From to ISOTiger – Community Driven Developments for Syntax Annotation in SynAF

Sonja Bosch, Kerstin Eckart, Gertrud Faaß, Florian Zipser, Antonio Pareja-Lora, Ulrich Heid, Laurent Romary, Andreas Witt, Amir Zeldes, Kiyong Lee, et al.

From <tiger2/> to ISOTiger – Community Driven Developments for Syntax Annotation in SynAF

Sonja Bosch, University of South Africa (UNISA) – Kerstin Eckart, University of Stuttgart – Gertrud Faaß, University of Hildesheim – Ulrich Heid, University of Hildesheim – Kiyong Lee, Korea University – Antonio Pareja-Lora, Universidad Complutense de Madrid – Laurette Pretorius, University of South Africa (UNISA) – Laurent Romary, Humboldt-Universität zu Berlin – Andreas Witt, Institut für Deutsche Sprache – Amir Zeldes, Georgetown University – Florian Zipser, Humboldt-Universität zu Berlin

Abstract

In 2010, ISO published a standard for syntactic annotation, ISO 24615:2010 (SynAF). Back then, the document specified a comprehensive reference model for the representation of syntactic annotations, but no accompanying XML serialisation. ISO's subcommittee on language resource management (ISO TC 37/SC 4) is working on making the SynAF serialisation ISOTiger an additional part of the standard. This contribution addresses the current state of development of ISOTiger, along with a number of open issues on which we are seeking community feedback in order to ensure that ISOTiger becomes a useful extension to the SynAF reference model.

1 Introduction

In 2010 an ISO[1] standard on the syntactic annotation framework SynAF was published, ISO 24615:2010. Even though this ISO standard specified a comprehensive reference model for the representation of syntactic annotations, it did not provide an accompanying XML serialisation for this type of annotations [1].

[1] thus presented <tiger2/>, an XML serialisation for SynAF, enhancing the existing TIGER-XML format [8] from the TIGER treebank [2] to meet the specifications of the SynAF model, such as being able to handle not only constituency-based representations but also dependency analyses and others which make use of extensible types of nodes and edges. [1] described the <tiger2/> format and presented examples of its use in the modelling of linguistic constructions, including e.g. contractions, elliptic subjects or compound sentences as they appear in Zulu.

[1] International Organization for Standardization, `http://www.iso.org`

In the meantime, ISO's subcommittee on language resource management (ISO TC 37/SC 4) is working on making the serialisation an additional part of the standard. For this reason, it was agreed in 2014 to rename the standard to ISO 24615-1 *Language resource management – Syntactic annotation framework (SynAF) – Part 1: Syntactic model* and start a new standard project for *Part 2: XML serialization (ISOTiger)*[2].

The SynAF serialisation ISOTiger is the continuation of <tiger2/>, pursuing two objectives: i) including feedback from the community, cf. [1], and ii) aligning SynAF even more closely with other existing standards such as the *Linguistic annotation framework* (LAF) [7], the *Morpho-syntactic annotation framework* (MAF) [6] and the combined ISO and TEI standards on *feature structures* (FSR, FSD) [4, 5].

The main purpose of this contribution is to explore how the two objectives of ISOTiger are met in a consistent, non-contradicting way. Section 2 briefly describes the SynAF reference model, Section 3 addresses the current state of development of ISOTiger and Section 4 discusses some open issues on which we are seeking community feedback, in order to ensure that ISOTiger becomes a useful extension to the SynAF reference model.

2 SynAF components

The SynAF – Part 1 metamodel specifies syntactic annotations as consisting of *SyntacticNodes*, *SyntacticEdges* and their corresponding *Annotations*. The model distinguishes between terminal nodes (*T_node*) for morpho-syntactically annotated word forms (or empty elements when appropriate) and non-terminal nodes (*NT_node*), which can be annotated with syntactic categories from the phrasal, clausal and sentential level. Edges can be established between (both terminal and non-terminal) nodes and can also be annotated. While this metamodel can be implemented on its own, it is recommended to express morpho-syntactically annotated terminal nodes following the MAF standard [6] and to apply a data category registry [3] to specify the syntactic categories that are part of the annotation.

3 XML serialisation

Figure 1 shows an excerpt of an XML-encoded syntactic annotation example[3]. The `<annotation>` element of the header makes reference to an external annotation declaration, cf. Figure 2. Furthermore, the example utilizes a standoff notation where the `terminals` refer to `wordForms` from a MAF document, cf. Figure 3.

[2]SynAF – Part 2 is currently at the stage of a committee draft (ISO/CD 24615-2). For an overview of the stages in the development of ISO standards see:
`http://www.iso.org/iso/home/standards_development/resources-for-technical-work/support-for-developing-standards.htm`

[3]For more elaborate examples in different languages see [1].

```
<head>
  <!-- ... -->
  <annotation>
    <external corresp="annot_decl.xml"
  </annotation>
</head>
<body>
  <s xml:id = "s1">
    <graph xml:id="s1_g1">
      <terminals>
        <t xml:id="s1_t1" tiger2:corresp ="m1.maf#wf1"/> <!-- we -->
        <t xml:id="s1_t2" tiger2:corresp ="m1.maf#wf2"/> <!-- can -->
        <t xml:id="s1_t3" tiger2:corresp ="m1.maf#wf3"> <!-- see -->
          <edge tiger2:type ="dep" label="nsubj" tiger2:target ="#s1_t1"/>
          <edge tiger2:type ="dep" label="aux" tiger2:target ="#s1_t2"/>
        </t>
      </terminals>
      <nonterminals>
        <nt xml:id="s1_nt1" cat="NP">
          <edge tiger2:type ="prim" label="HD" tiger2:target ="#s1_t1"/>
        </nt>
        <nt xml:id="s1_nt2" cat="VP">
          <edge tiger2:type ="prim" label="HD" tiger2:target ="#s1_t3"/>
        </nt>
        <nt xml:id="s1_nt3" cat="VP">
          <edge tiger2:type ="prim" label="--" tiger2:target ="#s1_nt2"/>
          <edge tiger2:type ="prim" label="HD" tiger2:target ="#s1_t2"/>
        </nt>
        <nt xml:id="s1_nt4" cat="S">
          <edge tiger2:type ="prim" label="SBJ" tiger2:target ="#s1_nt1"/>
          <edge tiger2:type ="prim" label="--" tiger2:target ="#s1_nt3"/>
        </nt>
      </nonterminals>
    </graph>
  </s>
</body>
```

Figure 1: Excerpt from an example encoded in <tiger2/> (version V2.0.5).

While there will be changes on the transition from <tiger2/> to ISOTiger, it is planned to still allow for inline notation in terminal nodes.

The example shows some main characteristics of the current format.[4] This format includes both a header (to describe the tags utilized in the annotations) and a body. In the body, the `<s>` element denotes a segment of the primary data, which is a more generic version of the respective TIGER-XML element denoting a sentence. A segment can contain several `<graph>` elements for syntactic graph structures, and a graph may include terminal nodes (`<t>`), non-terminal nodes (`<nt>`) and edges (`<edge>`). Nodes and edges can be typed and can be annotated by generic attribute-value-pairs defined in the `<annotation>` element of the header. Terminal

[4] There are also additional features, such as corpus structuring and corpus metadata elements.

nodes refer to a textual segment or to a word form in a morpho-syntactically annotated corpus (the latter is shown in the example), thus implementing *T_Node* from the SynAF reference model. Non-terminal nodes implement SynAF's *NT_node* and help represent hierarchical structures. `<edge>` elements are embedded in the element that denotes their start node, and they specify their target node by means of the `@target` attribute. The start node of an edge may not only be a non-terminal node, as stipulated in TIGER-XML, but also a terminal node, thus implementing the *SyntacticEdge* from the SynAF reference model. This allows representing e.g. constituency trees as well as dependency relations such as in Figure 1. The `@type` attribute distinguishes between different kinds of nodes and edges, e.g. *dep* vs. *prim* for dependency and constituency edges respectively in Figure 1.

Typing nodes and edges also allows to define specific attribute-value-pairs for the different node and edge types. The attributes @domain and @type of the feature element in the annotation declaration specify if the respective annotation can be applied to a terminal node, a non-terminal node or an edge (@domain), and, if applicable, to which user defined type of these (@type). Hence, the feature name *label* in the above <tiger2/> example can have different value sets for dependency and constituency edges, cf. Figure 2.

Since annotations are user-defined attribute-value pairs, there are also no restrictions with respect to specific linguistic theories; however, the semantics of the annotations needs to be specified. Accordingly, every feature and feature value can be linked to a specific data category, which in the ISO setup should come from a data category registry compliant to ISO 12620:2009 [3], e.g. ISOcat[5] (see the feature value definition for *NP* in Figure 2).

To inspect more <tiger2/> examples one can also make use of a web service client[6] described by [9] that generates MAF and <tiger2/> encoded analyses for Spanish sentences.[7]

4 Open issues

The current state of the SynAF XML serialisation is still closely related to the original TIGER-XML format. This closeness was a main concern in the development of <tiger2/>. In this way, an already utilized and accepted treebank format was taken into account and enhanced, instead of inventing a completely new format.

However, considering the new flexibility of treebank annotation possibilities that is offered by the current format, the annotation declarations, such as shown in Figure 2[8], fall short in two respects: the generic attribute-value-pairs neither offer the full descriptive power of feature structures as defined in standards from ISO

[5] `www.isocat.org`

[6] `http://quijote.fdi.ucm.es:8084/ClienteFreeLing/`

[7] The annotations themselves are generated by means of FreeLing (`http://nlp.lsi.upc.edu/freeling/demo/demo.php`), a multilingual part-of-speech tagger and a parser for both phrase structure and dependency analyses.

[8] This example is based on <tiger2/>, but already includes the dcr namespace.

```
<annotation>
  <feature  name="cat" domain="nt"
     dcr:datcat ="http://www.isocat.org/datcat/DC-1506">
    <value  name="NP" dcr:datcat="http://www.isocat.org/datcat/DC-2256"/>
    <value  name="S" dcr:datcat="http://www.isocat.org/datcat/DC-2295"/>
    <value  name="VP" dcr:datcat="http://www.isocat.org/datcat/DC-2255"/>
  </feature>
  <feature  name="label" domain="edge" type="prim"
     dcr:datcat ="http://www.isocat.org/datcat/DC-5596">
    <value  name="HD" dcr:datcat="http://www.isocat.org/datcat/DC-2306"/>
    <value  name="SBJ" dcr:datcat="http://www.isocat.org/datcat/DC-2261"/>
    <value  name="--"/>
  </feature>
  <feature  name="label" domain="edge" type="dep"
     dcr:datcat ="http://www.isocat.org/datcat/DC-2304">
    <value  name="nsubj">nominal subject</value>
    <value  name="aux" dcr:datcat="http://www.isocat.org/datcat/DC-2262"/>
  </feature>
</annotation>
```

Figure 2: Document annot_decl.xml containing external annotation declarations.

```
<wordForm xml:id="wf1" lemma="we" tokens="#t1"/>
<wordForm xml:id="wf2" lemma="can" tokens="#t2"/>
<wordForm xml:id="wf3" lemma="see" tokens="#t3"/>
```

Figure 3: Excerpt from a MAF document (m1.maf).

and TEI [4, 5], nor do they match the standard representation. Utilizing the FSR and FSD standards as in MAF (ISO 24611 - sections 7.2 and 7.4)[9] would however go far beyond the original TIGER-XML format.

Figure 4 shows the *NP* node and an outgoing edge, where the annotations are encoded as feature structures. On the one hand, we would no longer have to deal with generic XML attributes for nodes and edges, and the <tiger2/> elements `<feature>` and `<value>` would no longer be needed. On the other hand, we would (i) introduce structured annotations, which might not be completely mappable onto formats with non-structured annotations and (ii) introduce a slightly more verbose representation. However, utilizing FSR would of course also allow for the use of libraries, which could be declared centrally (or externally) and be referred to by a new ISOTiger attribute of nodes and edges. Furthermore, applying the ISO and TEI standards on feature structures fosters an integration of the different standardization approaches.

A standoff notation making reference to external feature structure declarations could also allow for structured annotations as an option, while still keeping the possibility of specifying simple attribute-value-pairs.

The second aspect under discussion is a reference mechanism to primary data,

[9] Section 7.4 in MAF states how to declare and reuse FSR libraries and Section 7.2 defines how to actually annotate word forms with feature structures.

```
<nt xml:id="s1_nt1">
  <fs>
    <f name="cat" dcr:datcat="http://www.isocat.org/datcat/DC-1506">
      <symbol value="NP" dcr:datcat="http://www.isocat.org/datcat/DC-2256"/>
    </f>
  </fs>
  <edge xml:id="s1_e3" type="prim" target="#s1_t1">
    <fs>
      <f name="label" dcr:datcat="http://www.isocat.org/datcat/DC-5596">
        <symbol value="HD" dcr:datcat="http://www.isocat.org/datcat/DC-2306"/>
      </f>
    </fs>
  </edge>
</nt>
```

Figure 4: Open issue: feature structures in ISOTiger

```
Locations in the document:
|w|e|  |c|a|n|  |s|e|e|
0 1 2 3 4 5 6 7 8 9 10

<terminals>
  <t xml:id="s1_t1" from="0" to="2"/> <!-- we -->
  <t xml:id="s1_t2" from="3" to="6"/> <!-- can -->
  <t xml:id="s1_t3" from="7" to="10"/> <!-- see -->
</terminals>
```

Figure 5: Open issue: reference mechanism to primary data in ISOTiger

for cases where there is no morpho-syntactic annotation, yet SynAF terminals are required to be represented in a standoff way. Therefore, for such cases, ISOTiger could refer to LAF [7], where the generic reference mechanism introduces virtual anchors in between base units of the primary data representation (e.g. characters), which can be referenced to select a region from the primary data. Figure 5 includes an example utilizing possible new ISOTiger attributes `@from` and `@to`, together with the idea of the virtual anchors. A related representation has been proposed in MAF [6]. However according to the SynAF – Part 1 metamodel, terminals in SynAF are equivalent to word forms, and can thus for example also be defined over multiple spans. Furthermore, pointing directly from a terminal node to the primary data might hide the essential distinction between tokens and word forms. Therefore a direct reference from terminals to primary data would only be allowed in exceptional cases.

It should be noted that the two ISOTiger examples in Figure 4 and Figure 5 only provide suggestions for further developments to transform <tiger2/> into the ISOTiger standard, and are likely to undergo changes before the standardization process is complete. A discussion in the community on these open issues, as well as on the current state of ISOTiger, would help to meet the requirements of the users in this ongoing standardisation work.

References

[1] Sonja Bosch, Key-Sun Choi, Éric La de Clergerie, Alex Chengyu Fang, Gertrud Faaß, Kiyong Lee, Antonio Pareja-Lora, Laurent Romary, Andreas Witt, Amir Zeldes, and Florian Zipser. <tiger2/> as a standardised serialisation for ISO 24615 – SynAF. In Iris Hendrickx, Sandra Kübler, and Kiril Simov, editors, *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (TLT11)*, pages 37–60, Lisbon, Portugal, 2012. Edições Colibri, Lisboa.

[2] Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620, 2004.

[3] ISO 12620:2009 Terminology and other language and content resources – Specification of data categories and management of a Data Category Registry for language resources.

[4] ISO 24610-1:2006 Language resource management – Feature structures – Part 1: Feature structure representation.

[5] ISO 24610-2:2011 Language resource management – Feature structures – Part 2: Feature system declaration.

[6] ISO 24611:2012 Language resource management – Morpho-syntactic annotation framework (MAF).

[7] ISO 24612:2012 Language resource management – Linguistic annotation framework (LAF).

[8] Esther König, Wolfgang Lezius, and Holger Voormann. *TIGERSearch 2.1 User's Manual. Chapter V - The TIGER-XML treebank encoding format*. IMS, Universität Stuttgart, 2003.

[9] Antonio Pareja-Lora, Guillermo Cárcamo-Escorza, and Alicia Ballesteros-Calvo. Standardisation and interoperation of morphosyntactic and syntactic annotation tools for spanish and their annotations. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).