NATURAL LANGUAGE PROCESSING SOFTWARE TOOLS AND LINGUISTIC DATA DEVELOPED BY FRANCE TELECOM R&D

Emilie Guimier De Neef, Malek Boualem, Christine Chardenon, Pascal Filoche, Jérôme Vinesse

Abstract:

The NLP tools developped by France Télécom Research and Development's "Natural Languages" team are based on a text analyser which constitutes the core technology of the team. Lexicons and grammars are the linguistic resources necessary for the analyser to parse texts. Processing a new language supposes to create these linguistic resources.
In this paper, FTR&D NLP tools and text analyser are presented. The content of lexicons and grammars is explained in order to sketch what the addition of a language like Hindi would involved.

Introduction:

France Télécom Research and Development's "Natural Languages" team is made up of about a dozen people whose speciality is the natural languages processing of written documents. Many years of work have allowed the development of a set of NLP tools that facilitate the access of texts content. Most of these tools are built on the output of a text analyser which constitutes the core of the technology. They are available for different languages and can easily be adapted to new languages.

The purpose of this article is not only to present these software tools and the text analyser, but also to present underlying linguistic resources in order to sketch out what the adaptation of these tools to a language like Hindi would involve.

NLP Tools

Six tools have been developed in FTR&D. Their purpose is briefly described below.

Language and charset identification
This software makes it possible to identify the language used in a text as well as the encoding of the characters in the document. The length of the document may vary from several words to several thousand words. It can process text files or word sequences typed on the keyboard. It can not only recognise the language used in a text according to the score obtained, but also returns the score obtained for every other language constituting the language referential. The software can be integrated into a larger application that needs this language identification functionality.
In its present version, it can identify 25 languages and 11 character encoding systems. The addition of a new language or a new encoding system is a very simple and rapid process, which is carried out by a learning method on a corpus.

Linguistic information filtering
The software represents the relevant contents of a text by extracting and standardising the right descriptors. It is to be used as a pre-processing tool for various processes

manipulating texts: indexing, text-mining, information retrieval, classification or document filtering.
It gives the linguistic analysis of a sentence or a text written in a natural language. It identifies parts of speech: word categories (nouns, verbs, articles), numbers, dates, mail addresses, smileys, etc.
It corrects some errors (typing or spelling errors), normalises forms (verbs in the infinitive, nouns in the singular), recognises compounds, generates sets of forms (inflectional paradigm of verbs etc.).
Relevant descriptors can be chosen according to the task (delete articles, keep or delete dates, etc.)

Text summarisation
The text summarisation tool produces a shortened version of a given text. It extracts relevant sentences and key words in the original text.
Summarisation makes long and voluminous texts accessible and allows them to be consulted rapidly. It can make long texts accessible via small terminals.

Text thematic classification
This software analyses a text or part of a text and classifies it according to a thematic categorisation; examples of themes: medicine, computer science, chemistry, biology, finances, etc. The themes are classified according to their importance.
We have defined about 180 domains which can be refined depending on the desired precision. The data can be adapted to specific domains.

Question/answering in natural language
This software finds the precise answer to simple questions asked in natural language. The system can search for the answer on the Web or on a particular intranet. It facilitates access to large textual databases by using natural language.
The search for the answer is made through the Internet via a search engine available on the Web or into a textual database situated on a particular intranet (news documents, etc.) with a specific search engine.

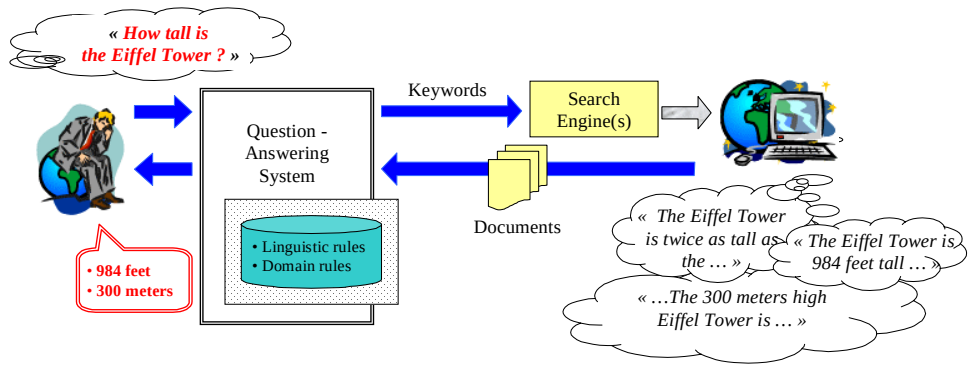

Fig.1. Searching the Internet and intranets using Q&A on search engines

Natural language is used to ask a question to which the system finds the accurate answer. The questions under consideration are factual questions. They relate to answers located in a single document, within a single sentence (nouns of people, nouns of cities, places, dates, monetary values, etc).

Examples of questions that can be asked to the system:

Questions about people:
Who built the Eiffel Tower ?
Who wrote Moby Dick ?
Questions about companies, etc.:
Which are the Japanese mobile operators ?
Which are the subsidiaries of France Telecom ?
Questions about acronyms:
What does VXML mean ?
What does ADSL mean ?
Questions about places:
Where did Gandhi die ?
Where is Montmartre ?
Questions about dates of events:
When did Colombus discover America ?
When was Gustave Eiffel born ?
Questions about quantities:
How high is the Eiffel Tower ?
For how much did France Telecom buy Orange ?

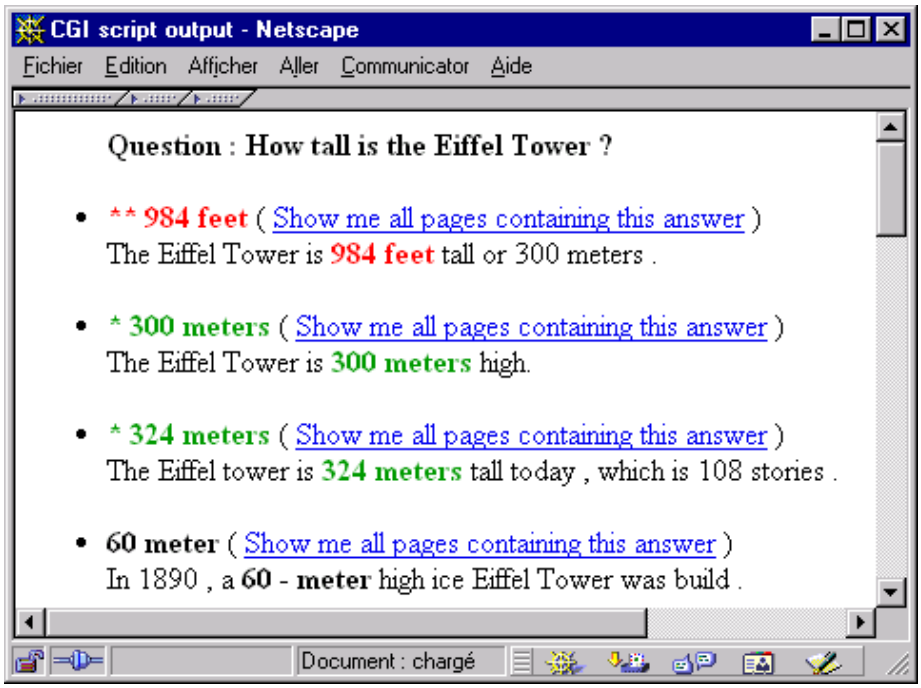

Fig.2: Answers are ranked and displayed to the user

Question/Answering systems give pertinent answers. They allow time gain: unlike search engines which only return a set of documents susceptible of presenting the answer, the tool directly returns the answer to the question asked by the user.

Generation of linguistic exercises for language learning
This tool has educational aims. It allows users to generate their own linguistic gap exercises starting from a given text. It automatically generates linguistic exercises online and corrects answers. It allows users (such as school pupils) to learn in both supervised and unsupervised situations.
The current version is available for English, and proposed exercises concern the following categories: subject pronouns, prepositions and verbal forms.

This set of linguistic software and services is particularly suited to any kind of on-line and in-door applications and services addressing large amounts of texts written in different languages and belonging to many different cultural and thematic contexts. They are based on the latest advanced natural language processing technologies and adapted to the Internet context. These software tools can run on the most frequently used OS platforms (Linux, Solaris, Windows)

The core technology: a text analyser

The core technology of these tools is a text analyser which has been developed in FTR&D for several years. A text analyser allows a transition from a text, that is a sequence of typographic symbols, to a representation of this text that can be used by a machine.

Ambiguity characterizes natural languages: a word cannot be interpreted alone, contextual information must be taken into account. The aim of the text analyser is to resolve this ambiguity in order to obtain the correct interpretation for the words of a sentence.

Architecture of the text analyser

Segmentation: The text is divided into paragraphs, sentences and segments (words, punctuation marks, figures etc.)

Interpretation: Segments are interpreted: words are found in the lexicon, compound-words are recognized etc. When a word is not found in the lexicon[*], several kinds of corrections are available:

> *Morphological analysis* serves to cut into pieces an unknown word in order to identify neologisms produced by usual derivational rules of the language. Example: "inactionability" is not listed in lexicons but it is understandable because it is a regular derived word that comes from the adjective "actionable" to which the prefixe "in" and the suffix "ility" have been added. This correction can also identify accidental agglutination of words such as "thepeople".
>
> *Accents correction* can find forms that are identical but for the accents. Example: "peche" is unknown in French lexicons but "pêche", "péché", "pêché" are not.

[*]Unknown words arise very frequently in real corpora like texts from the press, databases, requests sent to search engines etc.

Typographical misprints correction finds all words identical to an unknown form by the insertion, substitution, permutation or deletion of characters. Example: "analysys" gives "analysis", "analyses" and "analysts".

Phonetic correction can find graphical forms that can be pronounced in the same way as the unknown word. Example: the French form "bato" can correspond to the words "bateau" and "bateaux".

Morpho-predictive correction can guess the part of speech of an unknown word by analysing its ending. For instance, the English ending "or" generally corresponds to a noun ending.

Morpho-syntactic analysis: parts of speech are disambiguated using a shallow parser and lemmas are found for each word of the text.

Syntactic analysis: syntactic relations between words of a sentence are identified and the output is a tree.

Semantic analysis: the logical structure of the sentence is extracted in the form of a graph.

Example of analysis "flying planes can be dangerous"

The *interpretation* of the five words of the sentence "flying planes can be dangerous" shows that the words "flying", "planes" and "can" are ambiguous[*]: "flying" can be a verb, a noun and an adjective, "planes" may be a noun or a verb, "can" may be a modal auxiliary and a noun. "be" is a verb and "dangerous" an adjective.

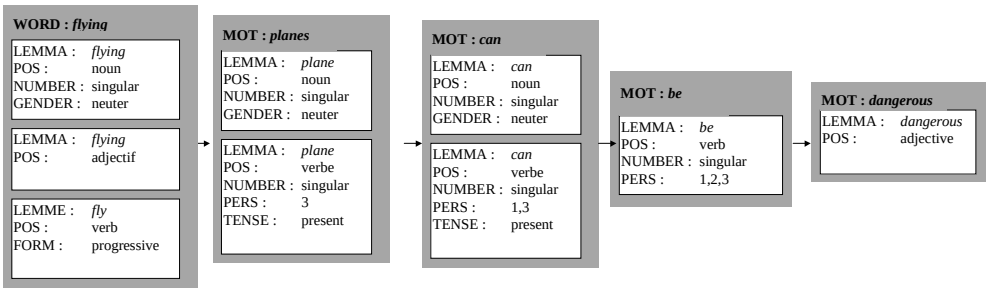

Fig.3: Lexical ambiguities

The *shallow parsing* of this sentence gives 2 interpretations which can be summed up in the following results showing part-of-speech tagging:

"flying"/"flying"/A "planes"/"plane"/N "can"/"can"/V "be"/"be"/V "dangerous"/"dangerous"/A

"flying"/"fly"/V "planes"/"plane"/N "can"/"can"/V "be"/"be"/V "dangerous"/"dangerous"/A

[*] We only deal here with part-of-speech ambiguity, semantic ambiguity is put aside at this stage.

These outputs respectively correspond to the interpretations "planes that are in the air can be dangerous" and "it can be dangerous to fly planes". The first field is filled by the initial word form, the second one corresponds to the lemma and the third one to the part of speech (V = verb, A = adjective, N = noun).

The syntactic parsing of the sentence produces a tree which identifies the syntactic relations between words in the sentence. In the first interpretation, *plane* is the subject of *be* whereas in the second one, *flying* is the head of the subject phrase:

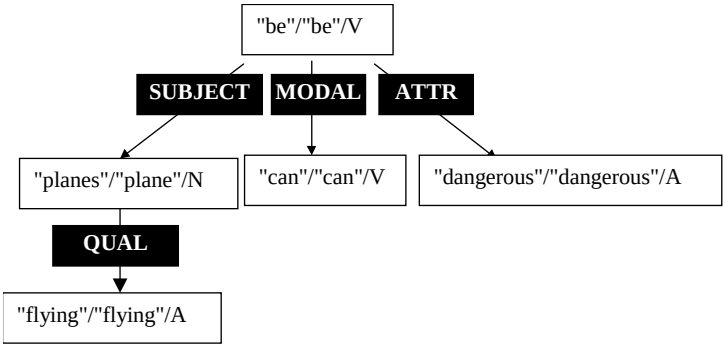

Fig.4: Syntactic analysis for the interpretation equivalent to "planes that fly are dangerous"

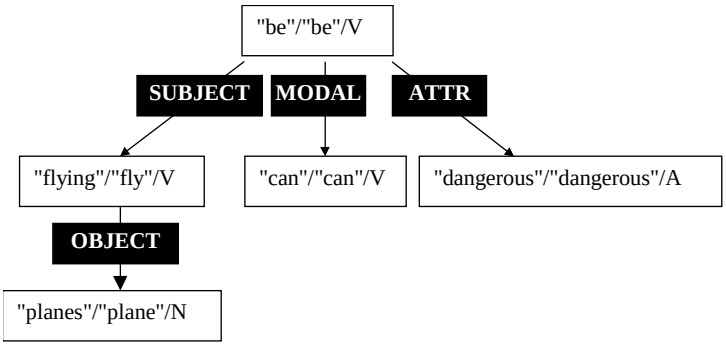

Fig.5: Syntactic analysis for the interpretation equivalent to "it is dangerous to fly planes"

The semantic analysis gives rise to 2 interpretations which can be summarised in the following logical formulae:

DANGEROUS(FLY(x,PLANE))
DANGEROUS(FLYING(PLANE))

This level of analysis[*] allows a certain freedom from syntax and it is therefore possible to obtain similar representation for sentences that are not close syntactically. The semantic representation of the following pairs of sentences will be identical since a syntactic transformation (passivisation and nominalisation) links them:

"Mary gave a book to Peter" / "Peter was given a book by Mary"
"The production of films" / "Films are produced"

Linguistic resources

The efficiency of the text analyser mostly depends on the linguistic resources that are connected to it. Two kinds of linguistic resources must be distinguished:

- lexicons that list and give information on the language's vocabulary
- and grammars that explain the grammatical rules of the language and allow to obtain a representation of the sentences.

Lexicons

Lexicons needed by the text analyser must not be considered as the electronic form of a paper dictionary. The information has to be explicit and exhaustive in order for the grammar to interpret it. Classically, three levels of information are distinguished: a morphological level, a syntactic level and a semantic level.

Morphological information for each word consists of:

- the lemma or main graphical form,
- the inflectional paradigm,
- the part of speech (verb, adjective, adverb, noun...)

Syntactic information gives information on the behaviour of words in sentences. It typically consists of:

- information relative to the position of the word in a sentence:
  - is a particle necessary ?
  - which auxiliary may be used ?
  - position of adjectives with respect to nouns
  - ...
- information relative to the complements governed by the word:
  - number of complements
  - case or governed preposition
  - kind of complements (are they nominal, sentencial ?)
  - ...

Semantic information formalizes the meanings of words. For each meaning, the following information should be encoded:

- domain (medicine, law, education...)
- semantic class (human, artefact, plant...)
- predicative representation (type of the predicate, number and types of its arguments)

---

[*] This level of analysis has been used in MKBEEM: an IST research program whose aim is the questioning of multilingual commercial databases using natural language.

- relation with other meanings (synonymy, hyponymy...)

In our data, morphological and syntactic information is encoded in one database whereas semantic information is encoded in another one.

Morpho-syntactic lexicons

Lexical data are handled using ASCII formatted files. Two kinds of files are to be distinguished:

- Files containing lexical entries and morphological codes,
- Files describing morphological codes.

In a file containing lexical entries, one line corresponds to one lexical entry. Each line is divided into 5 fields conforming to the following description:

<graphical form>	[<phonetical form>]	<morphological code>	<syntactic features >	<sense>

Syntactic information is encoded using features represented with a string in capitals.

Examples of entries:

abate, [], V2, -INTRANS, $abate_1
abbreviate, [], V2, -TRANS, $abbreviate
abbreviation, [], N4, , $abbreviation_1
can, [], V63, -AUX_MODAL, $can_10
capitalise, [], V196, -V_PREP/ON-TRANS, $capitalise_on
believe, [], V2, -TRANS-SUBCAT/THAT_SP, $believe_3

Here follows the meaning of features displayed:

INTRANS:	the verb is intransitive (no governed complements)
TRANS:	the verb is transitive (an object governed)
AUX_MODAL:	the verb is a modal auxiliary
V_PREP/ON:	the preposition "on" introduces the complement of the word
SUBCAT/THAT_SP:	a completive subordinate clause introduced by "that" is required.

Phenomena of inflection are described using an 'add-and-remove' formalism. Each lexical entry is associated with one morphological code which itself refers to one radical and to one or several pairs of inflectional suffix and morphological features.

Features combinations indicate:

- grammatical category (eg. NOUN, VERB, ADJ…)
- and morphological features associated with the inflectional suffix (GENDER, NUMBER, CASE, TENSE, MOOD, PERSON, VOICE…)

The information is divided into 2 files. The first one serves to calculate the radical of the word whereas in the second one available pairs of suffix and features are declared.

Format for the radical model file:

<morphological code identifier>	<number of characters to be removed from the graphical form in order to obtain the graphical radical>	<number of characters to be removed from the phonetic form in order to obtain the phonetic radical>	"<example>"

Format for the inflectional suffixes model file:

<suffix to be added to the graphical radical>	<suffix to be added to the phonetic radical>	<morphological code identifier>	"<example>"	<features combination>

Examples from the radical model file:

```
V2    1    0    abase
N33   0    0    child
N4    0    0    dog
```

Examples from the inflectional suffixes model file:

```
"e"      ""    V2    "abase"    VERB-INF
"e"      ""    V2    "abase"    VERB-PRES-1PRS
"e"      ""    V2    "abase"    VERB-PRES-2PRS
"es"     ""    V2    "abase"    VERB-PRES-3PRS-SINGULIER
"e"      ""    V2    "abase"    VERB-PRES-3PRS-PLURIEL
"ing"    ""    V2    "abase"    VERB-PART-PRES
"ed"     ""    V2    "abase"    VERB-PRETERIT
"ed"     ""    V2    "abase"    VERB-PART-PASS
""       ""    N33   "child"    NOM-SINGULIER
"ren"    ""    N33   "child"    NOM-PLURIEL
""       ""    N4    "cruel"    NOM-SINGULIER
"s"      ""    N4    "cruel"    NOM-PLURIEL
```

Inflectional forms of the verb "believe":

LEMMA: believe RADICAL: believ MORPHOLOGICAL CODE: V2

VERB-INF	believe
VERB-PRES-1PRS	believe
VERB-PRES-2PRS	believe
VERB-PRES-3PRS-SINGULIER	believes
VERB-PRES-3PRS-PLURIEL	believe
VERB-PART-PRES	believing
VERB-PRET-1PRS-2PRS-3PRS	believed
VERB-PART-PASS	believed

Semantic information

Semantic information is contained in a multilingual thesaurus which is a sort of catalogue of meanings.
Its purposes are:

- to list the different meanings of words in the languages we study,
- and to structure this information so as to exploit its content with NLP tools.

Our thesaurus is divided into 180 domains like politics, agriculture, food etc. The domains are themselves divided into themes. Themes contain synsets which are multilingual bullets of synonyms.

Synsets are structured and give information about:

- the lexicalisation(s) of the meaning in different languages (with a capital letter indicating the corresponding language)
- the semantic class of the meaning (introduced by SRT)
- lexical semantic information (introduced by SEM): with this information it is possible to obtain relationships between meanings (ex: between "sow" and "sower") and to associate a predicational representation with the meaning.

Example of synset corresponding to "sow":

SEM [X] ACTION agent patient
SRT [X] ACTION
VBA [F] semer
VBA [F] ensemenser
VRB [E] sow
VRB [P] ssiec
VRB [D] säen
VRB [A] zaraãa
VRB [L] semear

These three levels of lexical data make up a huge amount of knowledge which is interpreted by the grammars and collated to texts.

Grammars

Shallow parsing and syntactic analysis involve two kinds of grammars. The shallow parsing requires a grammar that describes available series of grammatical categories in the

language; the syntactic grammar requires more complex rules in order to determine the relations that link words.

Our shallow parser uses a regular grammar that produces chunks. These grammatical rules are completed by agreement rules and by linear order constraints.

Example of rules:

Règle Prep Det => GP-D
Règle GP-D Adj => GP-AN
Règle GP-D Noun => GP-NC
Règle GP-AN Noun => GP-NC

This grammar allows to identify prepositional phrases chunks like "with the beautiful colors", "in the kitchen".

Example of agreement rules:

ContraintesIntraGroupe GP-NC 2
Propriétés NUMBER NUMBER
Contrainte AgreementBetween
On Det Noun

This rule of agreement specifies that in a chunk tagged GP-NC, words belonging to the Det and Noun parts of speech must agree in NUMBER.

Shallow parsing cannot solve all syntactic puzzles. It can identify chunks and decide on the part of speech of words, but it cannot determine the sentence structure. For instance, the shallow parsing of a sentence like "I want to go to Lille without changing trains" will show 5 chunks and identify parts of speech but it will not indicate that "to Lille" is the complement of the verb "go", and "trains" that of "changing". And this is crucial information if high level NLP applications are aimed at:

(I/Pro want/V) (to/Pn go/V) (to/P Lilles/Np) (without/P changing/Ving) (trains/N)

This limitation of the shallow parsing is mainly due to the fact that non-local information like the governing of complements cannot be taken into account. The formalism used in the syntactic parser allows this.

The result of our syntactic parser is a tree. The formalism used for the grammar rules is that of dependency grammars.

Example of rule:

RègleAttachement OBJD-2 OBJD
Schéma GV-PT << GN-NC
ConditionsPrincipal (TRANSITIF/OUI SY_OBJD/!)
AutresConditions ((P += SY_OBJD/+))

This rule controls the attachment of an object to a verb. The relation OBJD is created between a verb (symbol GV-PT) and a noun (symbol GN-NC) if the

following conditions are fulfilled: the verb must be transitive (feature TRANSITIF/OUI) and must have not already met a direct object (SY_OBJD/!).

The tree resulting from the syntactic analysis may be mapped into a semantic graph using specific sets of rules that are not detailed here.

Available languages

At the present time, we have linguistic resources for French, English, German, Spanish, Polish and Arabic. Resources for Portuguese are being developped. The following table sketches more precisely the progress of work for each language:

	morpho-syntactic lexicon	advanced syntactic information	semantic information	Chunking grammar	dependency grammar
French	X	X	In part	X	X
English	X	X	In part	X	X
Spanish	X		In part	X	under construction
German	X		In part	X	under construction
Polish	X	In part		X	X
Arabic	X		In part	Under construction	
Portuguese	under construction				

Apart from our language identifier, all our NLP tools necessitate text analysis and the development of linguistic resources. Precise NLP tools requirements are given below as well as an estimation of the time needed for adapting the tool to a new language:

NLP Tools	**linguistic resources required**	**Available languages**	**Time needed for adding a new language**
Language and charset identification	_	Any language	1/2 weeks
Text summarisation	chunking	Fr, En, Sp, Ge, Po	3 / 6 months
Thematic classification	chunking and semantic information	Fr	12 / 15 months
Linguistic information filtering	chunking	Fr, En, Sp, Ge, Po	6 / 7 months
Question/Answering system	chunking and specific data	Fr, En, Po, Ge, Sp	7 / 9 months
Generation of linguistic exercises	chunking and specific data	En	7 / 9 months

Adding a new language

Several reasons make the addition of a new language only a matter of data:

- the architecture of the text analyser is modular: the linguistic data and the algorithmic component are separated from each other,
- the system can be parameterised: analysis strategy can depend on the language,
- a large panel of languages have already been studied and lots of mechanisms are available: agglutination of words (like in German) is dealt with, rich inflectional languages (like Polish) poses no real problems, complex morphological systems (like Arabic) have been solved in adapting our model of lexicons etc.

As far as Hindi is concerned, though we have not studied the specificities of this language in any depth, our initial impressions are that the adaptation of our NLP tools to this language would necessitate no specific developments. The creation of linguistic resources (lexicons and grammars) would be enough to parse Hindi texts. Here follows a few comments on aspects of Hindi that we have already noticed:

Our model for inflectional morphology would be sufficient to deal with Hindi's morphology as it seems to be close to that of French or Spanish (prefixes+radicals+suffixes). Hindi has case inflection but its case system seems far from being as complex as that of Polish. On the other hand, the Hindi verbal system seems rich and specific with the frequent presence of the "honâ" auxiliary. This question needs more investigations but the mechanisms used for analysing the compound verb forms in French or in German may be sufficient to cope with this problem.

As far as syntax is concerned, Hindi is a S O V language and this words order is also encountered in German subordinate clauses. It has postpositions, contrary to the languages we have treated so far. A more noteworthy fact concerns a syntactic phenomenon that we have never dealt with: ergativity. In certain circumstances, transitive verbs agree with their object and not with their subject as is commonly the case. Though we have never encountered this phenomenon before, it may not be out of reach since studies have shown that dependency grammars are able to cope with it.

Conclusion

With regard to the languages already studied by the FTR&D NLP team (French, English, Spanish, Polish, German and Arabic), the Hindi language seems to present no insurmountable difficulties. The flexibility of our text analyser should allow the processing of Hindi texts provided that Hindi linguistic resources are created. These resources consist of lexicons and grammars.

Once text parsing is possible for Hindi, minor developments would allow most FTR&D NLP tools to run: Question/Answering system, Information retrieval, Automatic Summarisation etc. The addition of semantic resources makes thematic classification possible.

Text parsing not only allows our tools to run, but also renders other applications possible: pre-processing of text before text to speech applications for instance. It offers a lot of

possibilities and constitutes a key component of all NLP developments. The constitution of linguistic resources for Hindi is the first step to reach this goal.

REFERENCES

M. Boualem, *Hahooa Arabic Directory and Arabic Information Retrieval using NLP*, workshop *ARABIC Language Processing: Status and Prospects*, http://www.elsnet.org/acl2001-arabic.html, ACL conference, Toulouse, 2001

C. Chardenon, *Un environnement de développement pour le TALN*, journée ATALA *Environnements de développement d'applications de TAL : état, enjeu*, http://www.atala.org/je/011215/Chardenon.pps, 2001

F. Duclaye, P. Filoche, J. Sitko, O. Collin, *A Polish Question-Answering System for Business Information*, 5th International Conference on Business Information Systems, Poznan, 2002

F. Duclaye, F. Yvon, O. Collin, *Using the Web as a Linguistic Resource for Learning Reformulations Automatically*, posters corpora and corpus tools, Third International Conference on Language Resources and Evaluation, Las Palmas, 2002

E. Guimier, *L'analyse de textes : entre données linguistiques et robustesse*, journée METIL http://www.apil.asso.fr/metil.htm, 2002

M. Plu, C. Chardenon, L. Maupeu, *On-line hypermedia contents for learning and practicing foreign languages*, http://www2002.org/educationtrack.html, the eleventh WWW conference, Hawai, 2002