LinguaStream: An Integrated Environment for Computational Linguistics Experimentation

Frédérik Bilhaut
GREYC-CNRS
University of Caen
fbilhaut@info.unicaen.fr

Antoine Widlöcher
GREYC-CNRS
University of Caen
awidloch@info.unicaen.fr

Abstract

By presenting the LinguaStream platform, we introduce different methodological principles and analysis models, which make it possible to build hybrid experimental NLP systems by articulating corpus processing tasks.

1 Introduction

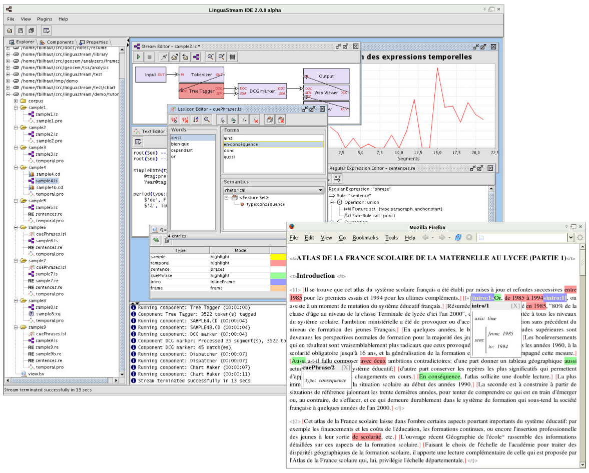
Several important tendencies have been emerging recently in the NLP community. First of all, work on corpora tends to become the norm, which constitutes a fruitful convergence area between task-driven, computational approaches and descriptive linguistic ones. On corpora validation becomes more and more important for theoretical models, and the accuracy of these models can be evaluated either with regard to their ability to account for the reality of a given corpus (pursuing descriptive aims), either with regard to their ability to analyse it accurately (pursuing operational aims). From this point of view, important questions have to be considered regarding which methods should be used in order to project efficiently and accurately linguistic models on corpora.

It is indeed less and less appropriate to consider corpora as raw materials to which models and processes could be immediately applicable. On the contrary, the multiplicity of approaches, would they be lexical, syntactical, semantic, rhetorical or pragmatical, would they focus on one of these dimensions or cross them, raises questions about how these different levels can be articulated within operational models, and how the related processing systems can be assembled, applied on a corpus, and evaluated within an experimental process.

New NLP concerns confirm these needs: recent works on automatic discourse structure analysis, for example regarding thematic structures or rhetorical ones (Bilhaut, 2005; Widlöcher, 2004), show that the results obtained from lower-grained analysers (such as part-of-speech taggers or local semantics analysers) can be successfully exploited to perform higher-grained analyses. Indeed, such works rely on non-trivial processing streams, where several modules collaborate basing on the principles of incremental enrichment of documents and progressive abstraction from surface forms. The LinguaStream platform (Widlöcher and Bilhaut, 2005; Ferrari et al., 2005), which is presented here, promotes and facilitates such practices. It allows complex processing streams to be designed and evaluated, assembling analysis components of various types and levels: part-of-speech, syntax, semantics, discourse or statistical. Each stage of the processing stream discovers and produces new information, on which the subsequent steps can rely. At the end of the stream, various tools allow analysed documents and their annotations to be conveniently visualised. The uses of the platform range from corpora exploration to the development of fully operational automatic analysers.

Other platform or tools pursue similar goals. We share some principles with GATE (Cunningham et al., 2002), HoG (Callmeier et al., 2004) and NOOJ[1] (Muller et al., 2004), but one important difference is that the LinguaStream platform promotes the combination of purely declarative formalisms (when GATE is mostly based on the JAPE language and NOOJ focuses on a unique formalism), and allows processing streams to be designed graphically as complex graphs (when GATE relies on the pipeline paradigm). Also, the

[1] Formerly known as INTEX.



LinguaStream IDE 2.0.0 alpha
File View Plugins Help
Explorer Components Properties
Stream Editor – sample2.lc
Input Tokenizer Tree Tagger DCG marker Output Web Viewer
des expressions temporelles
Segments
Lexicon Editor – cuePhrases.lxl
Text Editor
Words
Forms
Semantics
rhetorical
<Feature Set>
type:consequence
Regular Expression Editor – sentences.re
Mozilla Firefox
File Edit View Go Bookmarks Tools Help
intro/1
axis: time
sem:
from: 1985
to: 1994
cuePhrase/2
type: consequence
Stream terminated successfully in 13 secs

Regular Expression : "phrase"
⇒ Rule : "sentence"
Operator : union
Feature set : {type:paragraph, anchor:start}
Sub-Rule call : ponct

4 entries

Type	Mode	
sample	highlight	🟨
temporal	highlight	🟥
sentence	braces	
cuePhrase	highlight	🟩
intro	inlineFrame	🟦
frame	frame	🟧

```
Running component: Tree Tagger (00:00:00)
Component Tree Tagger: 3522 token(s) tagged
Running component: SAMPLE4.CD (00:00:04)
Running component: SAMPLE4B.CD (00:00:04)
Running component: DCG marker (00:00:04)
Component DCG marker: Processed 35 segment(s), 3522 to
Component DCG marker: 45 match(es)
Running component: Dispatcher (00:00:07)
Running component: Dispatcher (00:00:07)
Running component: Chart Maker (00:00:07)
Running component: Chart Maker (00:00:11)
Stream terminated successfully in 13 secs
```

<t>ATLAS DE LA FRANCE SCOLAIRE DE LA MATERNELLE AU LYCEE (PARTIE 1)</t>

<t>Introduction </t>

<§2> [Cet atlas de la France scolaire laisse dans l'ombre certains aspects pourtant importants du système éducatif: par exemple les financements et les coûts de l'éducation, les formations continues, ou encore l'insertion professionnelle des jeunes à leur sortie de scolarité, etc.] [L'ouvrage récent Géographie de l'école* rassemble des informations détaillées sur ces aspects de la formation scolaire.] [Faisant le choix de l'échelle de l'académie pour traiter des disparités géographiques de la formation scolaire, il apporte une lecture complémentaire de celle qui est proposée par l'Atlas de la France scolaire qui, lui, privilégie l'échelle départementale.] </§>

Figure 1: LinguaStream Integrated Environment

low-level architecture of LinguaStream is comparable to the HoG middleware, but we are more interested in higher-level aspects such as analysis models and methodological concerns. Finally, when other platforms usually enforce the use of a dedicated document format, LinguaStream is able to process any XML document. On the other hand, LinguaStream is more targeted to experimentation tasks on low amounts of data, when tools such as GATE or NOOJ allow to process larger ones.

2 The LinguaStream Platform

LinguaStream is an integrated experimentation environment targeted to researchers in NLP. It allows complex experiments on corpora to be realised conveniently, using various declarative formalisms. Without appropriate tools, the development costs that are induced by each new experiment become a considerable obstacle to the experimental approach. In order to address this problem, LinguaStream facilitates the realisation of complex processes while calling for minimal technical skills.

Its integrated environment allows processing streams to be assembled visually, picking individual components in a "palette" (the standard set contains about fifty components, and is easily extensible using a Java API, a macro-component system, and templates). Some components are specifically targeted to NLP, while others solve various issues related to document engineering (especially to XML processing). Other components are to be used in order to perform computations on the annotations produced by the analysers, to visualise annotated documents, to generate charts, etc. Each component has a set of parameters that allow their behaviour to be adapted, and a set of input and/or output sockets, that are to be connected using pipes in order to obtain the desired processing stream (see figure 2). Annotations made on a single document are organised in independent layers and may overlap. Thus, concurrent and ambiguous annotations may be represented in order to be solved afterwards, by subsequent analysers. The platform is systematically based on XML recommendations and tools, and is able to process any file in this format while preserving its original structure. When running a processing stream, the platform takes care of the scheduling of sub-tasks, and various tools allow the results to be visualised conveniently.

Fundamental principles

First of all, the platform makes use of **declarative representations**, as often as possible, in order to define processing modules as well as their connections. Thus, available formalisms allow linguistic knowledge to be directly "transcribed" and used. Involved procedural mechanisms, committed to the platform, can be ignored. In this way, given rules are both **descriptive** (they provide a formal representation for a linguistic phenomenon) and **operative** (they can be considered as instructions to drive a computational process).

Moreover, the platform takes advantage of the complementarity of analysis models, rather than considering one of them as "omnipotent", that is to say, as able to express all constraint types. We indeed rely on the assumption that a complex analyser can successively adopt several points of view on the same linguistic data. Different formalisms and analysis models allow these different points of view. In a same processing stream, we can successively make use of regular expressions at the morphologic level, a local unification grammar at the phrasal level, finite state transducer at sentential level and constraint grammar for discourse level analysis. The interoperability between analysis models and the communication between components are ensured by a **unified representation** of markups and annotations. The latter are uniformly represented by **feature sets**, which are commonly used in linguistics and NLP, and allow rich and structured information representation. Every component can produce its own markup using preliminary markups and annotations. Available formalisms make it possible to express constraints on these annotations by means of unification. Thereby, the platform promotes **progressive abstraction from surface forms**. Insofar as each step can access to annotations produced upstream, high level analysers often only use these annotations, ignoring raw textual data.

Another fundamental aspect consists in the **variability of analysis grain** between different analysis steps. Many analysis models require a minimal grain to be defined, called *token*. For example, formalisms such as grammar or transducers need a textual unit (such as character or word) to which patterns are applied. When a component requires such a minimal grain, the platform allows to define **locally** the unit types which have to be considered as tokens. Any previously marked unit can be used as such: usual tokenisation in words or any other beforehand analysed elements (syntagms, sentences, paragraphs...). The minimal unit may differ from an analysis step to another and the scope of the available analysis models is consequently increased. In addition, each analysis module indicates antecedent markups to which it refers and considers as relevant. Other markups can be ignored and it makes it possible to partially rise above textual linearity. Combining these functionalities, it is possible to define different **points of view** on the document for each analysis step.

The **modularity** of processing streams promotes the **reusability** of components in various contexts: a given module, developed for a first processing stream may be used in other ones. In addition, every stream may be used as a single component, called macro-component, in a higher level stream. Moreover, for a given stream, each component may be **replaced** by any other **functionally equivalent** component. For a given subtask, a rudimentary prototype may *in fine* be replaced by an equivalent, fully operational, component. Thus, it is possible to compare processing results in rigourously similar contexts, which is a necessary condition for relevant comparisons.

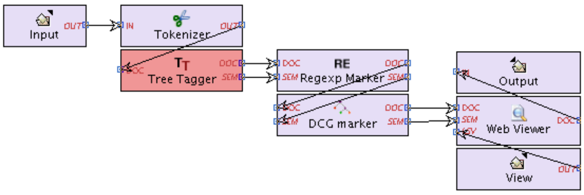


Figure 2: A Simple Processing Stream

Analysis models

We indicated above some of the components which may be used in a processing stream. Among those which are especially dedicated to NLP, two categories have to be distinguished. Some of them consist in ready-made analysers linked to a specific task. For example, morpho-syntactic tagging (an interface with TreeTagger is provided by default) consists in such a task. Although some parameters allow to adapt the associated components to the task (tag set for a given language...), it is impossible to fundamentally modify their behaviour. Others, on the contrary, provide an **analysis model**, that is to say, firstly, a formalism for representing linguistic constraints by means of which the user can express expected processing. This formalism will usually rely on a specific **operational model**. These analysis models allow constraints to be expressed, on surface form as well as on annotations produced by the precedent analysers. All annotations are represented by feature sets and the constraints are encoded by unification on these structures. Some of the available systems follow.

- A system called EDCG (*Extended-DCG*) allows **local unification grammars** to be written, using the DCG (*Definite Clause Grammars*) syntax of Prolog. Such a grammar can be described in a pure declarative manner even if the features of the logical language may be accessed by expert users.

- A system called MRE (*Macro-Regular-Expressions*) allows patterns to be described using **finite state transducers** on surface forms and previously computed annotations. Its syntax is similar to regular expressions commonly used in NLP. However, this formalism not only considers characters and words, but may apply to any previously delimited textual unit.

- Another descriptive, prescriptive and declarative formalism called CDML (*Constraint-Based Discourse Modelling Language*) allows a constraint-based approach of formal description and computation of discourse structure. It considers both textual segments and discourse relations, and relies on expression and satisfaction of a set of primitive constraints (presence, size, boundaries...) on previously computed annotations.

- A **semantic lexicon** marker, a configurable **tokenizer** (using regular expressions at the character level), a system allowing linguistic units to be delimited relying on the XML tags that are available in the original document, etc.

3 Conclusion

LinguaStream is used in several research and educational projects:

- Works on discourse semantics: discourse framing (Ho-Dac and Laignelet, 2005; Bilhaut et al., 2003b), thematic (Bilhaut, 2005; Bilhaut and Enjalbert, 2005) and rhetorical (Widlöcher, 2004) structures with a view to information retrieval and theoretical linguistics.

- Works on Geographical Information, as in the GeoSem project (Bilhaut et al., 2003a; Widlöcher et al., 2004), or in another research project (Marquesuzà et al., 2005).

- TCAN project: Temporal intervals and applications to text linguistics, CNRS interdisciplinary project.

- The platform is also used for other research or teaching purposes in several French laboratories (including GREYC, ERSS and LIUPPA) in the fields of corpus linguistics, natural language processing and text mining.

More information can be obtained from the dedicated web site[2].

References

Frédérik Bilhaut and Patrice Enjalbert. 2005. Discourse thematic organisation reveals domain knowledge structure. In *Proceedings of the Conference Recent Advances in Natural Language Processing*, Pune, India.

Frédérik Bilhaut, Thierry Charnois, Patrice Enjalbert, and Yann Mathet. 2003a. Passage extraction in geographical documents. In *Proceedings of New Trends in Intelligent Information Processing and Web Mining*, Zakopane, Poland.

Frédérik Bilhaut, Lydia-Mai Ho-Dac, Andrée Borillo, Thierry Charnois, Patrice Enjalbert, Anne Le Draoulec, Yann Mathet, Hélène Miguet, Marie-Paule Péry-Woodley, and Laure Sarda. 2003b. Indexation discursive pour la navigation intradocumentaire : cadres temporels et spatiaux dans l'information géographique. In *Actes de Traitement Automatique du Langage Naturel (TALN)*, Batz-sur-Mer, France.

Frédérik Bilhaut. 2005. Composite topics in discourse. In *Proceedings of the Conference Recent Advances in Natural Language Processing*, Borovets, Bulgaria.

Ulrich Callmeier, Andreas Eisele, Ulrich Schäfer, and Melanie Siegel. 2004. The DeepThought Core Architecture Framework. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Stéphane Ferrari, Fréférik Bilhaut, Antoine Widlöcher, and Marion Laignelet. 2005. Une plate-forme logicielle et une démarche pour la validation de ressources linguistiques sur corpus : application à l'évaluation de la détection automatique de cadres temporels. In *Actes des 4èmes Journées de Linguistique de Corpus (JLC)*, Lorient, France.

Lydia-Mai Ho-Dac and Marion Laignelet. 2005. Temporal structure and thematic progression: A case study on french corpora. In *Symposium on the Exploration and Modelling of Meaning (SEM'O5)*, Biarritz, France.

Christophe Marquesuzà, Patrick Etcheverry, and Julien Lesbegueries. 2005. Exploiting geospatial markers to explore and resocialize localized documents. In *Proceedings of the 1st Conference on GeoSpatial Semantics (GEOS)*, Mexico City.

Claude Muller, Jean Royaute, and Max Silberztein, editors. 2004. *INTEX pour la Linguistique et le Traitement Automatique des Langues*. Presses Universitaires de Franche-Comté.

Antoine Widlöcher and Frédérik Bilhaut. 2005. La plate-forme linguastream : un outil d'exploration linguistique sur corpus. In *Actes de la 12e Conférence Traitement Automatique du Langage Naturel (TALN)*, Dourdan, France.

Antoine Widlöcher, Eric Faurot, and Frédérik Bilhaut. 2004. Multimodal indexation of contrastive structures in geographical documents. In *Proceedings of Recherche d'Information Assistée par Ordinateur (RIAO)*, Avignon, France.

Antoine Widlöcher. 2004. Analyse macro-sémantique: vers une analyse rhétorique du discours. In *Actes de RECITAL'04*, Fès, Maroc.

[2] `http://www.linguastream.org`