Transatlantic Collaboration between LAPPS and CLARIN
M6 Milestone Report

WP3, Task T3.1

Inventory of Objects and Features

This document reports on the progress in WP3 in the first 6 months of the project. The goal of task T3.1 was to create an initial inventory of features used in LAPPS, WebLicht and LINDAT. We focus mostly on the objects and features used in some of the more basic LAPPS and WebLicht services since those services were the ones that we decided to integrate first.

Categories and Features for LAPPS and WebLicht

Both the LAPPS Grid and WebLicht use a set of annotation types (or categories or objects) to represent instances of common linguistic categories. For the Language Application Grid, the annotation objects and features used are defined by the LAPPS Vocabulary, also known as the Web Services Exchange Vocabulary (WSEV), which is available at http://vocab.lappsgrid.org. The set of categories and features is kept as small as possible and is driven by the LAPPS services available at the moment. As of May 2017, the relevant objects and features are as below.

```
Annotation {id}
   Region {targets, start, end}
      Paragraph
      Sentence {sentenceType}
      NounChunk
      VerbChunk {vctype, tense, voice, neg}
      NamedEntity {category, type, gender}
      Token {pos, lemma, tokenType, orth, length, word}
      Markable
   Relation {label}
      GenericRelation {relation, arguments}
      SemanticRole {head, argument}
      Constituent {parent, children}
      Dependency {governor, dependent}
   Coreference {mentions, representative}
   PhraseStructure {constituents, root}
   DependencyStructure {dependencyType, dependencies}
```

Features are listed between curly brackets after the annotation type. Since the objects are arranged in a hierarchy, all features defined at a higher level are inherited at the lower level. All these objects and features have URIs and these URIs are referenced in the data structures that are exchanged between LAPPS services. We are ignoring for now the meta data features and we are also ignoring elements from the vocabulary that describe documents.

On the WebLicht end, the annotation objects and their features are defined implicitly by the Text Corpus Format format (TCF)[1] and its schema[2]. Unlike LIF and the LAPPS Vocabulary there are no explicit links from TCF representations to URIs. Similar to LIF in the LAPPS case, TCF is used for communications between services in WebLicht and the categories used are a reflection of what services are provided. Relevant objects and features are listed below. This listing is based mostly on TCF examples given on the TCF page, in some cases extra attributes are printed that are only mentioned in the schema, these are in italics.

```
token {ID, start, end}
sentence {ID, start, end, tokenIDs}
tag { ID, tokenIDs }
lemma { ID, tokenIDs }
parse { ID }
constituent { ID, cat, tokenIDs, edge, cref}
dependency { func, depIDs, govIDs }
emptytok { ID }
entity { ID, class, tokenIDs, start, end }
reference { ID, tokenIDs, mintokIDs, type, rel, target }
gpoint { tokenIDs, alt, lat, lon, continent, country, capital }
```

As with the WSEV categories, we do not list meta data features. It should also be noted that all of these objects are expressed in special lists, for example, *tag* objects are expressed in a dedicated *POSTags* list, and the list names are part of the schema. We do not enumerate the list objects here for brevity, but one should keep in mind that they exist and at times are associated with meta data features.

Two more remarks on the TCF categories listed above. One is that in some cases, the actual value that would be associated with a category is not expressed in an attribute but in the CDATA of the XML tag. For example, the actual part-of-speech of a *tag* is enclosed by the tag. The same holds for the *token* and *lemma* categories. The other is that TCF categories are not explicitly organized in an inheritance hierarchy.

[1] https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/The_TCF_Format
[2] https://github.com/weblicht/tcf-spec/blob/master/src/main/rnc-schema/textcorpus_0_4.rnc

Discussion

In general, there is considerable overlap between LAPPS and WebLicht categories, which is not surprising given that both include basic NLP services. In particular, similar categories and features are used for tokens, sentences, lemmas, parts-of-speech, named entities, coreference, syntactic structure and dependency structure, even though naming and particulars may be different. Looking across categories, there are obvious mappings of identifiers and start and end offsets. In addition, the WSEV *targets* feature serves the same purpose as the TCF *tokenIDs* feature.

The following discusses in some more depth where the categories and features overlap or diverge, grouped in a couple of annotation categories.

1. Tokens, tags and lemmas. This involves the *Token* category for LAPPS with features *pos* and *lemma* and the *token*, *pos* and *lemma* categories for WebLicht. Even though there is a difference in that TCF gives first class citizen status to all these categories and LAPPS demotes some to features, there is a straightforward conceptual mapping and no changes will need to be made to the vocabularies on either side.

2. Other simple annotations. This include sentences and named entities. Both TCF and WSEV have these categories, but the features are somewhat richer on the WSEV end so some features may need to be added to TCF's *sentence* and *entity* categories if those features are used in LAPPS services and the decision is made that WebLicht users should have access to those features. Also note the notational difference that TCF has a *class* feature while WSEV uses *category*. The *gpoint* category in TCF does not seem to have a counterpart in WSEV, but extending the optional features allowed on *NamedEntity* would account for the information that is expressed in the *gpoint* features.

3. Phrase structure and dependency structure. Again, the similarities are greater than the differences. WSEV has PhraseStructure and DependencyStructure in its vocabulary and there is no direct equivalent listed for TCF, but recall that TCF groups its annotations in specialized lists and it has special XML tags for these. Two things are worth noting and may need attention on the WSEV side of things: (1) TCF uses the *emptytok* category for dummy tokens in dependency structures and there is no equivalent on the WSEV end, (2) dependencies in TCF are potentially between lists of identifiers and WSEV cannot capture that at the moment.

4. Coreference. This is handled differently between TCF and WSEV and this needs to be studied further once we start sharing the output of coreference modules.

We conclude with pointing out that there is a fair number of categories that are expressed in either WSEV or LIF but not in both.

Exploring the TCF links given in the beginning of this document will reveal that many of the categories used in TCF were not listed in the previous section. For example, TCF has categories for morphology, orthography, phonetics, lexical semantics, word sense disambiguation, query matching and discourse connectives. None of these have counterparts in the LAPPS vocabulary. We are not planning to make WebLicht services that use those categories available to LAPPS users in the early stages of the project, so for now we ignore them, but it is clear that the LAPPS vocabulary needs to be extended when we start focusing on some of those services.

Similarly, TCF has no examples for chunks and semantic roles (although the TCF schema do have a relation category), but since we have no current plans to include LAPPS services that produce these we will ignore this for now.