MT Summit VIII

Santiago de Compostela (Spain), 18 - 22 September 2001

New Generation Systran Translation System

Jean Senellart, Péter Dienes, Tamás Váradi

SYSTRAN, HIS
1, rue du Cimetière - BP.7
95237, Soisy-sous-Montmorency Cedex
France
senellart@systran.fr, dienes@nytud.hu, varadi@nytud.hu

Abstract

In this paper, we present the design of the new generation Systran translation systems, currently utilized in the development of English-Hungarian, English-Polish, English-Arabic, French-Arabic, Hungarian-French and Polish-French language pairs. The new design, based on the traditional Systran machine translation expertise and the existing linguistic resources, addresses the following aspects: efficiency, modularity, declarativity, reusability, and maintainability. Technically, the new systems rely on intensive use of state-of-the-art finite automaton and formal grammar implementation. The finite automata provide the essential lookup facilities and the natural capacity of factorizing intuitive linguistic sets. Linguistically, we have introduced a full monolingual description of linguistic information and the concept of implicit transfer. Finally, we present some by-products that are directly derived from the new architecture: intuitive coding tools, spell checker and syntactic tagger.

Keywords:

machine translation architecture, implicit transfer, declarativity, management of linguistic resources

Introduction

The Systran machine translation (MT) system today, generally classified as a transfer translation system, is the result of 30-years worth of accumulation of linguistic and technical expertise. The tremendous amount of development resulted in very large multilingual linguistic resources. During this time, the system has been rewritten several times, progressively integrating the latest MT technology, for example, backtracking capacity, introduction of a fine-grained semantic description, etc. The evolution of the systems shows that the major problem inherent in Systran is not so much a question of technical or linguistic capacity, but more a question of a) maintainability of linguistic resources, b) reusability of the resources, c) global mastering of the linguistic process during translation, and d) opening of the system to the very demanding market.

In today's market, MT is facing more demanding requirements than ever:

- To rapidly develop new languages and cross-language translation systems;
- To provide personalized translation service, e.g., personalized dictionaries, customerizable format filters and specific grammars;
- To expand the translation service to more general natural language processing, e.g., multilingual indexing, document authoring, intuitive integration with translation memory;
- To provide easy access for direct terminology exchange, e.g., exchange between different systems, integration of linguistic acquisition tools.

In response to the above requirements, we have initiated a thorough redesign of the Systran translation engine from the ground up. The development work is carried out in the framework of the MATCHPAD project, a 5th Framework EU funded project, which is aimed at extending the suit of Systran MT systems into Hungarian and Polish. So far, we have developed new language pairs involving Hungarian and Polish (i.e. English to Hungarian and Polish, and Hungarian and Polish to French), while the Arabic (i.e. English and French to Arabic) and new cross-language systems are in progress.

Although the new technical structure is quite different from the traditional systems, the most important feature of the redesign lies in the connection of the system to linguistic description. Especially, the traditional systems and linguistic resources are used as the basis for the new development in order to ensure the continuity in translation quality. While at the same time, the continued improvement of the existing language systems can also benefit from the renewal.

In this paper, we use the two language pairs, English-Hungarian and Hungarian-French, to demonstrate the design. The development done is in collaboration with the Hungarian Academy of Sciences.

System Overview

The overview architecture of the new system design is presented in Figure 1. The new generation of the systems continues to use the transfer approach.

$$Analysis \rightarrow Transfer \rightarrow Synthesis$$

The most prominent changes are the presence of four different independent external resources, the concept of implicit transfer, and the radical changes in the internal organization of each module.

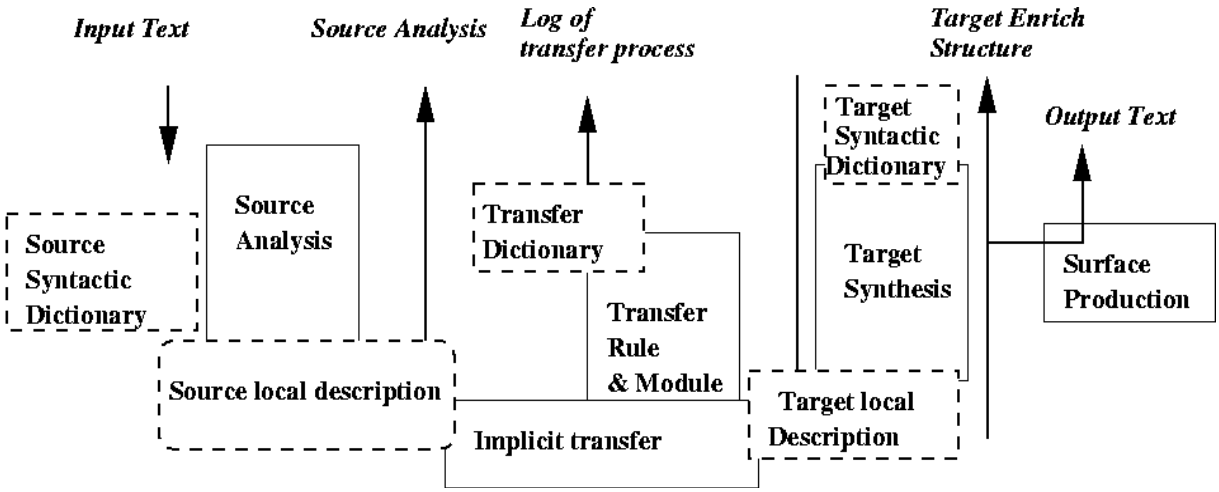

Figure 1. Architecture of the new systems.

From a theoretical point of view, the design of the new architecture reflects a higher-level transfer system (Figure 2). Nevertheless, the theoretical behavior of the system is not our interest. Our express objectives focus on the capability of handling principled linguistic resources, and the pragmatic incrementability of such resources.

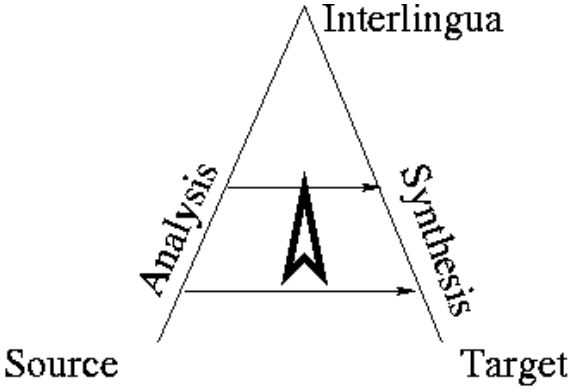

Figure 2. Representation of MT structure. The arrow shows the shift of the new systems on this axis.

Technically, the main characteristics of the system are:

Efficiency

Finite automaton libraries are extensively used for lookup purposes. These Unicode-compliant libraries integrate localized optimization strategies. The optimization enables us to store fully inflected lists for languages such as German, which has a significant number of case combinations. For even more highly inflected languages such as Hungarian, the same libraries allow a two-level lookup strategy.

The finite-state automata provide a constant lookup time at any level, independent of the size of the resources. They also give access of lookup variants, for example, the spelling-check feature. Finally, the choice of finite-state representation allows the natural extension from linear dictionary to complex linguistic resources.

Declarativity

Declarative systems define the behavior of the system without entering into the details of internal programming languages. Generally, declarativity requires a meta-language of description. The extensive experience from developing the traditional systems proves that the use of the meta-language and the power of representation of the language are not really the most critical point. Whatever the power of representation of a language is, there exist some linguistic phenomena requiring exceptional processing, or lexical rules that will require to be coded at a lower level (or the meta-language itself will have such a complexity that it will be a real programming language by itself). In parallel, for a given level of coding, the distribution of the typology of the rules shows a "Zipf"-like behavior. This leads to the focus on frequent patterns in order to increase readability. The right level of meta-description is *the key* for allowing efficiency and productivity.

For that reason, we have several different levels of coding depending on the various linguistic phenomena. Each level is formalized according to a single textual file, which describes all of the linguistic features that the systems deal with.

Table 1 gives a sample of feature definitions. By way of automatic generation of internal structure and lexical parser, this textual representation is the same for direct access to internal rules, transfer rules (Table 2), or disambiguation rules (Figure 5). This external structure definition is linked at runtime with a strict internal and external type checking to maintain overall coherence.

```
<syntax_HU>
      enum functional_category (noun, adj, verb,
adverb, pro, det, adv, conj, prep, part, intj)
      boolean void_translation
      boolean not_found
      reference<WORD> predicate_linked
      boolean predicative
      reference<WORD> direct_object
      reference<WORD> predicate
<syntax_HU_Noun:syntax_HU>
      enum type (commonnoun,propernoun,acronym)
      enum case (ill, ine, ela, dat, ade, sub, del,
abl, all, cau, ter, sup, ess, fac, ins, acc, dis, soc,
nom)
      enum number (singular,plural)
      enum possessor_number (p_singular,p_plural)
      enum possessor_person (poss_1,poss_2,poss_3)
      reference<WORD> modified_by_number
      reference<WORD> possesses
      boolean anaposs
      boolean nodet
      reference<WORD> determinant
      reference<WORD> modified_by_adj
      reference<WORD> dirobj_of
      reference<WORD> governs_relpro
```

Table 1. Sample of Hungarian syntactic features. The formalism describes a minimal semantic in attribute and inheritance between structures.

Finally, in order to reduce the surface complexity of some descriptions, graph representations (as in Figure 5, 6) are introduced. These graphs give a visual representation of coding on any level (except for the lowest, i.e. the programming, level), and they are easily understood.

Modularity

The system has been designed in order to be more modular. The modularity means that we can extract each component from the system and use it for other purposes. This characteristic is in agreement with the organization of linguistic resources. For example, Figure 3 is an output of source analysis of a sentence. The generated file can be modified and used as the input text for the transfer function. We have the same capability for the tagged target text before surface synthesis (Figure 4).

```
<clause subject=[239] predicate=[240]>
(236) How many [<syntax type="DET">+functional_pos
=funcadjright+pronoun_type=interrogative+type=pron
oun+modifies_right=[237]</syntax>]
(237) apples [apple<syntax type="N">+functional_po
s=funcnoun+object_of_verb=[240]+dirobj_of=[240]+co
ncrete+countable+number=plural+type=commonnoun+mod
ified_by_adj_left=[236]</syntax>]
(238) would [will<syntax type="V">+auxiliary+tense
=past+mood=question+type_aux=aux_conditional</synt
ax>]
(239) he [he<syntax type="PRO">+functional_pos=fun
cnoun++human+number=singular+type=personal+perspro
_type=subject+person=3+agent_of_verb=[240]</syntax
>]
(240) eat [eat<syntax type="V">+functional_pos=fun
cverb+conditional+tense=past+mood=question+object_
of_action=[237]+direct_object=[237]+agent=[239]</s
yntax>]
</clause>
<clause subject=[242] predicate=[243]>
(241) if [if<syntax type="CONJ">+type=subordinate<
/syntax>]
(242) he [he<syntax type="PRO">+functional_pos=fun
cnoun+human+number=singular+type=personal+perspro_
type=subject+person=3+agent_of_verb=[243]</syntax>
]
(243) was [be<syntax type="V">+meaningid=m1+functi
onal_pos=funcverb+tense=past+modified_by_adv=[244]
+agent=[242]</syntax>]
(244) at home [<syntax type="ADV">+functional_pos=
funcadv+subject_of_clause=[242]+type=simple+modifi
es_verb=[243]</syntax>]
</clause>
```

Figure 3. Tagged sentence after analysis.

Implicit transfer

The concept of implicit transfer has been developed and integrated into the new engines. The principle is to build a source description of some linguistic phenomena: mainly support verb description or various local expressions (for example, the expression of dates, see Figure 6). The description of such phenomena is very complex using the traditional transfer rules. The implicit transfer module uses a parallel target description of the same phenomenon, aligns source and target sentences, and thus synthesizes a syntactically correct target expression (Senellart et al. 2001).

```
(247) Hány [hány<syntax type="PRO">+type=interrogati
ve</syntax>]
(248) almát [alma<syntax type="N">+number=singular+t
ype=commonnoun+case=acc+modified_by_number=[247]</sy
ntax>]
(249) enne [esz<syntax type="V">+subject=[252]+quest
ion+person=3+object_type=indefinite+number=singular+
tense=past+mood=conditional</syntax>]
(253) ha [ha<syntax type="CONJ"></syntax>]
(255) otthon [<syntax type="ADV"></syntax>]
(250) lenne [van<syntax type="V">+person=3+number=si
ngular+tense=present+mood=conditional+modified_by_ad
v=[255]</syntax>]
```

Figure 4 - Target sentence after transfer and synthesis.

Linguistic Resources

In the traditional Systran systems, the average size of the main dictionaries for European languages is about 200,000 entries. Moreover, one single external dictionary (e.g., the EEC dictionaries) can have up to an additional 200,000 entries. Handling and maintaining such huge dictionaries are a very complex task. For this reason we have focused on suppressing redundancy by introducing, for instance, real monolingual dictionaries. Moreover, in a traditional Systran expression dictionary, more than 90% of the entries could be considered as simple entries (i.e. the entries can be easily re-coded using generic terms. This fact suggests that in order to increase the quality of the dictionaries, the problem implies not so much a need for additional power but the need for direct and intuitive access to these frequent patterns. Therefore, we have introduced an intuitive coding tool described in the External Component section.

Monolingual resources

The principle of monolingual resources is to give access to a localized reference that will lead to reduce redundancy, and to give a basis for coding new entries. The monolingual dictionary contains simple words as well as compound words. It is maintained independently of the multilingual dictionaries.

The reduction of redundancy comes from the reference mechanism used in both external resources and monolingual dictionaries (as a cross-reference). For example, the French dictionary contains the following

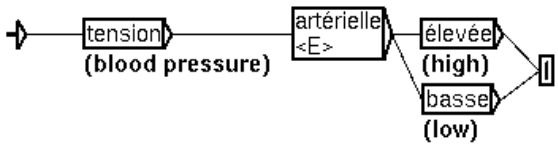

Figure 5b. Factorization and enrichment of linear entries.

entries:

<entry>*pilote<cat>noun<syncode>31<sem>human,occupation*

<entry>pilote^*[1] de (course automobile)[2]<cat>noun<\entry>*

"*pilote*" is fully described in the first entry, and the second entry only describes any additional information and identification information for its structure. The whole entry would be something like:

<entry>pilote.N31[1]^ de (course.N21*[2]*

automobile[2].A31).N*

<cat>noun<sem>+human+profes

where *[i] represents the internal agreement mechanism, the ^ points out the headword, and the figures the flexional code. In fact, during the construction of the runtime dictionary, the second structure is automatically generated from the first one. This minimality in information, requires self-coherence tests. Indeed, if we want to add a new "*pilote*" entry to our dictionary (for example, in the sense of "*computer driver*"), we will need to update the first representation and give an additional *identification* tag:

<entry>pilote^*[1+human] de (course automobile)[2]*

<cat>noun<\entry>

This overhead in the maintenance of monolingual dictionaries guarantees that we can maintain coherent and "light" resources.

This work on the monolingual dictionary reflects the general idea in the organization of linguistic resources. By structuring the information, we avoid redundancy, and have better control over the description. More generally, the different ways of reducing the entropy in the description are: a) the use of graphs for factorizing similar entries, and b) the use of implicit transfer rules. We give here examples of both points:

Factorization of graphs. The entries "haute tension=high tension", and "faible tension=low tension" are, in a traditional dictionary described on the same level as "tension artérielle=blood pressure". Gathering these entries in the two graphs (Figure 5a and 5b), technically, is only a matter of location of the information. From a linguistic point of view, we have added information gathering the three modifiers of the noun "tension" and reducing the size of the description (in term of number of entries).

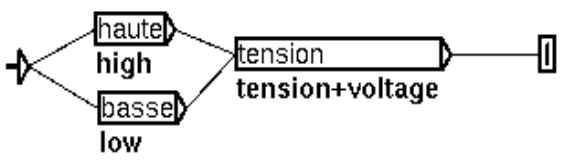

Figure 5a. Simple factorization of two linear entries.

Implicit transfer. Let us consider again the "*haute tension=high tension*" entry. In terms of information, there is very little information in the equivalence, because we know that "*high*" is one of the potential translations for "*haut*", and "*tension*" is one of the potential translations for "*tension*". In that case, there is some redundancy in the whole entry. The implicit transfer module could thus retrieve this link using source and target lists of compounds and an alignment dictionary. In fact, we have not implemented implicit transfer for this kind of entries, but this example points out the implicit transfer mechanism and the potential entropy reduction by organizing the database with such a mechanism.

Transfer rules formalism

The transfer rules (Table 2) are represented in a very simple language, in which the formal grammar is generated from the syntactic feature definition file. This formalism allows us to have several levels of representation, from the simplest non-conditional transfer, to transfer with context constraints, and conditional actions. The transfer rules give access to basic features defined in the syntactic features definition file, and to frequent additional conditions/actions corresponding to complex instructions. For example in Hungarian, setting the case of the verb's object will be a basic coding feature.

Exchange format and readability

In order to allow exchange of transfer dictionaries, we are working on a projection filter of internal resources to external exchange formalism (for instance OLIF2 format). This conversion is not merely a conversion of format. To be able to exchange non-trivial transfer entries, we need to keep most of the linguistic information without knowing how this information will be used. So far, we can extract a readable XML-like format from the linguistic resources losing nevertheless most of the linguistic organization: implicit transfer is made explicit; graph entries are de-factorized; and cross-references to monolingual dictionaries are substituted. This readable format can also be used at runtime to visualize the linguistic resources used for the translation of a specific text.

Reciprocally, external glossaries can be imported using mainly the coding tool, potentially modifying for the coding process, the list of "linguistic clues" to map Systran internal representation.

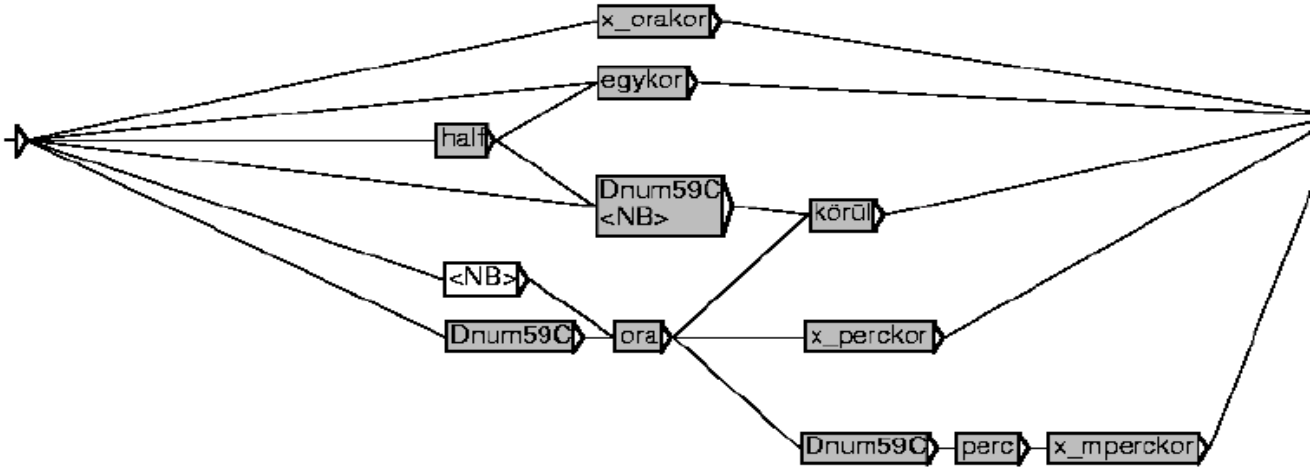

Figure 6. Local grammar described some temporal adverbial expression in Hungarian (e.g.,; fél nyolc körül, or hat óra húsz perc tíz másodpercig). This is the source language description for the implicit transfer of dates.

External Components

The modularity required in the design of the new system, combined with the organization of the linguistic resources to avoid redundancy and to increase accessibility of resources, extend the translation engine itself, to several by-products. We describe here two of them.

Syntactic Tagger

The interface between the analysis and the transfer module is restricted to the internal structure corresponding to the analysis of each sentence. This structure can be output to show the state of the internal analysis (Figure 3). Reciprocally, we have introduced the possibility of running the transfer on a textual file with the same format describing the sentence. The output and the parsing functions are generated automatically from the syntactic field description file. We have thus access to a functional external syntactic tagger, and we could even consider to apply the transfer module on a file generated by an external analyzer.

Intuitive Coding Tool

To open the system to any external customization, we have developed an intuitive coding tool named *AllCoding*. This tool utilizes:

- The large monolingual dictionaries
- A derived statistical guesser computing for any word (found or not found), a list of potential category and morphological codes
- A statistical context-free description of the compound structures. For example, to analyze "*moyenne tension électrique*" as a A(NA) noun , the system will use the rules:

  noun adjective$\rightarrow_{0.99}$ *noun*
  noun noun$\rightarrow_{0.3}$ *noun*
  adjective(+left) noun$\rightarrow_{0.99}$ *noun*
  adjective(-left) noun$\rightarrow_{0.6}$ *noun*

- A set of intuitive clues and conventions. ("`to save`" refers to a verb, that "`believe (in)`" indicates that "`believe`" could expect an "`in`" complement)

`"penguin"->"pinguin".`	Simplest transfer rule. Neither source, nor additional information is given (it is retrieved from source and target monolingual dictionaries)
`<strike.V>->"üt";`	Minimal source information is provided to avoid ambiguity.
`<end.N>->"vég";`	
`"bright"->"fényes".`	Condition on "human" features differentiates the two rules. The "modifies" reference is defined as a macro-features.

`"bright"->"okos";`
`$1~<syntax type="A">+modifies=$2</syntax>;`
`$2~<syntax>+human</syntax>.`	

`"allergic"->"allergiás";`
`$1~<syntax type="A">+modified_by_prep1=$2</syntax>;`
`$2="to";`
`$2~<syntax type="PREP">+direct_object=$3</syntax>;`
`$3~<syntax type="N"></syntax>;`
`$3-><syntax type="N">+case=sub</syntax>;`
`$2->$$$.`
	More complex rule to set a transfer action setting case.

`"there"->"oda";`
`$1~<syntax type="ADV">+predicate_of_clause=$2 </syntax>;`
`$2~<syntax type="V">+illative</syntax>.`	

Table 2 - Example of some EN-HU transfer rules.

- A list of the more frequent transfer patterns in the main transfer dictionaries
- An alignment module trying to align source and target information in a bilingual entry.

This tool converts an input file like the following to a real transfer dictionary.

```
an alligator peach=un avocat
bright (human)=brilliant
to save (a file)=sauvegarder
to believe (in)=croire (en)
a sample=des échantillons
an advocate=un avocat
expert fighter pilot=[pilote de combat] confirmé
```

Conclusion

The design and implementation of a new generation MT engine, has mainly led us to work on the organization and the accessibility of linguistic resources. In this short overview, we have presented the technological choices serving that purpose. Tested on a few real sized systems, this approach has so far proven to facilitate the coding of main dictionaries and keep them more intuitive. The same transformation is in progress on several existing systems. The major expectation from these systems is their easy maintainability and their external enhancement.

Moreover, the focus on monolingual data and the concept of implicit transfer are an axis for extensive development of cross-language systems

Finally, the obtained modularity is a first step towards more generic natural language processing systems in which linguistic resources are foremost critical in an incremental frame, and the fundamental condition for high quality translation systems.

References

Senellart J. Plitt, M., Bailly, C. and Cardoso, F. (2001) In Proceedings of the MT Summit VIII. Resource Alignment and Implicit Transfer. September 18-22, 2001. Santiago de Compostela, Spain.

Acknowledgements

We would like to thank our colleagues Jin Yang and Elke Lange at Systran La Jolla for editing the paper.

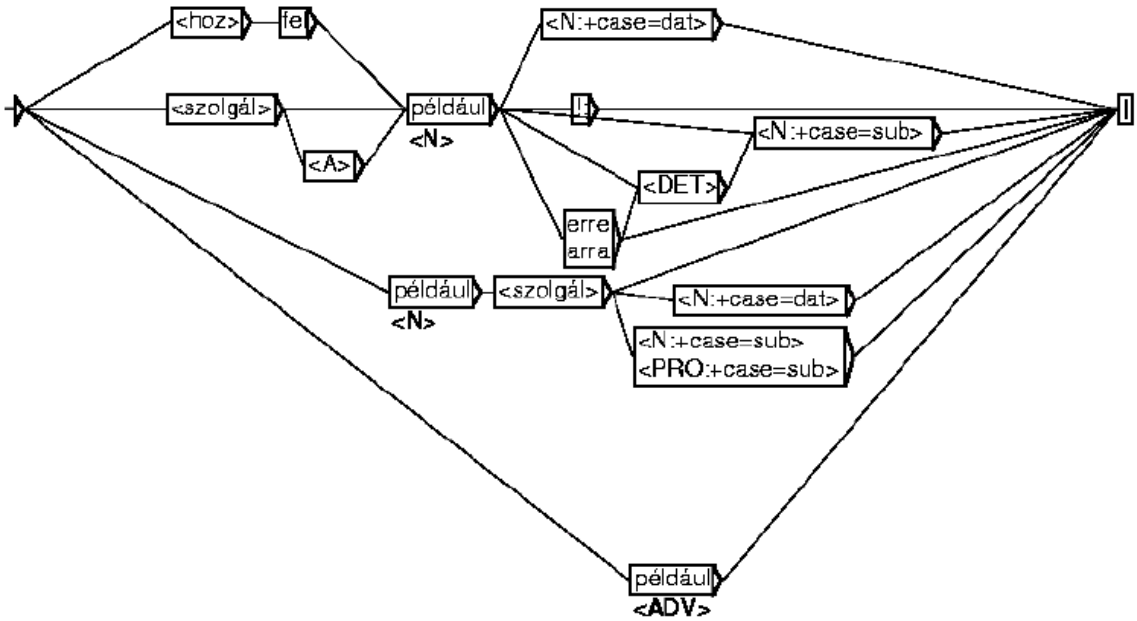

Figure 7 - Graph of syntactic ambiguity of "például" in Hungarian : each path represent a rule, the output of the automaton yields the disambiguisation choices.