HUNGRAM VS. ENGGRAM IN PARGRAM
On the Comparison of Hungarian and English
in an International Computational Linguistics Project

TIBOR LACZKÓ
GYÖRGY RÁKOSI
ÁGOSTON TÓTH

Institute of English and American Studies
University of Debrecen
{laczko, rakosig, tagoston}@delfin.unideb.hu

-- DRAFT VERSION OF 15 APRIL 2009 --

1. Introduction

The goal of this paper is to offer an overview of a recently launched computational linguistic project at the Department of English Linguistics at the University of Debrecen. The project aims at developing a syntactic and morphological analysis of the Hungarian language within the basic linguistic framework of Lexical-Functional Grammar (LFG), with special emphasis laid on a comparative English vs. Hungarian aspect. This initiative is a contribution to a large-scale LFG-based international computational linguistic collaboration: Parallel Grammars (PargGram), which is coordinated by Palo Alto Research Center (PARC) in California. In this paper, we discuss the most crucial features of LFG and the computational linguistic apparatus developed by ParGram (in a framework called XLE), and we illustrate them by the help of a sample analysis of some fundamental properties of English and Hungarian noun phrases.

The structure of the paper is as follows. In Section 2, we briefly outline and illustrate the design principles of the LFG framework. In Section 3, we provide an overview of the history of the ParGram project, and introduce the key features of the XLE grammar writing apparatus. In Section 4, we discuss the history and the main objectives of the HungGram project, and demonstrate the capabilities of our system by an analysis of a special type of Hungarian noun phrases and their English counterparts. In Section 5, we round up with an outlook on problems and challenges in the domain of natural language processing that we aim to address in our project. Section 6 contains our concluding remarks.

2. Lexical-Functional Grammar

LFG, as an alternative, non-transformational generative linguistic theory was developed in the late 1970s by Joan Bresnan and Ronald Kaplan. One of the principal
Aims of the two co-founders of LFG was to create a linguistic framework suitable for massive computational applications. We here restrict ourselves to a brief introduction that will serve our current purposes, and refer the reader to Bresnan (1982) for a comprehensive summary of the initial state of research in LFG, as well as to Bresnan (2001) and Dalrymple (2001) for more recent and updated overviews.

The basic design principles of LFG are as follows.

• **Strong lexicalism.** Every morphological variant of a stem (including derivational and inflectional variants alike) is stored as a separate, richly annotated entry in the lexicon. The lexicon is regarded as an active component of grammar within which operations changing the featural makeup of the entries may apply. This view entails that grammatical behaviour is assumed to be lexically determined to a large extent.

• **Modularism.** Linguistically relevant information is represented in distinct modules, governed by module-specific rule systems. The key syntactic modules assumed are *c*(onstituent)-*s*tructure, which represents information about possibly language-specific surface word order and constituency patterns, and *f*unctional-*s*tructure, which encodes featural information concerning the grammatical relations and predicate-argument structures underlying the surface strings of expressions. The number of modules is not pre-determined by the theory: a phonological and a semantic module is generally assumed, and, more recently, arguments have been made for a distinct morphological module and a module representing discourse information.

• **Parallel architecture.** As opposed to mainstream Chomskyan generative grammar, LFG is not derivational: any correspondence is legitimate among the modules if it turns out to be useful in the analysis. For example, *c*-structure and *f*-structure can communicate with each other in either direction.

• **Mathematical formalism.** Computational application being a key target for LFG, it is essential that the rule system should be rendered in a mathematical formalism or at least it should be written in a form that is directly renderable in a mathematical formalism.

Let us now illustrate how syntactic information is represented in LFG by a comparative analysis of a Rumanian and an English noun phrase. As is well-known, Rumanian, together with other Romance languages, systematically allows for the post-head occurrence of adjectival modifiers in noun phrases (1a). In English, however, this is a very restricted option (cf. *the people present*), and the order noun+adjective is certainly not grammatical in the case at hand (1b).

(1) a. \[
\begin{array}{c}
\text{NP} \\
\downarrow \\
N' \\
\downarrow \\
\text{NumP} \\
două \\
two \\
\text{N} \\
cărți \\
books \\
\text{AP} \\
frumoase \\
nice \\
\end{array}
\]

(1) contains simplified *c*-structure representations that will serve our current purposes.
The annotation ↑=↓ means that the daughter node has the same grammatical properties as the mother node. These two c-structures capture in an obvious way the difference between the two languages in terms of the linear ordering of the noun head and the adjectival modifier.

The respective f-structures, by contrast, are going to be much more similar for the reason that it is here that the underlying functional and featural information is represented. At this level, the only important difference between the two languages is that gender and case are grammatically active features across the whole nominal paradigm in Rumanian, and adjectives agree with the noun head with respect to number, case and gender. Therefore the f-structure of the Rumanian example (2a) includes more features, but there is no other essential difference between (2a) and (2b) beyond that. We use English as a semantic metalanguage, and omit details that are not directly relevant.

(2) a. două cărți frumoase

\[
\begin{align*}
\text{PRED} & : 
\begin{cases}
\text{PRED} & : \text{book} \\
\text{NUM} & : \text{pl} \\
\text{CASE} & : \text{nom} \\
\text{GENDER} & : \text{fem}
\end{cases} \\
\text{ADJUNCT} & : 
\begin{cases}
\text{PRED} & : \text{nice} \\
\text{NUM} & : \text{pl} \\
\text{CASE} & : \text{nom} \\
\text{GENDER} & : \text{fem}
\end{cases}
\end{align*}
\]

b. two nice books

\[
\begin{align*}
\text{PRED} & : 
\begin{cases}
\text{PRED} & : \text{book} \\
\text{NUM} & : \text{pl}
\end{cases} \\
\text{ADJUNCT} & : 
\begin{cases}
\text{PRED} & : \text{two} \\
\text{PRED} & : \text{nice}
\end{cases}
\end{align*}
\]

Adjuncts are generally introduced in LFG analyses as an unordered set. This rests on the underlying assumption that their interpretation does not depend on linear order but is simply a matter of set intersection. In general, no linear order is defined over the constituents at the level of f-structure, and it is the c-structure (cf. (1) above) where this sort of information is stored.

As should be evident from this brief presentation, f-structure serves in LFG as a representational level where significant cross-linguistic generalizations can be captured. Since it also serves as a storage place for the grammatically relevant features associated with a particular natural language expression, it is not unexpected that LFG-based computational applications center around generating and parsing f-structure representations.
3. The ParGram initiative
3.1. A brief historical overview

The ParGram project was launched by PARC in 1996 and has been organized and coordinated by PARC ever since. It is an LFG-based computational linguistic initiative, which capitalizes on the theory’s flexible general linguistic and computationally implementable architecture. The main objective is to analyze more and more languages on a maximally uniform platform, focusing on both parsing and generating tasks.

It is a truly international project which includes researchers from different parts of the world. The list of languages that have been incorporated into the ParGram grammar development efforts includes English, German, French, Norwegian, Japanese, Chinese, Urdu, Malagasy, Arabic, Vietnamese, Spanish, Welsh, Indonesian, Turkish, Georgian. As a result of our project, Hungarian has now also been added to this list.

3.2. The software environment

In our work, we use a licensed copy of the Xerox Linguistic Environment (XLE), which includes a state-of-the-art LFG parser and adds functions that make it a very efficient linguistic workbench for creating, testing and editing grammars.

XLE is able to create and work with two basic representations: the c- and f-structure (see Section 2. for a discussion). In addition to the parser function, the environment has a generator mode that can generate strings from f-structures.

As far as its morphological capabilities are concerned, XLE supports finite-state transducers that carry out lemmatization and morphological analysis. Of course, ambiguous morphological analyses do appear when available.

In Natural Language Processing (NLP), parsing is often “shallow”: it is statistics-oriented rather than rule-oriented, it has problems with long-term dependency and precision is secondary to other factors including speed and high recall. XLE carries out deep-parsing, which is superior to many existing solutions. Since parsing time increases exponentially with every new input item in XLE, the parser offers solutions for controlling running time, too.

For a detailed description of the computational linguistic apparatus developed within the framework of the PargGram project, we refer the reader to Butt et al. (1999).

4. On the HunGram project
4.1. History and goals

Debrecen’s participation in the international ParGram collaboration began when Tibor Laczkó received a Fulbright research grant to the Linguistics Department of Stanford University for the academic year 2005/2006. While at Stanford, he was invited to PARC to launch a Hungarian subproject, so he carried out research at two host institutions simultaneously. At PARC he had two major objectives. On the one hand, he set out to acquire familiarity with the XLE formalism, and, on the other hand, he started working on the XLE implementation of the results of his research on the morpho-
syntax of Hungarian noun phrases in an LFG framework. This previous research included the investigation of the structure of noun phrases, the nature of possessive constructions, nominalization, the properties of pre- and postmodification with particular attention to participial constituents. At PARC the following main aspects of the implementation of the findings were completed: basic noun phrase structure, the treatment of possessive constructions, and, as a new area, the analysis of various elliptical constructions.

In 2007 Tibor Laczkó and two colleagues of his, György Rákosi and Ágoston Tóth, formed an LFG Research Group (LFGRG) at the Department of English Linguistics at the University of Debrecen. PARC provided LFGRG with the necessary XLE software licence.

LFGRG successfully applied for a research grant to OTKA (Hungarian Scientific Research Fund) for a four-year period: from July 2008 to June 2012 (grant number: K 72983). The two fundamental goals of this research project, called HunGram, are as follows:

- developing a comprehensive LFG grammar of the Hungarian language in the areas of morphology, syntax and lexical representation by also addressing relevant semantic issues;
- implementing it in HunGram and ParGram in continuous collaboration with the international ParGram community.

It is our future plan to launch an English vs. Hungarian comparative research project on the ParGram platform after the successful completion of the HunGram project. We also intend to incorporate the results of the comparative research in various course materials at the English Linguistics Department.

In the next subsection, we demonstrate along what lines we expect to achieve our short and long term objectives by discussing the analysis of a special Hungarian noun phrase type and its English counterpart within the ParGram framework.

4.2. Demonstration: elliptical Hungarian noun phrases and their English counterparts

In this section we discuss how we envisage carrying out our HunGram research project as appropriately embedded in the collaborative context of ParGram. We also show how this can be developed into a comparative research project, and, finally, how the findings can be utilized to enrich our course materials at the English Linguistics Department. We illustrate all this through a brief example of the analysis of a particular Hungarian elliptical noun phrase construction and its English counterpart. For detailed discussion, see Laczkó (2007) and Laczkó (this volume).

Compare the following examples.

(3) a. (az) öt nagy zöld golyó-t
    the five large green ball-ACC
    ‘the five large green balls’
b. (az) öt nagy zöld-et
   the five large green-ACC
   ‘the five large green ones’

(3b) exemplifies the Hungarian elliptical noun phrase type in question. In these constructions the noun head is missing from the expression entirely, cf. (3a) and (3b). The rightmost modifier in the “remainder” of the expression (whether an adjective or a numeral) functions formally as the head. The formal head status of this element is manifested by the fact that all the nominal suffixes are attached to the head of this final constituent: plural markers, possessive agreement suffixes and case endings. Note that in non-elliptical noun phrases, numerals and adjectives take none of these suffixes, cf. (3a). Also note that in the English counterpart of this Hungarian construction there is a pronominal element occupying the noun head position: one(s), cf. the English translation in (3b).

Laczkó (2007) proposes an LFG analysis of this Hungarian construction type and claims that it has been successfully implemented in HunGram. Consider Laczkó’s (2007) phrase structure rules in (4) and the functionally annotated c-structure of (3b) in (5) (for simplicity of this exposition, we leave out the determiner and concentrate on the N’ portion of noun phrase structure).

(4a) is the (simplified) phrase structure rule for the N’ portion of the (non-elliptical) noun phrase. The \( \downarrow \in (\uparrow \text{ADJUNCT}) \) annotation means that the constituent it is
associated with is a member of the adjunct set of the noun head (associated with LFG’s functional head annotation: \( \uparrow \)). The \( \neg(\uparrow \text{CASE}) \) and \( \neg(\downarrow \text{NUM}) \) annotations express constraints to the effect that in non-elliptical constructions none of the adjuncts can be marked for either case or number. (4b) is the rule for the N’ portion of the elliptical noun phrase. There is no noun head in it; therefore, the structure is exocentric (which is readily admitted by the phrase structure principles of LFG). There can be several adjuncts (adjectival phrases and also a number phrase) in the construction as well and none of them can be marked for case and number except for the final one, which, by contrast, must be marked for both features. The \( (\uparrow \text{CASE})=(\downarrow \text{CASE}) \) and \( (\uparrow \text{NUM})=(\downarrow \text{NUM}) \) annotations encode that the case and number features carried by the final adjunct are the relevant features of the entire noun phrase. The key annotation associated with the final adjunct is \( (\uparrow \text{PRED})=\text{‘pro’} \). It introduces an LFG-style pronominal element which does not have a phonetic form and does not occupy a syntactic position in constituent structure. However, as a result of this annotation, this pronominal element does appear in the f-structure of our elliptical noun phrase, cf. (6).

\[
\begin{array}{c}
\text{PRED} \quad \text{‘pro’} \\
\quad \begin{cases}
\text{PRED} \quad \text{‘five’} \\
\quad \begin{cases}
\text{PRED} \quad \text{‘big’} \\
\quad \begin{cases}
\text{PRED} \quad \text{‘green’} \\
\text{NUM} \quad \text{sg} \\
\text{CASE} \quad \text{acc}
\end{cases}
\end{cases}
\end{cases}
\end{array}
\]

As has been pointed out, this analysis has also been implemented in HunGram. Now consider the c-structure and f-structure HunGram provides for (3b). (The numbers associated with the nodes in the c-structure are randomly generated and they are used to identify these nodes uniquely. In the f-structure they are used to indicate which parts of the f-structure correspond to which c-structure nodes.)
In ParGram/HunGram it is also possible to extend the c-structure representation to the sublexical level, where it can be indicated what information is associated with individual morphemes in the form of tags:
Now consider how the English Grammar of ParGram (EngGram) analyzes the English counterpart of this Hungarian elliptical construction, cf. the English translation in (3b).

Figure 3

Figure 4
There are significant differences and similarities between the Hungarian and the English constructions and their ParGram analyses. Obviously, the major difference between the constructions themselves is that the Hungarian one is elliptical, while the English one has an overt pronominal head. Another typical contrast is that the Hungarian construction is marked for accusative case as usual while the English counterpart is not. As regards the analyses, EngGram and HunGram postulate partially different c-structures for noun phrases (EngGram: NP, HunGram: DP) on the basis of the (partially) different properties of noun phrases in the two languages. This is fully legitimate both in LFG and on the ParGram platform. In the f-structure of EngGram there are much more features than in that of HunGram. This is fundamentally due to the fact that EngGram is much larger, much more fully developed. As HunGram is being developed, more and more features, motivated and justified by the relevant Hungarian facts, will be introduced. As has already been mentioned, the most crucial aspect of the HunGram analysis is that in c-structure it employs an exocentric phrase (one without a structural head) and uses a special functional annotation to introduce a pronominal head in f-structure. By contrast, the pronominal head in the English counterpart is overt; thus, EngGram provides a regular NP analysis: the pronoun has its own lexical form, and it appears both in c-structure and in f-structure.

While LFGRG’s Hungarian Grammar is being developed in an LFG framework and it is being implemented in HunGram we intend to compare Hungarian and English systematically, observing both similarities and dissimilarities. In the context of our current example: case-marking, headed and headless noun phrases, overt and covert pronominal elements, endocentric and exocentric constructions are of outstanding significance. We consider ParGram an ideal platform for carrying out this comprehensive comparative investigation.

Figure 5
It is also one of our important future objectives to incorporate the linguistic tools, the results and the findings of our research project in our course materials at the English Linguistics Department at various levels. For instance, in our BA programme there is a course entitled *English vs. Hungarian – Patterns and Contrasts*. The main findings of the comparative investigation can be included in it in the form of fundamental empirical generalizations: e.g. Hungarian elliptical constructions and their English counterparts. In our MA programme LFG and ParGram/EngGram can be taught at an introductory, basic level, while in our PhD programme they can be taught at an advanced level.

5. **Challenging natural language phenomena and system design**

There is a significant set of natural language phenomena that continues to cause difficulties for algorithm formulation attempts. They include various forms of ambiguity, the productivity of word formation, the recognition of named entities, anaphora resolution, scope problems and ellipsis. We continuously collect data about these phenomena so that we can formulate strategies and prepare solutions to address them (see subsection 4.2. on elliptical constructions). Of course, only the most complex and heavily meaning-dependent NLP tasks face all of these problems, but Machine Translation, which is treated as a possible major area of application of the ParGram effort, is one of them. In this initial phase of our investigation, we focus on lexical and structural ambiguity.

5.1. **Lexical ambiguity**

Cruse argues that ambiguous words have multiple senses that exhibit the phenomenon that he calls *antagonism*: you cannot focus your attention on two or more readings at the same time (Cruse 2000:108). When you utter or hear the sentence in (7), for instance, it is either the “financial institution” or the “riverbank” sense that becomes active for the word *bank*.

\[(7) \text{We finally reached the bank.}\]

In some cases, the disambiguation can be carried out within the scope of the given sentence; in other cases, you must resort to sentence-external context.

Even those words that do not show antagonism may have multiple discrete readings that can be detected via various tests offered by Cruse (2000). Other discussions of lexical ambiguity make references to polysemy and homonymy. The polysemy-homonymy distinction is not as unproblematic as it seems for the first sight, however, because grasping the notion of relatedness of meanings is challenging, especially when etymology is excluded for lack of psycholinguistic relevance. NLP applications usually resort to the “maximize homonymy” approach (cf. Lyons 1977) leaving out polysemy from the description. WordNet (Miller et al. 1990) is a full-scale lexical database that excludes polysemy from the description but implements a host of lexical and semantic
relations. WordNet is also instructive in a different way: even tiny sense variations are kept distinct, and the database is probably as fine-grained as possible; therefore, the disastrous effect of lexical ambiguity on real-world applications is made obvious in WordNet-based projects. Mihalcea and Moldovan (2001) point out that it is not uncommon that WN “word senses are so close together that a distinction is hard to be made even for humans” (Mihalcea and Moldovan 2001:454). Reaching the level of sense-granularity present in WordNet is not a design goal in our project. Instead, along the lines drawn by Palmer (1998:7), we would like to enumerate only those sense distinctions that can be identified by differences in argument structure and/or selectional restrictions. This is still a formidable task when you aim to build a non-toy lexicon of a natural language, and sense enumeration in general has its own pitfalls.

5.2. Structural ambiguity

In many cases, multiple interpretations come from the different possible relationships among the constituents of a sentence.

(8) *I saw the girl with the telescope.*

The sentence in (8) has at least two interpretations:

(a) the speaker had the telescope (see Figure 6 for its XLE analysis)
(b) the girl had the telescope (see Figure 7)

Interpretation (a) may seem more natural because of the ‘associational’ connection (cf. Ide and Veronis 1993:263) between the words *see* and *telescope* (or between the phrases *see a girl* and *with the telescope*, for that matter). Nevertheless, both interpretations are possible, and the parser should return both of them. Notice that we have ignored the effect of the lexical ambiguity caused by an alternative reading of *saw* (which involves ‘cutting with a saw’).
5.3. Why bother?

In the context of a grammar-writing project that has the primary goal of writing and implementing a comprehensive grammar of the Hungarian language, our effort to tackle the above problems may seem irrelevant. But it is our understanding that
currently accepted system design, which is based on a highly modular view on language and on linguistic tasks and relies on interfaces that may not be possible to implement, has inherent limitations. Word Sense Disambiguation, for instance, is usually considered to be a necessary and well-encapsulated linguistic task. When put to the hard test of WordNet-based sense identification, however, it turns out that the best strategy is to decrease the number of senses (i.e. to eliminate senses rather than disambiguate them, see, for instance, Seagull 2000). At the same time, Dolan, Vanderwende and Richardson (2000) seem to manage without WSD: they use “raw” (ambiguous) input in the MindNet project. They argue that “the traditional view of WSD as involving the assignment of one or more discrete senses to each word in the input string” cannot be used to implement broad-coverage NLP systems (Dolan, Vanderwende and Richardson 2000:5), and “like humans, machines cannot be expected to perform reliably on a task that is incorrectly formulated” (ibid.). They point out that not even humans seem to excel in carrying out word-sense disambiguation tasks; their position is supported by the data in Fellbaum, Grabowski and Landes (1998). Therefore, correct formulation of linguistic tasks and careful system design should be of primary concern. Our plans include participating in the international effort of developing semantic structures based on the ParGram syntactic representations (ParSy), and we also try to devise new strategies, relying on novel, connectionist methods (cf. Tóth 2008) in handling ambiguity using artificial neural network representations.

6. Concluding remarks

In this paper we offered an overview of the traits of LFG and the most significant aspects of ParGram, an international collaborative generative and computational linguistic project. This provided the relevant context for the discussion of launching HunGram, a new research project embedded in ParGram. In the characterization of HunGram, including the short and long term objectives, we used the example and analyses of a special Hungarian elliptical construction and its English counterpart. Finally, we briefly discussed what general problems and challenges we expect to encounter and (hopefully successfully) tackle in our research.

References


