ARMENIAN LANGUAGE
IN COMPUTATIONAL LINGUISTICS

by

Vachagan Gratian

Presented to the

Department of English & Communications

in Partial Fulfillment of the

Requirements for the Degree of Bachelor of Arts

American University of Armenia

Yerevan, Armenia

6 May, 2017

Within a computer, natural language is unnatural.

Alan Perlis

1. Introduction

Armenian is an Indo-European language spoken by about 7-8 million people in the world and is the official language of the Republic of Armenia. Often cited as one of the oldest surviving languages of the world, Armenian became a literary language in A.D. 406-407 when the Armenian alphabet was developed and Christian and Greek literature was translated into Classic Armenian. With a significant non-Indo-European legacy (Hewsen, 2001, p 29), it was influenced by many other languages; modern Armenian has plentiful loanwords from Greek, Syriac, Latin, Arabic, Persian, Turkish and Russian. It saw a literary blossoming in the mid 19th century and has two main literary standards ever since: Western Armenian and Eastern Armenian. Modern Armenian is widely used on the Internet. Armenian is the 36th by number of articles among the 295 languages of Wikipedia. It is often noted that Armenia's socio-political discourse and opinion forming chiefly occur online, including on social media (Baghdasaryan, 2013). Consequently, a significant amount of Armenian text is available online, but is rarely used for computerized analysis – either for academic research or business purposes. This paper deals with Armenian as a subject of computational linguistics: it provides an overview of the basic tools and resources that are required to manipulate language in a computer system, it furthermore recounts ongoing and completed Armenian NLP projects.

Computational linguistics is the branch of computer science that is concerned with making human language "understandable" for a computer system, a process known as *natural language processing* (NLP). As a multidisciplinary field, it involves techniques and theories from various disciplines, including applied linguistics, discourse analysis, syntax, morphology and lexicology, mathematics and mathematical logics, cognitive science, statistics and finally also *machine learning*. Computational linguistics is not a recent phenomenon, the first *machine translation* software was described by the mathematician Warren Weaver in 1949. The term *computational linguistics* itself was coined in the 1950s by David G. Hays who also authored the first textbook on computational linguistics, *Introduction to Computational Linguistics* (published in 1967). The first Armenian-English translation engine was developed in 2000 and the largest multilingual translator, *Google Translate*, was launched in 2006. NLP has a wide range of applications. A few more examples are:

- Software that improve human-generated text (spell-checking and grammar-checking tools, automatic hyphenation, etc.).
- Information extraction and retrieval. NLP allows extraction of information from very large data that would be impossible to analyze with human labor.
- Similar to the former are: sentiment analysis and summarization.
- Machine conversation (conversational agents and chatbots).
- Speech recognition

Nearly half a century after its emergence, natural language processing remains highly complex and challenging. The standard medium for human-computer interaction are still graphical interfaces and artificial languages (such as programming languages). A natural language is highly ambiguous, dependent on context and a general knowledge of the world. It, furthermore, changes constantly: new words are invented, old words disappear and the meaning, connotations of existing words changes over time. An everyday use of language includes many figures of speech, sarcasm, wordplay and idioms that usually go unnoticed for a native speaker. Finally, a natural language has many styles and varieties: formal and colloquial variations, prose and poetry, dialect, jargon and slang.

As a consequence of this *affluence* of natural languages, computational linguists are still primarily concerned with relatively “basic” NLP tasks, such as syntactic parsing. Only when these basic tasks are performed on high accuracy, more complex NLP tasks – such as machine conversation or information harvesting – can be developed. When it comes to rare languages, as the Armenian language, many of the basic NLP tasks are either not developed, incomplete or not performing well. A further point of concern is that scarce amount of research and software that has been written, the treebanks and text corpora that have been created, are either inaccessible or not suitable for a researcher or software engineer who enters the realms of Armenian NLP.

This paper aims to provide an initial overview and guideline into the computational aspects of the Armenian language. It describes the basic components and tools necessary for Armenian NLP and sums up the fields where work has already been done or is currently worked on.

2. Literature review

Computational linguistics is a field of research almost as old as Computer science itself. As such, it has been a subject of extensive academic research since the 1960s. The journal of the Association of Computational Linguistics only hosts over 41,000 papers in their online anthology (ACL, May 2017). NLP researchers investigate on the wide amount of issues concerning the computation of human language, the methods and algorithms of decomposing spoken or written human language and converting it into structured and computable data. While a significant amount of NLP publications don't deal with the processing of any specific natural language, they usually include examples from at least one language and sometimes are specifically focused on one or more languages.

The scarcity of publications on Armenian in NLP is in contrast with the few number of Armenian NLP tools and resources developed over the past few years. Two such significant projects are a machine translation tool initiated in 2000 (translator.am) and a national language corpus launched in 2006 (eanc.net). Although an enormous amount of work is invested in both projects, no scientific papers appeared dealing with NLP issues of Armenian, nor is any source-code of the developed software online publicized.

One publication that specifically deals with Armenian NLP tasks is *A Lexical Database: Construction and Implementation* by Lilit Arakelyan. This paper is the description of an formalistic Armenian dictionary that would include a lemmatisation tool. The paper is accompanied by an XML template that can be used for creating and populating the dictionary database which subsequently can serve as a lemmatisation tool for the Armenian language (both Eastern and Western versions).

Computational Linguistics. Models, Resources, Applications by Igor A. Bolshakov and Alexander Gelbukh is handbook for computer science students with no expertise in linguistics. The book has a strong emphasis on the historical, linguistic and theoretical background of computational linguistics and chiefly focuses on two grammar formalisms: meaning-text theory (MTT) and head-driven phrase structure grammar (HPSG). It also includes a broad overview of existing and potential NLP applications. Although the book is mainly based on Spanish NLP examples, it has a quite generic approach and uses a relatively accessible jargon. The book is an excellent introduction for anyone with basic CS background who wants to enter the field of computational linguistics.

Evaluating Natural Language Processing Systems: An Analysis and Review is a highly praised publication about NLP. This book provides a comprehensive coverage of existing NLP applications, evaluates their advantages and deficiencies. The authors review two major methodologies for evaluating NLP systems – the Wizard-of-Oz method and the Neal-Montgomery System Evaluation Methodology – and provide their own methodologies with an extensive description of their criteria.

A publication that is more recent and provides some new insights is *Natural Language Processing: 8 Lectures by Ann Copestake of Cambridge University*. It includes lecture notes of the Introduction to NLP course at University of Cambridge. The course aims to introduce students with the basic techniques of NLP systems such as stemming and spell-checking, information extraction and machine translation. Provides the set of skills and knowledge necessary to develop and understand simple NLP applications.

A book that addresses the programming aspects of NLP is *Natural Language Processing with Python*. This is a practical introduction to advanced NLP using Python, a programming language that is widely used for NLP research and software development. It describes the existing NLP resources and libraries that are available for Python developers and evaluates the most important NLP algorithms and data structures.

A highly accessible introduction to Lucien Tesnière's grammar theories is the article *Dependency Grammar And Valency Theory* by Vilmos Ágel and Klaus Fischer in *The Oxford Handbook of Linguistic Analysis.* This article goes through the central concepts of Tesnière's dependency structure — such as *connexion*, *junction* and *translation* — and exemplifies them with English, German, French and Hungarian phrases. Dependency grammar is often compared with constituency grammar although the the authors emphasize that they don't consider these two models as contending ideas, but rather as complementing ideas.

A widely acclaimed work on the grammar and syntax of Armenian is Hrachia Acharyan's *Universal Grammar of Armenian in Comparison with 562 Languages: Semantics, Lexicology and Syntax*. This book, posthumously published in 2005, includes the last 3 volumes of his *Universal Grammar of Armenian* (the first volume of the series was published in 1952). Although some of Acharyan's ideas and theories are contended and even refuted by contemporary linguists, his work is considered a monument in Armenian language studies. It analyses a wide aspect of the Armenian language starting from the 5th century.

Another publication providing substantial linguistic background of the Armenian is *Modern Eastern Armenian* by the Austrian linguist Jasmine Dum-Tragut. This handbook provides an comprehensive overview of the Armenian grammar, including its morphology, phonology, syntax and word formation. It addresses the literary and conversational languages that are spoken in the Republic of Armenia. The book is intended for both language learners as well as researchers.

3. Research Methodology

The purpose of this paper is to provide a general overview of the Armenian language as an object of computational linguistics. The questions it aims to answer are the following: What is the current state of the Armenian language in the context of language processing technologies? What efforts have been made to create NLP tools and resources for the Armenian language?

The findings of this paper are based on scholarly research and interviews. At one hand, a number of publications that were relevant to the topic were consulted. Roughly speaking, these publications fall into three categories:

1. Publications on computational linguistics, grammar theories and NLP programming. Some of these publications are quite technical and are aimed at computer scientists and software engineers, while others are intended for linguists and deal with theoretical concepts of

computational linguistics.

2. Publications about the Armenian language, including Armenian grammar, syntax and lexicology. Although this group is not directly related to NLP, it provides data that is essential to assess the computation of Armenian.

3. Third group of publications are those directly related to computational linguistics *and* the Armenian language, however because of the scarcity such publications, this category did not play a significant role.

During the research concepts from group 1 were compared and projected to the overview of the Armenian language from group 2. The resulting findings and insights were then synthesized into a general background of computational linguistics and into the basic concepts and definition of Armenian NLP.

Interviews were the main source of information about existing Armenian NLP projects and helped to understand the challenges and opportunities of the Armenian language in this field. My most important interviewees who provided many useful recommendations and insights were:

- Tom Samuelyan, author of Arak29 and AUA professor

- Marat Yavrumyan, Armenian Dependency Parser and Treebank Project

- Eduard Manukyan, ISMA (transator.am)

- Hayk Saribekyan, MIT student in artificial neuroscience

4. Research Findings and Analysis

4.1. Overview of NLP tasks

Natural language processing can be subcategorized into *core* NLP and *extended* NLP. Core NLP includes basic analysis of language: dividing a string of text into individual words and sentences, and identifying the syntactic roles of words in a sentence. Extended NLP uses this information to further manipulate language examples of which are machine translation, speech recognition or dialog systems. This kind of NLP systems thus depends on the performance of core NLP tools. Even when it comes to a language as English, where the lion's share of NLP research and engineering is focused, high accuracy of basic NLP tasks is still a challenge. A vivid example of this are machine translation systems which make an abundance of mistakes even when they are assisted by statistics and recently also machine learning (G. Lewis-Kraus, 2016).

Below is a brief description of what is usually defined as core NLP tasks:

- Tokenization is the process in which text is broken down into sentences, phrases and words, i.e. *tokens*. Although it might seem a simple task, it is often problematic to set the rules of what an individual word is. Traditionally, everything that is combined with whitespace or punctuation (including a hyphen) is divided into tokens. But sometimes the English word *co-operate* can also be spelled as *cooperate* and the noun *ice skate* is not the same as the verb *ice-skate*. Similarly, in Armenian almost any two words can be combined into a compound word

(Yavrumyan, interview). E.g. *գցել-բռնել, խելք֊խելքի, ծանր֊ծանր։* Noteworthy, hyphenized words not only get a new meaning, but, as in the case of ծանր֊ծանր, the lexical category of the word can change (ծանր is an adjective, while ծանր֊ծանր is an adverb).

- Lemmatisation (sometimes also named *morphological parsing*) is the restoring of the lemma from an inflected word. E.g. the lemma of ընկերներով is the noun ընկեր and the lemma of գնացինք is the verb գնալ. Lemmatisation is usually done using dictionaries that include all inflected forms of a word. A description of Armenian lemmatisation is done by Lilit Arakelyan in her master's thesis (Arakelyan, 2016).

- POS-tagging is often done simultaneously with lemmatisation. In this step each word gets a number of labels that include information about its original inflected form such as word category (noun, verb, adjective...), number, tense, case, etc.

- Named-entity recognition is the identification of two or more tokens as a proper noun. E.g. Հայաստանի Հանրապետություն, Նյու Յորք or Մհեր Մկրտչյան.

- Parsing (or syntactic parsing) is the final process of defining the syntactic and semantic relationships between words in a sentence or phrase. Since any sentence or phrase is inherently a hierarchic structure, usually this NLP task is represented in the form of a syntactic tree.

It is important to mention that these tasks usually form a pipeline which means successive tasks are dependent on preceding tasks, for example *lemmatisation* is possible after completion of *tokenization* and *NE-recognition* can only be successful after these two tasks are completed.

In this paper, I will refer to the tasks preceding parsing as *pre-parsing NLP*. There are several pre-parsing tools for Armenian that are quite successful: the EANC parser, ISMA, Arak29, several spell-checking and other specialized software. Conversely, very few initiatives have been taken to create an Armenian (syntactic) parser. None of the pre-parsing software for Armenian are available as free or open-source software, which means that any initiative to create an Armenian parser has to start with developing the preceding pre-parsing tools from scratch. Developing any of these core NLP tasks is a laborious work and often requires very specific expertise and sophisticated linguistic resources. The main challenge for Armenian NLP is therefore not so much *more* resources and tools, but more *free* and *open-source* resources.

Among the core NLP tasks, parsing proves to be the most precarious and problematic one. This is the point when ambiguity, complexity and innovativeness of natural languages come to play. Newspaper headlines are often a vivid example of syntactic ambiguity. Consider the famous headline “Teacher Strikes Idle Kids” that is understandable for a native speaker of English, but can be easily "misunderstood" by a computer software; correctly identifying the subject and the predicate of this sentence would require more than knowing basic grammar rules of English. In a free word order language, such as Armenian, it would be tricky to identify «Արամը բացեց դուռը» and «Դուռը բացեց Արամը» as two sentences with the exact same syntactic structure.

4.2. Parsing and Grammar theories

Different approaches in NLP to decipher the syntactic structure of a sentence trace their origins back to grammar theories. On a superficial level these approaches might seem quite similar, but on a deeper level they are based on quite different ideas about what language is and how it conveys meaning (Bolshakov and Gelbukh, 2004, p. 42).

Grammar theories try to give an “objective” description of language. The earliest linguists concerned with this problem in the 1920s, the Structuralists, assumed that word grouping and word order were key in decoding the syntax of a sentence. Structuralists mainly focused on English and their approaches didn’t work for many other languages, especially languages with free word-order. Starting from the 1950s Noam Chomsky’s ideas started to play an important role in linguistics. Chomsky considered language as a finite set of signs and rules that can generate an infinite set of strings. His initial model, generative grammar, was the first purely mathematical approach to language. An example of a generative grammar of a hypothetical language can serve as an illustration.

S = [NP, VP]

VP = [V, NP]

NP = [A, N]

NP = [N]

This is the set of rules (grammar rules) that our hypothetical language has. It states that a sentence is constituted by noun phrases (NP) and verb phrases (VB). Verb phrases on their turn are constituted by verbs (V) and noun phrases. Finally, noun phrases can consist of either nouns (N) and adjectives (A) or only nouns. Our language has no other syntactic roles, such as adverbs.

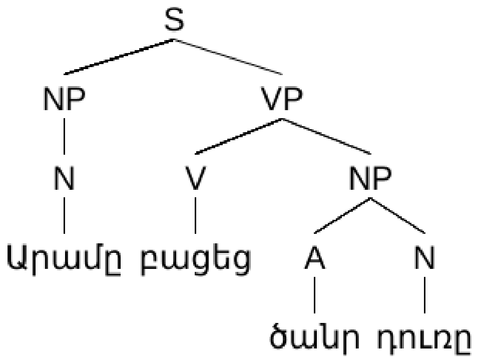
Subsequently, generative grammar defines a set of signs (i.e. words). Supposing that our language has only 6 words (which happen to be Armenian words), we could define this set as follows:

N = [Արամ, դուռ, ձանձրույթ]

V = [բացել, ակնկալել]

A = [ծանր]

The generative grammar that we created can then produce a number of sentences, such:



This is the *constituency tree* of the sentence «Արամը բացեց ծանր դուռը», we use this term because the syntactic structure is broken down from top to bottom: the sentence at the top is constituted by noun phrases and verb phrases, which in turn are constituted by other noun phrases, nouns, verbs, adjectives, adverbs, determiners and other *constituencies*.

Chomsky's model was a *universal grammar* model. It could be applied to any language. Although his model and its derivations are widely used in NLP, it has many shortcomings since it provides only a formalistic description of grammar and syntax. For instance, the generative grammar that we draw above can a nonsensical sentence as «Ծանր ձանձրույթը ակնկալեց Արամը».

Chomsky later proposed another approach, transformational grammar, that solved some of these problems, but was predominantly English-oriented and still assumed fixed word-order (Bolshakov et al, p. 45). Overall, Chomsky's grammar models are lack a full explanatory capacity. The concept of language as finite sets of signs rules was contended by other linguists who considered it more likely that language allowed *infinite* rules. On the other hand, Chomsky's grammar models were perfectly applicable to artificial languages and played an important role in the formulation of programming languages.

Dependency Grammar was formulated by the French linguist Lucien Tesnière starting from the late 1930s and completed in the 1940s. In a nutshell, this theory states that all words in a sentence depend on some other words and only one word, the *head* is on the hierarchic top. Tesnière's model was brought into computer science by David Hays who pioneered in machine translation in the 1950s and who in fact coined the term *Computational Linguistics*. Hays based his algorithms on Tesnière's grammar model and he created the first corpus of dependency trees with about 1 million Russian words.

In contrast to constituency grammar where the top of sentence is the sentence itself, in dependency grammar the top is defined as the word of which all other words have syntactic and semantic dependence. A dependency tree essentially neglects word order and focuses entirely on the syntactic and semantic relationships between its nodes. It would be a hasty conclusion to claim that dependency grammar is the most accurate model in explaining grammar. All existing grammar models are to a certain degree successful in describing certain aspects of grammar and none of them does that exhaustively (Ágel and Fischer, 2015, p. 234). However, as dependency grammar is more effective when it comes parsing of Armenian, a language with free word order, it will get a wider attention in this paper.

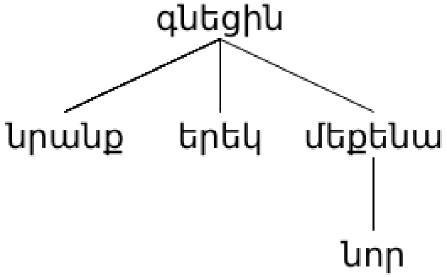
Dependency grammar is also the basis of Universal Dependencies, currently the largest multilingual NLP project with parsers and treebanks for about 50 languages. An initiative to add Armenian to this project started in late 2016.

4.3. Dependency grammar

Most of Tesnière’s ideas were formulated in his posthumously work *Éléments de syntaxe structurale* published in 1959, two years after Chomsky’s *Syntactic Structures* (Kahane and Osborne, 2015, p. xxix). Tesnière believed that he invented a universal grammar model which is applicable to any natural language, not only existing ones, but also historical languages. Generally speaking, dependency grammar disregards the traditional lexical and syntactic categories of words in favor of their syntactic *functions*. This principle comes especially handy for languages with free word-order. Consider, for instance, the Armenian sentence «Երեկ նրանք նոր մեքենա գնեցին» which may be rephrased in almost as many forms as there are words in it:

1. Նրանք երեկ նոր մեքենա գնեցին։
2. Նրանք նոր մեքենա գնեցին երեկ։
3. Երեկ նրանք գնեցին նոր մեքենա։
4. Նոր մեքենա գնեցին նրանք երեկ։

From the point of view of a human speaker who is proficient in Armenian it is obvious that these sentences are semantically identical (with slightly different emphases). While from the point of view of a computer software that is not so straightforward. However, because the dependency relations between the words are the same, the dependency model would allow us to recover the same semantic structure from all these sentences. The dependency tree of all five sentences would look as following:



A parsed sentence like this can be then used by a computer program. For instance, a machine translation system can translate the individual words into a second language, then restore then reconstructing the word order that is usual for that language.

The basic idea behind dependency grammar is to consider a sentence as a “small drama”. The most important feature of this drama the “event”, hence on the top of the hierarchic structure is always the verb. Other properties of the drama – the “actors” and the “circumstances” – all tell something about this event, therefore they are its *dependents* although they can also have dependents on their own. The relationships between words is thus always hierarchic. As Ágel and Fischer put it, “seen from the top, they are government relations, seen from the bottom they are dependency relations.” (Ágel and Fischer, 2015, p. 225).

A dependency tree, as the one above, illustrates the hierarchic relationships between words. This relationships are established following a few strict rules which are based on the syntactic roles of each words. In Tesnière’s original model, syntactic roles are limited to the following four:

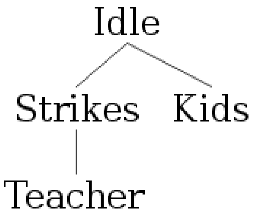
- verb
- noun
- adjective

- adverb

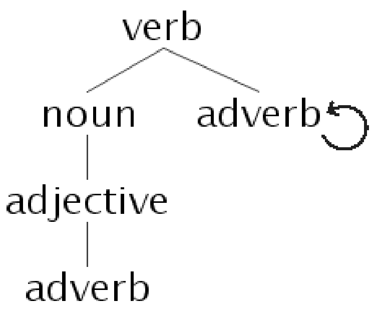
And they always obey the following four rules:

- a verb is always the head of the clause
- a noun is only a dependent of the verb
- an adjectives is only a dependent of a noun
- an adverb can be a dependent of a verb, an adjective or itself

It is important to note, that these syntactic functions do not coincide with the classical parts of speech. For instance, in the sentence “Teacher Strikes Idle Kids” the word “Teacher” would not be labeled as a *noun*, but as an *adjective* (since it tells something about the word “Strikes”). The words “Strikes” and “Kids” have both the syntactic role of a noun, because they complements the head of the sentence (which can only be complemented by nouns and adverbs). The dependency tree of this sentence would like as following:



The above mentioned four syntactic rules of dependency grammar predict a hierarchical semantic structure that, according to Tesnière, can be applied to any natural language, regardless of its grammar or lexical rules. Any sentence can be represented in the following dependency structure (Ágel and Fischer, 2015, p. 226):



It is important to note, that this structure represents only the backbone of the syntax of a language. For practical NLP tasks, more information about the morphological and syntactic categories of words are necessary. The Universal Dependencies, defines 40 dependency relationships between words in a sentence (Nivre e.a., 2016).

4.4. Valency theory

In addition of dependency grammar, Tesnière also formulated *Valency theory*, that predicts the dependency structure of a sentence from the perspective of its head. In short, valency defines how many and what kind of “actants” the verb is expected to rule over in the syntactic hierarchy (Bolshakov and Gelbukh, 2004, p. 39). Noteworthy, the term *Valency* comes from chemistry where it is used to describe the ability of a chemical element to combine with other atoms to form molecules. An example of valency annotation for the verb *ակնկալել* would be:

- *ակնկալել* N_1, N_2

This means that the *ակնկալել* has slots for two noun actants (subject and direct object). It would be very unusual to add a third actant, such as an adverb, to this verb. E.g. the sentence “Մենք անհամբեր ակնկալում ենք ձեր վերադարձը” would be incorrect. Furthermore, the sentence “Մենք ակնկալում ենք” would also be incorrect, since the two noun slots of ակնկալել are mandatory.

Conversely, the synonymous verb *սպասել* has slots for three actants and only one of them is mandatory. The sentences “Մենք անհամբեր սպասում ենք ձեր վերադարձին” and “Մենք սպասում ենք” are both be correct.

A comprehensive description of valency classes of Armenian verbs is formulated by Michael Daniel and Victoria Khurshudian (Daniel and Khurshudian, 2015).

4.5 Universal Dependency

Even when NLP tools are based on the same grammar model, the actual syntactic and morphological annotation scheme considerably varies in various language projects. This is partly due the different lexical and syntactic categories of different languages. A standard annotation scheme is a prerequisite for many extended NLP tasks such as multilingual translation or comparative linguistic studies. Universal Dependencies is a project that aims to create a universal annotation scheme for all languages. It combines earlier such efforts, including the Stanford Dependency and the Google dependency scheme (Nivre e.a., 2016). The Universal Dependency annotation scheme consists of 16 POS-tags and 40 dependency relations. In addition to these, language-specific tags can also be used.

Universal Dependency currently includes *treebanks* for more than 50 languages. *Treebanks* are databases with a large number of sentences that are annotated as a dependency tree and where each word is labeled with its morphological and lexical classifications. Treebanks are used to “train” specific *neural networks* which are then able to parse sentences in that language. The larger the treebank, the higher accuracy of these neural networks will be.

4.6 Armenian Treebank and Dependency Parser

Armenian Treebank and Dependency Parser (ATDP) is an initiative by Marat Yavrumyan, a Yerevan State University professor, Hrant Khachatryan, a Yerevan State University aspirant, to develop a fully functional set of core NLP tools for the Armenian language following the Universal Dependency guidelines.

ATDP is the first Armenian NLP that will be fully open-source under a free non-commercial license (Yavrumyan, 2017 interview). During the first phase of this project a large number of Armenian sentences will be manually annotated by linguists to become an Armenian dependency treebank. In the second phase a neural network will analyze these *treebank* and learn to predict the dependency structure of new Armenian sentences. The accuracy level of this predictive parsing will be only tested at the end of the project. A possible continuation of the project will expand the treebank and thus also improve the accuracy of parsing.

Currently the ATDP corpus includes more than 5,000 sentences with more than 100,000 words. These sentences are citations from university textbooks, literary works (prose only) and online media. The ATDP team was at the time of writing this paper working on the tokenization of this raw text material. Tokenization is mostly done automatically: a computer program searches for separators (empty spaces, punctuations and hyphens) within the text to identify sentences, phrases and words. Problematic in this stage is the tendency of Armenian to combine almost any two words into a compound word (e.g.

գիշեր֊ցերեկ, ծանր֊ծանր, մանր֊մունր, սուս֊փուս, քշել֊տանել, ասել֊խոսել). When treated separately these words either get a different meaning or even become nonsensical.

Once tokenization is completed, lemmatisation and POS-tagging tools will be developed by the engineering team. The linguistic team will manually add syntactic annotations to all 5,000 sentences. And finally a neural network will be “trained” on this treebank.

The project is expected to be completed by January 2018.

5. Limitations and Avenues for Future Research

This paper is an initial overview of Armenian in computational linguistics, however it only touches the tip of the iceberg. This paper will mainly serve those who are not familiar with Armenian NLP, but will not provide many new insights to those who are already in the field. Many Armenian NLP projects are either mentioned briefly or not mentioned here at all. A comprehensive research on the topic could contain detailed description of the algorithms that have been used, exemplify more extensively on the peculiarities of the Armenian language and propose recommendations for future strategies. Future studies that address specific NLP tasks and problems would be of much help for researchers and engineers dealing with Armenian NLP.

6. References

- Robert H. Hewsen (2001). *Armenia: A Historical Atlas*. University of Chicago Press. ISBN 978-0-226-33228-4.

- Laura Baghdasaryan et al (2013). *Facebook in Armenia: Users and Using.* OSCE. Retrieved from: https://www.osce.org/yerevan/108535?download=true

- ACL Anthology: A Digital Archive of Research Papers in Computational Linguistics. Retrieved from: https://www.aclweb.org/anthology/

- Vilmos Ágel, Klaus Fischer (2015). *Dependency Grammar And Valency Theory. In The Oxford Handbook of Linguistic Analysis (pp. 223-255)*. Oxford University Press. Retrieved from: https://www.uni-kassel.de/.../Dependency_Grammar_and_Valency_Theory_2010.pdf

- Lilit Arakelyan (2016). *A Lexical Database: Construction and Implementation. Master's Thesis*. Yerevan State University.

- H. Atcharyan (2005). *Universal Grammar of Armenian in Comparison with 562 Languages.* Yerevan University Press. ISBN 5-8084-0697-8. Retrieved from: http://www.armenianlanguage.am/images/menus/387/acharyan.pdf

- Jasmine Dum-Tragut (2009). *Armenian: Modern Eastern Armenian*. John Benjamins Publishing Company.

- Ralph Debusmann (January 2000). *An Introduction to Dependency Grammar. Universität des Saarlandes*. Retrieved from: https://pdfs.semanticscholar.org/b9cc/6e5f82d30162f9f1857e77c3045542fcf0d3.pdf

- Karen Sparck Jones, Julia R. Galliers (1995). *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer-Verlag Berlin Heidelberg

- Anna Copestake (2002). *Natural Language Processing: 8 Lectures*. Cambridge University. Retrieved from: https://www.cl.cam.ac.uk/teaching/2002/NatLangProc/nlp1-4.pdf

- Ewan Klein, Steven Bird, Edward Loper (2009). *Natural Language Processing with Python*. O'Reilly Media.

- Slav Petrov, e.a. (May 2012). *A Universal Part-of-Speech Tagset. European Language Resources Association (ELRA)*. Retrieved from: http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf

- Joakim Nivre, e.a. (2016). *Universal Dependencies v1: A Multilingual Treebank Collection*. In Proceedings of LREC. Retrieved from: http://www.lrec-conf.org/proceedings/lrec2016/pdf/348_Paper.pdf

- Gideon Lewis-Kraus (14 December 2016). *The Great A.I. Awakening.* The New York Times. Retrieved from: https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html

- Igor A. Bolshakov and Alexander Gelbukh (2004). *Computational Linguistics. Models, Resources, Applications*. Instituto Politecnico Nacional, Mexico City. Retrieved from: http://www.gelbukh.com/clbook/Computational-Linguistics.pdf

- Michael Daniel, Victoria Khurshudian (2015). *Valency classes in Eastern Armenian*, in Valency Classes in the World's Languages, edited by Andrej Malchukov e.a. De Gruyter.