# An Artificial Natural Language

*John Doe*
John.Doe@machin.truc.com

ABSTRACT

We describe in this article an Artificial Natural Language and discuss how it could benefit to the domain of semantic representation.

KEYWORDS : Artificial natural language, machine translation, parsing

## 1 Introduction

For a long time, the goal of computational linguistics was to design and implement formal models to describe as accurately as possible the way natural languages work. The syntactic parsers, which were implemented on the top of these formal theories, were usually slow, cumbersome and quite difficult to debug. Grammars were viewed as a bag of rules, which were combined on the fly to produce all possible analyses. Most of the time, the system would turn out either nothing or… thousands of solutions. One possible response to these issues was to annotate huge amount of texts to form large databases of pre-parsed sentences, such as the Penn tree bank (Mitchel & al), in order to train different parsers along strategies. Since computers and storage had dramatically improved at the end of the eighties; it became possible to test many different machine learning models to produce a vast range of syntactic parsers. Another solution was to order rules by hand within the grammar, to make the parser deterministic. But these *shallow parsers* (Aït et al. 2002), did not have the same appeal as machine learning algorithms, despite a relative popularity at the end of the nineties. Actually, how do these approaches differ from one another? Basically, the main difference is that in one case you have an *implicit* grammar (from within the annotator's mind), which machine learning programs extract and weight down with *explicit* probabilities, while on the other hand, you have an *explicit* grammar and *implicit* probabilities under the guise of rule priorities (often the order in which rules are executed in the grammar). But, as many campaigns have shown, the quality of their output is rather similar.

### 1.1 Semantics

Moreover, attempts to reconcile semantics and these parsers outputs have proven a rather frustrating task. Errors accumulate at each stage of the syntactic parsing at such a rate (see Welty 2008 for a discussion on the topic), that, often, nothing relevant can be extracted. Traditional machine translation systems, based on pure symbolic description, are the perfect example of how terribly these stages combine. To avoid these obstacles, modern MT techniques utilize complex alignment algorithms to build language models that map word chunks from a source language into equivalent word chunks in the target language. These chunks are then stored in huge translation memories together with their probability to combine with each other. Translation consists then of finding the proper chunk boundaries within the source text to find the adequate chunks in the target language. Syntactic parsers can be used in this approach during the alignment stage, to find out the probability with which source chunks align (or translate) in target chunks. However, if these systems work pretty well for English, they usually fail to render proper translations for highly inflected languages. Worse, they require a lot of aligned texts to function

properly. If language pairs such as French-English or Spanish-English pose little problems, we usually lack enough bilingual texts for less frequent language pairs such as Japanese-Dutch for instance, to provide an adequate coverage to train an efficient language model. In the context of the European Commission, with 23 official languages, the combination of all these languages makes it almost impossible to reach a stage where one could translate from Hungarian into Danish. Furthermore, the documents, on which alignment is done, usually come either from news agencies or from administrative bodies, which introduces a bias on both the coverage and the content on which the MT System is trained. For instance, if the sentence "*Cars are too heavy*" is correctly translated in French[1] as "*Les voitures sont trop lourdes*", the inclusion of one word as in: "*Cars are way too heavy*" yields a gender error on the adjective: "*Les voitures sont beaucoup trop LOURDS*". The sentence is still comprehensible, but you cannot rely on these translations without any supervision, even for sentences as simple as the one above. "*way*" in this context belongs to a rather casual style, which does not occur enough in corpora to be fully captured by the language model. Yet, French and English belong to the happy few, the rare language pairs for which large bilingual corpora are available, such as the Canadian Hansard or Europarl. Furthermore, news and dispatches are translated everyday from one language into another by companies such as Reuters or AFP. Still, despite the amount of linguistic data available, the system often fails to accurately translate very simple sentences.

## 1.2 Conlang or ANL

The disconnection between syntax and semantics in the early systems explains in large part the errors in translation. Modern systems have largely got rid of both syntax and semantics and rely only on loose correspondences between chunks from bilingual documents. Still, their huge translation memory cannot be indefinitely extended to keep tracks of all possible translations for all word combination. John McCarthy in 1976 proposed an original idea to tackle this problem: an Artificial Natural Language (ANL hereafter), which was possibly an answer to Sapir's exclamation: "*Were a language ever completely "grammatical" it would be a perfect engine of conceptual expression. Unfortunately, or luckily, no language is tyrannically consistent. All grammars leak.*" ANLs are not exactly new, during the last two centuries many people have created artificial languages like Esperanto to help people from different nations communicate with each other. Creating new languages or conlangs (constructed languages) remains a surprising popular activity, such as the Klingon, which was created by Marc Okrand for the Star Treck universe, or more recently the Dothraki for the series *Game of thrones*. Languages were even invented to test scientific theories such as Lojban (Cowan 1987) which is a syntactically unambiguous human language based on predicate logic. The inventor of the language wanted to test the Sapir–Whorf hypothesis, which stated that languages had an impact on the way people conceptualize their world. However, conlangs have been little studied in the context of NLP. Why would a constructed language be of any interest to the domain? After all, their grammars are perfectly regular, morphology is consistent and lexicons are unambiguous. Actually, you could use the most primitive parser to encode any ANL grammars in a few days. No part of speech ambiguity, no ambiguous words, no PP-attachment issues: one simple grammar that embodies both syntax and semantic into a fast light parser with 100% coverage and precision. A sentence in an ANL is a representation of a piece of information both syntactically and semantically. This is of course anathema to most semantic theories, as the goal of semantics is to represent sense

[1] With the help of a translation tool from a well-known internet company

independently from its realization. However, in the case of an ANL sentence, the transformation into any abstract representations is almost straightforward, since it contains only one interpretation. In contrast, if you want to keep track of time, aspect or gender in a graph representation (Blanc 2001), you are forced to enrich the structure with many features, either inherited or computed on the fly, features which often complicate the semantic analysis.

## 2 Lingvata

The design of Lingvata, which means *language* in our ANL, was guided by a simple principle: each utterance in this *conlang* must have one single unambiguous interpretation by a computer program. This constraint implies a grammar whose properties are quite different from a language like Esperanto, which was invented to allow human beings to communicate with each other. Human beings can handle sentences in which relations between words are implicit. A sentence such as "*I see a man with a telescope*" has two interpretations in English, "with a telescope" is either a verb complement or a noun complement. *A non-ambiguous language implies that each possible interpretation of the above example must have a different surface realization.*

### 2.1 Lexicon

For simplicity reasons, the Lingvata lexicon was automatically built out of multi-lingual dictionaries of Esperanto. The choice was motivated in part by the careful way in which words have been devised in this language. In Esperanto, a word is composed of a semantic root, which can be enriched with semantic suffixes and transformed into a word thanks to a restricted set of part of speech endings. With this method, part of speech ambiguity and homophones are totally absent from the vocabulary. There is a word for *bank* as a financial institution (*banko*) and a word for *bank* as a river side (*bordo*). However our constraints are rather different from those of Esperanto, since our goal is to communicate with a computer. For instance, Esperanto uses accented consonants (ĉ, ĝ etc.), which we did not want to keep in our language. We have also chosen a different set of morphological markers. We have applied some simple rules to transform these dictionaries into our own lexicons. With this method, we have created two bilingual dictionaries, one for English (about 13,000 words) and one for French (about 16,000). These dictionaries are quite sufficient for our tests and can be enriched at will.

### 2.2 Morphology

Lingvata is a highly inflected language with declensions. For those who have painful memories of learning Latin or German, this choice might sound uselessly complex. In fact, what makes learning these languages difficult is not really their declension system, but the number of rules and exceptions one must remember. In our ANL, rules are straightforward and know no exception. There are only four cases, with the same ending whatever the word. Furthermore, the part of speech of each word is identified through a unique category suffix.

Suffixes for determiners, adjectives and nouns are implemented as a triple: $C_1$-V-($C_2$)

where C is a consonant and V a vowel.

1. $C_1$ defines a part of speech in [l,c,t], with "l" for determiners, "c" for adjectives and "t" for nouns.

2. V is a semantic marker in [a,e,i,o,u], with “a” for regular words, “e” for locations, “i” for temporal expressions, “o” for proper names and “u” for numerals.
3. $C_2$ is a case-marker in [∅,n,d,s], with ∅ nominative, “n” accusative, “d” dative and “s” genitive.

Basically, any root can be transformed into a word with the proper suffix. For instance, if we take a proper name such as *John*, then “*Johnton*” is a proper noun at the accusative.

### 2.2.1 Case Markers

We have chosen only four cases, with nominative as the neutral form.

- Nominative is the sentence or sub-clause subject case, with no suffix.
- Accusative is the direct object case, with “n” as suffix.
- Genitive is the noun complement case, with “s” as suffix.
- Dative is the case of any words within a PP, with “d” as suffix.

Example:

*Kateta dometas* cat of house

### 2.2.2 Verbs

The verb suffixes embed the following notions:

- Tense: past, present, future, conditional
- Verb in the main clause or in a sub-clause

However, theses suffixes do not reflect any agreement with the verb subject, as in English or in French. A verb suffix is of the form: “iVC”, with V in *[a,e]* and C in *[p,g,f,d]*.

- V defines the position of the verb: “a” the verb is in the main clause, “e” the verb is in a sub-clause.
- C is the tense: “p” for past, “g” for present, “f” for future and “d” for conditional.

### 2.2.3 Pronouns

The basic pronoun suffix is of the form “Vi$C_1$er$C_2$”, with V in [a,e,i,o,u] and $C_1$,$C_2$ in [∅,n,s,d]. $C_1$ records the antecedent case and $C_2$ is the case maker of the pronoun itself. Thus “kainers”, for instance, is the genitive relative pronoun (meaning “whose”), with as an accusative antecedent. V is a semantics marker, which modifies the pronoun meaning. For instance, “kiier" means “when” since “i” is the temporal vowel (see above), “kuier” means “how many” or “how much”, “keier” means “where” and “koier” means “who”.

### 2.2.4 Prepositions

The basic preposition suffix is of the form “i($C^1$)ar($s^1$)”, with $C^1$ in [∅,n,s,d]. Basically, the “s” at the end of the preposition means that the PP is attached to a noun phrase, thus the genitive marking. If the “s” is omitted then the PP is attached to the main clause verb, in that case the only form accepted is “iar”.

Examples:

The following examples show how the suffixes are used to infuse structure within the different sentences.

| | |
|---|---|
| *Kateta laktetan **iniar** dometad **trinkeiag**.* | A cat **drinks** some milk **in** the house |
| ***Kateta** laktetan **iniars** dometad trinkeiag* | A **cat in** the house drinks some milk. |
| ***Kateta** laktetan **ininars** dometad trinkeiag* | A cat drinks some **milk in** the house. |

### 2.2.5 Coordinating Conjunctions and Question Mark

We have taken our inspiration from Latin to implement our coordination system. Actually, the Latin language provides two ways to handle coordination: "et" which is still used in most Latin languages today (except in Romanian) and "-que" as in "*Senatus Populus**que** Romanus*"[2]. Our conjunction system is based on the latter. The two basic conjunctions "and" and "or" have been respectively translated into two suffixes: "-que" and "-quo", which are always appended at the end of the last word of the coordination.

| | |
|---|---|
| ***Kateta grandeca hundetaque** laktetan trinkeiag* | A cat and a big dog drink some milk. |
| *Kateta **laktetan krudecan viandetanquo** ameiag* | A cat likes milk or raw meat. |
| *Romaneca senetata Romaneca popoletaque* | Roman senate and Roman people |

Actually, there is a third one "-qua" which is used to mark the interrogation as in Japanese.

| | |
|---|---|
| *Kateta laktetan trinkeiagqua?* | Does the cat drink some milk? |

## 2.3 Word Order

Most natural languages with a declension system admit a rather free word order. In Latin for instance, "*domina rosam amat*" is equivalent to "*rosam domina amat*"[3], thanks to the accusative suffix "*am*" in "*rosa*", which helps analyze the word as a direct object. However, when the sentence becomes too complex, the free position of these words makes the analysis quite tricky, as different verbs for instance might claim different direct objects. To solve this problem, we have imposed a simple constraint. The last element of a phrase is always its head, except for the prepositional phrase. Thus, the verb is always at the end of its clause and the noun is always the last element of a nominal phrase. Determiners and adjectives always agree with their nominal head, which simplifies the detection of a nominal phrase. The order in which nominal or prepositional phrases occurs before the verb is rather free, with the following constraint: *the antecedent of a pronoun or a preposition is always placed before this pronoun or this preposition. In case of ambiguity, the closest antecedent is the one selected.*

Examples:

*Ala **kateta** bonecan laktetan **kaier** deiar farmetad **veneieg** trinkeiag.*
The cat, which comes from the farm, drinks some good milk.
***Bonecan laktetan** ala kateta **kainer** deiar farmetad **veneieg** trinkeiag.*
The cat drinks some good milk, which comes from the farm.
*Ala kateta **bonecan laktetan kainern** farmeta **produkteieg** trinkeiag.*
The cat drinks some good milk which the farm produces.

---

[2] *Roman senate and people,* usually shortened as "SPQR" was the Roman republic motto.
[3] *The lady likes the rose*

These examples show how fiddling with different case markers modify the sentence interpretation. The form of the relative pronoun (*ka*…) reflects the nature of its antecedent. In the second sentence, for instance, the antecedent is the direct object, which is indicated by the presence of an "n" within the pronoun realization, while in the third example the final "n" indicates its role within the sub-clause as a direct-object pronoun.

A thorough examination of these sentences explains in part some choices in our language. The goal of this ANL is to create a representation in which nothing is ambiguous, where part of speech and phrase-boundaries can be detected by the most rudimentary parser. This objective requires a specific trade-off between the number of features, which should not be too large as to avoid a steep learning curve for a human being, nor too small as to deprive the computer from the minimum information needed to parse any occurrences of that language reliably.

## 3 Example of Use: as a Pivot Language in a Machine Translation System

Most MT systems work as a black box, leaving human translators with very little control over the whole process. Roughly, a human being can only influence these systems either in the initial text or in the final text, without any access to their intermediary representations. However, if these intermediary representations are encoded in an ANL, used as a pivot language, then the problem is quite different. An ANL text presents the advantage to be a semantic representation and in the same time to be readable by a human being. Since a text in our ANL is only composed of ASCII characters, it can be easily modified in any editors to introduce or to modify existing relations, providing the next stages with a cleaned semantic version.

### 3.1 Implementation

To test this idea, we have implemented a tool to help people play with the language itself. This tool integrates different grammars to analyze or generate sentences in English, French and Lingvata. The peculiarity of our system is that *dependencies* are our pivot representation. The parser component produces as output a set of dependencies, which the generation component can take as input to generate new sentences. Thus, a user can type in a sentence in French, whose analysis as a set of dependencies is passed to the generation Lingvata grammar. *The generated Lingvata sentence* is then parsed to produce a new set of dependencies that is passed to the English generation grammar and also to the French generation grammar to regenerate the initial sentence and check any semantic drift. The user can modify the Lingvata sentence and re-generate the sentence in French or English, hence the necessity to have an intermediary stage in that language.

Example:

If the user proposes to the system the following sentence:

*La jeune fille parle avec le neveu du voisin.*

Then the system will analyze this sentence and translate it into a Lingvata sentence:

*ala yuneca **filineta** kumiar alad nevetad alas nayberetas paroleiag.*

This sentence is then re-generated into French and English:

*La jeune **fille** parle avec le neveu du voisin.*

*The young **daughter** talks with the neighbour's nephew.*

The word "*fille"* in French is ambiguous between *girl* and *daughter*, and the word that was selected means *daughter*. Actually, the system displays all possible translations of this word in Lingvata, from which the user can pick up a better translation:

*ala yuneca **knabeta** kumiar alad nevetad alas nayberetas paroleiag.*

The sentence will then automatically be retranslated in English with this new interpretation:

*The young **girl** talks with the neighbour's nephew.*

This word selection can then be recorded for the rest of the session, if necessary. What is even more interesting is that this choice can be recorded with its context, in order to trigger this translation only if the word is found again in a similar environment.

The user can also play on the preposition suffixes in order to launch a deep rebuilding of the whole sentence. For instance, below, the adjunction of "s" at the end of "kumiar" transforms the preposition into a nominal preposition:

*ala yuneca knabeta **kumiars** alad nevetad alas nayberetas paroleiag.*

Triggering a new translation:

*The young girl **with the neighbour's nephew**, talks.*

## 3.2 Evaluation

Our evaluation is not based on a strict metric as it is the case in information extraction systems or in machine translation. Since our goal is to obtain a framework in which we could test a new way to encode language; we use the structural richness of Lingvata as our main evaluation criteria. This ANL was developed through an iterative process, where at each stage we checked the number of phrases and clauses we could cram in a sentence. The two grammars, the formal language grammar and the computational grammar, were designed and implemented in parallel. We would then test our parser over larger and larger sentences to check whether the system would still extract the proper dependencies. Our first ideas, as it is often the case, proved too simplistic. Word position was not enough for instance to extract clear relations. Actually, this was also a problem people discovered in controlled language implementations. If you base your parsing only on word positions, then you quickly loose the possibility to connect distant phrases within the sentence. The declension system was part of the early experiments, but it evolved with the idea of having prepositions and pronouns bearing some of their antecedent features. In our tests, we managed to write sentences up to 40 words, using as models, sentences from news articles. Another interesting metric is the rather small number of rules that we need to parse any sentences from that language. Our grammar, which has been implemented using Xerox tools (Ait 2002), comprises only 45 rules to build both the chunk tree and the dependencies. The generation grammar is even smaller with only 30 rules. Furthermore, as the language is totally deterministic, any parsers, even the dumbest, can analyze a sentence in a fraction of a second.

# 4 Conclusion

In a certain way, the most compact way to store the semantic representation of a text is...the text itself. A text keeps more efficiently information about time, location or events than any RDF or

OWL (Van Harmelen F 2003) descriptions. However, texts are ambiguous, terribly ambiguous. Natural languages have evolved in a rather organic way, without any actual plan, hence this inextricable fabric, with which our programs have so many difficulties to deal. For a long time, linguists have tried to formalize languages into strict mathematical frameworks, but languages proved so elusive that most theories would "leak". Machine learning techniques, despite their careful injection of hard science into the domain, did bring some improvement, but the best systems still fail to provide a precise and reliable analysis for too many utterances. Because of the deluge of linguistic data to handle, it is quite difficult to devise a full representation of any texts that would be consistent over all documents, even those belonging to a common domain. On the contrary, an ANL representation keeps intact the whole spectrum of linguistic data with no or little loss of information. A paragraph or a sentence written in that language is in the same time a description as precise as any piece of text and the very semantic encoding of that text: a symbolic representation which is half-way between man and the machine, "*the perfect engine of conceptual expression"*. Furthermore, this symbolic representation being extremely regular, does not require a lot of computer power. A very simple machine is enough to process large amount of data written in that language.

## 5 References


WELTY C. (2008). Answering Questions from the Semantic Web. Actes de *Langtech 2008,* Rome.

VAN HARMELEN F., HENDLER J., HORROCKS I., MCGUINNESS D. L., PATEL-SCHNEIDER P. F., AND STEIN L. A. (2003). OWL web ontology language reference, *http://www.w3.org/tr/owl-guide/*

AIT-MOKHTAR S., CHANOD J.P. AND ROUX C. (2002) Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering, Cambridge Univ Press* , 8, 2-3, 121-144.

BLANC É. & SÉRASSET G. (2001) From Graph to Tree: Processing UNLGraphs using an Existing MT System*. Proc. The first UNL open Conference,* Suzhou, China, 22-26 November 2001, UNDL.

LASSILA O., SWICK R. (1999). Resource description framework (RDF) model and syntax specification. *recommendation, W3C, february 1999. http://www.w3.org/tr/1999/rec-rdf-syntax-19990222*

COWAN, J.W. (1997), The complete Lojban language. *Logical Language Group.*

MITCHELL P. MARCUS, MARY ANN MARCINKIEWICZ, Building a Large Annotated Corpus of English: The Penn Treebank,*University of Pennsylvania*.

MCCARTHY J. (1976). An example for natural language understanding and the AI problems it raises. *Formalizing Common Sense: Papers by John McCarthy. Ablex Publishing Corporation*, 355.

EDWARD SAPIR (1921), Language, p. 38.