and not the opposite.

In 2001, the *National Network of Combat Against Wildlife Traffic* (Renctas) has published a report [18] in which it analyzes, among other issues, challenges of the authorities in fighting wildlife traffic. One of the top issues is the monitoring of such activities in social networks in the Internet. This is astonishing, given the fact that the Internet was not widespread in Brazilian homes in 2001.

A quick visit to one social network, the Orkut, shows that the situation hasn't changed. Smugglers post ads of unlicensed animals, and the mail service is illegally used to deliver spiders, turtles, snakes and even iguanas. All of this is stated in plain sight, without masquerading by code words, abbreviations or any form of text deception.

Given this situation, this work describes a computer system that was built to monitor wild animal traffic negotiations in social networks. This system applies an ontology developed to this purpose that helps to understand the nature of the collected posts. The ontology helps to perform a matching between the sentences and the animal trafficking scene described by a semantic frame [6, 7] previously built [4]. This paper is organized as follows. Section 2 presents an outline about ontologies and concisely describes the domain ontology developed for wildlife traffic conducted over social networks. Section 3 presents the concept of semantic frames and frames developed for the system. Section 4 discusses the multiagent systems. Section 5 outlines the model proposed to monitor the traffic activities. Section 6 presents the module responsible for the analysis of the natural language. Section 7 presents the results achieved. Finally, section 8 presents the conclusions of the work.

2 The Ontology

This section discusses how an ontology was developed to be applied in a software system and its final structure. The main purpose of an ontology, as normally happens in Computer Science, is to formally describe the elements of a given domain and their relationships so that it can be shared and processed by automate systems. There are several ways to characterize and classify the different types of ontologies, but the distinction which is relevant for this work is the one that distinguishes top-level ontology from domain ontology. Domain ontologies focus on describing the elements from a particular domain. A domain ontology that aims more specifically on the taxonomic structure are referred to as lightweight ontologies [12]. On the other hand, top-level ontologies attempts to present an overview of the nature of the objects that make up the world. They are domain independent and are based on philosophical principles as the identity, uniqueness, etc. [9]. The alignment of a domain ontology with a top-level ontology can produce better structured ontology with better defined concepts, allowing a more natural merge with other domain ontologies. The ontology presented in this article was developed aiming to describe the elements of the domain of wild animals illegal trafficking held in websites and social networks. It was developed as a domain ontology, and then merged with the DOLCE ontology [3], in its light version, named DUL [8].

2.1 The Ontology Development

The specification of ontologies is not a trivial task. Difficult decisions regarding the scope and granularity of the ontology, for instance, have a tremendous impact on the outcome. Moreover, the philosophical principles underlining the task and the perspective from which the domain is seen will define the entire ontology class hierarchy. Without proper care, one could end up with an ontology which does not meet the proposed requirements, and therefore unsuitable for the role it should play. This fact may be detected only when the ontology is already in use, increasing considerably the cost of correction.

In order to support this work, several ontology development processes were analyzed. Some of these processes are summarized in [17], where they also presented a new process, developed from their previous analysis. This process is similar to a software development process, involving an initial stage of requirements analysis. Techniques such as brainstorming and interviews with domain experts are also employed. In the end, the authors

of the present paper concluded that the best strategy was to develop the ontology based on *corpus* evidence and afterward adjust it to a consolidated top-level ontology. The remainder of this section describes how this was done.

The basic criterion for selection of the concepts of the ontology was its occurrence in a *corpus* previously constructed for the domain. The corpus was constructed upon dialogues extracted from social networking sites dealing with the trading of wild animals. The ontology was expressed in OWL [14], using the Protégé editor (http://protege.stanford.edu/). After the domain ontology construction, it was adapted to fit in a top-level ontology to enable a better adjustment of its structure and future sharing. The top-level ontology selected was DUL [8].

During the ontology development, many changes were made. Entities were created, while others were replaced. No need to discuss each of the small structural changes, so this section will only present two versions of the ontology: Before the merging with DUL and after the merging. A serious conceptual error was to treat the classes *Deal*, *Proposal* and *Counter Proposals* as being subclasses of *Negotiation*. While certainly these items comprise a negotiation, they are not alone, a full Negotiation. In fact what exists is a part-whole relationship. Since this was the case, the classes *Deal*, *Proposal* and *Counter Proposal* now inherits from *Information Entit*y, which is a sibling of *Process*, the superclass of *Negotiation*. The use of the term *Animal* instead of *Nonhuman* was another correction done after the merger. *Nonhuman* encompassed all instances that do not belong to the class *Human* and would be more comprehensive then we intended. Thus, a message would also be a *Nonhuman* as a *Police action*. Other minor conceptual errors and some absences were corrected in the merge phase. Although the use of ontology DUL did not directly correct these issues, the necessity of having to adjust them to DUL caused the topology to be rethought by a different perspective. The final result is shown in Fig. 1. Some entities are duplicated, because OWL allows for multi-inheritance, and in these cases, they may appear under each superclass.

At the top level of the class hierarchy there is the class *Thing*, which is defined as being the standard OWL top level entity. The *Thing* class has just one subclass, the class *Entity*, which is the DUL top class. The *Entity* class is further divided in five subclasses (*Abstract*, *Object*, *Event*, *Quality* and *InformationEntity*) where two of them are noteworthy: *Object*, which represents any physical or abstract object that exists entirely in a given instant of time (an *endurant*), and *Event*, which includes all events. Events are entities whose parts do not occur together in a given time (a *perdurant*). A perdurant occurs in time, and can only be seen partially at any instant where it exists. An endurant, on the other hand may change over time, but at any instant is a complete entity. See [9] for a better explanation of the perdurant/endurant distinction. Scattered across the leaves of DUL ontology can be found the classes selected for the domain in question. For example, there are the *Buyer* and *Trafficker* class occurring as subclasses of the *Social Person* class which is a DUL class.

Besides the class hierarchy, the ontology also encompasses the relation between the classes and the attributes of the classes. Fig. 2 shows the relations hierarchy (Object property in OWL terminology). The segment of the hierarchy that is the most important to emphasize is the one that deals with the role of an entity in an event. One or more entities may participate in of an event and no event occurs without the action of at least one entity. This relationship is expressed by the object property *engagesIn*, which reflects the participation of an entity within an event. The subclasses of this property specify how such participation takes place, covering four possibilities: *forced participation* (e.g.: dogs in a dog fight tournaments), *event promotions* (e.g.: dog fight tournaments organizers), *conventional participation* (e.g.: the owners of the dogs in a dog fight tournaments and dog fight tournaments audience), and *repression* (e.g.: the police).

Once the ontology is created, we have now the types to be assigned to the lexical elements of sentences. These types will be used to establish the probability of a sentence being related to an event of trafficking. The event is described by a semantic frame.

3 Semantic frames

Semantic Frames are schematic scenes that are used to establish the meaning of each term in a sentence [6]. Each Frame has a well-defined context, where the semantics of events and roles involved in the discourse are fixed. Thus, the analysis of some words in a text may allow the deduction of the underlying context. The context set by a frame has a limited scope and is well defined. To describe a given domain a number of frames may be needed. Some terms of the sentence may be specific to a particular frame and its presence is enough to instantiate the frame.

Figura 1: Structure of the entities after the merge with DUL.

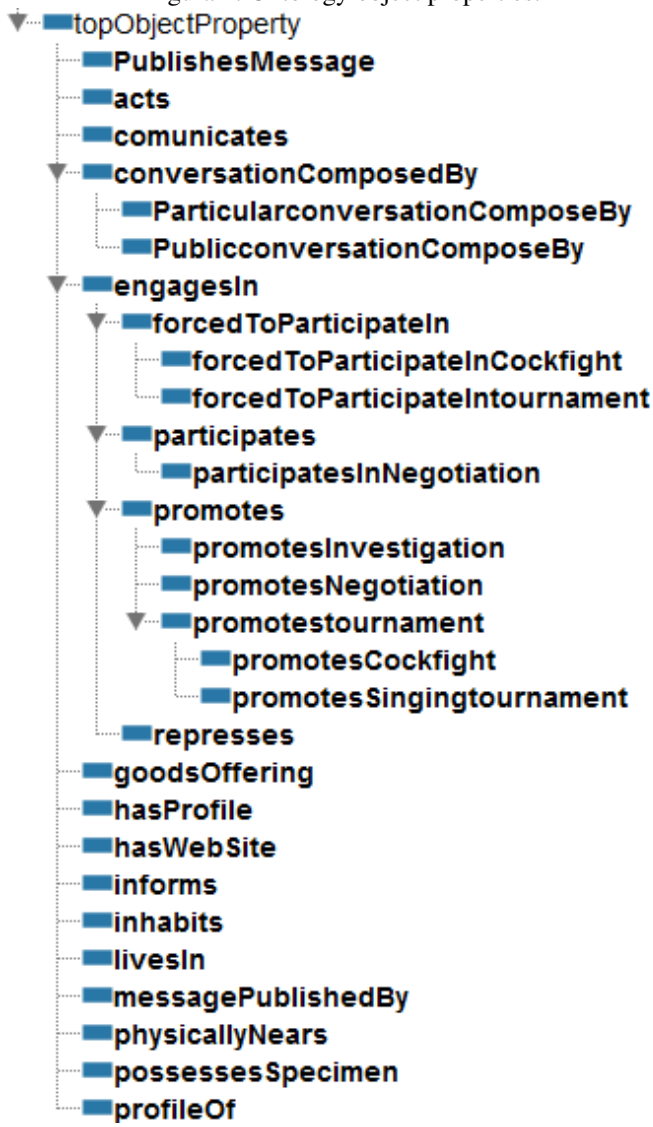
Interestingly, frames and ontologies may play a complementary role in natural language processing. While ontologies are responsible for formalizing the concepts in a domain, the different frames involved in the domain formalize the semantics of a speech, assisting in establishing and understanding the role of the entities participating in the scene. We have then, roughly speaking, the analysis of natural language could be done through two passes. At first tokens of the sentence would be annotated with its ontological type. From this analysis, another would be made where from this annotation would be calculated the probability of the sentence belonging to the scene described by the frame.

3.1 Obtaining the Frames

A semantic frame shall be supported by *corpus* evidence in order to establish the linguistic link with the semantics proposed by the scene described by the frame. Therefore, it is necessary first to obtain a *corpus* to provide the evidence needed for the frames creation. For this purpose it was created a *corpus* obtained from conversations on relationships sites that dealt with this kind of trade.

The animal traffic on the Internet is a type of commercial transaction, but has specific roles and peculiarities since it is an illegal trade. The goods is one or more animals and there are some elements that are not present in an ordinary commercial transaction, such as whether the animal is registered, the transfer to another communication

Figura 2: Ontology object properties.



environment to complete the transaction, the shipping method, the animal value when it is not registered, etc. The recognition of these elements is essential for the recognition of a trafficking scene. The analysis of the *corpus* sentences resulted in the frame shown in Fig. 3. In the representation of the frame was used the same notation of the Berkeley FrameNet , which is a lexical database for English based on the Semantic Frame Theory. The lexical units are shown with black background and frame elements are shown with colored background.

The types of transaction (*purchase* or *sale*), requires a perspective view at the scene of animal trafficking, which can result in a subdivision on the **Animal_Trafficking_Transaction** frame (Fig. 3).

In this work we are only interested in the perspective of the seller, since this is the one that can provide more evidence on the occurrence of trafficking. So we only designed the seller perspective *Illegal_Animal_Selling* frame, as shown in Fig. 4. This perspective is used in the system described in section 5.

Figura 3: Frame for Animal Trafficking Transaction.

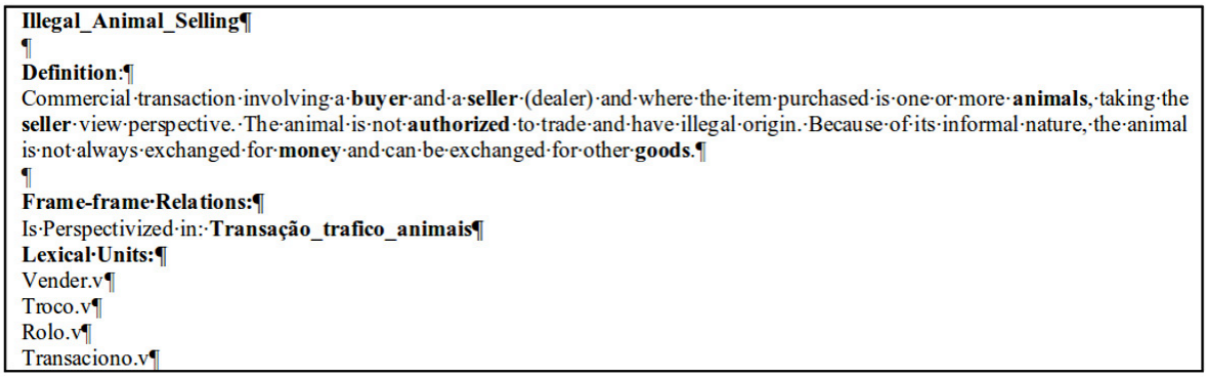
4 Multiagent systems

Multiagent Systems (MAS) is a paradigm for designing and implementing computer systems where the system is modularized into independent units called *agents*. The agents communicate with each other, working cooperatively and competitively, according to its own goals, allowing the system to meet their overall goal. This model was adopted during the modeling of the monitoring system, in order to reduce the complexity of system design. This paradigm is based on the concept of an agent, whose definition has not yet reached a consensus in the literature. One of these definitions is proposed in [21]: *An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators*. A MAS is a system composed by more than one agent [23]. Through the social interactions of its agents, a MAS expects to come to a solution for the problem it's trying to solve. The agents may not be aware of the system goal, and certainly will just try to get going its own business. This may conflict, resonate or have no relation at all with others agents goals. Nothing prevents two agents in the same MAS to compete with each other because of conflicting goals.

Since each agent is totally decoupled of each other and from the system as a whole, a MAS is inherently distributed [23], with the possibility of having a single instance executing in multiple computers. Each computer can have just a subset of every agent on the system, and yet all of them would be able to interact.

Given the vagueness of the agent definition, there is a great myriad of agents and MAS that can be deve-

Figura 4: Frame for illegal animal selling.



loped. This work is based on the cognitive MAS approach [23]. A cognitive MAS is based on elaborated agents, capable of sophisticated reasoning and communication. These agents interact directly, exchanging messages as if they were speaking to each other. Complex architectures were proposed for cognitive MAS, most of them based in psychology and anthropology. The cognitive MAS approach has several advantages that culminated on its choice and the strongest one was the elaborate nature of cognitive agents enable a high level view of some specific subset of the solution. It's possible to envision the system as an organization, with each agent working as a collaborator of the team, each one performing a specific duty through the desire to accomplish its individual goal.

5 Proposed system (WATES)

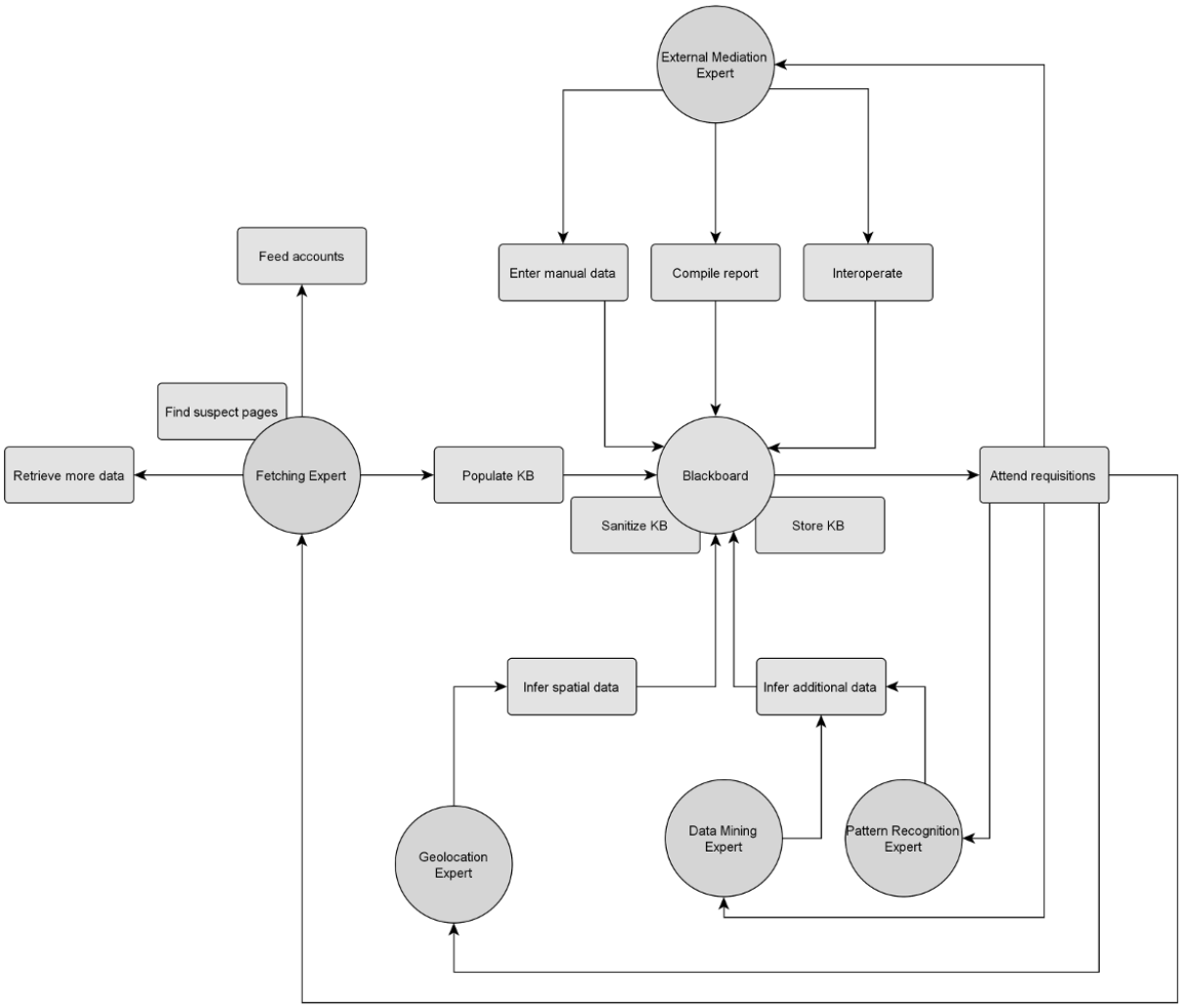
The ontology presented in this paper was developed in order to be added to a broader system of natural language processing. This system, named WATES (*Wild Animal Trafficking Evidence Seeker*) not only analyzes the messages of the social network, but also applies techniques of pattern recognition and data mining. To better understand the role of ontology in the system is important to understand the overall system architecture.

An *actor diagram*[22] was elaborated in order to obtain a better understand of the whole system. This simple model consists of circles representing each actor, and rounded rectangles representing goals. A goal that has one corner under an agent pertains to it. This means that this actor doesn't need any other to fulfill its goal. Its sole effort will suffice. In the cases where an actor needs the help of another one to accomplish some goal, an arrow points from the latter to the goal, and another leaves the goal to reach the former actor. The terms actor and agent mean the same thing in this context, and the former is preferred in this section because of the diagram name. Fig. 5 shows a central view of the obtained model.

In Fig. 5 there are two kinds of actors. The first kind is represented by the central *Blackboard* [10] actor. This agent is responsible for storing and delivering data to the system, centralizing all the exchange of information on it. The second kind of agents illustrated in 5 is the *Expert* agents. Each *Expert* can send data to the *Blackboard*. *Expert* agents can communicate with the *Blackboard*, but are forbidden to interact directly. This constraint makes communication simpler but, if the number of agents grows too much, it can become a severe bottleneck. Since the idea is that each *Expert* holds all expertise to act in itself, their interactions should be minimal. These architectural choices diminish the work involved in attaching a new *Expert* in the system. This happens mainly by lowering the coupling between the Expert actors. The *Expert* needs only to know how to reach and communicate with the *Blackboard*. Also, by sending its information to the *Blackboard*, each agent grants everyone else with its findings, with no need of direct communication. Each *Expert* can be actually a proxy between the *Blackboard* and another MAS, possibly developed to better implement a complex expertise.

The *Expert* agents illustrated in Fig. 5 are just an example of how the system can be organized. There is no constraint preventing that some of these gets suppressed in some implementation. Also other *Expert* agents can be included in the MAS at any time. Only the *Fetching Expert* and the *External Mediation Expert* are the exceptions

Figura 5: WATES system actor diagram.

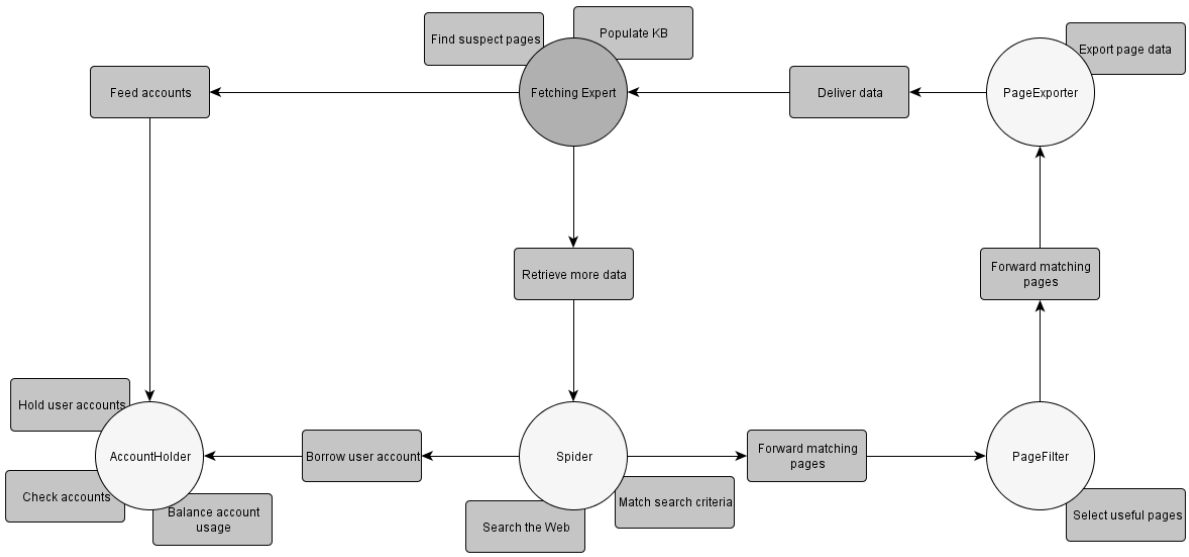


to this rule. The first one is needed in order to fetch new data from the Web, and the second to allow a direct communication with other systems. An agent which will be of great importance to the system, but that it is not yet implemented is the agent *Geolocation Expert*. It will be responsible for the attempt to establish the geographical location of the potential trafficker.

Given the importance of the *Fetching Expert*, this agent and its corresponding group will be addressed in the remainder of this section. The *Fetching Expert* task (locate and retrieve possible wildlife trafficking conversation over social networks) is central to the execution of every other *Expert* agents. If the *Fetching Expert* group fails to feed useful data into the *Blackboard*, there will be little to be done by others *Expert* agents. Fig. 6 presents an *Actor Diagram* model of the *Fetching Expert* group. In this group, The *Fetching Expert* acts just as a proxy between the *Blackboard* and the *Fetching* group, forwarding data in both ways. It gets profile accounts of the social networks that are listed in the *Blackboard*. It's expected that some accounts will be fed in the *Blackboard* by other *Experts* (specially the *External Mediator Expert*). Then the *Fetching Expert* should retrieve and use them to access the social networks, since most social networks only allows access to authenticated users. The agent *AccountHolder* is responsible for storing profile accounts. It should also respond to profile accounts requests of other agents, sending the appropriate accounts or no account at all, if it believes that the requisition comes from an untrusted source. The *Spider* agent receives data acquisition requests from the *Fetching Expert*. These requests may be to just search

some social network for evidences, or to follow specific conversation. In any way, the *Spider* may interact with the *AccountHolder* to find an appropriate account, if it is the case. After that it will search for new data, preferentially following the *Fetching Expert* directions. Any data that it finds relevant is forwarded to the *PageFilter* agent. The Spider should not be too selective, as this is the job of the *PageFilter*. Since each social network have a unique organization of its HTML code, spiders may specialize in only a fixed number of websites, refusing any request from the *Fetching Expert* to access a website that it is not capable to not understand.

Figura 6: The Fetching Expert group.



The *PageFilter* agent is responsible for filtering useful information from the pages selected by the Spiders agents. As is the case with the Spiders, *PageFilters* may act only on pages extracted from specific social networks, refusing to filter anything that it does not understand. To accomplish this task, the *PageFilter* should perform natural language analysis on the page content, discarding anything unrelated, such as navigational content and advertisements.

The filtered page content is converted to an internal canonical format and sent to the *Blackboard*. This data structure captures the posts, the threads, the authors and meta data, such as original URL and the access date. This job is performed by the *PageExporter*, which forwards this serialized data to the *Fetching Expert*, which, in its turn, sends it to the *Blackboard*.

Despite that each agent appears only once in Fig. 6 , while in execution the system is allowed to create any number of them. This enables the use of many *Spiders*, *PageFilters* and *PageExplorers*, each one constrained to probe just one social network or Web site.

6 The Natural Language Analysis

To verify the claiming that an ontology-based natural language analysis is able to detect wildlife traffic evidences, the natural language analysis module of the *PageFilter* agent was fully implemented. The module was built using Python 2.6.6 programming language, with the aid of NLTK (Natural Language Toolkit) version 2.0b9. The ontology was accessed using OWL API 3.2.4 and the reasoner HermiT 1.3.5. The module analyzes plain text files containing sentences in Brazilian Portuguese collected from a well-known social network, in 2011. About half of these posts negotiate animals in a way that was kind of suspect. The other half trade a wide range of goods, and none of them are animals. After the analysis, the system stores a HTML report with the evaluation of traffic for each sentence.

The Natural language module is composed of five main steps, united as a process by a Python script. Each step receives some files to process and outputs another file for each one of them. These outputs store the processing of the given step, and are used as input in the next step.

6.1 Step 1: Normalization

Before any processing takes place, it is necessary to fix orthographic errors, what is very common to occur in an informal communication medium. Each token is compared with a dictionary relating misspelled words and its correct writing. Not only common misspellings were put in this file, but also specific abbreviations used in traffic. For example, the bird Green-winged Saltator (*Saltator similis*), known in Brazil as *Trinca-Ferro* (iron cracker, in a free translation), is commonly called *tf*. These misspellings are placed in a table that serves as input and the normalization phase outputs a more correct version of the text.

The later stages assume the step 1 outputs as sufficiently correct, because there is no way to fix every possible misspelling. Also, there are instances where the misspelling simply cannot be detected. For example, consider another bird: the Rufous-bellied Thrush. This bird is known in Brazil as *Sabiá Laranjeira* (Orange-Tree Sabia), commonly called just *sabiá*. A recurrent misspelled form of *sabiá* is *sabia*, without the acute accent. Because the word *sabia* is the word in Portuguese to knew, the computer system can't state that it is a misspelling of *sabiá*. This is a kind of misspelling that cannot be fixed without a more powerful analysis, which is not in the current scope of this system.

6.2 Step 2: POS-Tagging

With the text fixed, it's viable to proceed with the analysis. The step 2 is where NLTK is used for its capabilities to POS-tag a token. "POS" stands for parts of speech, a grammatical evaluation which states the grammatical class of a given token. This step is in reality a POS-tagger that annotates each token, outputting the original text plus each token. The pair token and tag are stored as *token/tag*. For example a token *bird* would be tagged *bird/N*, where *N* stands for *noun*.

To better understand this step, it's useful to introduce some concepts first. The NLTK provides a number of POS-taggers, which one of them tags a token based on some criteria. A tagger may not be able to tag every word, but NLTK taggers are built in a way that allows composition. So a tagger may try to tag a word, and if it fails, it will pass the token to a fallback tagger, and so on. Last in this queue is a *DefaultTagger*, which tag anything as *N*.

Among the myriad of taggers that NLTK comprise, a common category is the *n-gram* taggers. An *n-gram* refers to the neighborhood of some token. A 1-*gram* (or *unigram*) is the given token. A 2-*gram* (or *bigram*) is the given token and the token before it. A 3-*gram* is the given token and the two tokens that precede it, and so on. The *n* can be arbitrarily large, but rarely is, since there is no use to tag a token based also on a token from the last sentence, as these two will not directly relate. Also, a large n will eventually only be able to tag a really specific combination of tokens that possibly will not happen in the entire text.

The *n-gram* taggers of NLTK based their analysis in a pos-tagged *corpus*. This *corpus* is statistically analyzed by the tagger, and will provide a basis for its tagging. The statistical nature of this analysis makes an *n-gram* tagger prone to error, since there is no way to guarantee that the same combination of tokens will really result in a given tag. This imprecision is inherent to natural languages.

The natural Language module employs a bigram tagger, that falls back to a unigram tagger, that falls back to a default tagger. The first two taggers base their analysis on the MAC-MORPHO *corpus* [1], which encompasses manually tagged sentences taken from popular newspaper in Brazil.

6.3 Step 3: Ontology Tagging

The POS tagging isn't of much use to spot wildlife traffic in conversations on social networks, since nothing can be extracted from the underling semantic of the conversation with just this information. In order to really understand what is happening, the step 3 tries to tag key tokens with some ontology class that may be related with the individual that it denotes. This process is aided by the POS-tags.

The output token of this step is the original POS-tagged token, plus a tag of the form [*individual:class*]. Tokens that were not tagged were marked so with an empty brackets, e.g. []. To determine the class of some token, a collection of criteria can be defined. Currently, the main one is to match a token with an individual that have it as a sign, if and only if the token was POS-tagged as *N*. The second criteria is to verify if the token is a class name, and then tag it as [:class]. For instance, if the ontology tagger finds a *Trinca-ferro* token it will tag it with the [*TrincaFerro: Animal*] tag. This task was extended by the use of the HermiT reasoner, mainly to discover the class of some individual looking over the ontology hierarchy.

6.4 Step 4: Value Attribution

Based on POS and ontological annotation the system identifies the occurrence of the frame elements. Then, the system checks whether in the sentences occurs values associated with FEs which generally occur in sentences that deal with wild animal traffic. For instance, if the animal is a wild animal and the shipping method is by post office, then there is a reasonable chance of being a case of trafficking. The values was selected manually from the *corpus* and is composed by following set: *Shipment via mail service, Low price, Not registered, wild animal, Commerce Transaction, use of slang*. The use of slang was included because of the perception that is often used in informal negotiations and authorized sellers, do not use this language resource.

The process is fairly straightforward, suppose a sentence with the following annotation:

Portuguese:	Vendo	Trinca-ferro	por	R$ 350
English:	*Selling*	*Trinca-ferro*	*for*	*R$ 350*
Ontology tags:	[action:sell]	[TrincaFerro:wildAnimal]		[Number:currency]
Frame elements		Animal		Good
Values	Commerce Transaction	Wild Animal		Low price

Then, since the instances of ontology shows that the average price of a TF is R$ 4,000.00, the system assigns the value “Low price” to the *good* frame element.

6.5 Step 5: Evaluation

The final step receives the results of the previous steps and verifies if there is a high probability for a given sentence to be an evidence of a wild animal illegal transaction. The occurrence of the frame elements and its values is used to connect the sentence to the semantic frame related with the traffic scene. The system, based on the probability of its occurrence, establishes the pertinence of the sentence in the trafficking scene.

The conditional probability that relates the occurrence of a value of a frame element (*vfe*) with the probability of traffic event (*P[te|vfe]*) was obtained using a training corpus of 45 sentences scrutinized by the WEKA[5] data mining software. These sentences were taken from the social network. It was used the WEKA Bayes net classifier to extract the probabilities. This technique is based in the one described by [15] and [16]. More details can be found in those works.

The criteria used to select the sentences to compose the corpus were: sentences with an intelligible argument structure and related with the offering of animals. Sentences related to the purchase or debates about animals were discarded. The table 1 shows the conditional probability obtained by the data mining software to the detection of illegal animal traffic scene, legal transaction scene or undefined, through the occurrence of the frame elements. Fig. 7 summarizes the system steps and it’s input/output.

7 Results Analysis

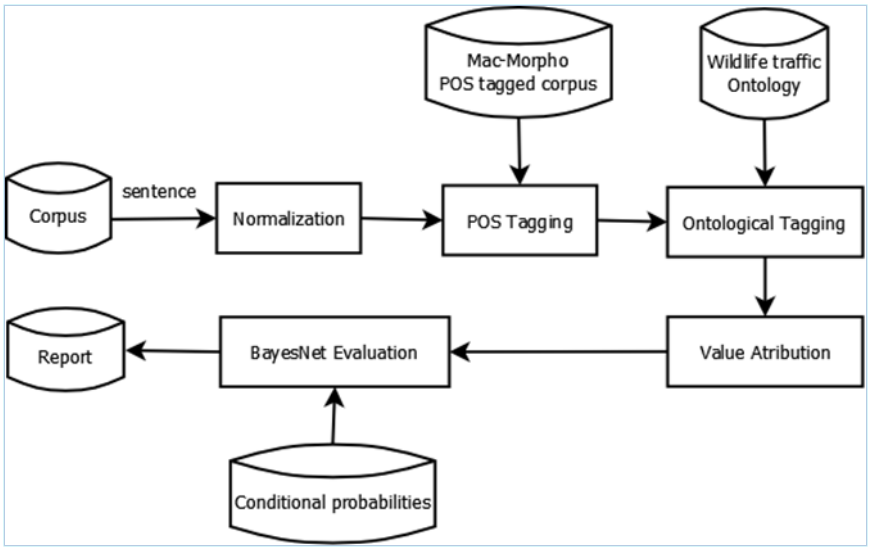
After training the network a test was carried out on a corpus composed by 63 sentences. The sentences were also taken from the social network. The sentences were extracted manually, however, the system foresees a specialized agent for the withdrawal of sentences in each site. These sentences were previously classified in three different types by a human expert: animal traffic, legal trade and undefined. The system was evaluated on the

[5] http://www.cs.waikato.ac.nz/ml/weka/

Tabela 1: Conditional probability for the frame elements values.

	Traffic	**Legal**	**Undef**
Shipment via mail service	0.719	0.3	0.011
Low price	0.781	0.1	0.278
Not registered	0.156	0.1	0.011
wild animal	0.969	0.1	0.922
Commerce Transaction	0.906	0.9	0.989
use of slang	0.156	0.1	0.122

Figura 7: Overview of the process steps.



basis of correct classification of these sentences compared with the specialist. The results are shown below. The first listing shows the network parameters and attributes listed, and the statistical results generated by the Bayesian network.

```
=== Classifier model (full training set) ===

Bayes Network Classifier
not using ADTree
#attributes=7 #classindex=6
Network structure (nodes followed by parents)
mail(2): frameTraffic
lowPrice(2): frameTraffic
unregistered(2): frameTraffic
wildAnimal(2): frameTraffic
Transaction(2): frameTraffic
InformalLang(2): frameTraffic
frameTraffic(3):
LogScore Bayes: -169.9823000776878
LogScore BDeu: -194.56037902488936
LogScore MDL: -206.03297377248649
LogScore ENTROPY: -164.60162650857117
LogScore AIC: -184.60162650857114

Time taken to build model: 0.01 seconds

=== Summary ===
```

```
Correctly Classified Instances          56               88.8889 %
Incorrectly Classified Instances         7               11.1111 %
Kappa statistic                          0.7166
Mean absolute error                      0.1051
Root mean squared error                  0.2243
Relative absolute error                 34.1675 %
Root relative squared error             57.7929 %
Total Number of Instances               63
```

The listing below shows the results per sentence. The *actual* column indicates the value assigned by the human analyst and the *predicted* column shows the value that the network estimated. The cross in the error column indicates that the predicted value did not match the expected value.

```
=== Predictions ontest split===

inst#,     actual, predicted, error, probability distribution
     1    1:undef    1:undef          *0.963  0.03   0.006
     2    1:undef    1:undef          *0.969  0.023  0.008
     3    1:undef    1:undef          *0.969  0.023  0.008
     4    1:undef    1:undef          *0.963  0.03   0.006
     5    1:undef    1:undef          *0.963  0.03   0.006
     6  2:traffic    1:undef      +   *0.818  0.18   0.002
     7  2:traffic    1:undef      +   *0.679  0.265  0.055
     8    1:undef    1:undef          *0.969  0.023  0.008
     9    1:undef    1:undef          *0.969  0.023  0.008
    10    1:undef    1:undef          *0.818  0.18   0.002
    11    1:undef    1:undef          *0.969  0.023  0.008
    12    1:undef    1:undef          *0.818  0.18   0.002
    13    1:undef    1:undef          *0.969  0.023  0.008
    14    1:undef    1:undef          *0.818  0.18   0.002
    15  2:traffic  2:traffic           0.02   *0.979  0.002
    16    1:undef    1:undef          *0.818  0.18   0.002
    17    1:undef    1:undef          *0.818  0.18   0.002
    18  2:traffic  2:traffic           0.002  *0.997  0.001
    19  2:traffic  2:traffic           0.02   *0.979  0.002
    20  2:traffic  2:traffic           0.02   *0.979  0.002
    21  2:traffic  2:traffic           0.001  *0.998  0.001
    22    1:undef    1:undef          *0.969  0.023  0.008
    23    1:undef    1:undef          *0.969  0.023  0.008
    24    1:undef    1:undef          *0.818  0.18   0.002
    25    1:undef    1:undef          *0.969  0.023  0.008
    26    1:undef    1:undef          *0.969  0.023  0.008
    27    1:undef    1:undef          *0.818  0.18   0.002
    28    1:undef    1:undef          *0.818  0.18   0.002
    29    1:undef    1:undef          *0.969  0.023  0.008
    30  2:traffic  2:traffic           0.149  *0.804  0.047
    31    1:undef    1:undef          *0.969  0.023  0.008
    32    1:undef    1:undef          *0.969  0.023  0.008
    33    1:undef    1:undef          *0.818  0.18   0.002
    34  2:traffic    1:undef      +   *0.969  0.023  0.008
    35    1:undef    1:undef          *0.969  0.023  0.008
    36    1:undef    1:undef          *0.969  0.023  0.008
    37    1:undef    1:undef          *0.969  0.023  0.008
    38    1:undef    1:undef          *0.969  0.023  0.008
    39    1:undef    1:undef          *0.53   0.005  0.465
    40    3:legal    1:undef      +   *0.53   0.005  0.465
    41    3:legal    1:undef      +   *0.53   0.005  0.465
    42    3:legal    1:undef      +   *0.53   0.005  0.465
    43    1:undef    1:undef          *0.53   0.005  0.465
    44    3:legal    3:legal           0.027  0.056 *0.916
    45    1:undef    1:undef          *0.969  0.023  0.008
    46    1:undef    1:undef          *0.969  0.023  0.008
    47    1:undef    1:undef          *0.969  0.023  0.008
    48  2:traffic  2:traffic           0.02   *0.979  0.002
    49    1:undef    1:undef          *0.969  0.023  0.008
    50    1:undef    1:undef          *0.969  0.023  0.008
    51  2:traffic  2:traffic           0.02   *0.979  0.002
```

```
52  2:traffic  2:traffic          0.02  *0.979  0.002
53    1:undef    1:undef        *0.818   0.18   0.002
54    1:undef    1:undef        *0.818   0.18   0.002
55    1:undef    1:undef        *0.969  0.023   0.008
56    1:undef    1:undef        *0.53   0.005   0.465
57  2:traffic  2:traffic          0.02  *0.979  0.002
58  2:traffic  2:traffic          0.02  *0.979  0.002
59    1:undef    1:undef        *0.963   0.03   0.006
60    1:undef    1:undef        *0.818   0.18   0.002
61    1:undef    1:undef        *0.969  0.023   0.008
62    1:undef    1:undef        *0.963   0.03   0.006
63  2:traffic    1:undef      +  *0.818   0.18   0.002
```

The score of 88.9% was quite impressive, but a few considerations are necessary. Most sentences were regarded as undefined by the specialist and all of these were classified correctly by the system. This should not be surprise since this classification stems from the failure to fit the sentence in the categories that really establish the semantic domain of the sentence. So, more tests will be needed with other *corpora* to confirm the accuracy of the system. Most classifications misleading occurred in the failure to frame a sentence judged by the expert as evidence of animal trafficking 4 on 15, which gives an accuracy of 73%. It's a good start, but there is room for improvement. An important point is that there were no false positives, i.e., no undefined or legal trade was classified as trafficking. Also, three sentences classified as legal trade by specialist was classified as undefined by the system, but this type of error is not harmful because there is no damage to mistakenly classify a sentence of legal trade as undefined.

Overall, these results are promising, and it is an evidence that natural language techniques, powered by domain-specific ontologies, can be applied to evaluate the association of a sentence to a scene, particularly to a wildlife traffic scene. It's even possible to replace the ontology with another domain ontology (like consumer satisfaction about some product), allowing the system to evaluate other domains without further changes.

8 Conclusions and Related Work

This paper presented a natural language analysis system based on ontology and frames aiming to find evidences of wild animal illegal trade in social networks conversations. The system was implemented as a Multiagent System in order to make it more adaptable to other sources of information and to better distribute the complexity of the job. The system is not fully operational, but the viability of the combination of the ontology with the frame can already be confirmed, once the natural language analysis module has been already implemented. The matching of the frame with the sentence, mediated by domain ontology allows us to see whether it may have been enunciated in a scene of animal trafficking. The system, once deployed, can help the authorities to fight this type of situation. The work being developed in parallel with the one presented in this article is the development of an agent with the purpose to establish of the geographical location of event.

The frame was obtained from a corpus developed especially for this purpose. The corpus was compiled from open conversations taken of social networking sites. From the sentences that make up the corpus were removed any reference that could identify the authors of the sentences. This was done because we have no legal power to investigate or indict citizens in any crime. The developed system should be used by people and agencies with that power. The scores achieved by the system in the previous tests hints that the combination of semantic frames and ontology is a viable choice to natural language analysis. Only the most basic features of ontologies and semantic frames were used by the computer system developed, so it is expected that even better results could be achieved if more features were explored.

Related works are difficult to find, since in general are developed by agencies to combat crime and intelligence agencies, which prefer to keep their work confidential. The FBI has the Carnivore system [13], which operates on private messages of Internet users, which leads to discussions of privacy violation. However, no systems were found in the literature that analyzes natural language sentences to detect illegal activities. As a somewhat related work one can cite the work in identifying associations of persons to documents [2], which analyze texts to identify people. More closely related is the work of Rodrigues et al. [19]. They propose a multiagent system to

follow-up the professional evolution of Graduates from their email posts. De Marchi et al. [5] described a Multiagent System (MAS) to monitor conversations in the chat tool of the Muzar Virtual Community (Comunidade Virtual do Muzar, CV-Muzar). They used conversational markers to analyze the conversations and did not used semantic frames.

Acknowledgment

We would like to thank the funding agencies FAPEMIG and CNPq for the financial support for this project.

Referências

[1] ALUÍSIO, S. M., PINHEIRO, G. M., MANFRIN, A. M., DE OLIVEIRA, L. H., GENOVES JR, L. C., AND TAGNIN, S. E. The lácio-web: Corpora and tools to advance brazilian portuguese language investigations and computational linguistic tools. In *LREC* (2004).

[2] BALOG, K., AND DE RIJKE, M. Associating people and documents. In *Advances in Information Retrieval*. Springer, 2008, pp. 296–308.

[3] BORGO, S., AND MASOLO, C. Ontological foundations of dolce. In *Theory and Applications of Ontology: Computer Applications*. Springer, 2010, pp. 279–295.

[4] CARRASCO, R. S., OLIVEIRA, A. P., LISBOA, J., MOREIRA, A., AND ARROYO, J. E. Linguistic structures to support an evidence tracking system for wildlife trafficking. CLEI 2011.

[5] DE MARCHI, A. C. B., RABELLO, R. D. S., ALBAN, A., CERBARO, V. A., AND BORDIGNON, J. M. Monitorando a comunicação na cv-muzar com o uso de agentes inteligentes. *Revista Brasileira de Computação Aplicada 2*, 1 (2010), 57–68.

[6] FILLMORE, C. J. Scenes-and-frames semantics. *Linguistic structures processing 59* (1977), 55–88.

[7] FILLMORE, C. J. Frame semantics. *Cognitive linguistics: Basic readings* (2006), 373–400.

[8] GANGEMI, A. Dolce+ dns ultralite, 2010.

[9] GANGEMI, A., GUARINO, N., MASOLO, C., AND OLTRAMARI, A. Sweetening wordnet with dolce. *AI magazine 24*, 3 (2003), 13.

[10] GARLAN, D., AND SHAW, M. An introduction to software architecture. *Advances in software engineering and knowledge engineering 1* (1993), 1–40.

[11] IBAMA. Procedimentos e consequências do tráfico. Tech. rep., IBAMA, 2003.

[12] MASOLO, C., BORGO, S., GANGEMI, A., GUARINO, N., OLTRAMARI, A., OLTRAMARI, R., SCHNEIDER, L., ISTC-CNR, L. P., AND HORROCKS, I. Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology.

[13] MCCARTHY, T. R. Don't fear carnivore: It won't devour individual privacy. *Mo. L. Rev. 66* (2001), 827.

[14] MCGUINNESS, D. L., VAN HARMELEN, F., ET AL. Owl web ontology language overview. *W3C recommendation 10*, 2004-03 (2004), 10.

[15] MOREIRA, A. *An Ontology Grounded Framework for Frames Detection*. Doctor thesis, Federal University of Juiz de Fora, Brazil, 2012.

[16] MOREIRA, A., AND SALOMÃO, M. M. M. Ontological analysis applied to frame development. *Alfa: Revista de Linguística (São José do Rio Preto) 56*, 2 (2012), 491–521.

[17] ÖHGREN, A., AND SANDKUHL, K. Towards a methodology for ontology development in small and medium-sized enterprises. In *IADIS AC* (2005), pp. 369–376.

[18] RENCTAS. Relatório renctas tráfico. Tech. rep., Renctas, 2001.

[19] RODRIGUES, D. F., OLIVEIRA, A. P., FILHO, J. L., AND MOREIRA, A. Semi-automatic follow-up of graduates. XXXI International Conference of the Chilean Computer Science Society (SCCC 2012).

[20] ROSEN, G. E., AND SMITH, K. F. Summarizing the evidence on the international trade in illegal wildlife. *EcoHealth 7*, 1 (2010), 24–32.

[21] RUSSELL, S. *Artificial Intelligence: A Modern Approach*. Prentice Hall, PA, 2009.

[22] SPANOUDAKIS, N., AND MORAITIS, P. The agent systems methodology (aseme): A preliminary report. In *Proc of the fifth European Workshop on Multi-Agent Systems, Hammamet, Tunisia, December* (2007), pp. 13–14.

[23] WOOLDRIDGE, M. *An Introduction to MultiAgent Systems*. John Wiley & Sons Inc, 2002.

[24] WYLER, L. S., AND SHEIKH, P. A. International illegal trade in wildlife: Threats and us policy. DTIC Document.