INTEX: A CORPUS PROCESSING SYSTEM

Max D. Silberztein

Laboratoire d'Automatique Documentaire et Linguistique
Université Paris 7

ABSTRACT

INTEX is a text processor; it is usually used to parse corpora of several megabytes. It includes several built-in large coverage dictionaries and grammars represented by graphs; the user may add his/her own dictionaries and grammars. These tools are applied to texts in order to locate lexical and syntactic patterns, remove ambiguities, and tag words. INTEX builds concordances and indexes of all types of patterns; it is used by linguists to analyse corpora, but can also be viewed as an information retrieval system.

INTRODUCTION

INTEX automatically identifies words and morpho-syntactic patterns in large texts. By using INTEX, one can:

— build the dictionary of the words of the texts; words may be *simple words* (sequences of letters, e.g. *table*), *compounds* (sequences of simple words which include a separator, e.g. *word processor*) or complete *expressions* (sequences of words which accept insertions, e.g. *to kick ... the bucket*);

— locate in texts all occurrences of a given word (even if inflected), a given category (e.g. all feminine plural adjectives) or a morpho-syntactic pattern (a regular expression);

— apply grammars represented by recursive graphs to texts; build indexes or concordances for all occurrences of the previous patterns;

— use local grammars to remove word and utterance ambiguities in texts, or to detect errors or deviant sequences.

While INTEX already includes several built-in dictionaries and grammars, it allows the user to create, edit and add his/her own tools, in order to increase coverage of texts and to remove additional ambiguities.

1. LINGUISTIC TOOLS

The user first loads a text and selects the working language[1]. INTEX counts the number of tokens in the text, the number of different ones, and sorts them by frequency. Then the user selects linguistic tools to parse the text. Tools are either dictionaries or finite state transducers (FSTs).

1.1. Dictionaries

INTEX is based on two large coverage built-in dictionaries:

— the DELAF dictionary contains over 700,000 simple words, basically all the simple words of the language[2]. Each entry in the DELAF is associated with explicit morphological information for each word: its canonical form (e.g. the infinitive for verbs), its part of speech (e.g. Noun), and some inflectional information (e.g. first person singular present). Here are three entries of the French DELAF:

a, avoir. V:P3s
abacas, abaca. N:mp
abaissa, abaisser. V:J3s

The token 'a' is the Verb 'avoir' conjugated in the Third Person Singular Present (P3s); 'abacas' is the masculine plural of the Noun 'abaca'; 'abaissa' is a verbal form of 'abaisser' conjugated in the third person singular "Passé composé" (J3s). Since the morphological analysis of each

1. At this moment, English, French and Italian dictionaries have been already included in INTEX. German, Spanish and Portuguese compatible dictionaries are under construction. We will give French examples.

2. For a discussion on the completeness of the DELAF dictionary, see in [Courtois; Silberztein 1989], [Clemenceau 1993].

token is performed by a simple lookup routine, INTEX guarantees an error free result (there is no guessing algorithm nor 'probabilistic' result). INTEX includes a few other dictionaries for proper names, toponyms, acronyms, etc.;

— the DELACF dictionary contains over 150,000 compounds, mostly nouns[3]. Each entry in the DELACF is associated with its canonical form, its part of speech, and some inflectional information. Here are three entries of the French DELACF:

à tout de suite, à tout de suite. ADV
cartes bleues, carte bleue. N:fp
pomme de terre, pomme de terre. N:fs

INTEX includes a few other dictionaries for compound proper names. The user may add his/her own dictionaries for simple words and compounds.

1.2. Finite State Transducers

FSTs are represented in INTEX by recursive graphs. Basically, the "input" part of an FST is used to identify patterns in texts; the "output" part of an FST is used to associate each identified occurrence with information. In many cases, FSTs represent words more naturally than dictionaries. For example, numerical determiners, such as *trente-cinq mille neuf cents trente-quatre*, formally are compounds which are naturally represented by graphs (see the graph *Dnum* in Appendix 1). FSTs may also be used to bring together graphical variants of a word in order to check the spelling coherency, to associate all the variants of a term with a unique canonical entry in an index, to represent families of derived words (see the graph *France* in Appendix 1), to associate synonyms of a term in an information retrieval system, etc. In the graph editor, gray nodes are graph names; tags written in white nodes are the inputs of the FSTs, outputs are written below nodes[4]. The user draws graphs directly on the screen; the resulting graphs are interpreted as FSTs by INTEX.

By selecting and applying dictionaries and FSTs to a text, the user builds the dictionary of the words of the text. Appendix 1 shows the resulting dictionary, as well as the list of all *unknown* tokens. Generally, these tokens are either spelling errors or proper names.

2. LOCATING PATTERNS

After having built the dictionary of the words of the text, the user can locate morpho-syntactic patterns in the corpus, index or build a concordance for all occurrences of the pattern. Patterns may be:

— a word, or a list of words. For example, one can locate in a text all occurrences of the verb *faire* (even when inflected), all the compound nouns (since most of them are non-ambiguous terms, their list constitutes a good index);

— a given category, such as *verb conjugated in the third person singular* (V:3s), or *noun in the feminine plural* (N:fp), etc. Here are several examples of categories[5]:

A:p (adjective in plural), ADV (adverb),
DET:f (feminine determiner),
V:Kms (past participle, masculine singular), etc.

— a syntactic pattern represented by a regular expression or a graph; the following is a regular expression:

<être> (<ADV> + <E>) <DET> <N>

This pattern matches any sequence beginning with a conjugated form of the verb *être*, optionally followed by an adverb (<E> stands for the null word), followed by a determiner and then a noun. Note that categories match simple *and* compound words. In particular, <ADV> also matches compound adverbs. More generally, the user may apply to the text grammars expressed by recursive graphs; graphs typically represent:

— sets of synonymous expressions, such as : *perdre la tête, l'esprit, le nord*, etc. Graphs in differ-

3. For a discussion on the completeness of the DELACF, see in [Courtois; Silberztein 1989].

4. For a description of the graph editor of INTEX, see [Silberztein 1993].

5. For a syntactic description of the categories, see [Silberztein 1993].

ent languages can be linked, so that each matching sequence in the source language could be automatically associated with the corresponding graph in the target language (e.g. *lose one's head, mind, bearings,* etc.). A graph may represent all the expressions which designate an entity, or a process; indexing such graphs allows one to retrieve information in large corpora;

— pieces of a large-coverage grammar of the language. Recursive graphs are easily edited; standard operations on graphs (union, intersection, differences, etc.) help to build an easily maintained system of hundreds of elementary graphs. This construction has begun in LADL; we already have graphs describing adverbial complements which express a measure (temperature, speed, length, etc.), a time or a date (e.g. *le 17 février 1993, le premier lundi du mois de juin*) (Maurel 1989), some locative structures (Garrigues 1993), etc.

3. REMOVING AMBIGUITIES

In order to disambiguate words in texts, INTEX uses *cache dictionaries* and *local grammars*.

3.1. Cache dictionaries

Since the DELAF and DELACF dictionaries included in INTEX have a very large coverage, they contain a number of words which only occur in some specific domains; in addition, some frequent words may be associated with generally inappropriate information. For instance, *par* is usually a preposition in French, but in some cases it may be a noun (a technical term in golf). By default, each occurrence of this token will be considered ambiguous (preposition or noun). Cache dictionaries are used as filters: if INTEX finds a word in a cache dictionary, it will not look up the selected dictionaries and FSTs. If the user knows that in a given corpus, the token *par* is always a preposition, he/she enters the following entry in a cache dictionary:

par, par. PREP

Hence, the user can avoid unnecessary ambiguities by putting frequent words (or conversely, specific terms) in cache dictionaries adapted to each processed text.

Most compounds are ambiguous, since they formally are sequences of simple words; for instance, the sequence *pomme de terre* is not necessarily a compound noun in the following sentence:

Luc recouvre une pomme de terre cuite
(Luc covers a cooked potato)
(Luc covers an apple with scorched earth)

However, a number of compounds are not ambiguous, either because they contain a non-autonomous constituent (e.g. *aujourd'hui*), or because they are technical terms (e.g. *un tube cathodique, un sous-marin nucléaire*). By entering these non-ambiguous compounds in a cache dictionary, the user prevents INTEX from looking up dictionaries and FSTs for simple words; hence INTEX does not process these compounds as ambiguous.

3.2. Local grammars

A local grammar is a two-part rule: if a given sequence of words is matched, then each word in the sequence is tagged in the proper way. For instance, in the sequence *s'en donne*, *s'* is a pronoun (not a conjunction), *en* is a pronoun (not a preposition), and *donne* is a verb (not a noun). The corresponding local grammar would be:

s'/<PRO> en/<PRO> <MOT>/<V>

<MOT> stands for any word. Local grammars are represented by FSTs, hence their length and their complexity have no limit. Any number of local grammars may be used at the same time to disambiguate texts (FSTs merge easily); hence it is best to create small ones. Local grammars use the dictionary of the words of the texts, so they correctly handle sequences with compounds. Appendix 2 shows a few local grammars. INTEX includes a dozen "perfect" local grammars, that is, grammars that will never give incorrect tagging solutions; the user may add his/her own perfect (or probabilistic) disambiguating grammars.

3.3. The result of the parsing

After having selected linguistic tools (either dictionaries or FSTs), the user can parse the text,

that is, insert in the text all the linguistic information required by a syntactic parser. For instance, the text: *il la donne* would at this step be represented by the following expression:

il,PRO
(la,PRO:fs + la,DET:fs)
(donne,N:fs + donner,V:P1s + donner,V:P3s
donner,V:S1s + donner,V:S3s + donner,V:Y2s)

la can be a pronoun or a determiner; *donne* is a noun, or 5 conjugated forms of the verb *donner*. INTEX then builds the corresponding minimal automaton: the number of transitions of this automaton corresponds to the number of lexical ambiguities of the text (in the above example: 9 transitions). By selecting and applying local grammars to the text, the user effectively removes transitions in the resulting automaton. For instance, thanks to a simple local grammar (which describes the preverbal particles), the above text can be parsed to give the following expression:

il,PRO
la,PRO:fs
(donner,V:P3s + donner,V:S3s)

The remaining ambiguity corresponds to the tense of the verb: indicative or subjunctive present. The corresponding automaton has only 4 transitions. Hence, the number of transitions can be used as a quantitative tool to measure the efficiency of the removal of ambiguities. By selecting one local grammar at a time, or by merging several, the user is able to apprehend exactly how each grammar covers the text, and performs in terms of deleting transitions.

CONCLUSION

INTEX is used for several purposes:

— lexicographers who build dictionaries for compounds (or technical terms) try to find new ones by applying characteristic patterns to big corpora, such as: *<N> (de + d' + de la + du + des) <N>*;

— linguists who study specific syntactic structures use INTEX to find attestations of these structures. For instance, one may search for the following structure in order to find predicative nouns associated to the support verbs *avoir, donner, être, faire*:

(<avoir>+<donner>+<être>+<faire>)
(<ADV>+<E>) <N>

— our objective is to build a large grammar which covers as much of the language as possible. By applying "pieces" of grammar to big corpora, and then studying the outputs, one can correct and refine each piece, and incrementally develop the global grammar;

— INTEX is used to find "semantic units" in large technical texts, hence it constitutes a good information retrieval system.

REFERENCES

Courtois, B., Silberztein M. Eds, (1989). *Les dictionnaires électroniques*. Langue française, Larousse : Paris.

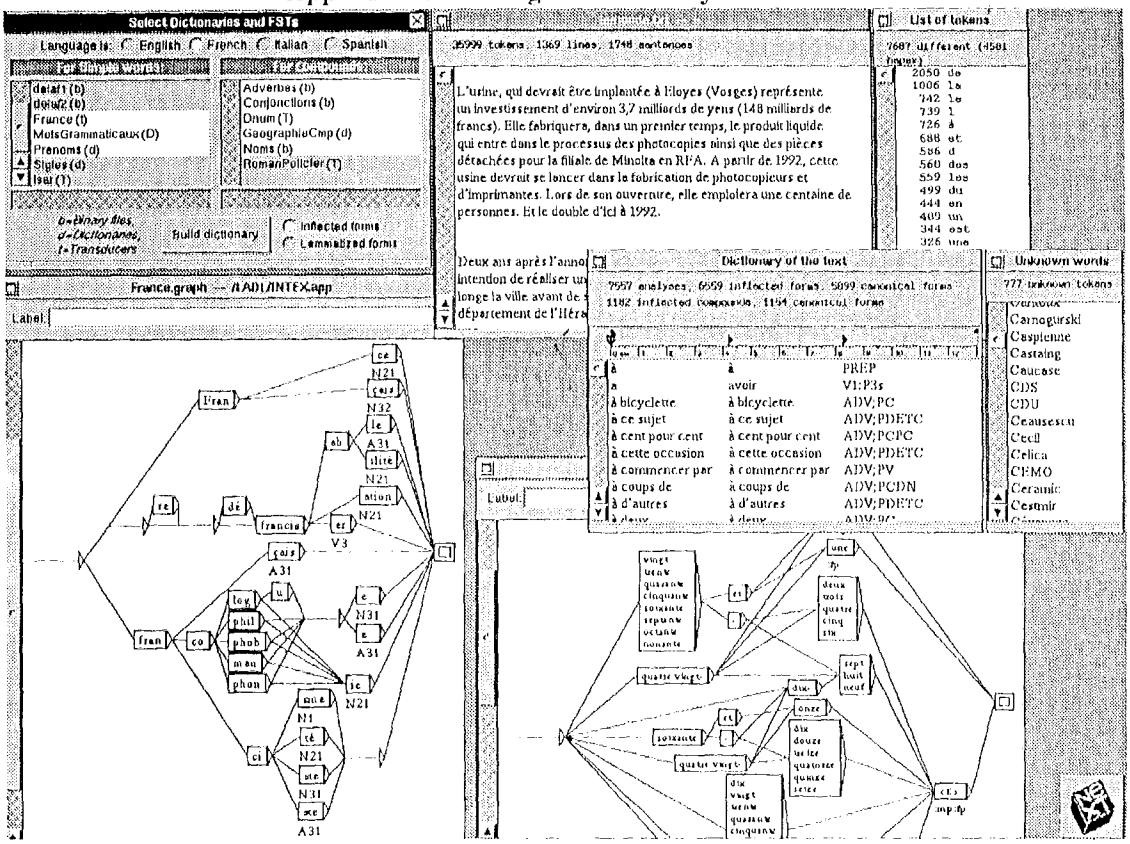
Clemenceau D. (1993). *Structuration du lexique et reconnaissance de mots dérivés*. Thèse de doctorat en informatique, LADL, Université Paris 7 : Paris.

Garrigues M. (1993). Prépositions et noms de pays et d'îles : une grammaire locale pour l'analyse automatique des textes. In *Lingvisticae Investigationes* XVII:2. John Benjamins: Amsterdam.

Maurel D. (1989). *Reconnaissance de séquences de mots par automate*. Thèse de doctorat en informatique, LADL, Université Paris 7 : Paris.

Silberztein M. (1993). *Dictionnaires électroniques et analyse automatique de textes*. Masson : Paris.

Appendix 1: Building the dictionary of the text



Appendix 2: Local grammars and the removal of ambiguities