Global Journal of Computer Science and Technology: D
Neural & Artificial Intelligence
Volume 23 Issue 2 Version 1.0 Year 2023
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Guideline for Including Unperceivable Knowledge in a Universal Ontology Experimentation Field: Ontology Malagasy

By Randriambololona Nivo & Andriamanohisoa Hery Zo
Université d'Antananarivo

Abstract- This article suggests a cognitive approach to initiate the inclusion of unperceivable worlds into established ontologies. We qualify as unperceivable any world that escapes automatic data exploration because of the leak of sufficient documentation. Initially based on knowledge engineering, the approach aims in the long term at the automatic production of knowledge vectors that can be assimilated to existing corpora. It leverages a proven and extendable universal ontology and is experienced in the emerging world of Malagasy culture.

Keywords: knowledge, ontologie, malagasy.

GJCST-D Classification: LCC: Subject: QA 76.9.K45

Guideline for Including Unperceivable Knowledge in a Universal Ontology Experimentation Field: Ontology Malagasy

Randriambololona Nivo [α] & Andriamanohisoa Hery Zo [σ]

Abstract- This article suggests a cognitive approach to initiate the inclusion of *unperceivable* worlds into established ontologies. We qualify as *unperceivable* any world that escapes automatic data exploration because of the leak of sufficient documentation. Initially based on knowledge engineering, the approach aims in the longterm at the automatic production of knowledge vectors that can be assimilated to existing corpora. It leverages a proven and extendable universal ontology and is experienced in the emerging world of Malagasy culture.

Keywords: knowledge, ontologie, malagasy.

I. Motivation

The new paradigms of Artificial Intelligence (AI) rely on innovative techniques to solve problems that exceed human capacity and sometimes even the real needs of humanity. These techniques require the massive presence of data intended to train an agent on the resolution of a particular problem [3]. An agent is no longer supposed to reason, instead he is supposed to learn and to automatically or statistically exploit the data made available to him in order to deduce the decision to be taken or the prediction that seems the most relevant [7]. AI, endowed with data science, has become "almost" sovereign. Despite its power, there is a catch. What about unknown but emerging *worlds* for which no one has ever thought of collecting data but which today aspire to be part of the lot, to also be *perceivable* by AI? Nowadays, there are performant trained models like *ChatGPT* that have the ability to interact in a conversational way and that have response to all kinds of questions. Unfortunately, they are not yet trained on such emerging exotic worlds.

This article proposes a solution to circumvent the absence of massive data for a specific emerging world. The idea is to use classic AI techniques like knowledge engineering, while exploiting as far as possible what the state of the art offers in terms of ontology. As experimentation field, we opt for the unperceivable world of Malagasy culture and name the project *Tontolo Malagasy.* In fact, it is an abbreviation for Taxonomy and *Ontology Malagasy. Tontolo* means at the same time Universe. That is to say that we try to put the Malagasy Universe into an ontology and will let an agent respond spontaneously to the most important questions concerning the Malagasy language, news, (historical) facts, events or personalities of Madagascar. This is our manner to perpetuate the access to the own cultural identity for Malagasy youth.Latter is increasingly immersed in attractive cultures that certainly promote open-mindedness but that overwrite at the same time precious cultural heritage. Often, we are only aware of the value of our culturewhen it disappears. In the present work, we start by introducing the main concepts behind knowledge engineering. Then, we will present examples to illustrate their compilation on our project. Finally, we will talk about a state of the art ontology named YAGO which will serve as a reference.

II. Introduction to the Concept of *Knowledge*

Knowledge comes to us not only from the information that is conveyed by our perceptions, but also by natural language. Traditionally, natural language is considered as a language for representing knowledge, but today we see it more as a medium of communication. Even if it is highly expressive, hypotheses revealthe delicacy of its use.

- The Sapir-Whorf hypothesis (1956) claims that the language we speak greatly influences our understanding of the world. In the language *Guugu Yimithirr* of the Aborigines in Australia, for example, there are no words to express the relative direction (such as left, right, ahead, behind) but only the absolute direction via the use of the cardinal points (north, south, east, west). Therefore, Aborigines excel at navigating open terrain but would be less comfortable if told to turn left in a corridor.
- Words are sometimes associated with non-verbal representations. In some cultures, a concept may be completely absent from the language (like the example told above). In the Malagasy language, the verb *to be* does not exist. However, in a sentence supposed to *mean* it implicitly, anyone speaking Malagasy would understand it.

Since language influences the apprehension of the world, knowledge is not always neutral, objective or complete. This reinforces our assertion regarding the

Author α σ: Laboratoire de recherche en sciences cognitives et applications (LRSCA) Université d'Antananarivo.
e-mail: nivoran@gmail.com

existence of *unknown worlds*. Existing ontologies are mainly imbued with Western culture, so as immense as they are, they are never complete [2]. Before showing how to add new worlds into pre-established concepts, let us first see how to represent knowledge mathematically for the purpose of its automatic processing.

a) Knowledge Base

Our goal is to model a knowledge-based agent that can form representations of a real world. The task is not about actually representing everything in the world. New representations are derived from existing ones through inference processes. These new representations are used to deduce what to do. A knowledge base (KB) consists of a set of *sentences* that are expressed in a *knowledge representation language.* Each sentence corresponds to some assertion about the world. When the sentence is considered as given without being derived from other sentences, we call it *an axiom*. The KB may initially contain some background axioms.

According to [1], two types of operation are used to manage knowledge in a KB:

- The standard operation TELL to add new sentences to the KB
- The standard operation ASK to query knowledge from KB.
- In the following section, we will study, how the expression of sentences and the definition of their semantics are achieved through *logic.*

b) Logic

i. *Syntax*

Logic governs the representation language and specifies through a grammar all the sentences that are syntactically correct (well-formed). According to [1], the syntax of First-Ordered-Logic is given in Figure 01.

$$
\begin{aligned}
&Sentence \rightarrow AtomicSentence \mid ComplexSentence \\
&AtomicSentence \ \rightarrow Predicate \mid Predicate(Term, \ldots) \ \mid \ Term \ = Term \\
&ComplexSentence \rightarrow (Sentence) \mid [Sentence] \mid \neg Sentence \mid Sentence \ \text{A}\, Sentence \\
&\qquad\qquad \mid Sentence \vee Sentence \mid \ Sentence \Longrightarrow Sentence \mid Sentence \Leftrightarrow Sentence \\
&\qquad\qquad\qquad\qquad \mid Quantifier\ Variable, \ldots Sentence \\
&Term \rightarrow Function(Term) \mid \\
&Constant \mid \ Variable \\
&Quantifier \rightarrow \forall \mid \ \exists Constant \ \rightarrow A \mid X_1 \mid \text{John} \mid \ldots \\
&Variable \rightarrow \alpha \mid x \mid s \mid \ldots \\
&Predicate \rightarrow True \mid False \mid After \mid Loves \mid Raining \mid \ldots \\
&Function \rightarrow Mother \mid LeftLeg \mid \ldots
\end{aligned}
$$

Figure 01: Syntax of First-Order-Logic in Backus-Naur-Form

ii. *Semantics*

A logic must also define the meaning (semantics) of sentences. Depending on the used logic, this task can be simple or more sophisticated. Propositional logic simply assumes that there are facts that either hold or do not hold in the world. Propositional logic has the advantage of using a declarative, context- independent and unambiguous semantics. It is sufficient to illustrate the basic concepts of logic and knowledge-based agents. Nevertheless, it is not suitable to represent knowledge of complex environments in a concise way. For this reason, first-order-logic is preferred. It builds a more expressive logic on the foundation of propositional logic, borrowing representational ideas from natural language and at the same time avoiding its disadvantages. Its language is built around objects and relations. It also forms the foundation of many other representation languages.

iii. *Model*

For every sentence, its truth or falsehood is specified through a model. The possible models are just all possible assignments to the concerned variables. If a sentence **α** is true in model m, we say that *m satisfies* **α** or *m is a model of* **α**. M(**α**) is the set of all models of **α**. For example, the sentence *a * 3 = 6* is true in a world where *a is 2*, but false in any other world.

iv. *Logical Entailment*

Logical entailment is a relation between two sentences the second of which follows logically from the first one. It is the basis of l*ogical reasoning.* In mathematical notation, we write **α c β** to mean that the sentence ***α*** *entails the sentence* **β**. The formal definition of entailment is

$$\alpha \ c\ \beta \textit{ if and only if } M(\alpha) \subseteq M(\beta)$$

In clear: **α c β** if and only if, in every model in which **α** is true, **β** is also true.

v. *Logical inference*

Entailment is applied to carry out *logical inference* (to derive conclusions). If an inference algorithm *x* can derive α from KB, we write $KB \vdash_x \alpha$, which is pronounced "α is derived from KB by x" or "x derives α from KB", where α is just one possibility among the set of all consequences of KB.

An inference algorithm is called *sound* or *truth-preserving* if it derives only entailed sentences. An unsound inference procedure essentially announces the discovery of nonexistent or false conclusions.

An inference algorithm is called *complete* if it can derive any sentence that is entailed.

vi. *Grounding*

The connection between logical reasoning processes and the real environment of an agent is called *grounding*. Since KB is just a set of sentences inside the agent's *mind*, how do we know that KB is true in the real world? The agent program achieves grounding by creating a suitable sentence whenever a perceptible event occurs. Then, whenever that sentence is in the knowledge base, it is true in the real world.

c) Knowledge engineering with First-Order-Logic

Knowledge engineering is the process of knowledge base construction. It includes the following steps [1]:

a. Identifying the task
b. Assembling the relevant knowledge
c. Deciding on a vocabulary of predicates, functions and constants
d. Encoding general knowledge about the domain
e. Encoding a description of the specific problem instance
f. Posing queries to the inference procedure and getting answers

Let us directly illustrate with the example of Tontolo Malagasy. *Tontolo Malagasy* is expected to inform us about Malagasy culture (including historical facts and events, language, personalities, places). For simplicity in this example, we will observe a restricted world, namely the world of the former presidents of Madagascar. We will further narrow down to the observation of a single President, Zafy Albert, a deceased President (to avoid any political controversy).

Suppose the agent gets from any source the information that he was born on *May 1 in 1927*, that he died on *October 13 in 2017* and that he had two prime ministers during his office: *Francisque Ravony (term of office 1993 to 1995)* and *Emmanuel Rakotovahiny (term of office 1995 to 1996)*.

We know furthermore that the legal duration of a presidential term is 5 years and that a President must be at least 18 years old. A President may nominate successively different Prime Ministers (PM) during his presidential term.

Choice of the vocabulary to use for predicates, functions and constants

- *The following predicates will be used:* Person, President, Prime minister (PM).
- *The following functions[1] will be used:* BirthDate, DeathDate
- The following relations will be used: *TermPeriod, Tenure*
- The following constants will be used: *Zafy Albert, Francisque Ravony, Emmanuel Rakotovahiny, 1993, 1995, 1996.*

Encoding general knowledge about the domain in the language of First-Order-Logic

1) $Presid(x) \Longrightarrow Person(x) \wedge (year2 - year1) \leq 5 \ \mathrm{A}\ Tenure(x, year1, year2) \ \mathrm{A}\ year1 \geq d + 18 \ \mathrm{A}\ BirthDat(x, d)$

Meaning: A president is a person, who is older than 18 and whose tenure is delimited betweenyear1 and year 2.

2) $(x, y) \Longrightarrow Perso(x) \ \mathrm{A}\ President(y) \ \mathrm{A}\ (year1_x \geq year1_y) \ \mathrm{A}\ TermPeriod(x, y, year1_x, year2_x) \ \mathrm{A}\ Ten(y, year1_y, year2_y) \ \mathrm{A}\ (year2_x \leq year2_y)$

Meaning: A prime minister is a person who is associated with a president and whose term of office is necessarily within the term of the president in question

3) $Per(x) \Longrightarrow BirthDate(x, dat1)$

4) $Per(x) \Longrightarrow DeathDate(x, dat2)$

Encoding of the Specific Problem Instance Description

1) $Presiden(Zafy\ Albert)$
2) $BirthDat(Zafy\ Albert,$ 1927/05/01$)$
3) $DeathDat(Zafy\ Albert,$ 2017/10/13$)$
4) $P(Francisque\ Ravony, Zafy\ Albert)$
5) $PM\ (Emmanuel\ Rakotovahiny, Zafy\ Albert)$
6) $P(Francisque\ Ravony,\ Zafy\ Albert) \ \mathrm{A}\ PM\ (Emmanuel\ Rakotovahiny,\ Zafy\ Albert) \ \mathrm{A}\ Francisque\ Ravony \neq Emmanuel\ Rakotovahiny \ \mathrm{A}\ \forall x\ PM(x,\ Zafy\ Albert) \Longrightarrow (x = Francisque\ Ravony \vee x = Emmanuel\ Rakotovahiny)$

   Meaning: Zafy Albert had no prime ministers other than F. Ravony and E. Rakotovahiny
7) $TermPerio(Francisque\ Ravony,$ 1993, 1995$)$
8) $TermStar(Emmanuel\ Rakotovahiny,$ 1995, 1996$)$

All this information (general or specific) encoded in the KB and to which others will be added via the agent'spercepts will remain simple sentences as long as no request comes to give them life. These are the queries that will trigger the inference procedure, which will use them as part of a sequence of entailments.

The goal of inference is to find out whether KBc α for some sentence α. Compared to conventional databases, the reward with a knowledge base is that we can let the inference procedure operate on the axioms and problem specific facts to drift by itself the fact we are interested in knowing. There are different inference algorithms like model checking, theorem proving, forward-chaining, Davis-Putnam algorithm, Hill-climbing search [1]. They have respectively their advantages and their drawbacks but their studies are beyond the scope of the present article. An upcoming article will be dedicated to the explanation as well as the demonstration of the operating mode of an inference algorithm.

So far we have studied knowledge as an abstract and a general concept with an abstract representation language. In this abstract form, knowledge reuse and knowledge sharing will not be possible. In the following section, we will learn knowledge implementation through the use of ontologies.

III. Ontology

In philosophy, the term ontology refers to the science that *studies being as being*. With the emergence of knowledge engineering and the Semantic Web, and emphasizing the importance of knowledge sharing and reuse, this definition has been extended. *"An ontology is a formal specification of a shared conceptualization" [6].* Ontology represents a means of materializing knowledge in a form and in a structure that makes its reuse and its sharing possible.

a) Taxonomy

A general ontology organizes everything in the world into a hierarchy of categories - called a taxonomy - such as Events, Time, Physical objects and beliefs. The organization of objects into categories is a vital part of knowledge representation since much reasoning takes place at the level of categories. Categories permit also to make predictions about classified objects. A taxonomy has a *tree* structure.

Categories (or classes) serve to organize and simplify the knowledge base through inheritance. If we say that all instances of the category *Persons* have the property *are Mortal*, and if we assert that *Women* is a subclass of *Persons* and *Mothers* is a Subclass of *Women*, then our agent will know that every mother is mortal. We say that the individual woman inherit the property of mortality, in this case from their membership in the *Persons* category.

b) Relationships

It is possible to bring more precisions into the characterization of the relations between categories. To state that two categories that are not subclasses of each other (e.g. *Males* and *Females)* have no members in common, we use the relation *Disjoint: Disjoint (*{Males, Females}). We can go further and precise that an animal that is not a male must be a female, and say therefore that males and females constitute an exhaustive decomposition of the animals. A disjoint exhaustive decomposition is known as a partition. *Partition* ({Males, Females}, Animals). We use the general *Part Of* relation to state that one thing is part of another. Through the introduction of such types of relations between categories, the tree becomes a graph.

c) Named Entities

For the purpose of our project, we need an ontology that should not only contain categories or concepts but also *named entities* designating among other people, organizations, places and other important things. It s̄hould be possible to establish relations between *individuals* of different kinds: *What is located where? Who was born where? Which sovereign reigned during which period*?

d) Ontology Language

The best known languages to write ontologies are OWL (Web Ontology Language) and RDFS (Resource Description Framework Schema). Both are computational logic-based languages such that knowledge expressed in OWL or in RDFS can be exploited by computer programs.

In the following section, we present the YAGO model that is a slight extension of RDFS. it is designed to be extendable by other sources (high quality sources, domain-specific extensions, or data gathered through information extraction from Web pages), what makes it highly interesting for our project.

IV. Yago (Yet-Another-Great-Ontology)

YAGO was developed at the Max-Planck-Institute for Informatics in Germany. It is able to express entities, facts, relations between facts and properties of relations, while it is at the same time simple and decidable.In contrast to other existing ontologies, that are limited to a single source of background knowledge, YAGO combines high coverage with high quality [8]. Its latest version YAGO 4 (2022) is a cleaned version of Wikidata that contains more than 50 million entities and 2 billion facts.

a) The Components of YAGO

In YAGO, all objects (concepts) are represented as *entities* which are organized according to a taxonomy. The higher classes come from *schema.org* [2]and the lower classes from Wikidata. In the leaves of the tree, we no longer have classes but named entities, that is, concrete objects, individuals. We will refer to entities that are neither facts nor relations as *common entities*.

An ontology represents also relationships between entities that have no hierarchical link between them. Example: In "An author writes a book", there is no hierarchical link between *author* and *book.* The two concepts are related by "writing". YAGO does not only allow relationships between entities but even between relationships or between a relationship and an entity. This is possible by considering a relationship itself as an entity.

The triple of an entity, a relation and an entity is called a fact. They represent respectively *a subject, a predicate, and an object.* Example: *yago: Zafy_Albert rdf: type schema: Person*. The two entities are called the *arguments* of the fact. In YAGO, each fact is given a fact identifier, that is one of its strengths.

To maintain the semantic integrity of the data, YAGO uses the SHACL [3]Standard which makes it possible toexpress semantic integrity constraints.

b) Mathematical Definition of YAGO

A YAGO ontology over a finite set of common entities C, a finite set of relation names R and a finite set of fact identifiers I is a function with the following *definition:*

y $y: I \rightarrow (I \cup C \cup R) \times R \times (I \cup C \cup R)$

For facts that require more than two arguments, it is assumed that for each n-ary relation, a primary pair of its arguments can be identified. The primary pair can be represented as a binary fact with a fact identifier: #1: AlbertEinstein HASWONPRIZE NobelPrize. All other arguments can be represented as relations that hold between the primary pair and the other argument: #2: #1 TIME 1921.

Now, it's time to see how to exploit YAGO for *Tontolo Malagasy.*

c) Exploiting YAGO for Tontolo Malagasy

In the mass of YAGO data, there is very little portion of information on Malagasy culture. After a brief test, we noticed that out of 1,048,576 facts, only 80 mention Madagascar as a subject or as an object. In addition, there are specific Malagasy concepts and relationships that are totally unknown to YAGO. And finally, and obviously, YAGO does not understand the Malagasy language.

However, thanks to the flexibility of YAGO, including all these specificities that it currently lacks is not an impossible mission. There are three things we can do to enable YAGO extension on *Tontolo Malagasy*:

1. Data integration
2. Implicit translation
3. Implicit specification

i. *Data integration*

The challenge is to know how to recognize everything that is essential to our project and also to know at what level we must act (adapt, adopt, add, personalize, withdraw). Actually we are more interested in the taxonomy and in the common entities than in the named entities or in the facts because we would like to contribute our own named entities and our own facts. However, it is quite tricky because even if Madagascar is an island, it is not isolated from the rest of the world, so we still have to leave room for everything related to the outside world. Let us for now mainly focus on persons, places and organizations. In the YAGO taxonomy, the class *person* has 1569 subclasses that correspond to 1569 different professions. They don't all interest us right away. We will start with a small number to be able to accommodate the historical and cultural characters of our knowledge base. Here are some examples of professions that we could integrate: *Ambassador, Contestant, Entrepreneur, Evangelist, Foreigner, Governer, Head_of_State, Historian, Humanitarian, Leader, Sovereign…*

Each person may be defined through YAGO-predicates like

<hasChild>, <hasGivenName>, <isCitizenOf>,
<hasFamilyName>, <wasBornIn>, <isMarriedTo>,
<worksAt>, <isLeaderOf>, <diedIn>,
<hasGender>, <holdsPoliticalPosition>,
<diedOnDate>, <livesIn>, <wasBornOnDate>,
<graduatedFrom> …

Concerning places, YAGO integrates data from *GeoNames* for places. *GeoNames* is a geographical user- editable database that covers all countries and contains over eleven million placenames that are available for download free of charge. The *Tontolo Malagasy* project could extract directly from *GeoNames* all the geographical entities that concern Madagascar, if they lack in YAGO.

In the YAGO taxonomy, the class *organization* has 278 subclasses that correspond to 278 different types of organization. As with persons, they don't all interest us. Here are some examples of organization types that we will certainly integrate: *academy, church_school, diplomatic_mission, deputation, educational institute, family, industry, institution, local*.

Finally, here is an example of how to describe the President Zafy Albert in the language of YAGO.

<Zafy_Albert > rdf:type <wordnet_president_110467179> <Zafy_Albert> <hasGender>
<male> <Zafy_Albert > schema:birthDate <1927-05-01>
<Zafy_Albert > schema:deathDate <2017-10-13>
<Zafy_Albert > <isCitizenOf><Madagascar>

ii. *Implicit Translation*

We have to create a new prefix- *tontoloMG* - to permit any extensions and to associate new relations. This step is necessary if we want to incorporate translations into Malagasy in our ontology.

The following relations will be used to indicate translation:

- *mg_classic:* to associate a concept (entity, predicate, relation) with its translation into classic and official Malagasy.

  Example:

  President ***tontoloMG : mg_classic*** Filoha
  <isMarriedTo> ***tontoloMG : mg_classic*** '(dia) manambady an'i'

- *mg_dialect:* to associate a concept with its dialect-version.

  Example:

  Olona ***tontoloMG : mg_dialect*** olo

- *mg_familiar. to* associate a concept with colloquial language. Colloquial language may include foreign words or words of foreign origin.
  Here are some examples :

  Boy ***tontoloMG: mg_familiar*** Kôfboay (from English: *cow-boy)*
  Money ***tontoloMG: mg_familiar*** Sôsy

iii. *Implicit Specification*

The *mg_specific* relation will allow us to introduce into our ontology concepts specific to the Malagasy culture. They are necessarily expressed in Malagasy. In the official Malagasy language, for example, thereis no single word to designate an uncle or an aunt. The Malagasy *specifies:*

- If it is an uncle who is the eldest in the siblings, he is called *dadatoa*
- If it is an uncle who is the youngest in the siblings, he is called *dadafara*
- If it is an uncle who is somewhere inbetween, he is called *dadanaivo*

In praxis however, certainly for simplicity, many people use familiar language which permit to call invariably an uncle *Tonton* (a French word).
We would therefore write:

Uncle *tontoloMG:mg_specific Dadatoa |Dadanaivo| Dadafara*

Uncle *tontoloMG:mg_familiar Tonton*

v. Conclusion

This project was motivated by the obvious exclusion of emerging worlds by the new paradigms of AI because of the lack of massive data describing them. These worlds are actually just as old as the world known to AI. It's just that they have never been taken into account because they are not part of dominant cultures. Consequently, there is very little, if any, data that could be exploited by statistics for the purpose of making decisions or making predictions automatically. The approach we propose to remedy this lack is cognitive in nature and relies on knowledge engineering. We are experimenting it on the case of Malagasy culture. Our objective is to build a historical and cultural knowledge base in order to conserve and preserve the essence of Malagasy cultural identity. We therefore adopt YAGO, a state-of-the-art ontology, and exploit it to the limit of the possible to then customize it manually and complete it with specific concepts and facts. By doing so, we also facilitate the integration of the Malagasy universe into an already multicultural ontology. Another very important achievement is the potential for using our ontology as an instrument for the automatic production of corpora in Malagasy language. We are aware that over time, as automatic processing advances on these emerging worlds, new, more sophisticated needs will arise that will need to be solved by contemporary AI techniques. In the meantime, we will have time to massively produce data (through projects like this one) so that we can one day catch up with the bandwagon. We must see in this type of project a way to start the road, where there is nothing, to facilitate the integration of emerging worlds into the known world so they can finally be detected by the new AI.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References Références Referencias

1. Stuart Russel, Peter Norwig (2010). Artificial Intelligence, a modern approach. Edition Stuart
2. B. Jack Copeland, Carl J. Posy, O. Shagrir (2015). Computability Turing,Gödel, Church and Beyond. The MIT Press
3. Stefan Manegold, Martin Kersten, Constantino Thanos. Special theme Big Data (2012) ERCIM News N°89, April 2012
4. N.J. Smelser, P.B. Baltes (2001). International Encyclopedia of Social and Behavioral Sciences
5. Guugu Yimithirr (2010). The Natural Navigator. https://www.naturalnavigator.com/news/2010/09/guugu-yimithirr
6. Willem Nico Borst (1997). Construction of Engineering Ontologies for Knowledge Sharing and Reuse. University of Twente
7. Ian H.Witten, Elbe Frank, Mark A. Hall (2011). Data mining. Practical Machine Learning Tools and Techniques. Morgan Kaufmann Third Edition
8. Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Gerhard Weikum (2012). YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia