



LIRICS

Deliverable <D1.3>

< Analysis of Syntactic Annotation Framework (SynAF) >

Project reference number	e-Content-22236-LIRICS
Project acronym	LIRICS
Project full title	Linguistic Infrastructure for Interoperable Resource and Systems
Project contact point	Laurent Romary, INRIA-Loria 615, rue du jardin botanique BP101. 54602 Villers lès Nancy (France) romary@loria.fr
Project web site	http://lirics.loria.fr
EC project officer	Erwin Valentini
Document title	Analysis of Syntactic Annotation Framework (SynAF)
Deliverable ID	D1.3
Document type	Report
Dissemination level	Public
Contractual date of delivery	
Actual date of delivery	26 June 2007
Status & version	Draft
Work package, task & deliverable responsible	WP1, Unis
Author(s) & affiliation(s)	Lee Gillam (UniS), Neil Newbold (Unis)
Additional contributor(s)	
Keywords	ISO, Terminology, Quality Assurance, Readability

Document evolution

version	date	version	date
1.0	02/03/07		

1 Automated Quality Assurance

This document accompanies LIRICS D1.3.

To demonstrate results of the analysis of automated quality assurance, a standard being developed within the LIRICS project has been analysed. The document 'Syntactic Annotation Framework (SynAF)' (at Working Draft stage¹) was chosen to show the output obtained from the various stages of the analysis.

1.1 Terminology Lookup

All known terms were annotated, including those containing another term. For example, 'data category' contains another known term 'data'. The annotation allows access to the definition for the term. All terms in 'ISO 1087-1', 'ISO 1087-2', 'ISO 12620' and 'LMF' were annotated. An example of annotated terms in 'SynAF' is shown in figure 1.

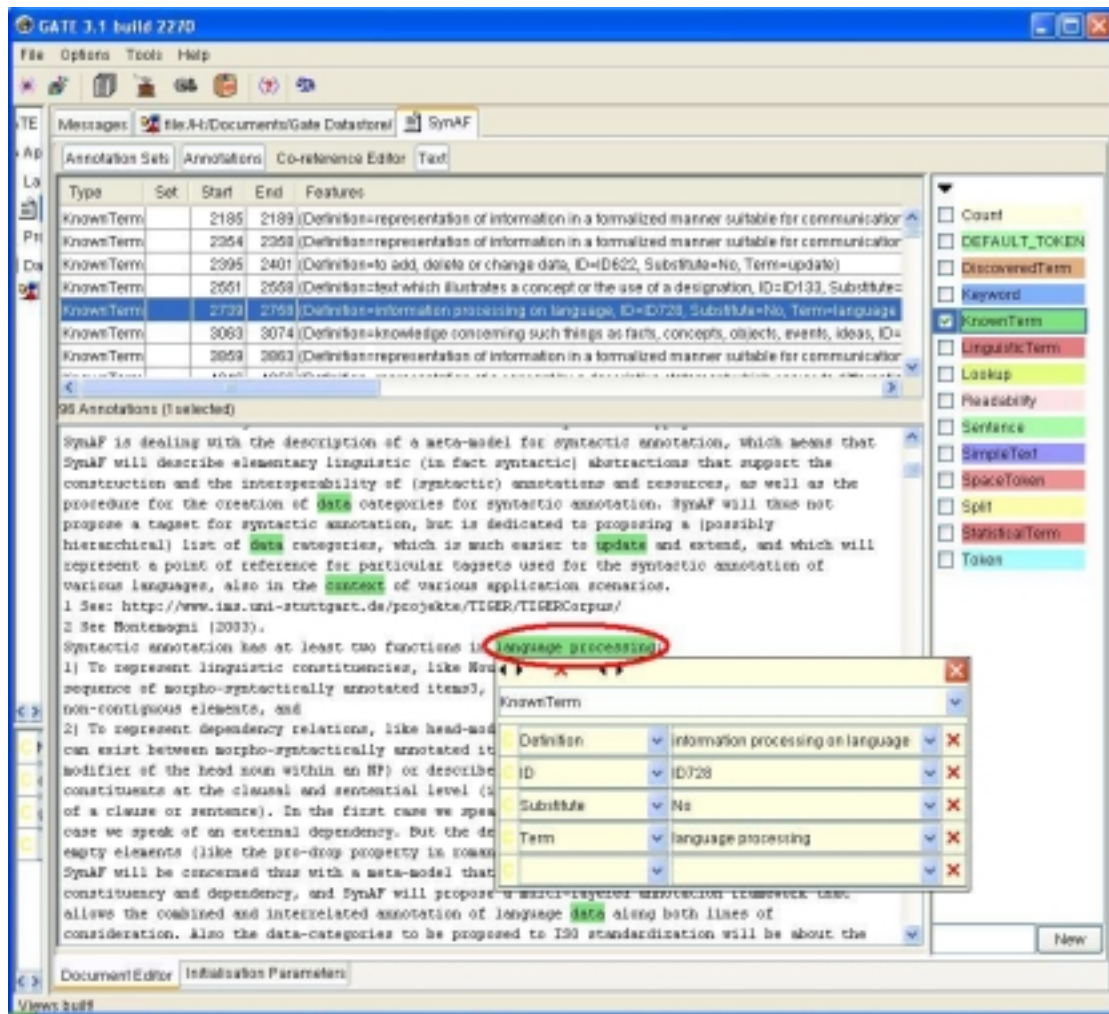


Figure 1: A screenshot of the 'SynAF' document in GATE displaying the terminology annotations and a popup window of annotation information for the term 'Language Processing'.

¹ Available at: http://lirics.loria.fr/doc_pub/SynAF_WD_2006-01-22.pdf

Term	Count
type	28
label	15
data	14
definition	9
object	9
information	6
merge	2
parsing	2
read	2
context	1

Table 1: The top 10 known terms, and their frequencies, in ‘SynAF’.

Discovered terms were further investigated to evaluate which could be considered new terms. Terms highlighted by both statistical and linguistic methods could be prioritised for consideration. Terms such as ‘syntactic annotation’, ‘annotation’, ‘SynAF’ and ‘morph’ were identified as items that may need to be defined. Further filtering of this list is required, but frequency information can be helpful here also; variations by part of speech can lead to duplications, for example for ‘SynAF’. Examples of discovered terms from SynAF are shown in Table 2.

Term	Linguistically Valid	Statistically Valid	Count
* annotation	false	true	42
head	true	false	33
value name	true	false	22
partec	true	true	21
* synaf	true	true	19
value	true	false	18
edge label	true	false	14
syntactic annotation	true	true	13
mod	true	false	11
morph	true	true	11
* synaf	false	true	11
word	true	false	11
* annotation	true	true	10
constituency	true	false	10
relation	true	false	10
data categories	true	false	9
Dependency	true	false	9

Table 2: Examples of highly frequent discovered terms in ‘SynAF’, including duplications due to different parts of speech (*).

The discovered terms demonstrated, at lower frequencies, linguistically valid multiword expressions that were surprisingly complex. Examples of such terms are shown in Table 3.

Term	Linguistically Valid	Statistically Valid	Count
multi-layered annotation strategy	true	false	2
tiger annotation framework	true	false	2
morpho-syntactically annotated items	true	false	2
morpho-syntactically annotated fragments	true	false	2

Table 3: Examples of potential multiword terms that were discovered in ‘SynAF’

Additionally, the linguistic and statistical methods for discovering terms found numerous valid two word expressions that were regularly used. Examples of these are shown in Table 4.

Term	Linguistically Valid	Statistically Valid	Count
value name	true	false	22
edge label	true	false	14
syntactic annotation	true	true	13
dependency information	true	false	4
annotated corpora	true	false	3
dependency annotation	true	true	3
starting point	true	false	3
multi-layered annotation	false	true	3
tiger annotation	false	true	3
annotation framework	true	false	2

Table 4: Examples of frequent discovered two-word terms in ‘SynAF’.

There were also notable keywords (single words) identified as valid either linguistically or statistically and frequently used throughout the document. Examples of these are shown in Table 5. Some of these may easily be filtered out.

Term	Linguistically Valid	Statistically Valid	Count
annotation	false	true	42
head	true	false	33
synaf	true	true	19
value	true	false	18
word	true	false	11
annotation	true	true	10
constituency	true	false	10
relation	true	false	10
dependency	true	false	9
tiger	true	false	9

Table 5: Examples of discovered single-word terms in ‘SynAF’.

The two methods of identifying potential new terms allowed for potential readability issues to be highlighted. Such a readability issue can be demonstrated by the first item in Table 3, the “multi-layered annotation strategy” – e.g. is it an “annotation strategy” which is “multi-layered”, or a “strategy” for a “multi-layered annotation”? Potential ambiguity could be demonstrated by specific differences in term recognition and entailment. The two terms shown in Table 6 were identified slightly differently by the two methods, and such differences may or may not be significant.

Term	Linguistically Valid	Statistically Valid	Count
multi-layered annotation	false	true	3
multi-layered annotation strategy	true	false	2

Table 6: Complexity in entailed terms.

The fact that the two methods recognise the term differently demonstrate that the two methods for recognising terms result in further considerations being required.

1.3 SimpleText Analyser

The ‘SynAF’ document was analysed to determine the potential for improvement. A report was produced of new replacements suggested by the SimpleText plug-in for words and phrases deemed unnecessarily complex. The first 100 replacements were analysed manually, resulting in 20 suitable replacements of which 10 substitutions were unique. Every further

instance of the unique substitutions was analysed throughout the rest of the document, 28 instances in total, to see if the replacements were appropriate in every instance. These replacements and their results are detailed in Table 7.

Phrase	Replacement	Appearances In Document	Correct Replacements	%Correct
activity	work	1	1	100.00%
difficulties	problems	1	1	100.00%
feature	property	4	4	100.00%
furthermore	also	1	1	100.00%
similar	almost the same	1	1	100.00%
thus	therefore	4	4	100.00%
various	different	4	2	50.00%
with respect to	for	1	1	100.00%
within	in	9	5	55.56%
would	will	2	1	50.00%

Table 7: The 10 unique replacements filtered from the initial 100 suggestions with the number of times the replacements were correct throughout the rest of the document.

The 9 occurrences of the word “within” are detailed in Table 8, which shows the full sentence where the word appeared and whether the replacement “in” was deemed suitable.

Sentence	Valid	Reason
This annotation strategy has reached in the meantime a kind of consensus within the corpus linguistics.	Yes	“in” is simpler
Within the eContent LIRICS project, a group of international experts has started the ISO process, called SynAF (Syntactic Annotation Framework), whereas SynAF has already been accepted at the ISO Level as a New Work Item.	Yes	“in” is simpler
The various “cat” values within the NT nodes of Tiger will build in SynAF the starting point for a list of data categories for constituency annotation.	Yes	“in” is simpler
Considering the examples we give just below, covering both a germanic and a romance language, we do not expect huge difficulties in our task, within the proposed meta-model of a multi-layered annotation strategy for syntactic annotation.	Yes	“in” is simpler
The dependency information can exist between morpho-syntactically annotated items within a phrase (an adjective is the modifier of the head noun within an NP) or describe a specific relation between syntactic constituents at the clausal and sentential level.	Yes	“in” is simpler
The dependency information can exist between morpho-syntactically annotated items within a phrase (an adjective is the modifier of the head noun within an NP) or describe a specific relation between syntactic constituents at the clausal and sentential level.	No	Loss of accuracy
The GRs csubj and xsubj may be used for clausal subjects, controlled from within , or without, respectively. ncsbj is a non-clausal subject.	No	Not the same meaning
The GR cmod is for when the adjunct is controlled from within , and xmod for control from without.	No	Not the same meaning
Within the edge label below, we can see a small list of dependency labels, which will also offer a starting point for data categories for dependency annotation.	No	Loss of accuracy

Table 8: The descriptions detailing when “in” was a suitable replacement for “within”.

The 4 occurrences of the word “various” are detailed in Table 9, which shows where the word appeared and whether the replacement “different” was deemed suitable.

Sentence	Valid	Reason
The various “cat” values within the NT nodes of Tiger will build in SynAF the starting point for a list of data categories for constituency annotation.	Yes	“different” is simpler
SynAF will thus not propose a tagset for syntactic annotation, but is dedicated to proposing a (possibly hierarchical) list of data categories, which is much easier to update and extend, and which will represent a point of reference for particular tagsets used for the syntactic annotation of various languages, also in the context of various application scenarios.	Yes	“different” is simpler
SynAF will thus not propose a tagset for syntactic annotation, but is dedicated to proposing a (possibly hierarchical) list of data categories, which is much easier to update and extend, and which will represent a point of reference for particular tagsets used for the syntactic annotation of various languages, also in the context of various application scenarios.	No	Loss of accuracy
As the starting point for this standardization activity, we consider both existing syntactically annotated tree-banks and the output of various existing parsing systems.	No	Loss of accuracy

Table 9: The descriptions detailing when “different” was a suitable replacement for “various”.

The 2 occurrences of the word “would” are detailed in Table 10, which shows where the word appeared and whether the replacement “will” was deemed suitable.

Sentence	Valid	Reason
Following this view, we would not deal with constituents like empty elements or traces generated by movements at the constituency level.	Yes	“will” is simpler
We would like to keep this as a point for the SynAF meta-model for syntactic annotation: all contiguous syntactic information can (possibly) be encoded using an embedded XML tree representation.	No	Doesn’t quite make sense

Table 10: The descriptions of when “will” was a suitable replacement for “would”.

1.4 Readability analysis

It was found that some SimpleText replacements were appropriate in every further instance such as “feature” and “thus”. However, other words rarely had correct replacements and in some cases were never suitable again. The majority of proposed SimpleText replacements were found not to be suitable and leaving much room for investigation in how to focus the substitutions more accurately. However, to investigate the extent that this limited number of substitutions would influence the readability scores of the document the Replacer plug-in was run. All the readability scores were slightly reduced except for the FOG and SMOG results which increased a little. The number of words in a document can actually *increase* due to some of the replacements. The prime example of this occurrence is the substitution of “similar” for “almost the same”. Readability scores before and after replacements are shown in Table 11. These tests were performed with known terms annotated in the document; these are later discounted from the complex word count in the FOG and SMOG readability scores.

Score	Before	After
Kincaid	18.813	18.810
Flesch	24.133	24.152
<i>FOG</i>	21.893	21.937
<i>SMOG</i>	17.982	18.026
ARI	19.746	19.728

Table 11: Readability scores before and after the SimpleText process.

FOG and SMOG were tested in two forms: first in “raw” form (Raw), second by removing known terms from the calculations (Surrey). Any word of more than 3 syllables which was part of a known term was not considered as a complex word in these calculations since it is assumed that inclusion in the terminology has already accounted for readability. This approach improves readability scores by reducing the number of complex words. The FOG and SMOG scores before and after the replacements, and omitting or including complex words, are shown in Table 12.

Score	Before Replacements		After Replacements	
	Raw	Surrey	Raw	Surrey
FOG	21.973	21.893	22.018	21.937
SMOG	18.061	17.982	18.105	18.026

Table 12: Readability scores omitting and utilising terminology before and after the SimpleText process.