TyPTex : Inductive typological text classification by multivariate statistical analysis for NLP systems tuning/evaluation

Serge Heiden, Sophie Prévost, Benoît Habert, Helka Folch, Serge Fleury, Gabriel Illouz, Pierre Lafon, Julien Nioche

TyPTex : Inductive typological text classification by multivariate statistical analysis for NLP systems tuning/evaluation

Helka Folch*, Serge Heiden*, Benoît Habert*,

Serge Fleury*, Gabriel Illouz†,
Pierre Lafon*, Julien Nioche*, Sophie Prévost*

* UMR8503 : Analyses de corpus linguistiques, usages et traitements
CNRS / ENS Fontenay/Saint-Cloud
92211 Saint-Cloud, France
{slh, folch, fleury, lafon, nioche, prevost}@ens-fcl.fr, habert@limsi.fr

†LIMSI, CNRS / Université Paris Sud
Orsay, France
illouz@limsi.fr

Abstract

The increasing use of methods in natural language processing (NLP) which are based on huge corpora require that the lexical, morpho-syntactic and syntactic homogeneity of texts be mastered. We have developed a methodology and associate tools for text calibration or "profiling" within the ELRA benchmark called "Contribution to the construction of contemporary french corpora" based on multivariate analysis of linguistic features. We have integrated these tools within a modular architecture based on a generic model allowing us on the one hand flexible annotation of the corpus with the output of NLP and statistical tools and on the other hand retracing the results of these tools through the annotation layers back to the primary textual data. This allows us to justify our interpretations.

1. Introduction

Natural Language Processing (NLP) is increasingly dependent on corpus-based methods. The availability of corpora is no longer a problem, as huge and annotated corpora are now readily available. The real problem has now become corpus heterogeneity. Several studies covering different areas of NLP suggest that the performance of Natural Language systems depends on the working corpus's homogeneity.

D. Biber (1993, p.223) has shown that the performance of a probabilistic tagger is related to the domain on which it operates after performing tests on the LOB corpus : the probability of a morpho-syntactic category is a function of the domain. Similarly, collocations were shown to differ significantly from one domain to another (for instance for *sure* and *certain*).

S. Sekine (1998) has shown that the performance of a parser is also dependent on the domain. He examined the results obtained by a probabilistic syntactic parser on 8 different domains of the BROWN corpus (documentaries, editorials, hobbies, learned, fiction, western, romance novels). He observed varying differences in the performance of the parser, in terms of recall and precision, depending on the learning domain and the test domain.

Similar observations have been made by J. Kalgren (1999) concerning the performance of information retrieval systems. He examined the dependence of the relevance of the queries 202 to 300 of the TREC evaluation campaign on various categories of articles from the *Wall Street Journal* part of the TIPSTER corpus. Those categories were based on a classification of some of their stylistic features : average word length, average word frequency, proportion of digit letters… Results suggest that the articles judged relevant to those queries appear in specific categories and moreover the articles chosen by all the systems for those queries (relevant or not) where found in specific categories too.

As suggested by these experiences, corpus heterogeneity induces 2 types of statistical errors (Biber, 1993, p.219-220) : random error (which occurs when the sample is too small to represent the population) and bias error (which occurs when one or several features of the sample are systematically different from the population under examination).

2. An architecture for text profiling

In this paper we present a methodology and a set of tools for text profiling, that is for "calibrating" different parts of a corpus in terms of linguistic features based on the internal properties of each text : the vocabulary, its morpho-syntactic categories, some patterns of those categories… The aim of our text profiling method is to produce measures of corpus homogeneity within the different parts of a corpus which enables us to construct homogeneous subsets of the corpus in terms of one or more parameters of our model.

Our approach is similar to D. Biber's work on text classification (1988)(1995). In his work, each text of a group of 4,814 contemporary english texts is represented by a vector of 67 features. Those features are based on 16 different categories (verb tense and aspect markers, interrogatives, passives, etc.) automatically extracted from the first 1,000 words of each text. Each text can then be seen as a point in a 67 dimensions space. Two texts close together in that space have similar properties regarding the features associated to the dimensions along which they are close. After computing each feature frequency to build the vector of each text, the dimensionality of the feature space

is first reduced by a discriminant analysis. The results of that discriminant analysis are n new features (where n << 67) each one composed by a mixture of the original features. Biber uses the 5 most discriminant dimensions. Clustering methods are then being used to group texts in terms of their location in this new space. The resulting clusters are types of texts which correspond directly neither to text "genres" nor to language styles or registers. The projection of a new text in that space then permits to assign it a type by choosing its closest cluster.

In the TyPTex project we continue and extend the work of Biber in several ways.

First we apply that kind of analysis to the French language. Our discourse analysis framework is based on Biber's work for English and on the ones made by J.-P. Sueur (1982) and J.-P. Bronckart (1985) for French. Within that framework we have atmost 200 linguistic features available to describe a text. But the features set on which the feature vectors are computed is tuneable during the texts analysis process.

Second we developed an architecture that enables us to work on any textual corpus. We can apply various taggers and feature extractors to any text. The SGML format is used to store all the various annotations that our NLP tools add to the texts and we developed an SGML aware specific tool to semi-automatically correct the errors made by those tools and master the quality of feature extraction.

Third we assembled a set of multi-dimensional statistical analysis tools and developed a specific tool to graphically analyse and tune the results of automatic clustering with the feature set used for classification.

On top of that generic architecture our efforts are focused on the bi-directional links to maintain between the textual data at the origin of the features used in classification and the various text types we obtain. We want to be able to follow the links back from a text type to any textual data at the origin of a specific feature used as a classification dimension. That permits us to tune specific features extractions to accentuate the contrast between the classes. The main goal being the stability of our interpretations.

We have already presented several results obtained using that architecture (Habert et al. 2000 and 2000b). In this paper we present and justify our architecture for profiling.

3. Architecture of the TyPTex project

3.1. Architecture modules

We have developed a modular architecture which provides a flexible framework for processing annotated texts necessary for the study of corpus heterogeneity. At the bottom level, this architecture consists of a collection of texts which are tagged according to the TEI (Text Encoding Initiative) recommendations. Each text has a descriptive header attached to it. We then perform queries based on the descriptive variables associated to the texts to extract a subset of texts (or text chunks) which are relevant to a certain study or application. These descriptive variables include information concerning the date, the author, the type of document or for instance, for the journalistic press included in the corpus, pre-existing categories describing the newspaper sections to which the articles belong (politics, arts, current affairs, etc). The next step is to perform a morpho-syntactic tagging which associates each lexical item (or a poly-lexical item) to a given word stem. The tagging process also associates a part of speech category and other morpho-syntactic information to each lexical item. We have used Sylex-Base (Ingenia, 1995) for tagging. It is a tagger/parser based on the work of P. Constant (1991) which has proved to be robust during the tagger evaluation programme GRACE. The lower level tagging, which is at the present day still limited, includes shifters , modals, presentatives, tense use, passives, certain classes of adverbs (negation, degree) , articles, etc. The category (or part of speech) is kept for those words or polylexical items which have not been otherwise tagged. We then perform *typological marking*, which consists of replacing the information generated by the morpho-syntactic tagger by higher-level categories. These new categories are calculated from the morpho-syntactic tags and vary according to which features we want to study. From the resulting tagged corpus several matrixes are generated, in particular the matrix containing the frequencies of each feature in each text of the corpus under study. The resulting tagged corpus is then analysed by statistical software programs. The analysis of this matrix is aimed, on the one hand, at identifying the relevant features to a certain opposition and on the other hand, at making an inductive or supervised classification of texts. At present, two types of statistical treatments are performed : The first type are aimed at exploring the significant correlations of linguistic features (Principal Component Analysis, Correspondance Analysis, Sammon Projection) ; They consist of observing one feature or a small group of features in order to determine their relevance in relation to a classification. It enables the observation of features which are not necessarily ruled by the same probability laws (Karlgren, 1999, p.153) This implies being able to visualise texts as points in a space, being able to change the point of view, the classification. The second type is that of supervised training. It implies being able to place a text in a pre-existent classification (via Quinlan's C4.5, for instance).

4. Evaluation of other architectures for corpus processing

We have tested the following 4 architectures :

4.1. TIPSTER

TIPSTER is an architecture for Natural Language Processing Systems (NLP) (Grishman, 1996) which is based on a data-driven approach. This means that all information on a given text is stored in a database separately from the text itself. Thus, the text itself remains unchanged. The information about the text or annotation is therefore not encoded in a SGML format but according to a database model. Annotations link arbitrary information to text segments in the document base. The relevant document segments are identified by character spans in the byte stream of the document specified in terms of start/end offsets. The database model for annotations is object-oriented. It defines classes representing queries, for

instance, or elements of information extraction and information retrieval. Different types of documents are grouped into collections and their annotations are described by different database models.
The TIPSTER architecture is not tied to any specific implementation, which makes it portable over a range of platforms. The GATE architecture described below is a specific implementation of TIPSTER.

4.2. GATE

GATE (Wiks & Gaiauskas,1999) This architecture, based on the TIPSTER model, is aimed at making heterogeneous NLP modules intercommunicate for the development of complex systems. Annotations, as in TIPSTER, are stored separately from the primary data to which they refer. The GATE architecture is composed of 3 main components :

- GDM, the GATE document manager. The GDM centralises all the descriptive information associated to the documents. It is the gateway for all queries from any language engineering component integrated in the architecture. In other words, components do not communicate directly but through API functions (for retrieving information or outputting results) directed at the GDM.
- CREOLE a Collection of Reusable Objects for Language Engineering. CREOLE modules are interfaces to resources. These resources may be programmes (taggers, parsers, etc) or data (a lexicon, a semantic tag list, etc). CREOLE modules are object-oriented and therefore encapsulate their functionality through an interface (containing attributes and methods). When an object's method is executed, it launches a call to the GDM API. This method can be a query to either obtain information concerning a document's primary data or it's annotations, or else to store the results of analysis or processing done by the module in the GDM database. The results of this module's analysis thus become available to other modules.
- GCI, the GATE Graphical Interface. The GCI is a graphical tool that displays the resources underlying GDM and CREOLE and makes the task of interconnecting components and exploring different combinations of existing modules easier. However an effort is required to develop tools to generate an intermediate format from the specific formats accepted and generated by existing modules.

4.3. IMS

IMS Corpus Workbench (Christ, 1994): This workbench has been developed around a search engine aimed at the study of tagged corpus. Textual data is accompanied of as many annotations as necessary and is treated as a database. This base is stored and indexed in order to allow queries to be answered promptly. Queries are expressed in terms of regular expressions concerning all of or a part of the annotations or sequence of annotations. This architecture is especially suited to efficiently handle a corpus whose annotation is stabilised.

4.4. LT XML

LT XML (McKelvie et al., 1997) LT XML is a generalisation of the approach based on successive UNIX filters (pipelines). The data, at all stages of the processing is tagged in SGML. The tree or the event sequence that constitute a parsed SGML document provides as precise a context as required for formulating queries. This architecture enables experimentation with different types of annotation whilst guaranteeing the formal validity of data throughout the different stages as well as an optimised parsing of the SGML event flow.
Two solutions are thus available for the use of multiple annotations : storage of the annotations in a single document (IMS-CWB) versus distribution of the annotations (GATE). The first approach facilitates the subsequent access to the documents and the establishment of connections between the different levels of annotation. The second one is favoured when the annotations diverge. It enables the articulation of a great number of simultaneous annotations. Furthermore, linking components one after another can be done using a pivotal format between 2 modules (GATE) –each module remaining « in control of itself »- or by rendering each module to a single format. The first solution favours the joint use of heterogeneous modules, the second one the homogeneity of the treatments.

5. Architecture constraints for a text-profiling platform

We believe that our architectural model fulfils the following requirements

5.1. Supporting multiple representations of linguistic phenomena

The aim of our project is to study the distribution and correlation between linguistic phenomena which can provide measures of text homogeneity and be at the basis of text typologies. Part of the task at hand is therefore to determine which particular linguistic events are statistically discriminatory yielding the most relevant results and leading to clearly defined text types. This can only be determined empirically. Testing this requires a flexible architecture that does not impose only one representation of the underlying linguistic phenomena. Our architecture therefore supports different types of segmentation and markings of the primary textual data. For instance, we plan to use parallel annotations for part of speech (POS) markings, each one is the output of a different POS tagger (SYLEX and CORDIAL 6 UNIVERSITES).
Likewise, typological marking is not based on a unique set of features. As mentioned above, part of the task at hand is to determine which features are statistically most discriminatory. Therefore, the set of features of the typological marking is constantly evolving, at pace with the results of our tests.
Further, not only is the set of profiling features open but it contains features corresponding to different levels of representation. For instance, at present the features employed belong to several different categories :

Characters : punctuation marks, capital letters and digits in particular (Illouz, 1999);
Closed lexical sets : categories of functional words (Brunet, 1981)(Biber, 1988), (Illouz et al., 1999) ;
Fine-grained typological categories (Sueur, 1982)(Bronckart et al., 1985)(Biber, 1988) ;
Text structure, titling, image presence, charts (Karlgren, 1999).
We have achieved the necessary flexibility for supporting multiple representations by building up different layers of annotations in a decentralised way. In other words, the primary textual data remains unchanged, whilst the successive annotation layers are stored in separate documents. Annotations are then connected to the corresponding primary data by intertextual links.
This approach shares aspects of both the LT-XML and TIPSTER architectures. On the one hand it is reminiscent of the TIPSTER approach in so far as the annotations are kept separately from the texts themselves. However it differs from this approach in that the annotations are themselves encoded in SGML (as in LT-XML).

5.2. Tracing back results

In the TyPTex architecture, annotation layers from segmentation to typological marking are built in a recursive way. For instance, typological marking builds upon part of speech and morpho-syntactic tags.
Each annotation layer is a document composed by a header and a body. The header contains information describing the annotation operation performed. For instance, if the annotation in question is that of a morpho-syntactic tagger, the header contains information relative to the specific software used, its parameters, how its output will be articulated with the typological marking and any other relevant decisions and choices made at that point. The body contains the annotation tags themselves and the elements to which they are applied, expressed indirectly in terms of links pointing to elements from other layers. In some cases, elements from other layers are merged literally into the body of the annotation document to speed up processing.
Annotations are then organised as a hyper-document recursively layered over the corpus primary data. This forms a tree-like structure, because multiple annotation layers can branch from the same element of the primary corpus or from some lower annotation level. For any given element in this structure, either a text chunk in the primary corpus or an annotated element of a higher layer, it is possible to access the complete sequence of treatments and annotations it has gone through.
Keeping track of all the operations performed on any subset of text chunks or sub-corpora implies being able to not only retrace these operations step by step but also to access the parameters and choices performed at each step which are documented in the descriptive header of each annotation document. This is crucial in order to correctly interpret the results of the statistical analysis.
It is especially important for two reasons. Firstly, the statistical methods we use (Sammon projection, factor analysis, clustering) are based on multivariate analysis which are contrastive in nature. This means that the results are valid only within the scope of a certain sub-corpus. Therefore the parameters relative to the sub-corpus extraction (in particular the query leading to the construction of the sub-corpus) as well as the particular parameters of the statistical method employed in the analysis of the sub-corpus provide the necessary contextual information necessary to the interpretation of the results.
Secondly, the typological marking which we use for text profiling is abstract and hard to interpret. The results of the statistical methods give patterns and oppositions between correlations of abstract linguistic features. In order to interpret these patterns it is essential to be able to backtrack each step in the construction of these features and to recover their context in the texts of origin. In other words, it is necessary, in order to check the proposed interpretations and hypothesis, to be able to examine the behaviour of these traits within the context of the sub-corpus under study.

5.3. Constructing principled sub-corpora

The aim of our text profiling method is to calibrate a corpus in terms of different criteria. In order to achieve this, calibration has to be performed on different views of the corpus, in other words on sub-corpora constructed in terms of different combination of parameters.
This imposes 2 requirements on the architecture. Firstly, that the corpus be finely grained. In other words, that the base level elements in the corpus be fine-grained textual units (paragraph, sentence, etc) and not whole documents. In this way, a sub-corpora can be built by extracting and assembling only the relevant textual units in relation to a given parameter. One could not achieve this by extracting entire documents, as a document can be heterogeneous in relation to a given criteria and can vary enormously in length. At present, the base level segmentation unit is the paragraph but the implementation of other segmentation schemes (the sentence, for instance) can be envisaged.
Secondly, the architecture must support annotation of the base-level structural units by arbitrary and possibly, conflicting annotation data. This is achieved by overlaying multiple annotation levels on the primary, segmented corpus as described above. The extraction of a sub-corpus results in a sub-corpus document that has its own descriptive header and whose body is composed of all the textual chunks relevant to the extraction query merged with the relevant annotations. Merging the relevant annotations associated to a given chunk into the document stream of the sub-corpus document can be seen as a flattening out or linearization of the chunk's annotation layers. Not all annotations associated to a chunk are merged into the sub-corpus document, some may not be relevant to the particular study, others may be parallel or conflicting annotations which can not be serialised into one single SGML stream.

5.4. Retro-projecting results into the corpus

Annotation of the corpus is not limited to the marking of the relevant linguistic phenomena under study (morpho-syntactic and typological marking) but also of the results of the statistical analysis on the extracted sub-corpora. These results are re-injected into the corpus in the form of

annotation documents with their own descriptive headers which specify the details of the kind of statistical analysis undertaken. The bodies of these annotation documents contain the tagged results which are connected to the corresponding textual chunks in the sub-corpora through backward links.
The status of the projected results is that of any other kind of annotation. They can be used as an extraction criteria in the construction of subsequent corpora. Further, their articulation to other types of descriptive information can be explored in order to establish correlations between statistical results and pre-existing classification schemes.

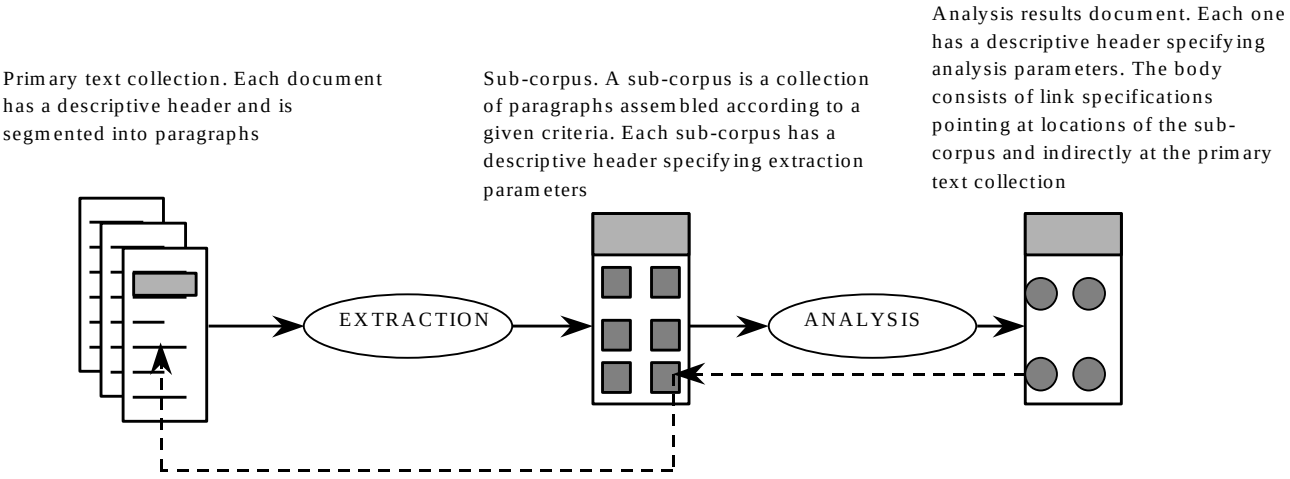


Figure 1

5.5. Modularity

TyPTex is designed to be a modular architecture providing an open testbed where different text analysis tools can be plugged in. The aim is to be able to test and compare different statistical treatments.
The tools which have been described here are to be found among different communities (data analysis, automatic learning) and are therefore difficult to use simultaneously. GATE can, in principle, articulate them, but encapsulates them to guarantee an interoperability of the treatments employed.
Our approach has been to use a standard and normalised SGML format for encoding the sub-corpora under study.
Generating an intermediate subcorpus document as described above, instead of directly outputting an application format may seem redundant and a waste of storage space. However, this intermediate document is crucial, firstly for tracing back results, and secondly as a normalised, self-describing, interchange format from which application formats can be easily generated, using SGML processing libraries for instance. From this sub-corpus document we generate a contingency matrix where the rows are the texts and the variables (columns) are the features. Outputting particular application formats from this matrix is straightforward.

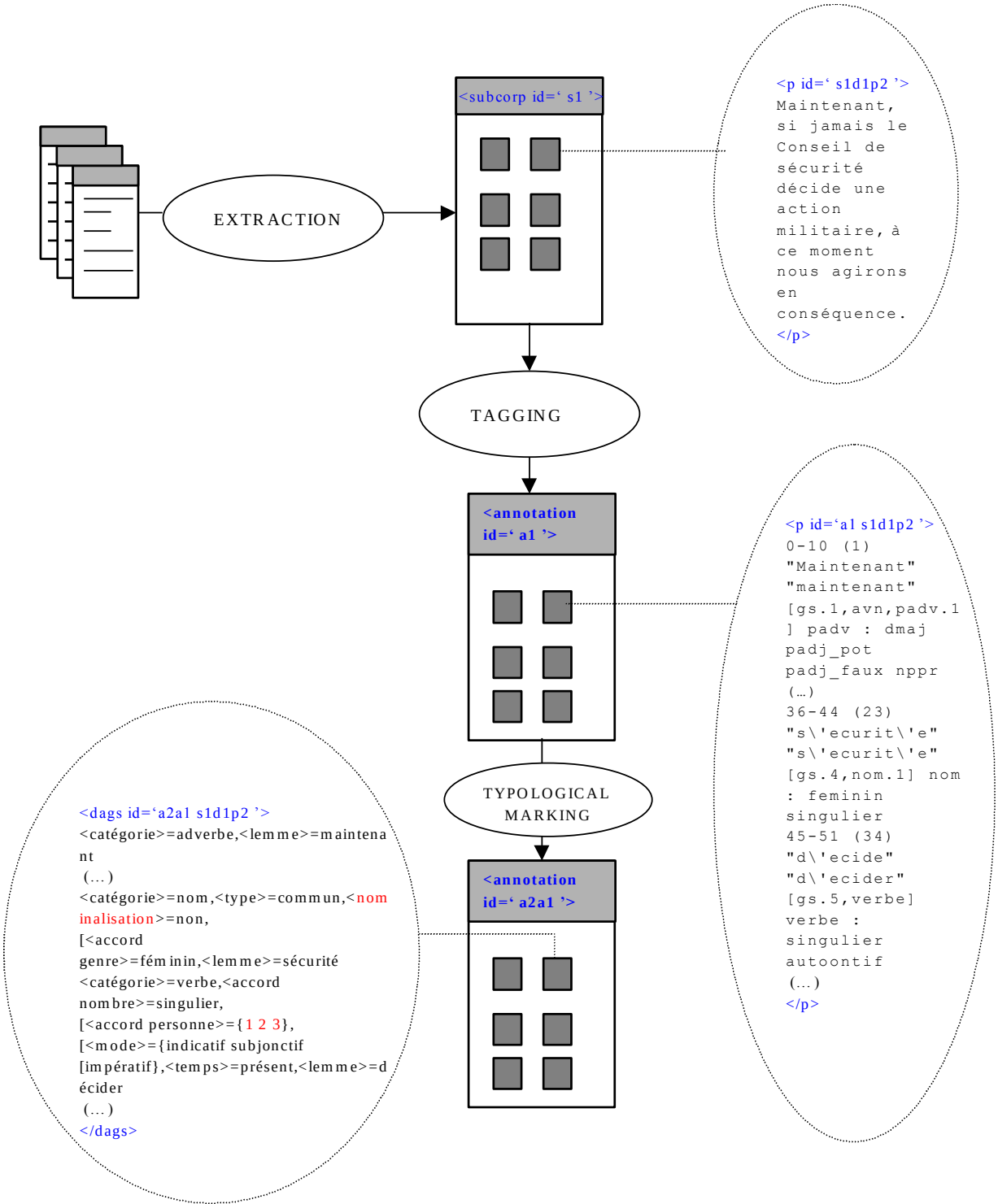
5.6. Annotations based on structured features

This experience of typological marking has enabled us however, to examine the features we have used in a critical light. They can be too fine-grained and lead to a scattering of occurrences which makes contrasts imperceptible. This has been the case concerning verb tenses in the current choice of features : the verb category is fragmented into some 50 features, most of which have a limited number of occurrences. Therefore we have no grip on the verb considered globally, nor on its tendencies with respect to the sections or to the articles. Inversely, certain features are too rough and probably hide real oppositions. This is the case for *nombres cardinaux* (cardinal numbers) that groups quantity indicators, as well as dates, which would probably be more effective to differentiate. This can also be the case for certain nouns which result from different nominalisations. Thus, it may be relevant to further specify the tagger's output with information indicating whether the noun is morphologically related to a verb (like *importation*) or an adjective.
In general, changing the granularity of the information outputted by the tagger, highlights contrasts between texts that were not directly visible from the results of the tagger. Our aim in fact is to manipulate structured features in order to be able to use the corresponding information totally or partially. Thus for instance, having the following kind of tag {category=noun, type=common, gender=masculin, person=singular…} enables us to select subsets such as {category=noun}, {category=noun, type=common}, or {genre=masculin}. Using feature structures such as those employed in unification grammars makes it possible to modelise more precisely the information resulting from marking, in the style of, for instance (Gazdar et al.,1990) as well as the operations that can be performed on them.
In consequence, we have adopted the PATR-II formalism (Shieber, 1986) to represent each word of the corpus as a feature structure. The advantage of this approach is that feature transformation can be carried out within the formal framework of unification grammars and feature logic and

can benefit from the transformation tools developed in this domain. Simplifying, enriching and re-organising the information outputted by a tagger can be more rigorously formulated in terms of operations such as conjunction or disjunction of features. Following this approach, we have tested the possibility to sum features up, thus creating super features which are expressed in terms of a disjunction or conjunction of elementary features. For instance, one can define a super-feature standing for the property of agency as a conjunction of elementary features such as nominalisation, active verbs, certain suffixes, etc.
Another formal quality of feature structures is the fact that they can express hierarchical information. Therefore, depending on what kind of oppositions one wants to highlight in a given study, one can choose features at different levels in the feature structure corresponding to different degrees of generality. The nested structure of the feature structure is flattened out again at the end of the transformation process.
The operations of feature transformation are performed through meta-rules. A meta-rule (Gazdar et al. 1987)(Jacquemin 1997) consists of a source (left hand side of the rule) and a target (right hand side). The source of the rule is matched against a given feature structure. If unification succeeds, the feature structure is transformed according to the specifications of the rule's target.

Figure 2 : The annotation flow

Figure 2 shows how transformations of the typological marking are built on the tagging results, forming thus



successive annotation levels on the subcorpus. Firstly, as shown at the top of the figure, an extraction produces a sub-corpus document which groups paragraphs (which are our atomic extraction and analysis unit) that satisfy the constraints in the extraction query. A sub-corpus document is an XML object with its own descriptive header and its unique identifier. The paragraphs included in the sub-corpus document also have a unique identifier which is a concatenation of the sub-corpus' identifier and its original identifier in the TEI text collection. For instance, if a paragraph's identifier in the TEI text collection is 'd1p2', once it is extracted and is integrated in a sub-corpus whose identifier is 's1' its own identifier becomes 's1d1p2'. This naming scheme is followed throughout all the annotation levels of the architecture. For instance, once this subcorpus has been tagged, the tagger's output is stored in a annotation document whose unique identifier is, for instance 'a1'. The paragraph's identifier then becomes 'a1s1d1p2'. The advantage of this naming scheme is that all the transformations that a text chunk has gone through are retraceable through its identifier.

In the example, the annotation document coding the tagger's results, is an illustration of the output of the robust Sylex-Base tagger (Constant, 1991). The raw output of the tagger for each paragraph is stored in this document. The following step is that of typological marking and the construction of feature structures. One can see how the tagger's results have been transformed and refined. For instance, nominalisation information has been added (« nominalisation » tag to nouns) and the ambiguity of the verb person has been expressed as disjunction of values ({1 2 3}).

6. Generic status of the architecture

The architecture presented here is based on a generic model which has been developed and tested within the framework of the Scriptorium project and will be implemented in the TyPWeb project. Both projects are presented below.

6.1. Scriptorium

Scriptorium (Lahlou et al., 1998), is a project developed in the Research & Development Division of *EDF* (*Electricité de France*) in collaboration with *ENS* (*Ecole Normale Supérieure*) *de Fontenay/Saint-Cloud*. The aim of this project is to extract prominent and emerging topics from the automatic analysis of the discourse of the company's (EDF) different social players (managers, trade-unions, employees, etc) by way of textual data analysis methods. The corpus under study in this project has 8 million words and is very heterogeneous (it contains book extracts, corporate press, union press, summaries of corporate meetings, transcriptions of taped trade union messages, etc).

Scriptorium is a modular architecture which provides an open framework where different text-mining tools can be plugged in. The results of these text mining tools are integrated into the corpus' architecture as structured layers over the corpus' primary data, and pointing to the relevant units in the corpus The architecture is structured on 3 levels. The first level consists of a collection of documents which are tagged according to the CES (Corpus Encoding Standard) recommendations. As defined in CES, each document at this level is provided with a descriptive header and is segmented into minimal textual units or chunks(which in our case correspond to paragraphs). We then use an extractor developed using the XML Python libraries to retrieve relevant text chunks and assemble them into homogeneous sub-corpora of exploitable size (< 10 Mb). This extractor runs queries concerning the descriptive parameters stored in each document's header as well as full text searching constraints. It is essential for text mining software to run on homogeneous corpora in order to yield relevant results.. These dynamically assembled corpora constitute the 2nd level in the corpus architecture. Finally, the results of the treatments performed by the statistical software are structured into annotation layers pointing to the textual primary data.

6.2. TyPWeb

A new project, TyPWeb, in collaboration with CNET, aims at adapting the TyPTex architecture to the processing of web sites and will mark the passage of the present prototype to a generic profiling architecture. The aim of this project is to provide a methodological and practical framework for web site profiling and the development of a fine-grained typology of these sites. The approach consists of characterising each site by a set of indicators concerning both content and structure. The first step is to define and subsequently enrich the description of sites in terms of these content and structural indicators : this information is pumped into the descriptive header of the analysed sites. The header remains open and extendable by any new information deemed relevant. TypWeb should subsequently lead to a proposition of a content typology (using predefined topic indexes or constructing new content categories by way of an inductive approach). The resulting analysis should be obtained by crossing the formal structure with the content typologies. It will also consist of describing the articulation between the formal and semantic description of the sites with the practical account of the agents involved (designers and visitors). This approach aims in particular at analysing the progressive establishment of implicit exchange rules over the web.

7. Conclusion

We believe that the TypTex architecture provides a modular framework for text profiling and text typology. It enables flexible text annotation and more importantly it allows documenting and backtracking the transformations and results of NLP and statistical analysis tools. This is essential in order to produce an explanatory and principled model for correlations of linguistic features and text typology. We want to pursue our tests to determine the relevance of the linguistic features used at present for describing text typologies and measuring text homogeneity. We will perform these tests within the representational framework of feature structures, using the expressive power of the operations performed on these structures to define combinations of features of different granularity. We shall further enlarge the scope and nature

of our features within the TypWeb project as we will consider both linguistic and structural markings.

8. References

Biber D. (1988). Variation accross speech and writing. Cambridge University Press, Cambridge.

Biber, D. (1993). Using register-diversified corpora for general language studies. *Computational Linguistics*, 19(2), 243–258.

Biber D. (1995). Dimensions of register variation : a cross-linguistic comparison. Cambridge University Press, Cambridge.

Bronckart, J.-P., Bain, D., Schneuwly, B., Davaud, C. & Pasquier, A. (1985). Le fonctionnement des discours : un modéle psychologique et une méthode d'analyse. Lausanne : Delachaux & Niestlé.

Christ, O. (1994). A modular and flexible architecture for an integrated corpus query system. In *Proceedings of COMPLEX'94 (3rd Conference on Computational Lexicography and Text Research)*, Budapest, Hungary. CMP-LG archive id 9408005.

Constant, P. (1991). Analyse syntaxique par couches. Doctorat de l'enst, Ecole Nationale Supérieure des Télécommunications, Paris.

Dunlop, D. (1995). Practical considerations in the use of TEI headers in large corpora. Computers and the Humanities, (29), 85–98. Text Encoding Initiative. Background and Context, edited by Nancy Ide and Jean Véronis.

Gazdar G., Pullum G. K., Carpenter R., Klein E., Hukari T. E., and Levine R. D. (1990). Les structures de catégories. In Miller P. and Torris T. editors, Formalismes syntaxiques pour le traitement automatique du langage naturel, Langue, raisonnement, calcul, chapter 6, pages 245--301. Hermès, Paris.

Grisham, R. (1996) . TIPSTER Architecture Design Document Version 2.2. Technical report, Defense Advanced Reseearch Projects Agency.

Habert B., G. Illouz, H. Folch, S. Fleury, S. Heiden, P. Lafon, S. Prévost, (2000). " Prendre Le Monde en main : choix d'architecture", RIAO'2000, Paris, Avril.

Habert B., Illouz G., Lafon P., Fleury S., Folch H., Heiden S., Prévost S., (2000b). «Profilage de textes : cadre de travail et expérience », JADT'2000, Lausane.

Illouz G., Habert B., Fleury S., Folch, H., Heiden S., and Lafon P. (1999). Maîtriser les déluges de données hétérogènes. In Condamines A., Fabre C., and Péry-Woodley M.-P. editors, Corpus et traitement automatique des langues : pour une réflexion méthodologique, pages 37--46,

Cargèse.

Ingenia (1995). Manuel de développement Sylex-Base. Ingenia – Langage naturel, Paris. 1.5.D.

Karlgren, J. (1999). Stylistic experiments in information retrieval. In T. Strzal.

Lahlou S., Folch, H. (1998). Quelques stratégies pour l'exploitation en ADT de grands corpus hétérogènes. In JADT proceedings. 1998.

McKelvie, D., Brew, C. & Thompson, H. (1997). Using SGML as a basis for data-intensive NLP. In *Proceedings 5th Conference on Applied NLP*, pp. 229–236: ACL.

Sekine, S. (1998). The domain dependence of parsing. In *Fifth Conference on Applied Natural Language Processing*, pp. 96–102, Washington: Association for Computational Linguistics.

Shieber, S. N. (1986). *An Introduction to Unification-Based Approaches to Grammar*. CSLI Lecture Notes 4. Stanford, CA: CSLI.

Sueur, J.-P. (1982). Pour une grammaire du discours : élaboration d'une méthode; exemples d'application. *MOTS*, (5), 145–185.