A robust linguistic infrastructure for efficient web content analysis: the ALVIS project

Sophie Aubin, Julien Derivière, Thierry Hamon, Adeline Nazarenko, Thierry Poibeau, and Davy Weissenbacher

Laboratoire d'Informatique de Paris-Nord, Villetaneuse, France,
first.lastname@lipn.univ-paris13.fr,
http://www-lipn.univ-paris13.fr

Abstract. This paper focuses on the design and the development of a text processing architecture exploiting specialized NLP tools, to produce linguistically annotated documents. This architecture is instanciated using existing NLP modules and resources which need to be tuned to specific domains. Taking as an example the biological domain, we show how a syntactic analyser can be adapted to this domain. We focus on parsing since it exhibits various kinds of adaptation, ranging from unknown words analysis to specific vocabulary (terms, named entities) and structure identification.

1 Introduction

Search engines like Google or Yahoo offer acces to billions of textual webpages. 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 solutions exist for these requests. There is thus a need for more specific search engines dedicated to specialized domain or users.

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), ALVIS is intended to process the query more accurately, taking into account the topic and the context of search to refine both the query and the document analysis. The goal is achieved through a peer-to-peer architecture: the system is made of general search capabilities enhanced with specialized, domain-specific, knowledge-based systems called crawlers.

Crawlers may use specific resources and tools to extract relevant information from the text. For example, recent developments in biology and biomedicine are reported in large bibliographical databases either focused on a specific species (e.g. Flybase, specialized on Drosophilia Menogaster) or not (e.g. Medline). This type of information sources is crucial for biologists but there is a lack of tools to explore them and extract relevant information. Content analysis tools (named entity and technical term recognizers) have recently gained a certain success on these domains. However, being able to fill structured databases from free texts (a task known as event-based Information Extraction) is still a challenge.

Given the specificity and the reliability of the information that is sought by scientists, it is clear than one needs more than existing search engines. For example, biologists working on genomic data mainly focus on biological interactions between genes and/or proteins. They more specifically look for the source and target of the interaction as well as for the type of that interaction. We think that looking for this kind of information requires a linguistic analysis and even a syntactic parsing of the documents.

We developed an NLP architecture which enriches documents (such as downloaded Medline abstracts) with linguistic information. This platform is designed to be generic for processing specialized documents. Actually, we carry out experiments and evaluate it on biomedical texts. It can be viewed as a framework integrating existing NLP tools. Each tool is wrapped in the platform. Most of them can be tuned by adding domain specific lexical resources.

This paper focuses on the design and the development of a text processing architecture exploiting specialized NLP tools, to produce linguistically annotated documents. This architecture is instanciated using existing NLP modules and resources which need to be tuned to specific domains. Taking as an example the biological domain, we show how a syntactic analyser can be adapted to this domain. We chose to focus on parsing since it exhibits various kinds of adaptation, ranging from unknown words analysis to specific vocabulary (terms, named entities) and structure identification.

In section 2, we give an overview of the existing architectures designed for document annotation. Then the platform is described in section 3. We enumerate the list of modules that are integrated and give a detailed description of the adaptation strategy for the syntactic module in section 5. Lastly, processing time, a key issue for this kind of architecture, is evaluated in section 6.

2 Background

Several text engineering architectures have been proposed to manage text processing over the last decade [1]. GATE (General Architecture for Text Engineering) [2] has been essentially designed for information extraction tasks. It aims at reusing NLP tools in built-in components. The interchange annotation format (CPSL – Common Pattern Specific Language) is based on the TIPSTER annotation format [3].

Based on an external linguistic annotation platform, namely GATE, the KIM platform [4] can be considered as a "meta-platform". It is an ontology population, semantic indexing and information retrieval architecture. KIM has been integrated in massive semantic annotation projects such as the SWAN clusters[1] and SEKT[2]. The authors identify scalability as a critical parameter for two reasons: (1) it has to be able 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.

[1] http://deri.ie/projects/swan

[2] http://sekt.semanticweb.org

UIMA[5], a new implementation architecture of TEXTRACT [6], is similar to GATE. It mainly differs from GATE in the data representation model. UIMA is a framework for the development of analysis engines. It offers components for the analysis of unstructured information streams as HTML web pages. These components are supposed to range from lightweight to highly scalable implementations.The UIMA annotation format is called CAS (Common Analysis Structure). It is mainly based on the TIPSTER format [3]. Annotations in the CAS are stand-off for the sake of flexibility. Documents can be processed either at a single document level or at a collection level. Collections are handled in UIMA by the Collection Processing Engine, which has some interesting features such as filtering, performance monitoring and parallelization.

The Textpresso system [7] has been specifically developed to mine biological documents, abstracts as well as articles. Focusing on *Caenorhabditis elegans*, the system processes 16,000 abstracts and 3,000 full text articles. It is designed as a curation system extracting gene-gene interaction that is also used as a search engine. NLP modules are integrated: tokenizer, sentence segmentation, Part-Of-Speech (POS) tagging, and an ontology tagging based on information provided by Gene Ontology[8].

While Textpresso is specifically designed for biomedical texts, our platform is near to GATE in its aim: proposing a generic platform to process large document collections.

Generally, very little information is given to evaluate the behavior of the systems on a collection of documents whereas from our point of view, this aspect is crucial for such a system. Our first test shows that GATE is not suited to process large collections of documents. GATE has been designed as a powerful environment for conception and development of NLP applications in information extraction. Scalability is not central in its design, and information extraction deals with small sets of documents. However, we have observed that problems appear on small sets of documents. We choose to propose a platform able to analyse large amounts of documents, and focus on the efficiency of the processing.

3 A modular and tunable platform

We developed a platform exploiting existing NLP tools rather than developing new ones[3], and quickly annotating a large number of documents. The platform allows us to test the various combinations of annotations to identify which ones have a significant impact on the extraction rule learning. In that respect, the platform can be viewed as a modular software architecture tunable according to the targeted domain.

[3] We developed NLP systems only when no other solution was available. We preferably chose GPL or free licence softwares when possible.

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, constraints concern above all the heterogeneity of 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. Testing various combination of annotations, including processing time of some linguistic analysis (the main pitfall is syntactic dependency parsing which is time consuming) incite us to propose a distributed architecture.

Proposing a platform to annotate biomedical texts implies also NLP constraints like the availability of lexical and ontological resources, or tuning of NLP tools to improve Part-of-Speech tagging or parsing. Specialized linguistic processing can be required according to specific domains. For instance, we have argued in [9] that identification of gene interaction requires gene name tagging, term recognition and a reliable syntactic analysis.

3.2 General architecture

The different processing steps are traditionally separated in modules [2]. Each module carries out a specific processing: named entity recognition, word segmentation, POS tagging and parsing. 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 heterogeneousness of NLP tools input/output (the DTD is fully described in [10]).

The modularity of the architecture simplifies the substitution of a tool by another. This implies module switching without any impact on the whole architecture.

Tuning to a specific field or species is insured by the resources used by each module. For instance, a targeted species gene list can be added to the biology-specific named entity recognizer to process Medline abstracts. It only depends on the availability of such a resource.

Figure 1 gives an overview of the architecture. The various modules composing the NLP line are represented as boxes. The description of these modules are given in section 4. The arrows represent the data processing flow. The dotted arrows represent alternative types of outputs that the platform may produce.

We assume that input web documents are already downloaded, cleaned, encoded into the UTF-8 character set, and formatted in XML [10]. Documents are first tokenized to define offsets for further linguistic units to annotate and 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, and parsing.

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

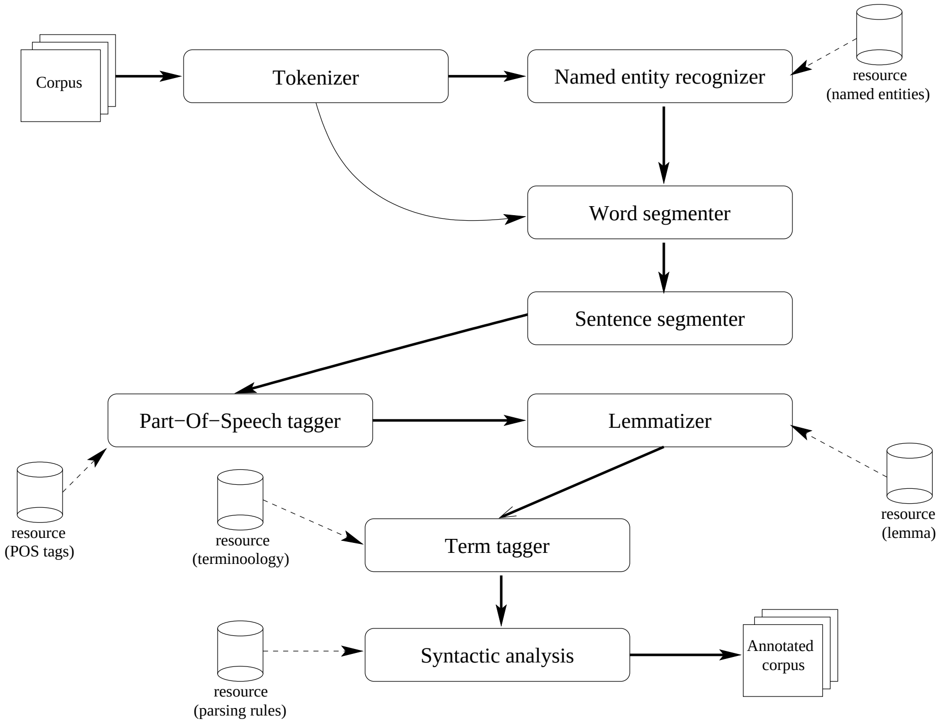

Fig. 1. Architecture of the text line processing.

- Tokenisation is a first step to compute a first basic non-linguistic segmentation, and is used for further reference. Those are the basic textual units in the text processing 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.

4 Description of the NLP modules

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 a document processing.

This section describes the general specification of the various modules of the NLP line that produces the various types of linguistic annotations. The tools wrapped in the modules are examples of NLP tools integration, and they can be substituted by others.

Name 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 will be tagged with a unique tag corresponding to its semantic value (for example a ”gene” type for gene names, ”species” type for species names, etc.). All these text sequences are also assumed to be equivalent to nouns: the tagger dynamically produces linguistic units equivalent to words or noun phrases. We use the TagEN Named Entity tagger [11], which is based on a set of linguistic resources and grammars.

Word and sentence Segmentation This module identifies sentence and word boundaries. We use simple regular expressions, based on the algorithm proposed in [12]. Part of the segmentation has been implicitly performed by the Named Entity tagging to solve some ambiguities: identification of the sequence ”B. subtilis”, and providing information on ”B.” as a short form of ”Bacillus”. Word and sentence segmentation step is simplified.

Morpho-syntactic tagging This module aims at associating the part of speech (POS) tag to each word.It assumes that the word and sentence segmentation has been performed. We are using the probabilistic Part-Of-Speech tagger TreeTagger [13].

Lemmatisation The module 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 apply), the information is omitted. It assumes that word segmentation and morpho-syntactic information are provided. While it is a distinct module, we currently exploit the TreeTagger's output which provides lemma as well as POS tags. An external resource could be required depending on the lemmatizer and the domain tuning requirements. This resource would provide association between the inflectional forms of a word and its lemma.

Terminology tagging This module aims at recognizing terms in the documents differing from named entities, like *gene expression*, *spore coat cell.* Term lists can be provided as terminological resources such as the Gene Ontology [14], the MeSH [15] or more widely UMLS [16]. They can also be acquired through corpus analysis. Providing a given terminology tunes the term tagging to the corresponding domain. Previous annotation levels as lemmatisation and word segmentation but also named entities are required.

Parsing The parsing module aims at exhibiting the graph of the syntactic dependency relations between the words of the sentence. The word level of annotation is required. Depending on the choice of the parser, the morphosyntactic level may be needed. Processing time is a critical point for syntactic parsing, but we argue that a good recognition of the terms can reduce significantly the number of possible parses and consequently the parsing processing time. Term identification is therefore performed prior to parsing. The Link Grammar Parser[17] is integrated. More details are given in the next section.

5 Tuning a syntactic analyzer to the biological domain

This section is specificaly devoted to the way the syntactic analyser has been tuned to analyze texts from the biological domain. It is however a typical example of the adaptation strategy we adopted for most NLP modules.

Parsing is a time and resource consuming task for NLP, especially when compared to other tasks like named entity recognition or part-of-speech tagging. This technology is not yet compatible with Information Retrieval: the system has to propose a set of relevant documents in less than one second and parsing in this context is simply not realistic. It is even not sure that parsing is really useful to provide more accurate answers to traditional, simple queries made of a few keywords.

However, in order to extract structured pieces of information from texts, one needs to link isolated chunks of texts together. Most of the time, chunks of texts correspond to named entity and relations are expressed through verbs or predicative nouns. We thus need a reliable and precise analysis of syntactic relations between phrases. For those reasons, we chose to integrate a symbolic dependency-based parser seemed (in contrast with a constituent-based parser).

Instead of redeveloping new parsers for each sublanguage, we try to define a method for adapting a general parser to a specific sublanguage. This section presents a strategy to adapt the Link Parser (LP) [18] to parse Medline abstracts dealing with genomics. More details are given in [19].

5.1 The initial parser choice

LP presents several advantages among which the robustness, the good quality of the parsing, the adequation of the dependency technique and representation with our IE task and the declarative format of its lexicon.

In order to test various parsers, different corpora were built from Medline[4] abstracts (in English) dealing with transcription in *Bacillus subtilis* [19]. This short description will be based on results obtained from the (`MED-TEST`) corpus. Our test corpus contains 212 randomly selected sentences (5,992 words). The sentences contain an average of 25.4 words (from 8 to 59). Despite its relatively small size, this corpus is a good sample of the sublanguage of genomics. Medline abstracts present the following characteristics : they are made of long and syntactically complex sentences, technical lexicon, scientific notations and numerous agrammatical constructions.

From the results of the evaluation that we did on different parsers, it turned out that dependency-based parsers have better results on long and complex sentences, particularly with coordinations. For example, LP seems to offer better performances than a constituent-based parser applied on Medline abstracts (see ([20] for an experiment using a GPSG parser). This conclusion is shared by [21] who also worked on te same kind of corpus. Other experiments, in the context of the ExtrAns project [22], showed that 76% of 2,781 sentences from a Unix manpage corpus were completely parsed by LP with no regard to the parsing quality, while we reach only 54% on the biological corpus. When looking at the quality of the parses, we noticed different kinds of errors depending either on the biological domain or on more general linguistic difficulties like ambiguous constructions. We propose three solutions to address these issues: text normalization, terminology analysis and lexicon/grammar adaptation.

5.2 Diagnosis and adaptation

Our analysis of the performance of the Link grammar on the biological corpus confirms previous works. The main problems can be classified along the following axes.

"Textual noise" Scientific texts present particularities that we chose to handle in a normalization step prior to the parsing. First, the segmentation in sentences and words was taken off from the parser and enriched with named entities recognition and rules specific to the biological domain. We also delete some extra-textual information that alters parsing quality. Finally, we use dictionaries and transducers to replace genes and species names by two codes, which averts extending the LP dictionary too much.

[4] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi

Unknown words In a corpus made of full Medline abstracts, we identified 6,005 out-of-lexicon forms (45,804 occurences) among 12,584 distinct words, *i.e.* 47.72%. They are mostly latin words, numbers, DNA sequences, gene names, misspellings and technical lexicon.

However, LP includes a module that can assign a syntactic category to an unknown word. It is based on the word suffix. Modifying the morpho-guessing (MG) module seemed a better strategy than extending the dictionary since biological objects differ from an organism to another ([20] also reports a similat process). We then created 19 new MG classes for nouns (*-ase*, *-ity*, etc.) and adjectives (*-al*, *-ous*, etc.) along with their rule. At the same time, we added about 500 words of the biological domain to the LP lexicon in different classes, mainly nouns, adjectives and verbs.

Specific constructions Some words already defined in the LP lexicon present a specific usage in biological texts, which implied some modifications including moving words from one class to another and adaptating or creating rules.

The main motivation for moving words from one class to another is that the abstracts are written by non-native English speakers. This point was also raised by [23]. One way to allow the parsing of such ungrammatical sentences is to relax constraints by moving some words from the countable to the mass-countable class for instance. Some very frequent words present idiosyncratic uses (particular valency of verbs for instance), which induced the modification or creation of rules. Numbers and measure units are omnipresent in the corpus and were not necessarily well described or even present in the lexicon/grammar.

Structural ambiguity We identified two cases of ambiguity that can be partially resolved by using terminology.

Prepositional attachment is a tricky point that is often fixed using statistical information from the text itself [24], a larger corpus [25], the web [26] or external resources such as WordNet [27].

The second major ambiguity factor is the attachment of series of more than two nouns. like in *"two-component signal transduction systems"*. We noticed that such cases often appear inside larger nominal phrases often corresponding to domain specific terms. For this reason, we decided to identify terms in a pre-processing step and to reduce them to their syntactic head. If needed, the internal analysis of terms is added to the parsing result for the simplified sentence. The strategy proposed by [28] that consists in the linkage of the words contained in a compound (for instance *"sporulation_process"*) was excluded. It makes the lexicon size augment and does not reduce complexity for reasons due to the implementation of LP.

Before practically integrating the use of terminology in our processing suite, we made a simulation of this simplification of terms.

5.3 Evaluation

We performed a two-stage evaluation of the modifications in order to measure the respective contribution of the LP adaptation on the one hand and of the term simplification on the other hand.

Corpus and criteria We used a subset (10 files[5]) of the MED-TEST corpus but, contrary to the first evaluation (choice of a parser), we wanted to look at the quality of the whole parse and not only at specific relations.

Table 1 (for the MED-TEST subset) shows the way that out-of-lexicon words (OoL), i.e. unknown (UW) and guessed (GW) words, are handled by giving the percentage of incorrect morpho-syntactic category assignations with the original resources (lp), those adapted to biology (lp-bio) and finally the latter associated with the simplification of terms (lp-bio-t).

	lp		lp-bio		lp-bio-t	
	a	b	a	b	a	b
UW	244	41.4%	53	52.8%	26	19.2%
GW	24	4.2%	72	0%	31	0%
OoL	268	38%	125	22.4%	57	8.8%

a : total MS assignations, b : % of incorrect assignations

Table 1. Incorrect MS category assignations

In Table 2, five criteria inform on the parsing time and quality for each sentence : the number of linkages (NbL), the parsing time (PT) in seconds, the fact that a complete linkage is found or not (CLF), the number of erroneous links (EL) and the quality of the constituency parse (CQ). (NbW) is the average number of words in a sentence which varies with term simplification. The results are given for each one of the three versions of the parser.

	lp	lp-bio		lp-bio-t	
crit.	avg	avg	%/lp	avg	%/lp
NbW	24.05	24.05	100%	18.9	78.6%
NbL	190,306	232,622	122.2%	1,431	0.75%
PT	37.83	29.4	77.7%	0.53	1.4%
CLF	0.54	0.72	133%	0.77	142.6%
EL	2.87	1.91	66.5%	1.15	40.1%
CQ	0.54	0.7	129.6%	0.8	148.1%

Table 2. Parsing time and quality

[5] 141 sentences, 2630 words

UW, GW, NbL, PT and CLF are objective data while EL and CQ necessitate linguistic expertise. The CQ evaluation consisted in the assignation of a general quality score to the sentence.

Results and comments The **extension of the MG module** reduced the number of erroneous morpho-syntactic category assignations (see Table 1) from 38% to 22.4%. 61% of the sentences where one or more assignation error was corrected by the MG module actually have better parsing results (15% have been degraded). More generally, the increase of guessed forms makes category assignation more reliable.

The **extension of the lexicon** and the **normalization of genes and species names** discharged the two modules from 143 assignations out of 268, 50 of which were wrong. 64% of the sentences where one or more assignation error was corrected by the extension of lexicon have better parsing results (18% of the sentences were degraded).

The effect of **rule modification and creation** is difficult to evaluate precisely though it is certain to play a part in the parsing improvement, especially the relaxing of constraints on determiners and inserts.

The most obvious contribution to the better parsing quality is the one of **term simplification**. The drastic reduction in parsing time and number of linkages gives an idea of the reduction of complexity. It is not only due to the smaller number of words since the number of erroneous links is reduced by 60% while the number of words is reduced by only 21.4%. This confirms previous similar studies that showed a reduction of 40% of the error rate on the main syntactic relations with a French corpus.

Remaining errors are mainly due to four different phenomena. First, the normalization step, prior to parsing, needs to be enhanced. Concerning LP, there are still lexicon gaps, wrong class assignations and a still unsatisfactory handling of numerical expressions. In addition, and like [28], we identified a weakness of LP regarding coordination. A specific study of the coordination system in LP and in the biological texts may be necessary. Finally, some ambiguous nominal and prepositional attachments still remain in spite of term simplification. These may be resolved in a post-processing step like in ExtrAns that uses a corpus based approach to retrieve the correct attachment from the different linkages given by LP for a sentence.

Other questions like the feeding of LP with a morpho-syntactically tagged text or the amelioration of the parse ranking in LP were not discussed in this paper but are interesting issues that we intend to study.

6 Performance analysis

We carried out an experiment on a collection of 55329 web documents from the biological domain. Figure 2 shows the distribution of the input document size (both axes are on a log scale). Most documents have an XML size between 1KB and 100KB. The size of the biggest document is about 5.7 MB.

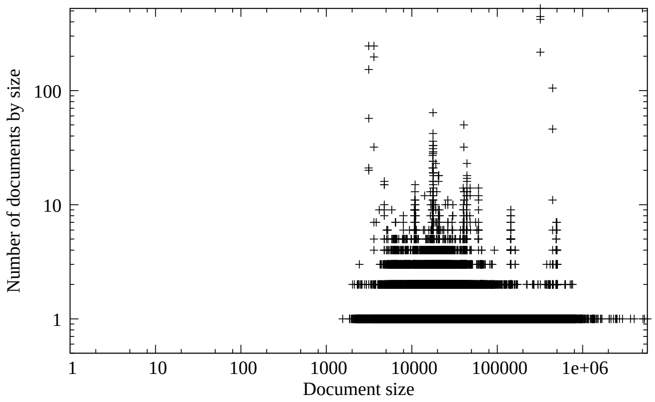

Fig. 2. Range of input document size

We used 16 machines to annotate these documents. Most of these machines are standard Personal Computers with 1GB of RAM and 2.9 or 3.1 GHz processor; Others are four elements of a cluster with a similar configuration and a computer with 8GB of RAM and two 2.8GHz Xeon (dual-core) processors. Their operating system is 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, with a low priority.

We consider these performances to be an indication of the platform processing time (a real benchmark would require several series of tests). Timers are run between each function call in order to measure the processing time of each module. We use the functions provided in the `Time::Hires` package. All the time results are recorded in the XML annotated documents, except for the XML rendering step. The annotation of the documents was completed in three days. Each client processed 2790 documents on average.

Table 3. Average and total of linguistic units.

	Average number of units by document	Total number of units in the document collection
Tokens	5,290.72	276,532,529
Named entities	136.61	7,202,367
Words	1,494.23	79,165,931
Sentences	67.96	3,639,945
Part-of-speech tags and lemma	992.79	53,594,958
Terms	326.29	17,193,097

Table 3 shows the total number of entities found in the document collection. 79 million words and 3.6 million sentences were processed, 7.2 million named entities and 17 million terms were identified. Each document contains, on average, 1494 words, 68 sentences, 136 named entities and 326 terms. 176 documents contained no words at all, they therefore underwent the tokenization step only. One of our NLP clients processed a 350,444 word document.

Table 4 shows the average processing time for each document. Less than one minute is required to process each document. The most time-consuming steps are term tagging (78% of the overall processing time) and named entity recognition (12% of the overall processing time).

Table 4. Average of document processing time in second

	Average document	Percentage
	time processing	Percentage
loading XML input doc.	0.67	1.2
tokenization	0.56	1
named entity recognition	6.68	12
word segmentation	1.39	2.5
sentence segmentation	0.38	0.7
part-of-speech tagging and lemmatization	2.2	4
term tagging	43.63	78.6
Total	55.52	100

These performances are reported for a very first version of the platform. Some bugs had a negative influence on the results and slowed down the indexing process. Only 27 documents out of 55329 have not been annotated (0.04%). Unfortunately, this is mainly due to an unidentified bug in a NLP tool we use, which froze some of our client machines. This problem noticeably increased processing times. We also noticed that certain external NLP tools may encounter difficulties with the UTF-8 character set when used on different machines, with various environments. Lastly, standard personal computers were being used by other connected users. In that respect, the CPU load varied and computers may have been rebooted.

However, despite these problems, this experiment shows that the platform is able to process large corpora on a distributed architecture. We have proven the efficiency of the overall process for semantic crawlers and its accuracy for a precise indexing of documents on the web.

7 Conclusion

We have presented in this paper a platform to enrich specialized 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 specialized documents and to be tuneable for any subdomain of biomedecine. 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 and term tagging. We are working on a better integration of syntactic parsing.

We also focus on the performances of the system, 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 performances (see [29]).

8 Acknoledgements

This work is supported by the EU 6th Framework Programme in the IST Priority under the ALVIS project. The material benefits from interactions with the ALVIS partners. Thanks also to Jean-Michel Daube for feedback.

References

1. Cunningham, H., Bontcheva, K., Tablan, V., Wilks, Y.: Software infrastructure for language resources: a taxonomy of previous work and a requirements analysis. In: Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2), Athens (2000)
2. Bontcheva, K., Tablan, V., Maynard, D., Cunningham, H.: Evolving GATE to meet new challenges in language engineering. Natural Language Engineering **10** (2004) 349–374
3. Grishman, R.: Tipster architecture design document version 2.3. Technical report, DARPA (1997)
4. Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A.: Kim – a semantic platform for information extraction and retrieval. Natural Language Engineering **10** (2004) 375–392
5. Ferrucci, D., Lally, A.: UIMA: an architecture approach to unstructured information processing in a corporate research environment. Natural Language Engineering **10** (2004) 327–348
6. Neff, M.S., Byrd, R.J., Boguraev, B.K.: The talent system: Textract architecture and data model. Natural Language Engineering **10** (2004) 307–326
7. Müller, H.M., Kenny, E.E., Sternberg, P.W.: Textpresso: an ontology-based information retrieval and extraction system for biological literature. PLoS Biology **2** (2004) 1984–1998
8. Consortium, T.G.O.: Creating the Gene Ontology Resource: Design and Implementation. Genome Res. **11** (2001) 1425–1433
9. Alphonse, E., Aubin, S., Bessieres, P., Bisson, G., Hamon, T., Laguarrigue, S., Manine, A.P., Nazarenko, A., Nedellec, C., Vetah, M.O.A., Poibeau, T., Weissenbacher, D.: Event-based information extraction for the biomedical domain: the caderige project. In: Workshop BioNLP (Biology and Natural language Processing), Confrence Computational Linguisitics (Coling 2004), Geneva (2004)
10. Nazarenko, A., Alphonse, E., Derivire, J., Hamon, T., Vauvert, G., Weissenbacher, D.: The alvis format for linguistically annotated documents. Alvis project deliverable, Université Paris-Nord (2006)
11. Berroyer, J.F., Poibeau, T.: TagEN, un analyseur d'entités nommées. LIPN Internal Report, Université Paris-Nord (2004)

12. Grefenstette, G.: Exploration in Automatic Thesaurus Discovery. Kluwer Academic Publishers, Boston, USA (1994)
13. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In Jones, D., Somers, H., eds.: New Methods in Language Processing Studies in Computational Linguistics. (1997)
14. Consortium, T.G.O.: Gene ontology: tool for the unification of biology. Nature genetics **25** (2000) 25–29
15. MeSH: Medical subject headings. WWW page `http://www.nlm.nih.gov/mesh/meshhome.html`, Library of Medicine, Bethesda, Maryland (1998)
16. National Library of Medicine, ed.: UMLS Knowledge Source. 13^{th} edn. (2003)
17. Sleator, D.D., Temperley, D.: Parsing English with a link grammar. In: Third International Workshop on Parsing Technologies. (1993)
18. Sleator, D., Temperley, D.: Parsing English with a Link Grammar. Technical report, Carnegie Mellon University (1991)
19. Aubin, S., Nazarenko, A., Nédellec, C.: Adapting a General Parser to a Sublanguage. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'05), Borovets, Bulgaria (2005) 89–93
20. Grover, C., Lapata, M., Lascarides, A.: A Comparison of Parsing Technologies for the Biomedical Domain. Journal of Natural Language Engineering (2004)
21. Ding, J., Berleant, D., Xu, J., Fulmer, A.W.: Extracting Biochemical Interactions from MEDLINE Using a Link Grammar Parser. In: 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03). (2003) 467–471
22. Mollá, D., Schneider, G., Schwitter, R., Hess, M.: Answer Extraction Using a Dependency Grammar in ExtrAns. Traitement Automatique de Langues (T.A.L.), Special Issue on Dependency Grammars (2000) 145–178
23. Pyysalo, S., Ginter, F., Pahikkala, T., Boberg, J., Järvinen, J., Salakoski, T., Koivula, J.: Analysis of link grammar on biomedical dependency corpus targeted at protein-protein interactions. In: Proceedings of the international Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA). (2004) 15–21
24. Hindle, D., Rooth, M.: Structural Ambiguity and Lexical Relations. In: Meeting of the Association for Computational Linguistics. (1993) 229–236
25. Bourigault, D., Frérot, C.: Ambiguïté de rattachement prépositionnel : introduction de ressources exogènes de sous-catégorisation dans un analyseur syntaxique de corpus endogène. In: Actes des 11mes journées sur le Traitement Automatique des Langues Naturelles, Fès, Maroc. (2004)
26. Volk, M.: Using the Web as Corpus for Linguistic Research. In Pajusalu, R., Hennoste, T., eds.: Tähendusepüüdja. Catcher of the Meaning. A Festschrift for Professor Haldur Õim. Publications of the Department of General Linguistics 3. University of Tartu, Estonia (2002)
27. Stetina, J., Nagao, M.: Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In Zhou, J., Church, K.W., eds.: Proceedings of the Fifth Workshop on Very large Corpora, Beijing, China (1997) 66–80
28. Sutcliffe, R.F.E., Brehony, T., McElligott, A.: The Grammatical Analysis of Technical Texts using a Link Parser. In: Second Conference of the Pacific Association for Computational Linguistics, PACLING'95. (19-22 April 1995)
29. Ravichandran, D., Pantel, P., Hovy, E.: The terascale challenge. In: Proceeding of KDD Workshop on Mining for and from the Semantic Web (MSW'04), Seattle, USA (2004)