The ALVIS Format for Linguistically Annotated Documents

A. Nazarenko, E. Alphonse, J. Derivière, T. Hamon, G. Vauvert, D. Weissenbacher

Laboratoire dInformatique de Paris-Nord (UMR 7030)
University Paris 13 & CNRS
99, Av. J.-B. Clément, 93430 Villetaneuse, France
{firstname.name}@lipn.univ-paris13.fr

Abstract

The paper describes the ALVIS annotation format and discusses the problems that we encountered for the indexing of large collections of documents for topic specific search engines. This paper is exemplified on the biological domain and on MedLine abstracts, as developing a specialized search engine for biologist is one of the ALVIS case studies. The ALVIS principle for linguistic annotations is based on existing works and standard propositions. We made the choice of stand-off annotations rather than inserted mark-up, and annotations are encoded as XML elements which form the linguistic subsection of the document record.

1. Introduction

One of the objectives of the ALVIS project[1] is to develop semantic-based search engines that achieve good performance in information retrieval in specialized domains. As one of our case study, we are developing a specialized search engine for the biological domain that should be able to handle complex queries (boolean and even relational queries including normalized gene names, for instance).
In this context, we are experimentally studying of the contribution of Natural Language Processing (NLP) in information retrieval. We are testing various indexing methods based on various types of linguistic annotation.
This paper presents and discusses the format that has been adopted in the ALVIS project for the linguistic annotation of the documents. It shows how NLP tools can add new annotations to a given document or exploit existing ones. The paper focuses on the linguistic part of the document annotations, disregarding the metadata associated with the document at the crawling step (Buntine et al., 2005).
The section 2 presents in more details the context of the ALVIS project and explains the need for linguistically annotated documents. The ALVIS format for linguistic annotatio is presented in section 3. The section 4 explains how such a rich levl of annotation can be achieved. The Section 5 discusses the advantages and limits of our format.

2. Context: the ALVIS project

The ALVIS project aims at building a peer-to-peer network of semantic search engines and at developing open source components to help the design of new topic specific search engines. Among these components, there is a Natural Language Processing (NLP) line, which goal is to enrich the crawled documents with linguistic annotations to enable a semantic and domain specific indexing of these documents. The type and quality of the annotations vary with the following factors:

- The way the annotated documents are used: for each specific topic, beside the collection of documents to index, we exploit a sample of documents for acquiring specialized linguistic resources (acquisition phase). The resulting resources (named entity dictionaries, terminologies, semantic tags) are then used to tune the generic NLP line for the corresponding specific domain (production phase). A deep analysis of the documents is required at the acquisition level whereas indexing necessitates an efficient and shallow analysis strategy.
- The availability of domain specific resources: the more domain specific knowledge is available (or acquired), the richer the document annotations can be.
- The language of the documents: the intrinsic complexity and the state of the art in NLP differ from one language to another. In ALVIS, four different languages are processed (English, French, Slovene, Chinese): the various processing steps are not equally important for all languages (e.g. traditional word segmentation is useless for Chinese, lemmatisation is more important for Slovene than for English and even for French);
- The volume of textual data to process: since NLP is known to be computationally expensive, the deepness of document analysis depends on the efficiency of the NLP components and the volume of documents to be analysed.

One of the objectives of the ALVIS project is to test the various combinations of annotations to identify which ones have a significant impact on Information Retrieval results, within a given specific domain. In this context, the definition of the format for the linguistic annotation of documents is a critical issue. It is also necessary to ensure the modularity of the ALVIS NLP line and the interchangeability of NLP tools.

3. ALVIS format for linguistic annotations

The linguistic annotation is represented as a layered set of textual units and linguistic properties.

3.1. Annotation principle

The ALVIS principle for linguistic annotations (Nazarenko et al., 2004) is based on existing works and standard propo-

[1] ALVIS is a FP6 STREP projet aiming at developing an open source prototype of a distributed, semantic-based search engine. See http://www.alvis.info

sitions (Grishman, 1997; Bird and Liberman, 1999). We made the choice of stand-off annotations rather than inserted mark-up, and annotations are encoded as XML elements which form the linguistic subsection of the document record (Buntine et al., 2005). The principle of stand-off annotations is to separate the text of the document to annotate. It has numerous advantages:

- The initial textual data may be read-only and/or very large, so copying it to insert mark-up may be unacceptable.
- The distribution of the initial data may be controlled whereas the mark-up is intended to be freely available.
- The stand-off annotations do not pollute the initial textual data.
- Stand-off annotations allow embedded and overlapping annotations that are incompatible with an inserted mark-up. It is therefore easier to encode concurrent annotations produced by different NLP tools, non linear elements, which may be relevant linguistic entities (such as "to... decide" in "to completely decide" or the French negation "ne... pas" in "je ne mange pas"), relations (grammatical functions, semantic relations) between elements belonging to various levels in the hierarchy of annotations.
- New levels of annotations can be added without disturbing the existing ones.
- Editing one level of annotation has minimal knock-on effects on others.
- Each level of annotation can be stored and handled separately, eventually in several files.

The main drawback of the standoff annotation principle is that it is difficult and computationally expensive to rebuild the textual signal from the list of annotations.
The problem of representing linguistic annotations is not new. It has been widely studied since the beginning of the nineties and several *ad hoc* formats have been proposed (see (Grishman, 1997; Bird and Liberman, 1999)). The efforts to unify these formats in order to allow interoperability among NLP tools are recent. An ISO proposition (TC37SC4/TEI) is currently under definition (Ide et al., 2004), which will include a Feature Structure Representation, a Morpho-Syntactic Annotation Framework, a Category Data Repository, a Linguistic Annotation Framework, a Lexical Mark-up Framework and some Data Category S-Electronic Lexical Resources.
Our goal is not as general as that of the TC37SC4/TEI: strictly complying with the norm would make our annotation formalism more complex whereas a light version is sufficient for ALVIS needs.

3.2. Textual entities

Different levels of textual units are relevant for NLP. In ALVIS, we take five levels into consideration. At a basic level, the text is segmented into tokens. The other levels are built on this first token level: we distinguish the words, the phrases, the semantic units and the sentences.
For sake of readability, the examples of the following subsections are given with traditional inserted annotation (slash of brackets) instead of stand-off annotations. The actual ALVIS format is shown on Figure 6.

3.2.1. Tokens

Tokens are the fundamental textual units in the ALVIS text processing line. This segmentation is not linguistically grounded. It serves no other purpose but to provide a starting point from which to implement further segmentation.
This level of annotation follows the recommendations of the TC37SC4/TEI workgroup. However, this workgroup proposes to insert pointer mark-up in the textual signal to mark the token boundaries whereas we refer to the character offset.
To simplify further processing, we distinguish different types of tokens:

- Alphabetical tokens: sequences of letters (`a-z` and `A-Z`) including accented characters;
- Numerical tokens: sequences of digits (`0-9`);
- Separating tokens: sequence of separator characters (`space`, `return` ...);
- Symbolic tokens: any other character.

The tokenisation is the basic stage of text analysis. Tokens are numbered from 1 for the first token. All others annotations refer directly or indirectly to that token numbering. In the example of figure 1, the slashes represent the token boundaries (note that blanks are tokens).

/Transcription/ /of/ /the/ /cotB/,/ /cotC/,/ /and/ /cotX/ /genes/ /by/ /final/ /sigma/(/K/)/ /RNA/ /polymerase/ /is/ /activated/ /by/ /a/ /small/,/ /DNA/-/binding/ /protein/ /called/ /GerE/./

Figure 1: Tokenization.

3.2.2. Words

Words are the basic linguistic units. They are made of tokens: every word is made of one or several tokens, numeric, alphabetic or symbolic. Words may contain spaces (i.e. *B. Subtilis* in biology or *pomme de terre* in French).
However, some character strings are not trivially split into words, for example "doesn't" is made of the words "does" and "not", which do not appear as such. In such a case, two words (does and not) are created independently from the corresponding tokens (*doesn*, the apostrophe (') and *t*).
In the following example (fig. 2), words are delimited by square brackets. Note that neither the punctuation marks nor the blanks are words. This segmentation can be compared with the tokenisation presented in the section above.

[Transcription] [of] [the] [cotB], [cotC], [and] [cotX] [genes] [by] [final] [sigma(K)] [RNA] [polymerase] [is] [activated] [by] [a] [small], [DNA-binding] [protein] [called] [GerE].

Figure 2: Word segmentation.

3.2.3. Phrases

A phrase is a group of words (or a single word) that functions as a syntactic unit. It is composed of a head (the main part of the phrase) and of optional modifier(s) that can be words or phrases. The syntactic properties of a phrase are derived from its head (a Noun for a Noun Phrase, an Adjective for an Adjectival Phrase, etc.).

At the phrasal level described here, we only delimit the unit and no syntactic category is assigned to the phrase (see the section **??** below).

In the following example (fig. 3), the phrases are delimited with inserted brackets.

[During [sporulation of Bacillus subtilis]], [[spore coat proteins] [encoded [by [cot genes]]]] [are expressed [in [the mother cell]] and [deposited [on [the forespore]]]]. Transcription [of [the cotB, cotC, and cotX genes]] [by [final [sigma(K) RNA polymerase]]] [is activated [by [a small, [DNA-binding protein] [called GerE]]]]. [The promoter region [of [each [of [these genes]]]]] [has [two [GerE binding sites]]].

Figure 3: Phrase identification.

3.2.4. Semantic units

The semantic units are the textual units that are considered as significant on a semantic point of view. They can be:

- Named entities that refer to well identified domain entities (often designated by proper names but not always)
- Terms that are the expressions referring to the concepts specific to the domain of the text.
- Undefined semantic units: other types of relevant semantic units can be identified, even if their semantic status is not established.

In the example of the figure 4, the named entities and terms are tagged as XML-like inserted mark-up.

3.2.5. Sentences

The sentences correspond to a traditional textual unit. They usually start from a word with a capital initial character and ends with a period. However various other types of sentences can be encountered in texts. In the ALVIS linguistic annotation format, we consider that titles, some list items and captions are sentences.

In the following, the sentences are identified by inserted brackets for sake of simplicity.

During sporulation of <NE>Bacillus subtilis<NE>, <term>spore coat proteins<term> encoded by <term> <NE>cot<NE> genes<term> are expressed in the <term>mother cell<term> and deposited on the <term>forespore<term>. Transcription of the <NE>cotB<NE>, <NE>cotC<NE >, and <NE>cotX<NE>genes by final <NE>sigma(K)<NE> <term>RNA polymerase<term> is activated by a small, <term>DNA-binding protein<term> called <NE>GerE<NE>. The <term>promoter region<term> of each of these genes has two <named entity>GerE<NE> <term>binding sites<term>.

Figure 4: Named entity and term tagging.

[During sporulation of Bacillus subtilis, spore coat proteins encoded by cot genes are expressed in the mother cell and deposited on the forespore.] [Transcription of the cotB, cotC, and cotX genes by final sigma(K) RNA polymerase is activated by a small, DNA-binding protein called GerE.] [The promoter region of each of these genes has two GerE binding sites.]

Figure 5: Sentence segmentation.

The ALVIS format for these textual units is homogeneous from one level to another. Except for tokens, each textual unit has an identifier, a list of components and an optional form in which the sequence of characters to which it corresponds can be copied.

3.3. Properties of textual entities

Various properties can be associated with textual units. They are encoded as separate XML entities referring to the textual entities to which they are associated:

- Morpho-syntactic tags: the lemmas, the stems, the syntactic categories[2], the morpho-syntactic features.
- Syntactic relations, which define the role (or function) played by two words between one another. These relations are represented as triplets: a relation type T, its head H (or governor) and its expansion E (or dependent, also called modifier or argument).
- Semantic tags: the semantic types, which are attached to semantic units that can be words (registered as undefined units by the tagger), named entities or terms.
- Semantic relations: the anaphoric relations and the domain specific relations. These relations are attached to semantic units (either named entities, terms or undefined semantic units) which may corresponds to words or phrases.

[2] A syntactic category is either a phrasal category, such as noun phrase or verb phrase, if the textual unit it refers to can be decomposed into smaller syntactic units, or a lexical category (also called "part of speech" or POS category) such as noun or verb, which cannot be further decomposed.

```
<documentCollection>
<documentRecord id="A79ACA58DEB7E6114747710B9A85059F">
  <acquisition>
    <acquisitionData>
      <modifiedDate>2004-11-21 15:59:14<modifiedDate>
      <urls>
        <url>http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&amp;db=pubmed&amp;dopt=MEDLINE&amp;
            list_uids=10788508<url>
      <urls>
    <acquisitionData>
    <canonicalDocument>
      <section>
        <section title="Combined action of two transcription factors regulates genes encoding spore coat proteins
            of Bacillus subtilis.">
          <section>Combined action of two transcription factors regulates genes encoding spore coat proteins of
              Bacillus subtilis.
          <section>
           ...
          <section>
        <section>
    <canonicalDocument>
  <acquisition>
```

```
 <linguisticAnalysis>
   <token_level>
   <token>
     <content>Combined<content>
     <from>0<from>
     <id>token1<id>
     <to>7<to>
     <type>alpha<type>
   <token>
   ...
   <token_level>
   <sentence_level>
   <sentence>
     <form>Combined action of two transcription
          factors regulates genes encoding spore coat
          proteins of Bacillus subtilis .<form>
     <id>sentence1<id>
     <refid_end_token>token30<refid_end_token>
     <refid_start_token>token1<refid_start_token>
   <sentence>
   ...
   <sentence_level>
   <semantic_unit_level>
   <semantic_unit>
     <named_entity>
       <form>Bacillus subtilis<form>
       <id>named_entity0<id>
       <list_refid_token>
         <refid_token>
           <refid_token>token27<refid_token>
         <refid_token>
         <refid_token>
           <refid_token>token28<refid_token>
         <refid_token>
         <refid_token>
           <refid_token>token29<refid_token>
         <refid_token>
       <list_refid_token>
       <named_entity_type>species<named_entity_type>
     <named_entity>
   <semantic_unit>
   ...
   <semantic_unit_level>
```

```
   <word_level>
   <word>
     <form>Combined<form>
     <id>word1<id>
     <list_refid_token>
       <refid_token>
         <refid_token>token1<refid_token>
       <refid_token>
     <list_refid_token>
   <word>
   ...
   <word_level>
   <lemma_level>
   <lemma>
     <canonical_form>combined<canonical_form>
     <id>lemma1<id>
     <refid_word>word1<refid_word>
   <lemma>
   ...
   <lemma_level>
   <morphosyntactic_features_level>
   <morphosyntactic_features>
     <id>morphosyntactic_features1<id>
     <refid_word>word1<refid_word>
     <syntactic_category>JJ<syntactic_category>
   <morphosyntactic_features>
   <morphosyntactic_features>
     <id>morphosyntactic_features10<id>
     <refid_word>word10<refid_word>
     <syntactic_category>NN<syntactic_category>
   <morphosyntactic_features>
   ...
   <morphosyntactic_features_level>
 <syntactic_relation_level>
   <syntactic_relation>
     <id>syntrel1<id>
     <syntactic_relation_type>NCOMPby
     <syntactic_relation_type>
     <refid_head>
       <refid_word>word26<refid_word>
     <refid_head>
     <refid_modifier>
       <refid_word>word35<refid_word>
     <refid_modifier>
   <syntactic_relation>
   ...
 <syntactic_relation_level>
 <linguisticAnalysis>
<documentRecord>
<documentCollection>
```

Figure 6: Example of the input and output of the linguistic annotation process.

4. NLP annotation

4.1. Architecture of the NLP line

In ALVIS project, the linguistic annotation processing line follows the following sequence of annotation steps:

- The first tokenisation step builds the reference segmentation of the document. It takes a rough document and produces the token level section of the linguistic annotation one (see figure 6).

- A preliminary semantic unit tagging (called the named entity tagging) aims at identifying named entities and various sorts of unanalizable character strings which would hinder the following linguistic analysis if it

were not identified as semantic units beforehand. This step may or may not associate semantic tags to the identified semantic units.

- The word and sentence segmentations build word and sentence units out of the token sequence. The word segmentation takes the pre-identified semantic units for granted and does not affect the semantic unit level of annotation.

- The morpho-syntactic tagging associates morpho-syntactic and syntactic features to the words identified at the previous step.

- The lemmatiser associates its lemma, *i.e.* its canonical form, to each word. If the word cannot be lemmatized (for instance a number or a foreign word where none of the rules applies), the information is omitted. This module assumes that word segmentation and morpho-syntactic information are provided.

- The terminological tagging identifies new semantic units. This module aims at recognizing terms in the documents differing from named entities, like *gene expression*, *spore coat cell*. It can exploit existing terminological resources. As opposed to the preliminary tagging step, it relies on the word and morpho-syntactic information.

- The syntactic parsing is obviously the most expensive NLP step. Parsing a large document base for indexing is impossible. The syntactic analysis is only applied on a subset of the document base in the acquisition phase. This step enriches the documents with syntactic relations.

- The semantic tagging associates semantic types to pre-identified semantic units according to a an existing ontology of the domain. Eventually, it also outputs new semantic units.

- The anaphora resolution exploits all the existing annotations to identify the antecedents of anaphoric pronoun occurrences. It produces new semantic relations.

- The semantic relation tagging is a very specific NLP steps that either exploits a set a extraction rules to tag semantic relations in the document or projects ontological relations on the document.

In this process of linguistic annotation, a given document is incrementally enriched with linguistic annotations : even of each NLP tool mainly operates on a specific annotation layer, they cooperate to the whole annotation. For instance, two different tools (the named entity and terminological taggers) are able to identify semantic units in the document, the latter enriching the raw tagging made by the former one.

4.2. Results

Our NLP line is intended to annotate large amount of documents and web pages to be indexed in a search engines. In our first experiments, more 18 000 different documents have been analyzed (19 850 000 words). In these experiments, all the NLP steps were applied up to the terminological tagging but only few terms were tagged since the terminological resource was very small.
The initial document size ranges from 1.8 Kbytes to 1 627 Kbytes. The size of the annotated document can be as much as 100 times the initial document size for large textual documents.

5. Conclusion

In this paper, we presented the format that has been adopted to encode the linguistic annotations of documents in the ALVIS project.
Besides incrementality and separability, we also argue that our format meets the requirements of openness, explicitness and consistency that any linguistic annotation framework is supposed to fulfil, according to (Ide et al., 2004).
The counterpart of explicitness is the huge size of the resulting documents, which may be more than 100 times larger than the initial one. This will led us to develop a compression method in order to handle large collections of documents.

6. References

S. Bird and M. Liberman. 1999. Annotation graphs as a framework for multidimensional linguistic data analysis. In Association for Computational Linguistics, editor, *Towards Standards and Tools for Discourse Tagging - Proceedings of the Workshop*, pages 1–10, Somerset.

W. Buntine, K. Valtonen K, and M. Taylor. 2005. The alvis document model for a semantic search engine. In Association for Computational Linguistics, editor, *2nd Annual European Semantic Web Conference*, Heraklion, Crete, May 29.

R. Grishman. 1997. Tipster architecture design document version 2.3. Technical report, DARPA.

N. Ide, L. Romary, and E. de la Clergerie. 2004. International standard for a linguistic annotation framework. *Natural Language Engineering*, 10 (3/4):211–225.

A. Nazarenko, E. Alphonse, S. Aubin, K. Derivire, T. Hamon, D. Mladenic, C. Ndellec, T. Poibeau, D. Weissenbacher, and Q. Zhou. 2004. Report on augmented document representations. Deliverable 5.1, ALVIS.