Ogmios: a scalable NLP platform for annotating large web document collections

Thierry Hamon, Julien Derivière, Adeline Nazarenko
LIPN – UMR 7030
CNRS – Université Paris 13 99 av. J.B. Clément, F-93430 Villetaneuse, FRANCE
Tél. : 33 1 49 40 28 32, Fax. : 33 1 48 26 07 12
firstname.lastname@lipn.univ-paris13.fr

1 Introduction

Search engines like Google or Yahoo offer access to billions of textual web pages. These tools are very popular and seem to be sufficient for a large number of general user queries on the Internet. However, some other queries are more complex, requiring specific knowledge or processing strategies: no really satisfactory solution exists for these requests. There is thus a need for more specific search engines dedicated to specialised domain or users.

Considering the case of text mining in Microbiology for example, it is clear that one needs more than existing search engines given the specificity and the reliability of the information that is sought by scientists. Even if recent developments in biology and biomedicine are reported in large bibliographical databases (e.g. Flybase, specialised on Drosophilia Menogaster or Medline), such databases and the associated searching functionalities are not sufficient to satisfy biologists' specific information needs, such as finding information on gene interactions in order to progressively figure out a whole interaction network. We previously argued that looking for this kind of relational information requires a domain-specific linguistic analysis and parsing of the documents (Alphonse *et al.*, 2004).

The ALVIS project aims at developing an open source search engine, with extended semantic search facilities. Compared to state of the art search engines (like Google, the most popular one), the ALVIS search engine is domain specific. It relies on a specialised crawler, which selects the web pages on terminological grounds. Indexing exploits various types of linguistic and domain specific annotation (cf. figure 1). A dedicated interface helps users to refine queries and analyse the content of the retrieved documents. The ALVIS search engine processes the query more accurately, taking into account the topic and the context of search to refine both the query and the document analysis.

This paper focuses on the design and the development of the text processing platform, Ogmios, which has been developed in the ALVIS project. The challenges were to handle rather large domain specific collections of documents (typical specialised collections gather hundreds of thousands of documents, rather than hundreds of millions of documents), to analyze documents from the web using a single platform, how heterogeneous they may be, to enrich documents with domain-specific semantic information to allow semantic querying. The present paper shows how the three constraints of genericity, domain semantic awareness and performance can be handled all together.

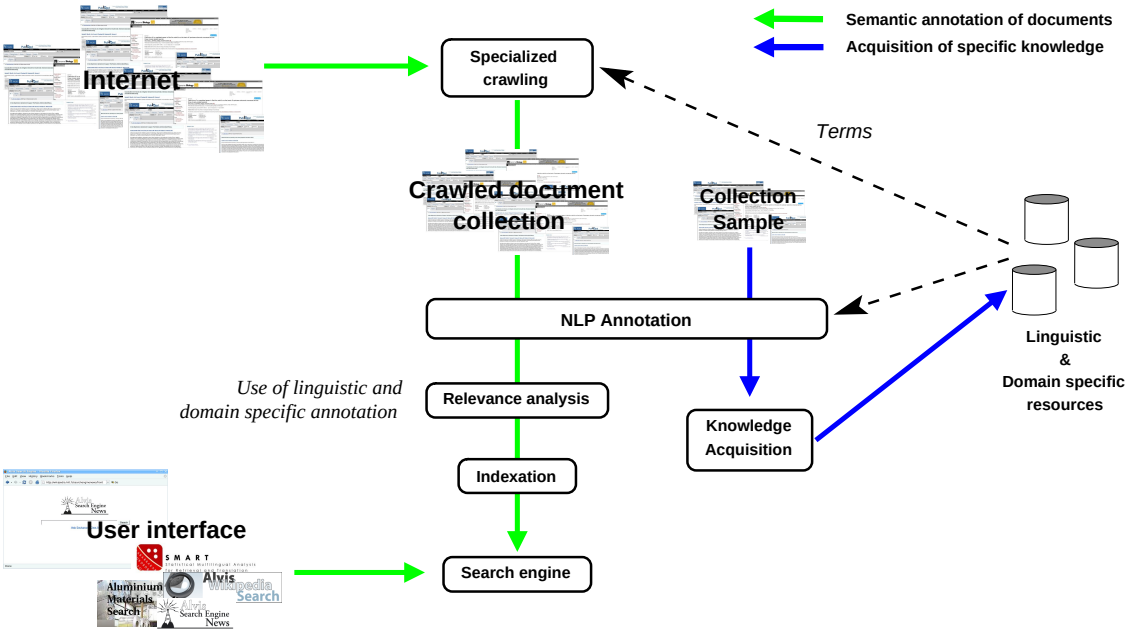

Figure 1. Role of NLP in the ALVIS semantic search engine

The Ogmios platform is a generic one. It is instantiated using existing NLP modules and resources, which can be tuned to specific domains. The figure 1 shows the role of the NLP annotation and the resource acquisition in the whole information retrieval process. For processing texts in the biological domain, we exploited specific named entity dictionaries and terminologies and we adapted a generic syntactic analyzer.

Section 2 gives an overview of the existing platforms designed for document annotation. Sections 3 and 4 describe the global architecture of the platform and its various NLP modules. Section 5 describes the performance of our system on a collection of crawled documents relative to Microbiology, and another collection of search engine news.

2 Background

Several text engineering architectures have been proposed to manage text processing over the last decade (Cunningham *et al.*, 2000) without being in the context of the information retrieval or the linguistic enrichment of very large corpora from the web. Thus, architectures like GATE (Bontcheva *et al.*, 2004), UIMA (Ferrucci *et al.*, 2004) or Textpresso system (Müller *et al.*, 2004), aim at linguistically annotating and exploring medium-sized corpus for the information extraction. LinguaStream (Widlöcher *et al.*, 2005) is designed for mining corpora and carrying out experiments with complex linguistic processing. Those linguistic platforms exploit existing NLP tools which are wrapped and insure the conformity of the intput/output streams. Defining a exchange format is crucial to insure communication between the modules and the integration of the results in external applications. Various exchange formats have been proposed. They are generally based on SGML and more recently on XML. For instance, the exchange formats of GATE, CPSL (Common Pattern Specific Language) and UIMA, CAS (Common Analysis Structure) are based on the annotation format of TIPSTER (Grishman, 1997). However, CAS annotations are stand-off for the sake of flexibility

Processing large collections of web documents imposes some specific constraints: genericity, reduced time processing, and easy tuning to a specialised domain. It appears that GATE or LignuaStream are not well adapted to the context of the specialised information retrieval, and especially to process very large corpora. GATE has been designed as a powerful environment for conception and development of NLP applications in information extraction. Scalability is not the main point in its design, and information extraction deals with small sets of documents. However, we have observed that problems appear even on small sets of documents. The Textpresso system (Müller *et al.*, 2004) pursues the same purpose as ours: developping a generic architecture to process specialised corpora. It has been specifically developed to mine biological documents, abstracts as well as articles. It is designed as a curation system extracting gene-gene interaction that is also used as a search engine. It has been evaluated on a medium-sized collection composed of 16,000 abstracts and 3,000 full text articles related to Caenorhabditis elegans. Based on an external linguistic annotation platform, namely GATE, the KIM platform (Popov *et al.*, 2004) can be considered as a "meta-platform". It is designed for ontology population, semantic indexing and information retrieval. KIM has been integrated in massive semantic annotation projects such as the SWAN clusters and SEKT. The authors identify scalability as a critical parameter for two reasons: (1) it was necessary to process large amounts of data, in order to build and train statistical models for information extraction; (2) it has to support its own use as an online public service. However, no information is provided to evaluate its scalability. Document collections could be processed in UIMA thanks to the Collection Processing Engine, which proposes amoung others performance monitoring and parallelization.

Those linguistic annoation platforms answer partly to our constraints. They are rather mining environment than platform designed to annotate very large collection of documents issus from the web. In that respect, we choose to propose a NLP architecture able to analyze large amounts of documents, and focus on the efficiency of the processing.

3 A modular and tunable platform

In the development of Ogmios, we focused on tool integration. Our goal is to efficiently exploit and combine existing NLP tools rather than developing new ones. But integrating heterogeneous tools and nevertheless achieve good performance in document annotation was challenging. We developped NLP systems only when no other solution was available. And we preferably chose GPL or free licence software.

Ogmios platform was designed to perform various combinations of annotations. In that respect, the platform can be viewed as a modular software architecture that can be configured to achieve various tasks of corpus design.

3.1 *Specific constraints*

The reuse of NLP tools imposes specific constraints regarding software engineering and processing domain-specific documents requires tuning resources to better fit the data.

From the software engineering point of view, the constraints mainly concern the input/output formats of the integrated NLP tools. Each tool has its own input and output format. Linking together several tools requires defining an interchange format. The second type of constraints is the cost linguistic analysis in terms of processing time (the main pitfall is the deep syntactic dependency parsing which is time consuming), which lead us to design a distributed architecture.

A domain specific annotation platform also requires lexical and ontological resources or the tuning of NLP tools such as the Part-of-Speech tagger or parser. For instance, we have argued in (Alphonse et al., 2004) that identification of gene interaction requires gene name tagging, which relates to traditional named entity recognition, term recognition and a reliable syntactic analysis.

3.2 *General architecture*

The different processing steps are traditionally separated in modules (Bontcheva *et al.*, 2004). Each module carries out a specific processing step: named entity recognition, word segmentation, POS tagging, parsing, semantic tagging or anaphora resolution. It wraps an NLP tool to ensure the conformity of the input/output format with the DTD. Annotations are recorded in an XML stand-off format to deal with the heterogeneity of NLP tools input/output (the DTD is fully described in (Taylor, 2006, Nazarenko *et al.*, 2006)). The modularity of the architecture simplifies the substitution of a tool by another.

Tuning to a specific field is insured by the exploitation of specialised resources by each module. For instance, a targeted species or gene list can be added to the biology-specific named entity recognizer to process Medline abstracts. In the ALVIS project, the problem of acquiring automatically these specialised resources from a training corpus is also addressed (see Figure 2 and (Alphonse *et al.*, 2004)) but this question falls out of the scope of the present paper.

Figure 2 gives an overview of the architecture. The various modules composing the NLP line are represented as boxes. The description of these modules is given in section 4. The arrows represent the data processing flow. Intermediary levels of annotations can be produced if the complete NLP line is not used. For instance, anaphora resolution is seldom activated.

We assume that input web documents are already downloaded, cleaned, encoded into the UTF-8 character set, and formatted in XML (Taylor, 2006). Documents are first tokenized to define offsets to ensure the homogeneity of the various annotations. Then, documents are processed through several modules: named entity recognition, word and sentence segmentation, lemmatization, part-of-speech tagging, term tagging, parsing, semantic tagging and anaphora resolution.

Although this architecture is quite traditional, few points should be highlighted:

– Tokenization computes a first basic non-linguistic segmentation of the document, which is used for further reference. The tokens are the basic textual units in the text pro-

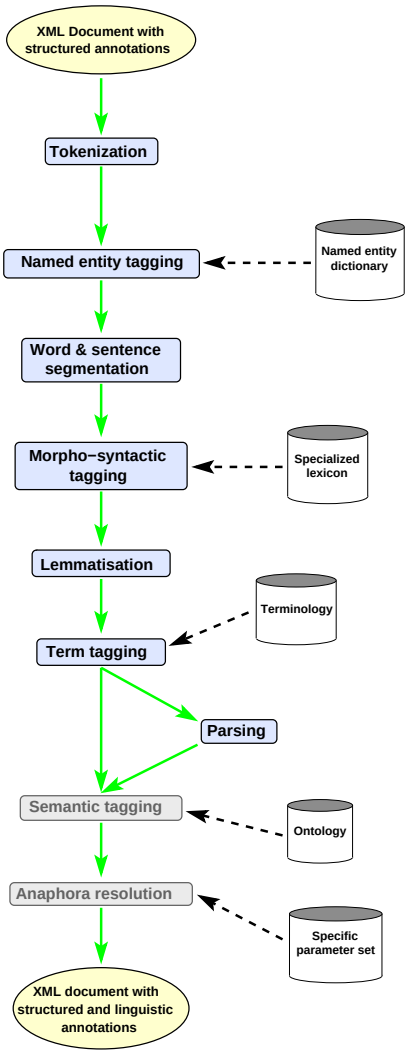

Figure 2. Ogmios architecture

cessing line. Tokenization serves no other purpose but to provide a starting point for segmentation. This level of annotation follows the recommendations of the TC37SC4/TEI workgroup, even if we refer to the character offset rather than pointer mark-up (TEI element ptr) in the textual signal to mark the token boundaries. To simplify further processing, we distinguish different types of tokens: alphabetical tokens, numerical tokens, separating tokens and symbolic tokens.

– Named Entity tagging takes place very early in the NLP line because unrecognized named entities hinder most NLP steps, in many sublanguages;

– Terminological tagging is used as such but is also considered as an aid for syntactic parsing. As this latter step is time consuming, we exploit the fact that terminological analysis simplifies the parsing cost.

For each document, the NLP modules are called sequentially. The outputs of the modules are stored in memory until the end of the processing. XML output is recorded at the end of the document processing.

The linguistic analysis of the documents are distributed according to the client/server model. The server manages the documents distribution by sending them to the clients and gathering the analysed documents coming from the clients. Each client performs the whole linguistic annotation described at the figure 2.

4 Description of the NLP modules

This section describes the different NLP modules. It also explains what is the expected impact of each linguistic annotation step on information retrieval (IR) or information extraction performance.

4.1 *Named Entity tagging*

The Named Entity tagging module aims at annotating semantic units, with syntactic and semantic types. Each text sequence corresponding to a named entity is tagged with a unique tag corresponding to its semantic value (for example a "gene" type for gene names, "species" type for species names, etc.). We use the TagEN Named Entity tagger (Berroyer, 2004) , which is based on a set of linguistic resources and grammars. Named entity tagging has a direct impact on search performance when the query contains one or two named entities, as those semantic units are have a high discriminative power in IR.

4.2 *Word and sentence Segmentation*

This module identifies sentence and word boundaries. We use simple regular expressions, based on the algorithm proposed in (Grefenstette *et al.*, 1994). Part of the segmentation has been implicitly performed during the Named Entity tagging to solve some ambiguities such as the abbreviation dot in the sequence "B. subtilis", which could be understood as a full stop if it were not analyzed beforehand.

4.3 *Morpho-syntactic tagging*

This module aims at associating a part of speech (POS) tag to each word. It assumes that the word and sentence segmentation has been performed. We are using a probabilistic Part-Of-Speech tagger: TreeTagger (Schmid, 1997). The POS tags are not used as such for IR but POS tagging facilitates the rest of the linguistic processing.

4.4 *Lemmatization*

This module associates its lemma, i.e. its canonical form, to each word. The experiments presented in (Moreau, 2006) show that this morphological normalization increases the performance of search engines. If the word cannot be lemmatized (for instance a number or a foreign word), the information is omitted. This module assumes that word segmentation and morpho-syntactic information are provided. Even if it is a distinct module, we currently exploit the TreeTagger output which provides lemma as well as POS tags.

4.5 *Terminology tagging*

This module aims at recognizing the domain specific phrases in a document, like *gene expression* or *spore coat cell*. These phrases considered as the most relevant terminological items. They can be provided through terminological resources such as the Gene Ontology (Consortium, 2000), the MeSH (Mesh, 1998)(MeSH) or more widely UMLS (National library of medicine, 2003). They can also be acquired through corpus analysis (see Figure 1). Providing a given terminology tunes the term tagging to the corresponding domain. Previous annotation levels as lemmatization and word segmentation but also named entities are required. The goal in identifying domain specific phrases in the documents is the same as for the named entitiy recognition, i.e. to identify the relevant semantic units. Even if previous experiments (see (Lewis, 1992) among others) have shown a little impact of the phrases on IR performance, we argue that terminology should have a more significant impact on specialised search engines, as a terminology is relevant for a specific domain. In addition to that, a normalization procedure can associate a canonical form to any phrase occurrence (e.g. *gene expression*, *expression of gene*, *expressed gene*). This normalization step is similar to the lemmatization one for words. Gathering associated variants under a single form modifies the phrase frequencies and thus affects IR.

4.6 *Parsing*

The parsing module aims at exhibiting the graph of the syntactic dependency relations between the words of the sentence. Parsing is a time and resource-consuming NLP, especially when compared to other NLP tasks like named entity recognition or part-of-speech tagging. As mentioned above, the syntactic analysis is especially important for the tasks that involve relations between entities (either information extraction or relational queries such as "*X's speeches as opposed to speeches on or relative to X*)". However, this technology is not yet fully compatible with information retrieval or extraction.

Even if processing time is a critical point for syntactic parsing, we argue that it may enhance the semantic access to web documents. On the one hand, it is usually not necessary to parse the entire documents. A good filtering procedure may select the more relevant sections to parse. We still have to develop a method for pre-filtering the textual segments that are worth parsing as proposed in (Nédellec *et al.*, 2001). On the other hand, as we will show in Section 3.2, a good recognition of the terms can significantly reduce the number of possible parses and consequently the parsing processing time (Aubin *et al.*, 2005).

In Ogmios, the word level of annotation is required in the parser input. Depending on the choice of the parser, the morpho-syntactic level may be needed or not. We chose to integrate the Link Grammar Parser (Sleator *et al.*, 1993). The parser presents several advantages among which the robustness, the good quality of the parsing, the underlying dependency formalism and the declarative format of its lexicon. We also adapt LP to parse Medline abstracts dealing with genomics. More details are given in (Aubin *et al.*, 2005)

4.7 *Semantic type tagging and anaphora resolution*

The last modules are currently under test and should be integrated in the next release of the platform. The semantic type tagging associates to the previously identified semantic units tags referring to ontological concepts. This allows a semantic querying of the document base.

The anaphora resolution module establishes coreference links between the anaphoric pronoun occurrences and the antecedents they refer to. Even if solving anaphora has a small impact on the frequency counts and therefore on IE, it increases IE recall: for instance *it inhibits Y* may stand for *X inhibits Y* and must be interpreted as such in a extraction engine dealing with gene interactions.

5 Performance analysis

We carried out an experiment on two collections ofweb documents. The first one gathers 55,329 web documents from the biological domain (henceforth BIO). Most documents have an XML size between 1KB and 100KB. The size of the biggest document is about 5.7 MB. Figure 3 shows the distribution of the input document size (both axes are on a log scale). The second document collection is composed of 48,422 news related to the search engines (henceforth SEN). All the documents have a size between 1 and 150 KB.

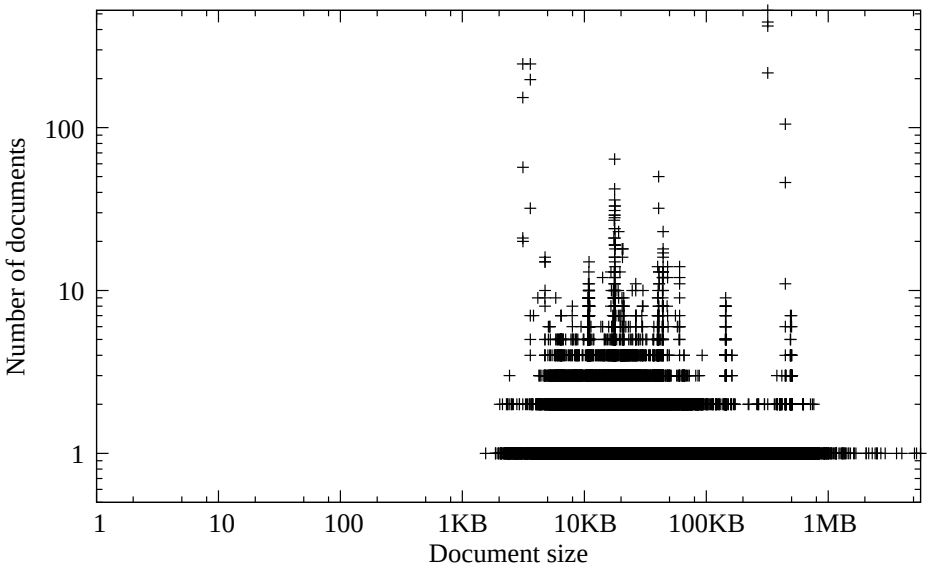

Figure 3. Distribution of document size

All the documents went through all NLP modules, up to the term tagging (as mentioned before, the goal is not to parse the whole documents but only some filtered part of them). To annotate the BIO collection, we used a 375,000 term list, issued from the MeSH and Gene Ontology, while for the SEN collection, the list was composed of 17,341 terms automatically extracted. In both cases, we exploited a 400,000 named entity list, including species and gene names for BIO, and names of person, software and company for SEN.

We used 20 machines to annotate these documents. Most of these machines were standard Personal Computers with 1GB of RAM and 2.9 or 3.1 GHz processor. We also used

a computer with 8GB of RAM and two 2.8GHz Xeon (dual-core) processors. Their operating system were either Debian Linux or Mandrake Linux. The server and three NLP clients were running on the 8GB/biprocessor. Only one NLP client was running on each standard Personal Computer.

Even if a real benchmark requires several tests to evaluate the performance, we consider this performance as an interesting indication of the platform processing time. Timers are run between each function call in order to measure how long each step is (user-time-wise). We used the functions provided in the Time::Hires Perl package. All the time results are recorded in the annotated XML documents.

	Average number of units by document	Total number of units in the document collection
Tokens	5,021.9	277,846,470
Named entities	81.88	4,530,368
Words	1,912.65	105,821,243
Sentences	85.41	4,726,003
Part-of-speech tags	1,883.5	104,208,536
Lemma		
Terms	250.76	13,874,089

Table 1. Average and total numbers of linguistic units

The documents of the BIO collection have been annotated in 35 hours, while the annotation of SEN was completed in 3 hours and 15 minutes.

Table 1 shows the total number of entities found in the BIO collection. 106 million words and 4.72 million sentences were processed; 4.53 million named entities and 13.9 million domain specific phrases were identified. Each document contains, on average, 1,913 words, 85 sentences, 82 named entities and 251 domain specific phrases. 147 documents contained no words at all; they therefore underwent the tokenization step only. One of our NLP clients processed a 414,995 word document.

Table 2 shows the average processing time for each document of BIO. Each document has been processed in 37 seconds on average. Due to the exploited resource, the most time-consuming steps are the term tagging (56% of the overall processing time) and the named entity recognition (16% of the overall processing time). The average time processing for the SEN documents is 2 seconds.

The whole BIO document collection, except two documents, has been analysed. Thanks to the distribution of the processing, the problems occuring on a specific document had no consequence on the rest of the process. The clients in charge of the analysis of these documents have been simply restarted.

The performance we get on this collection show the robustness of the NLP platform, and its ability to analyse large and heterogeneous collection of documents in a reasonable time.

	Average time processing	Percentage
Loading XML input doc.	0.38	1.02
Tokenization	0.7	1.88
Named entity recognition	6.12	16.42
Word segmentation	5.19	13.92
Sentence segmentation	0.18	0.48
Part-of-speech tagging Lemmatization	1.84	4.94
Terms tagging	20.83	55.89
Rendering XML output doc.	2.03	5.45
Total	37.27	100

Table 2. Average time for one document processing (in seconds)

6 Conclusion

We have presented in this paper a platform that has been designed to enrich specialised domain documents with linguistic annotations. While developments and experiments have been performed on biomedical texts, we assume that this architecture is generic enough to process other specialised documents. The platform is designed as a framework using existing NLP tools which can be substituted by others if necessary. Several NLP modules have been integrated: named entity tagging, word and sentence segmentation, POS tagging, lemmatization, term tagging, and syntactic parsing. Semantic type tagging and anaphora resolution are currently being under stress.

We also focused on the system performance, since this point is crucial for most Internet applications. We have experimented a distributed design of the platform, by splitting the corpus in equal parts: this strategy dramatically increased the overall performance (see (Ravichandran *et al.*, 2004). We have also shown that Ogmios is a robust NLP platform with respect to the high heterogeneity of the document sizes and types. These performances lead us to consider its combination with the specialised crawler to help the creation of annotated corpora from the web.

These first experiments show that a deep analysis of web documents is possible. Besides the necessary improvement the Ogmios platform, our next goal is to assess the impact of NLP on IR performance. Our hypothesis is that this impact should be higher in the case of a specialised search engines than for a generic IR framework, on which the IR-NLP cooperation has mainly been tested until now. Specific experiments are currently carried out in the ALVIS project to test the potential resulting enhanced functionalities on a microbiological search engine.

7 Acknowledgements

This work is supported by the EU 6th Framework Program in the IST Priority under the ALVIS project. The material benefits from interactions with the ALVIS partners, especially with INRA-MIG.

8 References

Alphonse E., S. Aubin, P. Bessieres, G. Bisson, T. Hamon, S. Laguarrigue, A. P. Manine, A. Nazarenko, C. Nedellec, M. O. A. Vetah, T. Poibeau and D. Weissenbacher (2004) Event-based Information Extraction for the biomedical domain: the Caderige project. *Workshop BioNLP (Biology and Natural language Processing), Conférence Computational Linguisitics (Coling 2004)*, *Geneva*, *2004*.

Aubin S., A. Nazarenko and C. Nédellec (2005) Adapting a general parser to a sublanguage. *The international conference RANLP 2005*, *Borovets, Bulgaria*, *2005*.

Berroyer J.-F., 'TagEN, un analyseur d"entités nommées : conception, développement et évaluation'. Mémoire de D.E.A. d'Intelligence Artificielle, Université Paris-Nord, 2004.

Bontcheva K., V. Tablan, D. Maynard and H. Cunningham. (Sept-Dec (2004)) 'Evolving GATE to meet new challenges in language engineering', *Natural Language Engineering*, vol. 10, num. 3-4, 349-374.

Consortium T. G. O. (may (2000)) 'Gene Ontology: tool for the unification of biology', *Nature genetics*, vol. 25, 25-29.

Cunningham H., K. Bontcheva, V. Tablan and Y. Wilks (2000) Software Infrastructure for Language Resources: a Taxonomy of Previous Work and a Requirements Analysis. *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2)*, *Athens*, *2000*.

Ferrucci D. and A. Lally. (Sept-Dec (2004)) 'UIMA: an architecture approach to unstructured information processing in a corporate research environment', *Natural Language Engineering*, vol. 10, num. 3-4, 327-348.

Grefenstette G. and P. Tapanainen (1994) What is a word, what is a sentence? problems of tokenization. *The 3rd International Conference on Computational Lexicography*, *Budapest*, *1994*. pp. 79-87.

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

Lewis D. (1992) An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. *proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, *Copenhaguen, Denmark*, *1992*.

MeSH. 'Medical Subject Headings'. Library of Medicine, Bethesda, Maryland, WWW page `http://www.nlm.nih.gov/mesh/meshhome.html`,, 1998.

Moreau F., 'Revisiter le couplage traitement automatique des langues et recherche d'information'. Thèse d'informatique, Université de Rennes 1, décembre (2006).

Müller H.-M., E. E. Kenny and P. W. Sternberg. (Nov (2004)) 'Textpresso: an ontology-based information retrieval and extraction system for biological literature', *PLoS Biology*, vol. 2, num. 11, 1984-1998.

National Library of Medicine (ed.). (2003) UMLS Knowledge Source. 13^{th} edition.

Nazarenko A., E. Alphonse, J. Derivière, T. Hamon, G. Vauvert and D. Weissenbacher (2006) The ALVIS Format for Linguistically Annotated Documents. *Proceedings of LREC 2006*.

Nédellec C., M. O. A. Vetah and P. Bessières (September (2001)) Sentence Filtering for Information Extraction in Genomics, a Classification Problem. *Proceedings of 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 01)*, *Freiburg, Germany*, *2001*.

Popov B., A. Kiryakov, D. Ognyanoff, D. Manov and A. Kirilov. (Sept-Dec (2004)) 'KIM – a semantic platform for information extraction and retrieval', *Natural Language Engineering*, vol. 10, num. 3-4, 375-392.

Ravichandran D., P. Pantel and E. Hovy (2004) The Terascale Challenge. *Proceeding of KDD Workshop on Mining for and from the Semantic Web (MSW'04)*, *Seattle, USA*, *2004*.

Schmid H. (1997) Probabilistic Part-of-Speech Tagging Using Decision Trees. D. Jones and H. Somers (eds.), *New Methods in Language Processing Studies in Computational Linguistics*.

Sleator D. D. and D. Temperley (1993) Parsing English with a link grammar. *Third International Workshop on Parsing Technologies*.

Taylor M. 'Report on metadata frameworks, including concrete representations, for network nodes and semantic document analyses'. ALVIS Deliverable 3.1, 2006.

Widlöcher A. and F. Bilhaut (juin (2005)) La plate-forme LinguaStream : un outil d'exploration linguistique sur corpus. *Actes de la conférence TALN 2005*, *Dourdan, France*, *2005*. pp. 517-522.