L. Vardanyan,

EXPEDITION TO ARMENIAN FORESTS: ANY UNIQUE TREE SPECIES GROWING THERE?
Towards Treebanking of Modern Eastern Armenian

ABSTRACT

This paper is aimed at sketching out an outline about the research work in progress in frames of PhD 2006-2008 program "Linguistica generale, storica, applicata, computazionale e delle lingue moderne", carried out at the Department of Linguistics "Tristano Bolelli", University of Pisa.

We propose and support the practical and theoretical background for developing a sample of a syntactically annotated corpus (otherwise called treebank) for Modern Eastern Armenian. The acknowledged lack and need for descriptive studies on Armenian language resources stands for good reason to set and accomplish such task. Basing on the conviction that only the annotated linguistic data may gain their real value for linguistic observations and research, we will present different annotation levels and formats of corpora. We target the dependency structures description in the form of syntactic functions (manual) annotation of the naturally occurring sentences in a corpus of written Eastern Armenian. The project's current focus of study is building an annotation scheme alongside setting forth a syntactic tagset to be further applied on a morphologically analyzed corpus on surface syntax level.

1. CORPORA ANNOTATION

1.1. Introductory remarks

Study of syntax is one of the milestones of general linguistic science. Throughout many decades traditional grammarians of various languages have built theories of syntax – most of them following prescriptive frameworks of grammar. The word syntax and syntactic analysis may bring to some of us bad memories on an exercise-book corrected by school teacher with red-pen underlines on text we exposed so convinced in its correctness and naturalness. In fact, our school grammar books are indeed valuable pool of grammatical rules governing the structures of a language, yet they inevitably lack the coverage of all possibly occurring grammatical structures in a natural language adopted by its speakers, being the source of any language's syntax in general. Such knowledge can be gained only through the descriptive framework of studies that compile data-driven, empirical recourses and rules from studied material.

Current progress in information technologies and availability of growing amount of digitally processed and machine-readable textual data of a particular language is providing support for holding empirical research. Corpora for variety of languages serve best source for a particular language sample, yet they are of limited use. They need to be not only larger in order to be representative, but also carefully encoded and enriched with linguistic descriptions - annotations, since the latter make the corpora valuable linguistic resources for searches and computations.

The annotation process of corpora is layered according to levels of aimed linguistic description. The following layers of annotation are generally differentiated in annotated corpora:

- Morphological annotation (lemmatization, inflection, derivation, compounding)
- Morpho-syntactic annotation (POS tagging, contextual morphological disambiguation)
- Syntactic annotation (parsing, constituency or functional dependency structure representation)
- Semantic annotation (word-sense disambiguation, anaphora, coreference resolution, information structure)
- Discourse annotation (dialog turns, speech acts)

To keep in frames of the current project objectives, we will limit ourselves at describing the syntactic level annotation of linguistic data with further projection onto the Eastern Armenian morphologically analyzed textual data.

1.2. Syntactic annotation of corpora

Syntactic annotation is commonly interpreted through the parsing – the process of analyzing an input sentential segment in order to determine its grammatical structure with respect to a given formal grammar. It presents a sentence as a data structure, usually ***a tree***, which captures the implied hierarchy of its structure according to either constituency or grammatical relations.

There are developed various automatic parser programs for carrying out such task on major languages. Yet, natural language sentences are not easily parsed by programs, as there is substantial ambiguity in the structure of a sentence composed by human. Despite the rise and efforts of computational technologies to build such linguistic resources automatically, there are actually no projects of syntactic annotation of corpora, implemented in fully automated fashion. Human annotator’s intervention is indispensable in developing a syntactically annotated corpus – a ***treebank***, from fully manual annotation up to post-editing of the parser’s output.

1.3. General Guidelines of Treebanking

To obtain the objective of building a treebank, certain guidelines should be followed. The quality and usability depends much on the corpus size, domain selection, as well as the motivation for which such databank is being created. These may vary from applying and testing of a given linguistic theory, up to developing a new resource, independent of any particular linguistic theory. Some projects also have a precise application goal, like inducing some linguistic resource (lexicon, grammar) for training and evaluating parsers. Such efficient tools have been developed and applied to a variety of studied languages. Nevertheless, these tools cannot either be made independent of a particular language to be treated. In other words, English languages tools, for example, may not work for Armenian once one will take a task of automatic parsing Armenian textual data.

Another and perhaps major issue is the choice of the ***annotation format*** – the representation fitting to a formalism of syntactic structures. A detailed ***annotation scheme*** should be designed where the descriptive rules of the language’s syntactic structures are drawn and the consistency of the application of each rule is strongly followed up (that is, the same constructions should receive the same analysis every time they occur). A ***tagset*** – the list of symbols used for syntactic categories represented in the corpus, should be build for applying throughout the annotation process and depending on the annotation type. This list may vary in size from a dozen up to millions depending on a specific language.

Currently the choice of annotation type for building treebanks is mainly distinguished between two trends: the linguistic resource is annotated according to either constituent or functional structure. The first efforts in building treebanks were done in the phrase structure – ***constituency-based*** frameworks, represented as tree-shaped constructions.

In the recent years there has been wide interest towards the grammatical function annotation of treebanks. The dependency grammar formalism appears fitting for this purpose and many ***dependency-based*** treebanks have been constructed. In addition, grammatical function annotation has been added to some constituent-based treebanks. A hybrid scheme is also applied by some projects, where the constituents are minimal phrased, called chunks, linked by dependency relations.

The major issue playing a determinative role for annotation format choice is mostly depending on theory-specific considerations of the project, and mainly the language-specific features to be represented in tree structures.

Briefly, the constituency structure annotation is rather suitable for (relatively) fixed word order languages, such as English. This annotation format describes the phrase structure of the entire clauses. Here the context-free grammar formalisms, otherwise called also phrase-structure grammars (e.g. HPSG, LFG) support and play pivotal role in the annotation scheme. A prominent example of a treebank of English built in this fashion is the Penn Treebank (Marcus et al. 1993) of 3 mln tokens skeletally parsed - syntactically bracketed text. Another project following this trend in its start phase is NEGRA treebank for German newspaper textual material (Brants et al. 1999).

In the constituency-based schemes adopted by these projects, the sequence of tokens (word order) in a phrase structure is followed. The occurring non-local dependencies, discontinuous structures and elliptical material are represented with trace-fillers (e.g. adding empty-nodes). In a language where such phenomena are of quite frequent and moreover specific nature, the phrase structure models do not perform efficiently in their constituency annotation. While such representation is quite compatible for a language like English, the annotation of a language with a (relatively) free word order under such format is not always optimal. Such languages are apt to perform a range of features causing problems, such as discontinuous constituents (e.g. topicalization, split phrases), ellipsis, etc. The structural handling of freer word order assumes well-formed constraints on structures involving many trace-fillers, which makes the rules and tree structure overloaded and far transparent.

Here the grammar frameworks based on functional representations of syntactic structures come to support the annotation. The grammar formalism representing dependency relations is the theoretical backbone of such annotation scheme. The description of dependency relations between the nodes (words) of a sentence tree and attachment of the functional labels make a rather powerful formalism to deal with annotation of free word order languages – a reasonable argument that is taken into consideration for further annotation choice of the Armenian language.

Perhaps the largest project representing this framework of treebanking is being implemented for Czech language – PDT (Prague Dependency Treebank) (Hajic 1998). This is a representative treebank that in its more than 10 years of development produced over 1.8 mln tokens (around 110 K sentences) of annotated material from newspaper, science, economics, literary and other domains. Worth to mention, that many aspects and experience of PDT project are followed up and inherited in taking the current project of Armenian treebank building. Other projects that have adopted dependency annotation schemata are implemented for Italian – ISST (Italian Syntactic-Semantic Treebank) (though the entire treebank exploits parallel layers of both constituent and dependency structure annotation), TUT (Turin University Treebank); for Dutch – the Alpino Treebank; the Dependency Treebank for Russian, etc.

2. FIRST STEPS TO TREEBANKING OF EASTERN ARMENIAN

Bearing in mind the guidelines described above, our project targets treebank building for Standard Eastern Armenian (henceforward referred to as Armenian in the frames of the project). A morphologically analyzed subcorpus from EANC (Eastern Armenian National Corpus) from press and fiction genres is targeted to be manually annotated. To take decisions upon choosing annotation format we deduce from the language specific requirements. From the typological point of view, Armenian is a pro-drop inflectional language showing agglutinative, synthetic and analytical features in word and phrase construction. The word order variation ranges from relatively fixed (mainly in NPs) up to very flexible inside a clause. The classical assumption about its basic word order is referred as SOV in neutral, unmarked sentence, though contextual occurrences of other combinations are rather frequent. Relevant to mention, that anyhow there is no data-driven evidence of the frequency of even basic order variation. In general many aspects of syntactic structures of Armenian language are not described and analyzed consistently enough. “There is a clear absence of a syntactic corpus study in previous work on Armenian standard varieties.” (Dum-Tragut 2002). A treebank as a linguistic resource may come useful to support such empiric research.

2.1 Annotation Scheme and Syntactic Tagset Development

Taking into account the syntactic characteristics of the target language and the past experience of treebankers, we were lead to choose the dependency representation framework for building an annotation scheme for Armenian sentences.

The core notion of dependency representation is the relational head-dependent structure. It can be graphically introduced by a tree – directed acyclic graph, which has one root and whose nodes have at least and most one parent node – the grammatical or technical (as the case may be) head.

In such dependency-based tree structure only the lexical words (nodes) are recognized, and the phrasal ones are omitted. All the lexical words together with the punctuation and other graphical symbols, i.e. every input token in a sentential segment is treated to be a node in the tree structure. No additional node insertion is allowed. There are no empty categories, which makes dependency structures more optimal and human-readable for the annotator to work on. The nodes in the sentence are linked with edges representing the dependency relation types in the form of syntactic function attribute values (given in the syntactic tagset in the section 2.2), which are attached to the nodes for technical data representation, yet in fact belong to edges (which is, by the way, visually seen on the tree view of the annotation tool as shown on Fig. 1 below).

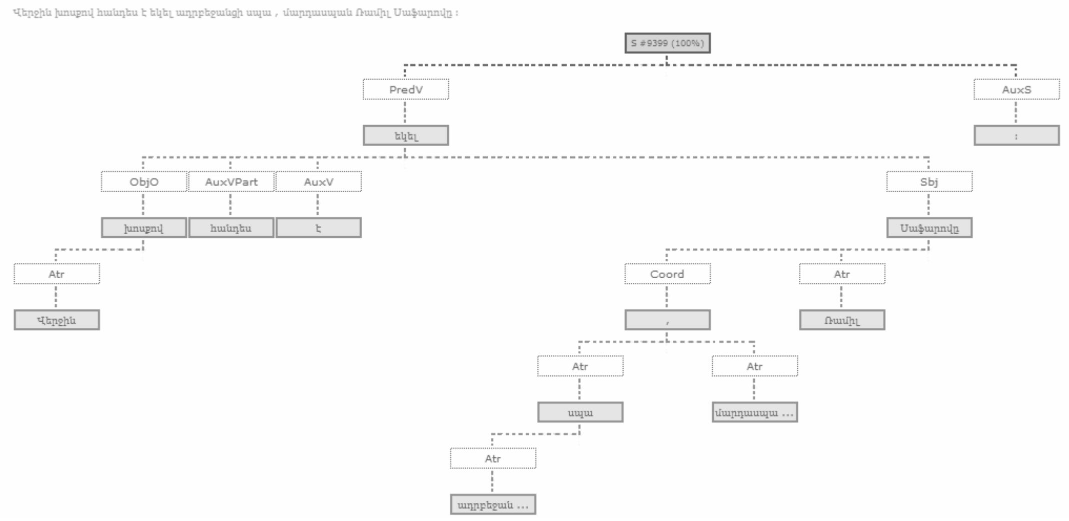


Fig. 1. Syntactic function and dependency representation in a tree structure

Following the basic assumption about the syntagmatic relationships, where two or more elements co-occur together, there is a dominant, principal element which is the primary determinant of the properties in such relationship. As regards to the sentence principal element, in our representation it is the predicate node taking the governing function in the entire utterance. Its valence (here it's worth to mention that whenever speaking of it, we always take into account the differentiation between the semantic and syntactic valence, and always refer to the latter), is determinant of the argument (again syntactic) structure of the whole sentence. Thus, in our representation, alongside with other arguments, the subject node is depending on the verb predicate (or other predication element). This very basic principle of annotation already differs from the traditional concept about sentence structure we find in Armenian grammar books that regard the subject as principal element together with the predicate. In general, wherever possible we follow the categorization and determination of grammatical relations and functions given in the textbook "Modern Standard Armenian" (Abrahamyan: 1981). Nevertheless, due to some inconsistency and incompatibility to formalism we come across in the grammar throughout developing the annotation scheme and tagset, we employ modifications to syntactic categories, either narrowing or broadening their delimitations, up to introducing new categories and structural representations.

Also, for the sake of formalism and computational approach, many nodes are labelled in a purely technical mode to represent structures and functions that are not linguistically competent. Due to the work being in preliminary stage and progress, many modifications and corrections are assumed to be done throughout their further exploitation in annotating the material.

One of the technical representations is the “dependence” of the predicate node on the sentence (<se>) node in the input file structure together with the sentence final punctuation mark. All the other elements are correspondingly suspended under this node in accordance with the valence frame of the governing verb or otherwise, being described in detail in the rules of annotation scheme. For keeping consistency, avoiding misinterpretations and clearing up dependency and syntactic function assignment, lexicons of verbs with their valence frame, as well as other relevant parts of speech with their assumed syntactic functions, are being compiled.

Another important non-linguistic dependency representation is drawn for linguistic phenomena like coordination and apposition, traditionally interpreted as equal sentence members. To represent such data, the members of coordination or apposition are represented as “depending” on the coordinating node (conjunction, punctuation), which itself hangs on the head node governing the coordination or apposition.

Many traditionally approached syntactic structures and functions are also getting more formal description and representation. The rules and tagset of syntactic functions are currently on the way of being formulated and populated, so that modifications and additions are in constant raise.

Below a brief list and description of the adopted syntactic tagset and functions are given. Once again it should be mentioned that the list is far from being complete and explanatory enough for many linguistic phenomena, moreover much more complex ones, which are unsurprisingly quite frequent in naturally occurring linguistic material to be annotated.

2.2 Syntactic Function Taglist and Brief Description

<Syn Fct> values	**Function.**	**Description, examples of expressions**
PredV	**Predicate**	The parent node of the sentence tree. Does not depend on other sentence member nodes, yet technically is suspended under the <se> node in data structure. Is expressed by simple finite verb forms or converbs that form finite predications. *Note:* The predicate of a subordinate clause does not take this function tag, but rather takes the function of the clause it represents.
PredN	**Predicative nominal element**	This is the predicative part in the compound nominal predication; always hangs on the AuxV, if latter is expressed overtly in the sentence.

AuxV **Auxiliary verb 'է'**

This is the function tag assigned to the copula when being a part of the main predication and always depends on PredV node. Otherwise stands as the governing node of the sentence with dependent compound nominal predicative. If not the previous two, this 'է' takes the function AuxPart hanging on modal particle '(չ)պետք' in neighbourhood with subjunctive verb. The particle itself takes the function AuxMod being suspended under verb.

Note: The occurrence of '(չ)պետք է' in a uniclausal occurrence with infinitive is represented as a construction of the case: PredN$_{[D]}$ AuxV$_{[H]}$

AuxVPart **Compound verbal particle**

This is the non-verbal element in compound analytic verb forms. Always is suspended under the main verb it "complements". Ex. '*ման*$_{[D]}$ գալ$_{[H]}$' '*գլուխ*$_{[D]}$ բերել$_{[H]}$', etc. A lexicon of such particles and respective verbs is envisaged to be compiled to avoid the mislabelling of these elements with other functions, as well as the fact that in such forms the main verbs, having lost their main lexical meaning, change their argument structure too.

Sbj **Subject**

The direct argument of the predicate which can be missing from the overt structure of the sentence. Compliant to our dependency representation it is depending on the predicate or auxiliary verb.

ObjD **Direct object**

An obligatory direct argument depending on the verb according to its valence frame: in active voice utterances it usually corresponds to the semantic role of patient. Usually morphologically is expressed by a noun, pronoun, other nominalizations in nom/acc(dat in animate nouns) case marking. The infinitival part of compound predicate also takes this function. Ex.: 'Ուզում$_{[H]}$ է *աշխատել*$_{[D]}$:'

ObjI **Indirect object**

An obligatory indirect argument depending on the verb according to its valence frame. Usually, but not always, the semantic role of recipient corresponds to this argument.

ObjO **Oblique object**

Adjunct or non-obligatory syntactic argument whose nature and behaviour are more describable in semantic terms than syntactic. They show different semantic relations to the verb and direct arguments and are likely to be most constrained in the semantic roles they may individually express, hence no further subclassification is given between kinds of oblique objects. This element is likely to be marked by an adposition, in which case the latter is the governing node, or case affix, in which case the ObjO is hung directly on the verb.

Comp **Complement**

Arguable it may seem, as it does not have an exact match in traditional grammar, or more correctly, it is presented under various categories. This is a direct argument with predicative role complementing itself one of the arguments of the verb; either the subject or object. In general the nature of this element is of double dependency both on verb and on the argument it refers to. Yet again, following the principles of annotation of unidirectional dependence, in the presence of the verb in the sentence, it gets the tag Comp and gets suspended under the node of argument it concerns, or the preposition ‘որպես’, as the case may be. Ex., ‘Պատերը$_{[H]}$ *սպիտակ*$_{[D]}$ ներկեցինք:’ ‘ Նա$_{[H]}$ *հիվանդ*$_{[D]}$ է ձնանում:’ ‘Նա$_{[H]}$ ելույթ ունեցավ որպես *տնօրեն*$_{[D]}$:’

Note: should not be confused with other functions like Obj, Adv, neither the overtly similar seeming apposition case with ‘որպես’, etc.

CompV **Verbal Complement**

When the governing node to which a Complement element refers is elided or missing at all, it gets the function tag CompV, and is suspended under the verb itself. Ex., ‘*հիվանդ*$_{[D]}$ է ձնանում$_{[H]}$’, ‘ելույթ ունեցավ$_{[H]}$ որպես *տնօրեն*$_{[D]}$’

CompAdv **Adverbial complement**

An obligatory argument dependent on the verb that is common to misinterpret with either adverbial or object: its usage is determined with the subcategorization frame of the verb. Depends on the verb it complements.

Note: a test to identify this function - the sentence is ill-formed, ungrammatical without it, unlike with adverbial: *Նա զտնվում է: *Գիրքը դնել:

Adv **Adverbial**

Optional arguments and adjuncts modifying the verbal elements. As the further subcategorization of these modifiers is according to their semantic roles, no further classification is given. Depends on the verb it modifies.

Atr **Attribute**

The traditional nominal modifying element. It depends on the nominal it modifies. Typically is expressed by lexical categories as adjectives, quantifiers, demonstratives, numerals, but also may be expressed by nominals marked with nominative, ablative, instrumental, locative case.

AtrGen **Genitive Attribute**

The possessor attribute which is actually a semantically constrained subcategory of Atr. Nevertheless a syntactic function is categorized as its inflectional case marking is easy for look-up. As the Atr, it is hanging on the nominal it modifies.

Apos **Apposition**

Again a technical node for representing the apposition. Is expressed by the punctuation mark "bouth", while the members of the apposition themselves are suspended under this node:

Note: The case of apposition introduced with ‘որպես’, ‘իբրև’ should not be misinterpreted with Comp: ‘Նա$_{[H]}$` որպես *տնօրեն*$_{[D]}$, ելույթ ունեցավ:’

Coord **Coordination**

The coordination node technically dependent on the higher element to which its own dependents refer, thus having those as child nodes. Is expressed by punctuation marks -comma, period, or coordinating conjunction.

AuxSub **Conjunctive element**

Subordinate conjunction or punctuation (period or "bouth" otherwise not specified as main apposition node).

AuxA **Adposition**

Prepositions and postpositions: their position in the tree is governing the nominal they refer to, thus they are suspended where the nominal introduced by them would take place and correspondingly the nominal is hung on them.

AuxMod **Modals and emphasizing words**

These can be suspended under any element member, according to the function they take to emphasize a particular word. A list of these words is assumed to be compiled. Should not be misinterpreted with adverbials or other similar elements. In case such word refers, emphasizes the whole utterance, without any clear-cut emphasize on a particular member, it gets always suspended under the predicate element of the sentence.

AuxPart **Particles of multi-token words**

These are parts of analytical word constructions that have no syntactic function; always suspended under their main lexical node. Ex., է in '(չ)պետք$_{[H]}$ *է*$_{[D]}$',

AuxP **Punctuation**

All punctuation marks with otherwise not specified above function they take (quotes, brackets, sentence start dash), abbreviation period and other graphical symbols. Their suspension depends on the position they take and are to be more detailed explained in tagging manual.

AuxS **Colon : full stop**

The sentence final punctuation mark; always hangs on the root of the sentence tree together with the PredV or in the absence of the latter, another sentence governing node.

ExD **External dependency**

A technically added function which labels the node that misses its actual governor in the sentence; typically the ellipses are handled by this function tag. Another example of such function tag exploitation is for the attributes referring to the lexical part of relational nouns, which are then suspended on the entire wordform. Ex. 'մեր *հարևան* $_{[D]}$ Արամինը$_{[D]}$'.

Voc **Vocative**

Depends on the verb of the imperative sentence.

2.3. Data representation and the annotation tool for manual tagging

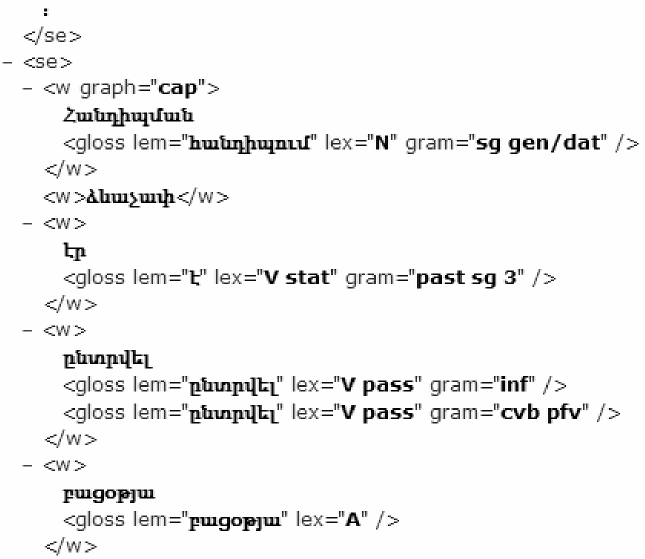
In data structure and representation the XML markup standard is followed up. The source input data is shared by Eastern Armenian National Corpus (EANC), and its format already has the corresponding structure. The textual data is analyzed morphologically; that is, the lexical nodes already bear morpho-syntactic information in the form of tags with attributes with corresponding values. Nodes appear in their linear order without contextual disambiguation.

Fig 2. shows the file format serving as an input to be pre-processed with slight modifications and be prepared for syntactic annotation by the application that is currently under development to serve the objectives of manual annotation.

The Tool is currently being developed with following goals:

- entitles to import morphologically tagged data from XML source
- enables to perform basic updates to the imported XML structure, as well as imports punctuation tags otherwise left untagged in the source files
- integrates the imported files into a database for easy data handling
- allows visualisation and edition data organized in paragraphs and sentences
- allows to perform manual syntactical annotation of the tokens
- displays real time visual tree structure of the syntactically annotated data

- re-exports original format files after modifications and syntactically annotated file to XML structures
-



```
      :
    </se>
  - <se>
    - <w graph="cap">
        Հանդիպման
        <gloss lem="հանդիպում" lex="N" gram="sg gen/dat" />
      </w>
      <w>ձևաչափ</w>
    - <w>
        էր
        <gloss lem="է" lex="V stat" gram="past sg 3" />
      </w>
    - <w>
        ընտրվել
        <gloss lem="ընտրվել" lex="V pass" gram="inf" />
        <gloss lem="ընտրվել" lex="V pass" gram="cvb pfv" />
      </w>
    - <w>
        բացօթյա
        <gloss lem="բացօթյա" lex="A" />
      </w>
```

Fig 2. Original source file with morphological analysis

The syntactical information is introduced by adding to each token a <Syn> tag with attributes <fct> and <dep>, respectively marking the function of a node and pointing to its governor node. As mentioned before, the syntactic function actually belongs to the edge of the tree structure, relating the node to its governor.

The tool enhancing the annotation is web based application with SQL Server database use. The web application will allow easier later sharing of knowledge, and easy cooperation with other projects without requiring special computer configurations or tools to install.

The system is made of several building blocks described below. As the application is under development the following list may not be up to date at the time of reading this paper.

2.3.1. Importation Routine

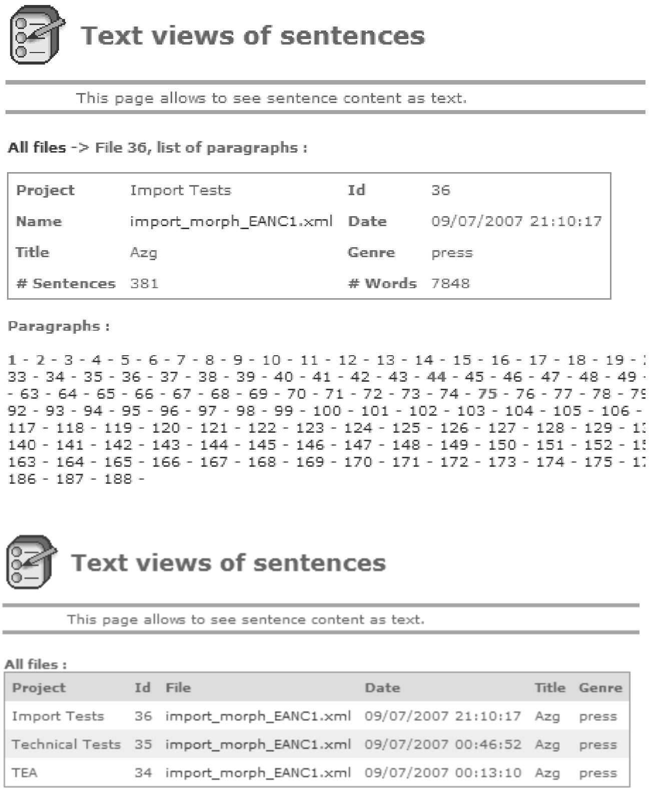
This imports morphologically analyzed data files and performs basic updates to make the data compliant with the syntactical layer needs. The input files are structured into paragraphs, sentences, words and “gloss” tags representing the morphological layer. As the morphological disambiguation is not carried out, there may be several glosses per word. The data is analyzed in linear manner and bears only the morphological information for the lexical items. The punctuation marks and digits in the imported files are left out tagging.

The initial pre-processing operates a preliminary parser on the source file and results in integrating punctuation and other graphical symbols into the structure (as the latter also take functions in a sentence structure and will need to be positioned into the tree), also adding explicit numbering of paragraphs, sentences and ordering the word sequences. Once updated, the file is parsed into a relational database.

Files are organized into projects that are created on user needs (per project, per annotator teams, etc.). Each file is divided into "paragraphs", subsequently divided into "sentences" and then into "tokens" - <t>, which have come to substitute the original <w> tags and also include punctuation marks and digits.

A Token contains the character string <txt>, which in the case of a word carries the morphological information about it within zero (if the word has not been recognized during the morphological analysis), one or several "glosses", its sequential order <ord> in the sentence, its <type> with values for lexical words, punctuation, graphical symbols and digits, and after annotation, the syntactical layer information within an additional <syn> tag.

2.3.2. Data handling tools



Text views of sentences

This page allows to see sentence content as text.

All files -> File 36, list of paragraphs :

Project	Import Tests	Id	36
Name	import_morph_EANC1.xml	Date	09/07/2007 21:10:17
Title	Azg	Genre	press
# Sentences	381	# Words	7848

Paragraphs :

1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 -
33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 -
- 63 - 64 - 65 - 66 - 67 - 68 - 69 - 70 - 71 - 72 - 73 - 74 - 75 - 76 - 77 - 78 - 7
92 - 93 - 94 - 95 - 96 - 97 - 98 - 99 - 100 - 101 - 102 - 103 - 104 - 105 - 106 -
117 - 118 - 119 - 120 - 121 - 122 - 123 - 124 - 125 - 126 - 127 - 128 - 129 - 1
140 - 141 - 142 - 143 - 144 - 145 - 146 - 147 - 148 - 149 - 150 - 151 - 152 - 1
163 - 164 - 165 - 166 - 167 - 168 - 169 - 170 - 171 - 172 - 173 - 174 - 175 - 1
186 - 187 - 188 -

Text views of sentences

This page allows to see sentence content as text.

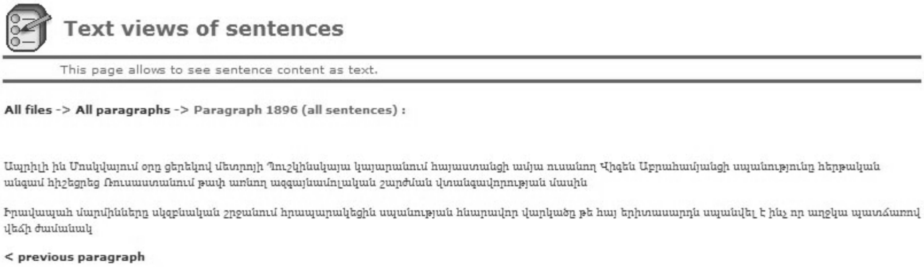
All files :

Project	Id	File	Date	Title	Genre
Import Tests	36	import_morph_EANC1.xml	09/07/2007 21:10:17	Azg	press
Technical Tests	35	import_morph_EANC1.xml	09/07/2007 00:46:52	Azg	press
TEA	34	import_morph_EANC1.xml	09/07/2007 00:13:10	Azg	press

Fig 3. Navigaition tools

This block consists of two main types: the structure and text data navigation tools and, the data edition tools.

The *structure and text navigation tools* allow the annotator to navigate within the data by browsing projects, files, paragraphs and sentences (Fig. 3). This navigation entitles the user to work on XML files omitting all technical XML / database tags, making the files readable, shown as plain text (Fig. 4),with or without morphological and syntactical information.



Text views of sentences

This page allows to see sentence content as text.

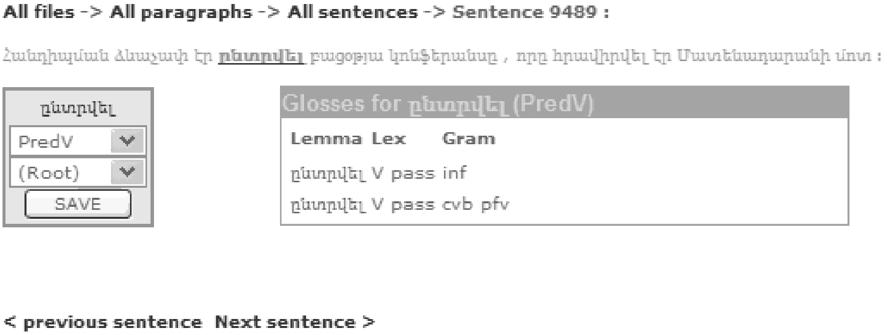
All files -> All paragraphs -> Paragraph 1896 (all sentences) :

Ապրիլի ին Մոսկվայում օրը ցերեկով մետրոյի Պուշկինսկայա կայարանում հայաստանցի ամյա ուսանող Վիգեն Աբրահամյանցի սպանությունը հերթական անգամ հիշեցրեց Ռուսաստանում թափ առնող ազգայնամոլական շարժման վտանգավորության մասին

Իրավապահ մարմինները սկզբնական շրջանում հրապարակեցին սպանության հնարավոր վարկածը թե հայ երիտասարդն սպանվել է ինչ որ աղջկա պատճառով վեճի ժամանակ

< previous paragraph

Fig. 4 . Text view of a file



All files -> All paragraphs -> All sentences -> Sentence 9489 :

Հանդիպման ձևաչափ էր **ընտրվել** բացօթյա կոնֆերանսը , որը հրավիրվել էր Մատենադարանի մոտ :

ընտրվել

PredV

(Root)

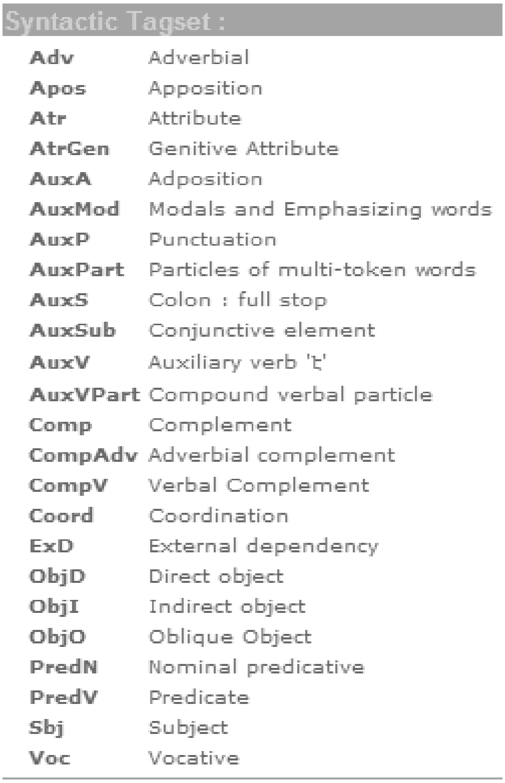
Glosses for ընտրվել (PredV)

Lemma	Lex	Gram
ընտրվել	V	pass inf
ընտրվել	V	pass cvb pfv

< previous sentence Next sentence >

Fig. 5. The annotation tool with the annotation toolbox with the imported morphological information

The annotator’s (manual) work can be performed from *data edition tools*. The application can work on token per token basis or full sentence annotation by means of annotation toolbox (to choose the syntactical function tag of a token and its governing node).



Syntactic Tagset :

Adv	Adverbial
Apos	Apposition
Atr	Attribute
AtrGen	Genitive Attribute
AuxA	Adposition
AuxMod	Modals and Emphasizing words
AuxP	Punctuation
AuxPart	Particles of multi-token words
AuxS	Colon : full stop
AuxSub	Conjunctive element
AuxV	Auxiliary verb 'է'
AuxVPart	Compound verbal particle
Comp	Complement
CompAdv	Adverbial complement
CompV	Verbal Complement
Coord	Coordination
ExD	External dependency
ObjD	Direct object
ObjI	Indirect object
ObjO	Oblique Object
PredN	Nominal predicative
PredV	Predicate
Sbj	Subject
Voc	Vocative

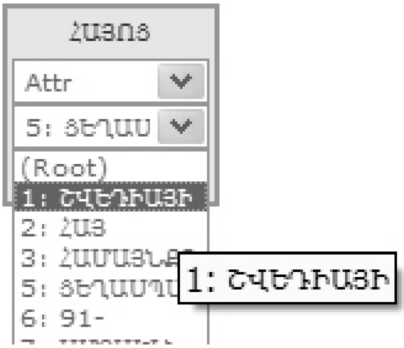


Fig. 6. Syntactic tagset display

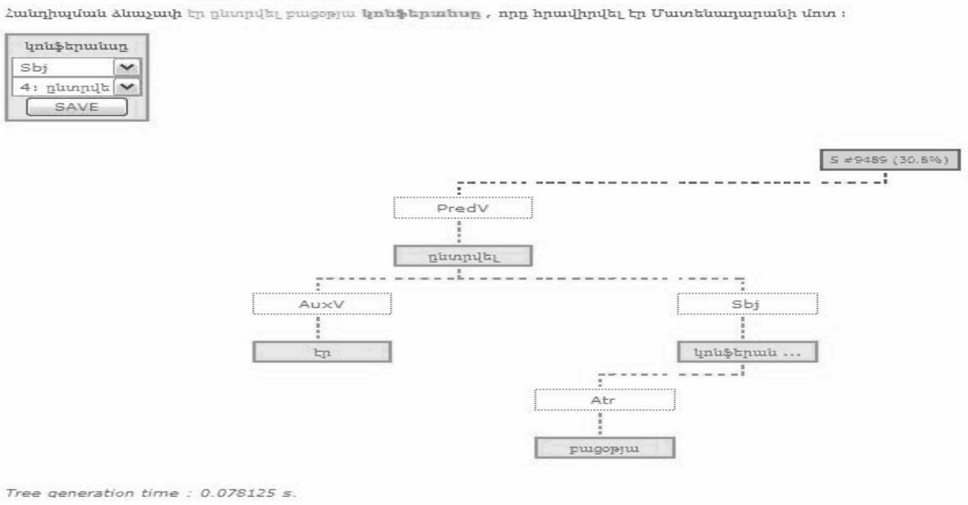
Fig. 7. Head selection

The list of syntactical function tags described in above sections (Fig. 6) is available in the first drop-down list of the annotation toolbox. These functions are stored into the database and can be managed by the user (add / edit / removed, if not used).

The second list (Fig. 7) displays the tokens in the sentence from which the annotator chooses the one, on which the token under analysis depends on, according to the annotation rules and principles.

The process of manual syntactical annotation allows creating the tree view of the sentence. The web based tree view component is developed in HTML, using <table> and simple images to be easily and widely usable, on all platforms, without software installation needed. The tree logic is all calculated on server, to produce light client application allowing use of the annotation tools on computers with lower resources or slow internet access.

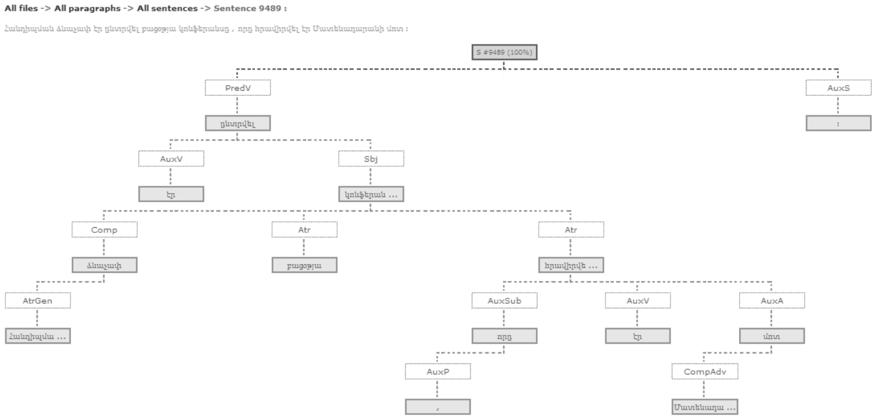
Throughout the annotation of a sentence, the tree is being generated and can be browsed in real time (Fig. 8). This figure displays the tree being built, with the root being the <s>entence.



Հանդիպման ձևաչափի էր ընտրվել բացօթյա կոնֆերանսը , որը հրավիրվել էր Մատենադարանի մոտ ։

Tree generation time : 0.078125 s.

Fig 8. The tree is being built while the annotator works on the sentence

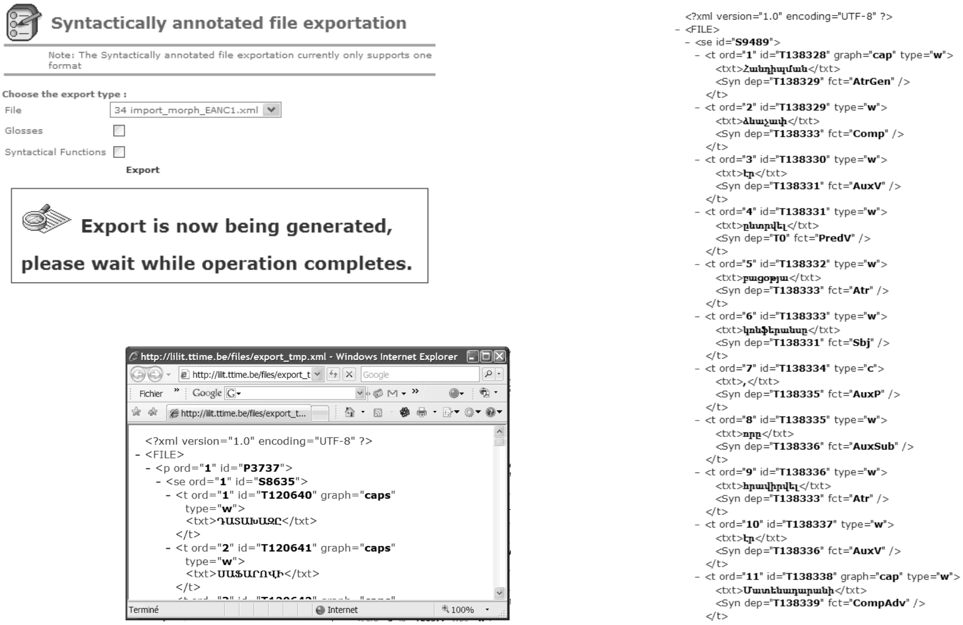


All files -> All paragraphs -> All sentences -> Sentence 9489 :

Հանդիպման ձևաչափի էր ընտրվել բացօթյա կոնֆերանսը , որը հրավիրվել էր Մատենադարանի մոտ ։

Fig 9. Dependency tree sample

The tree logic incorporates arches, that are made of images put in tables, with branches crossings and branches divisions etc., put in visual layout. Below an example of a fully annotated sentence tree is given:



Syntactically annotated file exportation

Note: The Syntactically annotated file exportation currently only supports one format

Choose the export type :

File 34 import_morph_EANC1.xml

Glosses □

Syntactical Functions □

Export

Export is now being generated, please wait while operation completes.

http://lilit.ttime.be/files/export_tmp.xml - Windows Internet Explorer
http://lilit.ttime.be/files/export_t
Google
Fichier
Google
http://lilit.ttime.be/files/export_t...
Terminé
Internet
100%

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <FILE>
  - <p ord="1" id="P3737">
    - <se ord="1" id="S8635">
      - <t ord="1" id="T120640" graph="caps"
          type="w">
          <txt>ԴԱՏԱԽԱԶԸ</txt>
        </t>
      - <t ord="2" id="T120641" graph="caps"
          type="w">
          <txt>ՍԱՖԱՐՈՎԻ</txt>
        </t>
```

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <FILE>
  - <se id="S9489">
    - <t ord="1" id="T138328" graph="cap" type="w">
        <txt>Հանդիպման</txt>
        <Syn dep="T138329" fct="AtrGen" />
      </t>
    - <t ord="2" id="T138329" type="w">
        <txt>ձևաչափի</txt>
        <Syn dep="T138333" fct="Comp" />
      </t>
    - <t ord="3" id="T138330" type="w">
        <txt>էր</txt>
        <Syn dep="T138331" fct="AuxV" />
      </t>
    - <t ord="4" id="T138331" type="w">
        <txt>ընտրվել</txt>
        <Syn dep="T0" fct="PredV" />
      </t>
    - <t ord="5" id="T138332" type="w">
        <txt>բացօթյա</txt>
        <Syn dep="T138333" fct="Atr" />
      </t>
    - <t ord="6" id="T138333" type="w">
        <txt>կոնֆերանսը</txt>
        <Syn dep="T138331" fct="Sbj" />
      </t>
    - <t ord="7" id="T138334" type="c">
        <txt>,</txt>
        <Syn dep="T138335" fct="AuxP" />
      </t>
    - <t ord="8" id="T138335" type="w">
        <txt>որը</txt>
        <Syn dep="T138336" fct="AuxSub" />
      </t>
    - <t ord="9" id="T138336" type="w">
        <txt>հրավիրվել</txt>
        <Syn dep="T138333" fct="Atr" />
      </t>
    - <t ord="10" id="T138337" type="w">
        <txt>էր</txt>
        <Syn dep="T138336" fct="AuxV" />
      </t>
    - <t ord="11" id="T138338" graph="cap" type="w">
        <txt>Մատենադարանի</txt>
        <Syn dep="T138339" fct="CompAdv" />
      </t>
```

Fig 10. Syntactically annotated file exportation and exported file sample

2.3.3. Exportation Routine

The exportation routine, as the names says, allows recreating XML files in various supported formats, with or without morphological information and with or without syntactical information.

CONCLUSION

The annotation of linguistic data with integration of syntactic structure information has been widely exploited during the past decade in the general and field linguistics. These efforts have proven to be a very successful basis for the progress in language theory and processing technology, from which apparently the broadly spoken and studied languages have favoured, while the minor ones perform lacks in such research experience.

Armenian language represents a separate node in the Indo-European language family tree, thus lacks any close relatives from the genetic point of view. Inevitably it has inherited structures proper to other family languages, yet not surprisingly it may have developed quite individual features. In this stage of linguistic science when language technologies afford accumulation of valuable knowledge, it is worth to build resources serving for such needs.

Treebank building is a challenging task. It is time and labour consuming, yet is very promising for language resource building and field linguistic knowledge updating to match the current linguistics trends. Although we perform manual annotation in the frames of the project, as a preliminary stage, the work assumes to give challenge and prospects for possible development and training of automatic tools for processing

Armenian language resources. Further development and fine-grained annotation scheme and rules are envisaged to be carried out, as well as manual annotation of reasonable amount of data. We believe it may come useful in its later exploitation for linguistic research and NLP tools development purposes.

REFERENCES

Ann Abellié (ed.) 2003. *Treebanks: Building and Using Parsed Corpora*. Kluwer

Jurafsky D., Martin J.H. 2000. *Speech and Language Processing*. New Jersey

Manning C., Schutze H. 1999. *Foundations of Statistical Natural Language Processing.* London

Jan Hajic, Joakim Nivre. 2006. *TLT 2006. Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories*. Prague

Jan Hajic. 1998. *Building a Syntactically Annotated Corpus: The Prague Dependency Treebank.* Prague

L.van der Beek (ed.) 2002. *The Alpino Dependency Treebank. Algorithms for Linguistic Processing NWO PIONIER Progress Report*. Groningen

Skut W. (ed.). 1997. *An Annotation Scheme for Free Word Order Languages*. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP).* Washington, D.C.

Tuomo Kakkonen. 2005. *Dependency Treebanks: Methods, Annotation Schemes and Tools.* In *Proceedings of the 15th Nordic Conference of Computational Linguistics (NODALIDA 2005).* Joensuu

Boguslavsky (ed.). 2002. *Development of a dependency treebank for Russian and its possible applications in NLP*. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*. Las Palmas

C. Bosco, V. Lombardo. 2004. *Dependency and relational structure in treebank annotation*. In *Proceedings of Workshop on Recent Advances in Dependency Grammar at COLING'04*.Geneve

Kiril Simov. 2003. *HPSG-Based Annotation Scheme for Corpora Development and Parsing Evaluation. Proceedings of RANLP*. Borovets

J. Weitenberg, S. Simonian. 1993. *Computers in Armenian Philology*. Yerevan

Van Valin. 2001. *Syntax*. Cambridge

J. Dum-Tragut. 2002. *Word-order correlations and word order change: An applied-typological study on literary Armenian variant.* München

S. Abrahamyan. 1981. *Modern Standard Armenian.* Yerevan

V. Arakelyan. 1958-62. *Syntax of Modern Armenian, I-II.* Yerevan

N. Kozintseva. 1995. *Modern Eastern Armenian*, München