Morphosyntactic Parser and Textual Corpora: Processing Uncommon Phenomena of Tibetan Language

Aleksei Dobrov
Saint-Petersburg State University
7-9 Universitetskaya emb.
Saint-Petersburg
+7 960 2321503
a.dobrov@spbu.ru

Anastasia Dobrova
LLC "AIIRE"
16 c.2 Korablestroiteley Street
Saint-Petersburg
+7 931 2240717
adobrova@aiire.org

Pavel Grokhovskiy
Saint-Petersburg State University
7-9 Universitetskaya emb.
Saint-Petersburg
+7 812 3287732
p.grokhovskiy@spbu.ru

Nikolay Soms
LLC "AIIRE"
16 c.2 Korablestroiteley Street
Saint-Petersburg
+79112514490
nsoms@aiire.org

ABSTRACT

This article analyzes the problems of parsing texts with linguistic phenomena of controversial nature which may rarely be encountered in NLP projects focusing on Indo-European languages, but are quite frequent in other languages, e.g. in the corpus of Tibetan Indigenous Grammatical Treatises, therefore, parsing texts with such phenomena is necessary for completeness of automatic morphosyntactic annotation of textual corpora. Development of the morphosyntactic analyzer for the Tibetan language started in 2016 and had already proved to be quite useful to deal with specific phenomena of Tibetan, and with previously unsolvable issues of tokenization. The ultimate goal of the project is to create a consistent formal grammatical description (formal grammar) of the Tibetan language, including all grammar levels of the language system from morphosyntax (syntactics of morphemes) to the syntax of composite sentences and supra-phrasal entities. The previously published version of the automatic morphosyntactic annotation was created on the basis of morphologically tagged corpora of Tibetan texts and had high, but not 100 percent coverage (the ratio of the amount of atoms covered by parse trees to the total amount of atoms), precision and recall. This article describes the problems that had to be solved after that, in order to develop the current version of the morphosyntactic parser which allowed to achieve complete and correct automatic annotation of the corpus, and the chosen ways of solving them, which allowed obtaining a complete morphosyntactic annotation of units previously treated as tokens (lexical tokens, words or other atomic parse elements), but required a substantial refactoring (restructuring existing code without changing its external behavior) of the formal grammar. Thus, not only the frequent, but all the constructions turned out to be important in the construction of the formal model.

CCS Concepts

• CCS → Computing methodologies → Artificial intelligence → Natural language processing.

IMS2017, June 21–24, 2017, Saint Petersburg, Russian Federation

ACM ISBN 978-1-4503-5437-0/17/06…$15.00
https://doi.org/10.1145/3143699.3143719

Keywords

corpus linguistics; Tibetan language; morphosyntactic analyzer; tokenization; immediate constituents; formal grammar; natural language processing

1. INTRODUCTION

The term 'morphosyntactic parsing' does not have a generally accepted definition in the scientific literature. In a number of publications (cf. [2], [16], [19], etc.), any analyzer that performs both morphological and syntactic analysis even separately, is called a morphosyntactic parser. Even a simple morphological tagger that assigns syntactic classes to word forms is called a morphosyntactic parser in some publications (cf. [14], [17], [4], etc.). In this article, the term 'morphosyntactic parser' means a parser that analyzes morphosyntax as a single level (not morphology or syntax separately) and tries to parse the morphosyntactic structure of the input text, that is, a single structure of immediate constituents that combines the internal structures of word forms, phrases and sentences. Morphological tagging for the Tibetan language is faced with the fact that there are no word or word form separators in Tibetan (like spaces in English), and there are no generally accepted solutions on word segmentation in Tibetan linguistics as well. Therefore, the proposed parser ignores the problem of word segmentation as such and is designed to build trees of immediate constituents not from word forms, but from morphemes (cf. [6]).

Theoretically, the principal indeterminacy of word segmentation in typological linguistics is vividly shown in [10] by M. Haspelmath on the material of world languages, as well as the problematic nature of the border between morphology and syntax. For Tibetan, which (like Chinese and many other syllabic languages) has not developed a traditional graphic segmentation of text into “words”, there is also no practical reason to make any ad hoc decisions to introduce any such segmentation.

The Tibetan language is currently spoken in parts of China, India, Nepal, Bhutan, and Pakistan as well as in diasporas worldwide by circa 4-6 million people. Genetically it is related to Chinese, Burmese and more than two hundred small Tibeto-Burman languages spoken in South and East Asia, many of them still unwritten. Typologically Tibetan combines features of isolating (amorphous) and agglutinative languages. Morphologically, real inflexion is only present in the system of verbal mood and tense categories (although these categories are expressed not by flexions, but rather by allomorphs of verbal roots, and verbal roots only sometimes have different allomorphs for different tenses and moods), generating not only finite verbal phrases, but also nominalizations that function both as attributes (like participles) and as noun phrases (like masdars), and converbs as well. Stable combinations of morphemes that are similar to words are being actively produced either by compounding (most often verb and noun roots), or by affixation, sometimes also combining these two techniques of derivation.

In order to develop a morphosyntactic analyzer of Tibetan texts, it is necessary to create a formal grammar, which includes all levels of the grammatical system of the Tibetan language from morphosyntax (syntactics of morphemes) to the syntax of sentences and supra-phrasal units.

A few currently available studies of the Tibetan language analyze mainly Tibetan morphology, the only notable exception being "The Classical Tibetan Language" by Stephan Beyer [3], which also includes an extensive presentation of Tibetan syntax. Still, this work does not fully describe the Tibetan system of syntactic units and often has a speculative character, since the conclusions are not supported by textual corpora.

The common problem of formal grammar development for less-resourced languages like Tibetan is that, in order to create an adequate formal model, a representative corpus of texts with a reliable annotation must be created first, but, in order to annotate a corpus, a formal model must already exist, to form the basis of annotation.

In this project, a new approach was adopted, which involves simultaneous development of formal grammar, dictionaries and corpora annotation. To implement this approach, a new corpus manager has been developed, which allows not only to work with morphosyntactic annotation instead of a simple morphological markup of texts, but also to debug all unrecognized units in the annotation, all breaks of syntactic trees, tree overlaps and cases of ambiguous parsing, leading to combinatorial explosions.

At the previous stage of the work, a formal grammar was created for the AIIRE linguistic processor (Artificial Intelligence Information Retrieval Engine), which covered the majority of the phenomena in the corpus and was relatively successful with ambiguity [6], and the texts annotated with it were loaded into the corpus manager. Further work consisted in eliminating the errors of the annotation found by the corpus manager by improving and, in some cases, substantially refactoring the formal grammar and dictionaries.

The uncommon phenomena of the Tibetan language dealt with in the present paper turned not to occur in the first place when developing a formal model, so the interpretation of them looked rather controversial. This article is devoted to the description and justification of those technical solutions that were adopted to solve these problems, to parse and annotate these units correctly.

The project uses an innovative approach to syntactic analysis, combining the immediate constituents structure (CS) and the dependency structure (DS) in an explicit way (with explicit distinction between heads, modifiers, and arguments according with object-oriented paradigm, unlike X'-theory, GPSG, HPSG or LFG, where subordinate phrases are only treated as arguments), and providing the possibility of non-projective linear orders of constituents [5]. A combination of CS and DS with explicit indication of dependencies was proposed in [7] for the first time, but no available mathematical model existed to allow its implementation in an algorithm. This study takes advantage of the AIIRE linguistic processor, which is one of successful computer implementations of combined CS and DS analysis [5]. Still, in order to be applied to the Tibetan language, it requires a new research on Tibetan syntax.

2. THE PROJECT'S CORPORA RESOURCES AND SOFTWARE

The databases available to the project at the moment it started (January 2016) comprised two corpora of the Tibetan language developed by the same team at the Saint-Petersburg University. The Basic Corpus of the Tibetan Classical Language includes texts in a variety of classical Tibetan literary genres. The Corpus of Indigenous Tibetan Grammar Treatises consists of the most influential grammar works, the earliest of them proposedly dating back to 7th-8th centuries. Both corpora are provided with metadata and morphological annotation.

The corpora comprise 34,000 and 48,000 tokens, respectively. Tibetan texts are represented both in a Tibetan Unicode script and in a standard Latin (Wylie) transliteration [8].

The formal model being developed should, ideally, cover both corpora; the corpus of classical Tibetan formed the basis of formal grammar at the initial stage, and at the present stage the main work was concentrated on the corpus of grammatical treatises, with the goal of achieving 100% coverage of its material.

The parallel Tibetan–Russian corpus of grammar treatises reflects the Tibetan grammatical tradition which presumably started to form in the 7–8th centuries AD with the creation of the first grammars *Sum cu pa* and *Rtags kyi 'jug pa*. This tradition is based largely on Buddhist grammars that were compiled by Indian scholars; thus, it goes back to the Indian grammatical tradition. In terms of methods of description and analysis of language phenomena these grammars significantly differ from Western linguistics. Modern Tibetan linguists continue to maintain and develop the tradition of the classical Tibetan linguistics.

The corpus includes grammar treatises, which are considered to be the most important grammatical works within the Tibetan grammatical tradition: (1) the first two treatises *The Thirty Verses* (Tib. Sum cu pa, presumably 7–9th centuries AD) and *The Application of Signs* (Tib. Rtags kyi 'jug pa, presumably 7–9th centuries AD), whose authorship is traditionally attributed to the creator of the Tibetan script, Thonmi Sambhota (Tib. thon mi sam b+hota), (2) the grammar by Smṛtijñakrti "The Speech Door, [Similar to] the Sword" (Tib. smra sgo mtshon cha, 11th century AD), (3) *A Beautiful String of Pearls: Necklace of the Wise* (Tib. mkhas pa'i mgul rgyan mu tig phreng mdzes, 18th century), the commentary by Situ Mahapaṇḍita (Tib. si tu mahA paN + Di ta) to the first two treatises, (4) *Verbal Instructions on the Work of the Great Scholar Situ* (Tib. mkhas mchog si tu'i zhal lung, 19th century), the commentary by Ngulchu Dharmabhadra (Tib. dngul chu dharma b+ha dra) to the first two treatises, (5) *Jewel Necklace of Fine Explanations* (Tib. sum rtags gzhung mchan legs bshad nor bu'i phreng ba, 18/19th century), the anonymous commentary to the first two treatises, and (6) *Clear Mirror: An Introduction to Tibetan Grammar* (Tib. bod kyi brda sprod rig pa'i khrid rgyun rab gsal me long, 20th century) the Tibetan grammar by Skal bzang 'gyur med.

These corpora were chosen for the project not only because they were free and open-source and developed by the same research team, but also because, at the moment when the project started, these were the only two Tibetan corpora that (1) had reliable (manually revised) morphological annotation and (2) were publicly available on the web. There are some published papers that present some future work on corpus development or some corpora that are intended to be published only in future (e.g., [13], [15], etc.), but none of these studies has led to any existing corpus by 2016, when this project started; there are also papers that present ‘corpora’ without any annotation (to be more precise, ‘textual collections’ seems to be a better term for resources of this kind, cf. [13] again, [9], etc.), that did not suit our project’s purposes.

The free open-source AIIRE linguistic processor (http://aiire.org) is used for the project. AIIRE implements the method of inter-level interaction, first proposed by G. Tseitin in 1985 [18], which ensures the effective rule-based ambiguity resolution. The principle of inter-level interaction helps to minimize the problem of combinatorial explosion, which is very important for NLP software. The formal grammar analysis produces a considerable rate of ambiguity, especially when ellipsis is possible. The principle of inter-level interaction, implemented in the AIIRE linguistic processor, allows applying upper-level constraints to lower-level ambiguity, and thus reduces the number of produced combinations.

The architecture of AIIRE and the developed algorithms of text analysis allow to apply this technology to languages of different types in the form of independent language modules, while the analysis algorithms are independent of the language. Besides the modules for the Russian language, modules for Arabic and Abkhaz languages were previously created, and the present project aims at developing a module for the Tibetan language, which is well known for the absence of formally marked word boundaries and ambiguity of word segmentation as such.

To operate with the corpora, the corpus manager is developed within the framework of this project, which allows: 1) analyzing syntactic (morphosyntactic) annotation and 2) finding the locations in this annotation that require improvement of the grammar and the dictionaries.

The corpus manager allows uploading plain texts or texts in "vertical" format for their further automatic annotation, reflecting the structure of the immediate constituents and the dependency structure, while units previously considered as tokens are divided into atomic units, which are then integrated into tree structures. Atomic units, or *atoms*, are minimal indivisible meaningful character sequences; atoms are not limited to allomorphs, they also include punctuation signs, digits, and also special Tibetan intersyllabic delimiters - upper dots (Tib. tshegs), like in (3) and (6). Tibetan is a syllabic language, all syllables are toned phonetically and separated by tshegs graphically. The units previously treated as tokens often contain several atoms of different types (e.g., allomorph, intersyllabic delimiter, and another allomorph), and the tree structures include atoms of different types, regardless of whether they are allomorphs, or just delimiters, or punctuation signs, or digits.

The search engine, developed for the corpus manager, allows finding morphosyntactic structures according to given models. Search is currently available for Tibetan inflection forms and word-formation models, however, the same search can be implemented for syntactic structures of any complexity (in Russian corpora, sentences and texts are also syntactically annotated, and the extension of the Tibetan corpus annotation is planned in the nearest future). As search is carried out not by words, but by morphosyntactic trees, search results are presented as a list of fragments of syntactic structures (morphosyntactic trees with grammatical characteristics and morphemic content).

The corpus manager includes support for corpora in various languages and includes a number of tools for automatic annotation of the corpus and detecting fragments of this annotation, that require improvements in linguistic data (annotation errors).

The corpus manager allows viewing the annotation of fully annotated text fragments. For partially annotated fragments, three additional types of errors are displayed: unrecognized units, breaks in syntactic trees and overlaps thereof. Unrecognized fragments are the fragments for which there are no syntactic trees in the annotation; breaks are the positions in which the tree can not be bound with any of its neighbors; overlaps are fragments of text in which the syntactic trees overlap, not completely covering the text: a fragment covered by one tree includes the position of the beginning of the fragment covered by the next tree, but not the position of its end.

This tool allows to simultaneously work on the annotation of the corpus and to improve the formal model behind the linguistic data being used. This is, in some respect, a new approach to the development of formal grammar and dictionaries for the linguistic processor, ensuring the constant verification of the formal model and its conformity to the corpora material. Consistently eliminating unrecognized fragments, breaks in the annotation, tree overlaps, and combinatorial explosions by improving linguistic data, a developer eventually achieves not only the complete corpus annotation, but also such a state of the formal model in which it explains all the phenomena present in the corpus.

3. REPRESENTATION OF TIBETAN MORPHOSYNTACTIC STRUCTURES IN AIIRE

The linguistic processor needs to recognize all the relevant linguistic units in the input text. For inflectional languages the input units are easy to identify as word forms, separated by space, punctuation marks etc. It is not the case with the Tibetan language, as there are no universal symbols to segment the input string into words or morphemes.

The developed module for the Tibetan language performs the segmentation of the input string into atoms by using the Aho-Corasick algorithm (by Alfred V. Aho and Margaret J. Corasick, cf. [1]), that allows to find all possible substrings of the input string according to a given dictionary. The algorithm builds a tree, describing a finite state machine with terminal nodes corresponding to completed character strings of elements (in this case, morphemes) from the input dictionary.

The language module contains a dictionary of morphemes, which allows the machine to create this tree in advance at the build stage of the language module, while in the runtime of the linguistic processor the tree is being loaded as a component of an executable module which brings its initialization time to minimum.

Two special files were created in order to analyze Tibetan morphosyntactic structures: the grammarDefines.py file determines types of atoms (atomic units), their properties and restrictions, while the atoms.txt file (the allomorphs’ dictionary) specifies the morpheme, the token type, and properties for each allomorph, also in accordance with grammarDefines.py file. For example, the following entry in the allomorphs dictionary: དགའ|morpheme=དགའ|type=v_root|has_mood=False|has_tense=False|fin_phoneme=vowel indicates that the དགའ (dga') allomorph is the basic allomorph of the (1) morpheme, that is to indicate that this verb root has no mood and no tense properties and ends in a vowel.

(1) དགའ

dga’

‘be glad’

At the present stage, 45 different types of atoms have been identified. All these types of atoms have their morphological and morphophonemic features indicated in the grammarDefines.py file. For example, the verbal root has such potential properties indicated as the mood (indicative, imperative), the tense (present, past, future), the availability of tense category (true / false), the availability of mood category (true / false), and the type of final phonemes defining the compatibility of the verbal root with suffix allomorphs. The restrictions for the verbal root require that the category of tense is available only if the respective parameter "has_tense" is set to "true", and the parameter of "mood" is set to "indicative", and the same is true for ‘mood’ and ‘has_mood’ categories.

These types of tokens and their grammatical features form the basis of the formal grammar being developed (the `grammar.py’ module), allowing the linguistic processor to build syntactic treebanks of various structure.

4. INTERSYLLABIC DELIMITERS

In Tibetan writing system, there are no delimiters of word boundaries. The only graphematic delimiters that are used are markers of pauses (similar to our punctuation marks) and also syllabic delimiters (because Tibetan is a syllabic toned language). Statistically, a syllable corresponds to a morph most frequently (*2*), polysyllabic morphs are also rather common (*3*), as well as morphs consisting of only one consonant and not forming a syllable (*r* ‘terminative case marker’, like in (*4*)).

(2) མི

mi

‘man’

(3) ར་བ

ra_ba

‘fence’

(4) མིར

mi-r

man-TERM

‘into a man’

Tibetan intersyllabic delimiters (*tshegs* in Tibetan) proved to be one of the most difficult problems for modeling in formal grammar, although the rules of intersyllabic delimiters usage were pretty much formalized and seemed to be rather simple at first. The main problem was that these rules were stated in terms of linear compatibility of symbols of Tibetan script, whereas the formal grammar dealt only with the hierarchical compatibility of Tibetan allomorphs and punctuation atoms. As AIIRE works with the universal Aho-Corasick algorithm, which is followed with a universal syntax parser, it doesn’t allow any language-specific rules to be hard-coded into any part of the linguistic processor; language-specific rules can only be embedded into formal grammar as classes of its immediate constituents, but not as “if-then-else”-like statements in the language processor’s core.

At the first stage, *tshegs* were incorporated into allomorphs as their final symbols, so that both a variant with a *tsheg*, and a variant without it were included into the dictionary, thus producing a copy almost for each allomorph and increasing the volume of the dictionary by two. Not merely the dictionary size was unreasonably increased by this solution, but it also produced a significant amount of incorrect versions of analysis, because multiconsonant allomorphs were often treated as multisyllabic combinations of smaller allomorphs, which were only possible with *tshegs* between them. E.g., (5) was treated as (6). The problem here lies in the fact that the [a] vowel has no graphematic representation in the Tibetan script, so the fact that (5) is (5) and not (6) is caused by the following graphematic rules: 1) there can be only one vowel in a syllable; 2) [ny] can not be a postinitial consonant, while [m] can be a preinitial consonant; 3) preinitial and final consonants can not be followed by a vowel in Tibetan.

(5) མཉན

mnyan

listen_FUT

‘will listen’

(6) མ་ཉན

ma-nyan

mother-listen_PRS

‘listens to the mother’

Nevertheless, technically, it is not necessary to model these graphematic or morphophonemic rules in order to exclude this version of analysis as an illegitimate one. In order to prohibit this combination, it is enough to include *tshegs* into syntactic trees as constituents and to reflect it by grammatical features of the allomorphs.

Thus, there were two possible ways to solve this problem. The first one supposed treating *tshegs* as parts of allomorphs and, therefore, introducing features like ‘having final *tsheg*’ for the allomorphs. The second approach was to treat *tshegs* as separate atoms, including them into syntactic trees as separate immediate constituents.

The first approach proved to be inconsistent. Firstly, there is no reason to include only final *tshegs* into allomorphs, and not to include, e.g., starting *tshegs*. Secondly, even if final *tshegs* are still treated as parts of the allomorphs, the problem remains that *tshegs* are not merely added to allomorphs, but rather are used between them (to be more accurate, between syllabic ones). Therefore, *tshegs* are used when there are two allomorphs between them, but are not used when one of the allomorphs is ellipsed. So if, e.g., the second allomorph is ellipsed, then the first allomorph loses its tsheg, but there is no such a mechanism in the formal grammar, that could have allowed such a shift in grammatical features of allomorphs. Thirdly, the first approach didn’t allow to eliminate redundant and artificial variants of allomorphs from the dictionary, whereas the second approach allowed to do it.

Therefore, the second approach was chosen, and *tshegs* were included into the dictionary as separate atoms.

This caused a significant refactoring of the grammar, as from this point of view every class of immediate constituents had to be specified in respect to *tshegs*. The general solution was to add *tshegs* to specifier constituents, so that *tshegs* were added together with the specifiers like on Fig. 1.

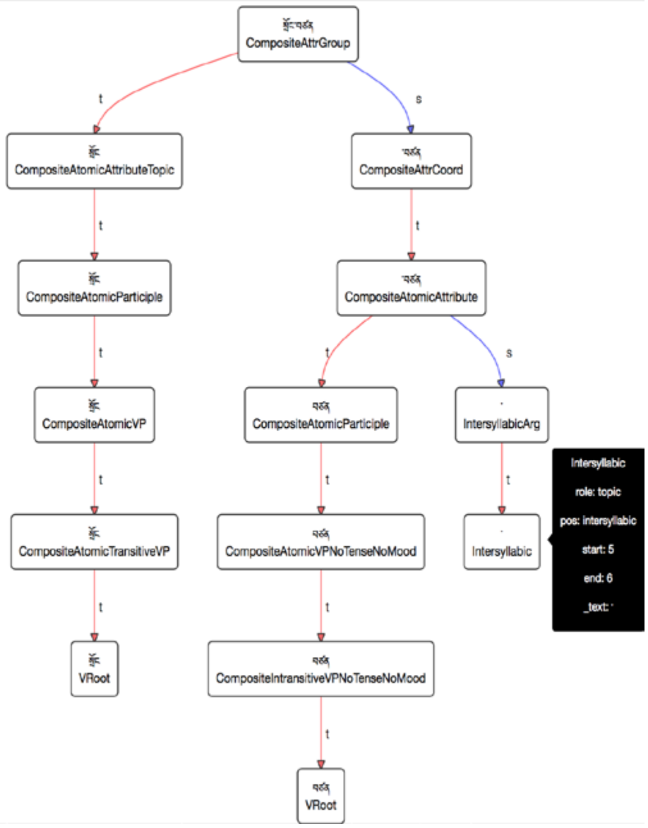

Figure 1. Intersyllabic delimiter in the morphosyntactic tree of a Tibetan compound སྲོང་བཙན (7)

The diagram on Fig. 1 depicts the structure of Tibetan compound (7).

(7) སྲོང་བཙན

srong-btsan

straighten-be_mighty

'One Who Straightens and Is Mighty' (name of Tibetan king)

The whole phrase on Fig. 1 is a CompositeAttrGroup (i.e., a Composite[1] Attributes Group, a group of superficially homogeneous attributes within a compound). This way of derivation is quite frequent for Tibetan personal names: the name consists of a set of epithets (attributes), without any explicit modified noun. As Kayne [11] and Johannessen [12] had shown, groups of homogeneous phrases are only superficially symmetric, and antisymmetric models like those proposed in X'-theory tend to better account for some phenomena like anaphora coreference in homogeneous phrases. Therefore, homogeneous phrases are treated as antisymmetric both in AIIRE grammars in general, and, in particular, in Tibetan grammar, including homogeneous parts of compounds. Thus, CompositeAttrGroup consists of CompositeAtomicAttributeTopic and CompositeAttrCoord, where the first part is the first attribute, and the second part is the rest of the second attribute attached as a specifier (a subordinate constituent). If there were other attributes, they would be attached in exactly the same way with CompositeAttrGroup self-embedding.

CompositeAtomicAttributeTopic stands for the main constituent ('topic') of an attribute group within a compound, represented by a single atom ('atomic'). The distinction between atomic and compound constituents of compounds had to be made due to the rule that a compound must have, at least, two atomic constituents, i.e., an atomic constituent itself can not form a compound.

CompositeAtomicParticiple stands for a nominalized verbal phrase that acts like a participle within a compound. Nominalized verbal phrases generally act both as masdars (noun phrases), denoting processes, and as participles (attributes), denoting properties, which is the case of homogeneous attributes within compounds. Generally, nominalized verbal phrases have a nominalizer, but nominalizers are elided, as well as case markers, in Tibetan compounds. Thus, participle can be atomic, but only within a composite.

CompositeAtomicVP means atomic verbal phrase within a compound. Generally, verbal phrases can attach circumstances, but, in this case, the verbal phrase has no specifiers. As in AIIRE in general, the distinction is made in Tibetan grammar between common (upper-level) verbal phrases that can attach only circumstances (with self-embedding allowed), and special (lower-level) verbal phrases (like transitive, dative, associative or terminative verbal phrases) that attach objects of different types. Objects tend to be attached lower in the tree than circumstances in many languages, including Tibetan, as the morpheme order shows. However, as the case markers are always elided in compounds, there is no formal difference between different types of objects within the compounds, and there are no circumstances in compounds at all. Technically, the CompositeAtomicVP constituent could be, therefore, skipped, but the decision was made to align the structures of compounds with that of free combinations where possible.

CompositeAtomicTransitiveVP stands for the phrase of a transitive verb root within a composite. Generally, this phrase attaches direct object (leftwards), but it can be ellipsed, like in (7): 'straightens' in general, with no indication of object of straightening needed.

VRoot means verb root; classes of atomic constituents have a one-to-one correspondence with types of atoms.

CompositeAttrCoord, in turn, stands for a coordinate attribute within a compound. As it is a specifier in this tree, it can only be a specifier anywhere else (this is one of the technical restrictions of AIIRE grammar), therefore, it can never be the main constituent, and an additional child constituent (CompositeAtomicAttribute in this case) has to be added.

CompositeAtomicVPNoTenseNoMood is a CompositeAtomicVP which does not have neither mood, nor tense, and CompositeIntransitiveVPNoTenseNoMood is a phrase of an intransitive verb root without tense and without mood within a compound. As it was already mentioned, Tibetan verb roots often do not have different allomorphs for differents moods and tenses, and it is the case in (7).

CompositeAtomicAttribute means an atomic attribute within a compound. Attributes are attached either to a noun phrase, or to a group of homogeneous attributes, and, as they always start with a syllable, they are always attached with an intersyllabic delimiter

[1] Here and further, the term 'composite' is used as an approximate synonym of 'compound'. The term 'compound' is not fully applicable to Tibetan for the above-mentioned reasons: this term means a *word* in Indo-European sense, but Tibetan complexes of morphemes are not exactly words. The term 'composite' has a more abstract meaning, but is rarely used as a noun in linguistics in English.

(tsheg). Atoms can only be main constituents in AIIRE grammar, so an additional IntersyllabicArg specifier has to be added between the Attribute and Intersyllabic constituents.

Table 1. Properties of intersyllabic delimiters according to atom types and subtypes

Type	Subtype	sep	prev_atom
v_suff	nom	False	sep
v_suff	semifinal	False	sep
v_suff	agent	False	sep
v_suff	coord	False	sep
v_suff	emph	False	any
punct	stop	True	text
punct	poetry_stop	True	text
punct	start	True	sep
p_ind_root		False	sep
n_suff	dim	False	any
p_pers_root		False	sep
case_marker_root	erg	False	any
case_marker_root	loc	False	sep
case_marker_root	gen	False	any
case_marker_root	term	False	any
case_marker_root	dat	False	sep
case_marker_root	ass	False	sep
case_marker_root	el	False	sep
case_marker_root	abl	False	sep
case_marker_root	comp	False	sep
num_suff		False	sep
v_root		False	sep
conj_root		False	sep
num_root		False	sep
p_def_root		False	sep
fin	narr	False	any
fin	inter	False	any
fin	command	False	sep
fin	informal	False	sep
fin	promise	False	sep
fin	warning	False	sep
fin	doubt	False	sep
posessive_root		False	sep
adj_root		False	sep
adv		False	sep
p_int_root		False	sep
quot		False	sep
square_bracket		True	any
round_bracket		True	any
angle_bracket		True	any
dig_root		True	any
whitespace		True	any

This allows to reflect the fact that *tsheg* is an entity that is situated exactly between two other entities, when they are both present. The problem with this solution, however, is that only specifier can be ellipsed together with *tsheg*, but not the head: if the head is ellipsed, then the specifier remains with *tsheg*, which is not right. Thus, separate classes of immediate constituents had to be created for all cases of possible head ellipsis.

This approach also supposed introducing new features to grammar definitions and to the dictionary of allomorphs.

Some atom types (especially subtypes) are definitely attached to previous atoms always with *tshegs*, some are always without *tshegs*, and some allow both ways of use. These atom types needed no special features to regulate *tshegs*, but there were different atom types, for which rules of *tshegs* usage proved to be atom-dependent: emphatic verb suffixes, diminutive noun suffixes, ergative, genitive, and terminative case markers, narrative and interrogative sentence ending markers, and some others. For these atom types, a new feature named 'prev_atom' was introduced, which can have three possible values: 'text', 'sep', and 'any', where 'text' means no intersyllabic delimiter, 'sep' means that the previous atom is a delimiter or bears its functions, and 'any' means that Tibetan grammar allows both types of usage.

The atom types themselves were also divided into two groups, according with another feature named 'sep', which is true for separators (delimiters and other atoms that have the function of delimiters), and false for all other atoms. The value of this feature is unambiguously deduced from the atom type, so, it is sufficient to create different rules for different atom types in the grammar, and it is not necessary to add this feature to grammar defines and to the atoms dictionary.

The values of the 'prev_atom' and 'sep' features for different atom types and subtypes are represented in Table 1.

The columns in Table 1 reflect, respectively, the atom type, its subtype (when applicable and necessary), the value of 'sep' feature (i.e., whether atoms of this type/subtype are separators), and the value of 'prev_atom' feature (i.e., whether the previous atom has to be a separator, a non-separating textual fragment, or any of them; in the latter case the value of the feature is individual for each allomorph.

5. SPECIFIC ATOM TYPES: NAMES, LETTERS, EXPONENTS, NONSENSE, AND ERRORS

Besides "typical" elementary constituents (morphs) some elementary constituents that are not so common and cannot be easily found in any given text in any given language, also occur in the Corpus of Grammatical Treatises. As language, speech and their elements are the subject of a grammatical treatise, it causes the so-called metalinguistic usage of linguistic means of expression. This is the cause why names of letters had to be included into the dictionary of allomorphs not only as graphematic counterparts of phonetic elements, but also as specific nouns with a narrower functional scope than full-fledged nouns. The same is true for allomorphs (mainly, grammatical affixes), as well as for exponents of possible graphemes combinations available for building existing allomorphs, but also about exponents of non-existent (erroneous) graphematic or morphological combinations.

Letters, exponents of Tibetan morphemes (metalinguistic usages of morpheme exponents in nominations of morphemes themselves in linguistic descriptions), names of different types, erroneous and nonsense usages were added to the dictionary of allomorphs as separate files with atoms and specifications of their features. As for letters, the decision to add them to the dictionary seemed natural, because there was no other way to represent them, but for names and, especially, for exponents this solution was not that obvious. Names, especially foreign names, can theoretically be of any kind, and, if the formal system is supposed to be universal (i.e., if it must be able to parse any name that is possible in any Tibetan text), then it may seem to be useless to list the names in a dictionary of any kind, and it may seem to be much more useful to create a system of rules that allow to determine arbitrary names in arbitrary texts. The problem with Tibetan, however, is that there are neither any word delimiters in Tibetan texts, nor any marks of personal names like upper case letters, so there are no simple deterministic rules for determining names in texts. Rules for determining names in Tibetan can be only contextually-dependent, strongly involving semantics, because of high ambiguity, but in order to build a working formal model of Tibetan semantics, a syntactically annotated corpus should be created first. Moreover, in this research, the purpose of creating a formal model of Tibetan grammar and a linguistic processor is not as ambitious as creating a universal tool of any kind; the main (and for now the only) purpose is to create a formal model that accounts for phenomena of a particular corpus of Tibetan texts, and a tool that is able to produce the complete and correct automatic annotation of this corpus. Therefore, it is normal for this tool to have all the names used in the corpus enlisted in the dictionary, and to fail with any other names, that are not included in the list. The same is true for the names of linguistic entities – exponents of Tibetan morphemes that are used metalinguistically in this particular corpus – and, moreover, the same is true for erroneous, or even nonsense usages, used as examples of incorrect Tibetan in particular texts. Thus, all these types of entities were included in the dictionary separately; the rules of determining them automatically and universally with respect to contextual semantics being an issue of further research, that will be possible only after finishing this one.

Atom types ‘n_foreign’ (borrowed noun), ‘pers_name’ (personal name), ‘pers_name_foreign’ (personal name borrowed from a foreign language), ‘letter’ (letter nomination), ‘exponent’ (metalinguistic usage of morpheme exponent), ‘place_name’ (toponym), ‘place_name_foreign’ (borrowed toponym), ‘text_foreign’ (a whole phrase or several phrases borrowed from a foreign language), ‘nonsense’ (nonsense usages) were added into grammarDefines.py. Attributes were specified for each new atom type: all of them were provided with the attribute ‘fin_phoneme’ (the type of final phoneme), nonsense atoms were also provided with the ‘only_in_composites’ attribute in order to distinguish between independent nonsense blocks and those used only as parts of compounds; n_foreign, pers_name_foreign, place_name_foreign were also provided, as noun roots, with ‘is_countable’ and ‘only_in_composites’ attributes.

All the new atom types were introduced in order to solve certain issues with automatic syntactic annotation of specific morpheme combinations in the corpus, mainly to get rid of the so-called gaps: the places where the parser could not bind two adjacent syntactic trees. Each type of the gaps proved to be a result of missing, insufficient or inadequate formal descriptions in the grammar, so new types of Tibetan morphosyntactic constructions were revealed, and new classes of immediate constituents (further – CICs) were created for them.

The CIC «Letter» was allowed to be the head component in CIC «CompositeClassNP» (signification nominal group, that may be expressed by several atoms, inside a compound). In this particular case, there are no grammatical requirements for head and specifier classes, but «Letter» has two additional attributes: «prev_atom» and «fin_phoneme». For example, in (8): *da* (‘letter d’) + *drag* (‘strong’) means ‘strong d’ (cf. Fig. 2), which is a compound from *da drag po.*

(8)

ད་དྲག

da-drag

letter_da-be_strong

‘strong d’

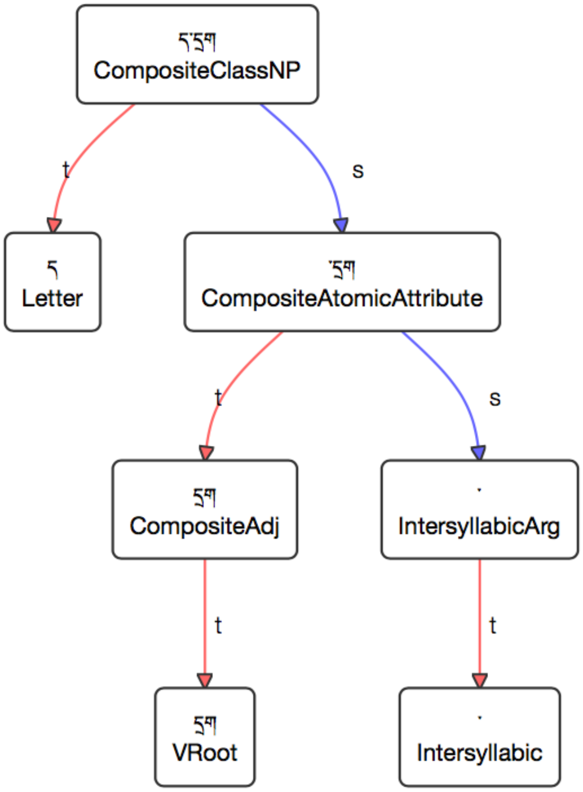

Fig. 2: Letter as a constituent within a composite class NP (8) ‘da drag’ (ད་དྲག)

CIC «PersNameForeign» (borrowed personal names) was allowed to be the head component in CIC «ClassNP» (Signification nominal group). There are no grammatical requirements for head and specifier classes, but «PersNameForeign» has two additional attributes: «prev_atom» and «fin_phoneme». For example, (9) is parsed as PersNameForeign, it is embedded into (10): (11) is a compound made of a contracted (composite only) form of n_foreign atom (12) plus a contracted adjective (13); so (10) means ‘Mati the Great Scholar’ (see Fig. 3)

(9) མ་ཏི

ma_ti

‘Mati’

(10) མ་ཏི་པཎ་ཆེན

ma_ti paN-chen

Mati scholar-great

‘Mati the Great Scholar’

(11) པཎ་ཆེན

paN-chen

scholar-great
‘great scholar’

(12) པཎྜི་ཏ
paN+Di_ta
‘scholar’

(13) ཆེན་པོ
chen-po
be_great-DER
‘great’ .

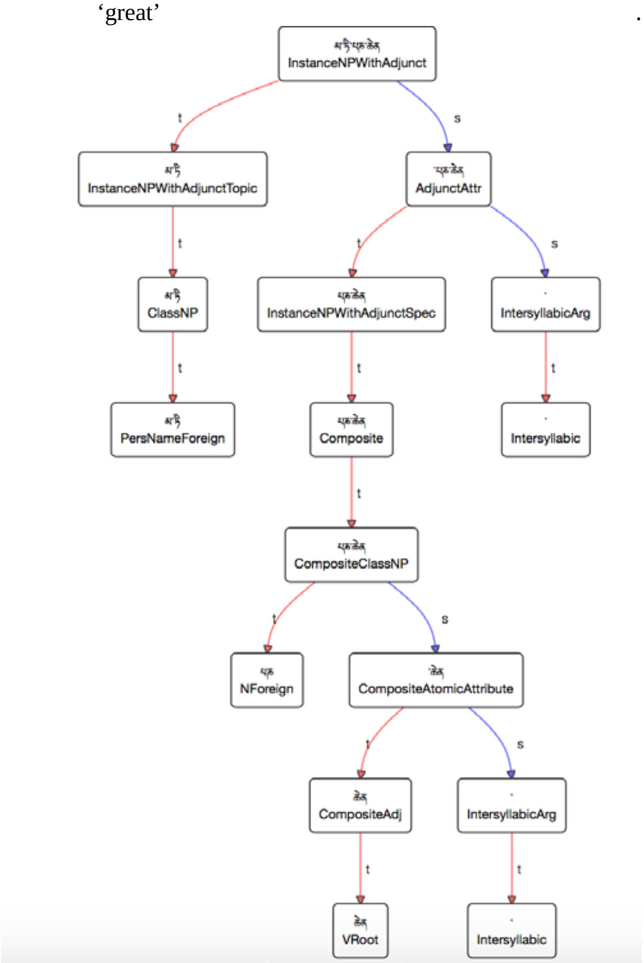

Fig. 3: Foreign personal name as a constituent of a class NP in a phrase (10) ‘ma ti paN chen’ (མ་ཏི་པཎ་ཆེན)

The CICs «NForeign» (foreign noun) and «PlaceNameForeign» (foreign toponym) were allowed to be heads in CIC «ClassNP» and also in compounds. For example (11) is a compound made of a contracted (composite only) form of n_foreign atom (12) plus contracted adjective (13) > (11) (cf. Fig. 4).

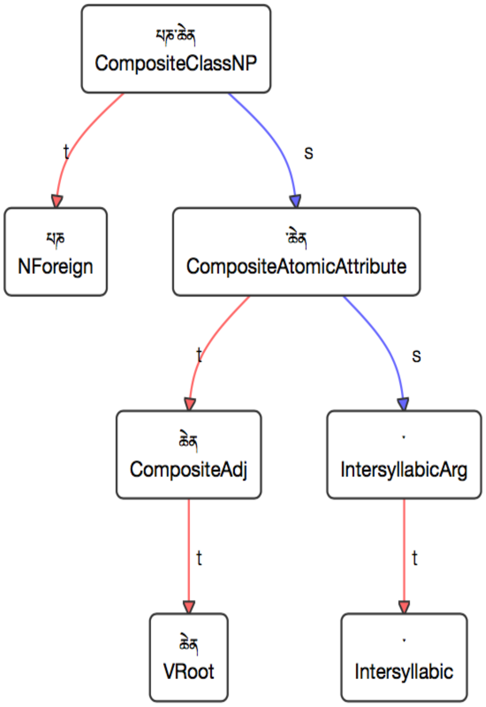

Fig. 4: Foreign noun as a constituent of a composite class NP in a phrase (11) ‘paN chen’ (པཎ་ཆེན)

The CIC «NamedEntityComposite» was made for combinations of letters or exponents of arbitrary Tibetan morphemes with «NRoot» (nominal roots). It was decided that the «Letter» or «Exponent» is the head component of «NamedEntityComposite». In this particular case there are no grammatical requirements for head and specifier classes, but ellipsis is prohibited for the both. The linear order of the specifier in relation to the head is right. The CIC is embedded into «ClassNP» CIC as a head component.

The «NamedEntityComposite» CIC is a class of named-entity nomination, where the name of the entity is a letter or an exponent of any Tibetan morpheme, e.g., (14) means (15) + (16), i.e., ‘the letter ‘*a*’ (cf. Fig. 5).

(14) འ་ཡིག
‘a-yig
‘a-letter
‘letter ‘*a*’

(15) འ
‘‘a’

(16) ཡིག
‘letter’

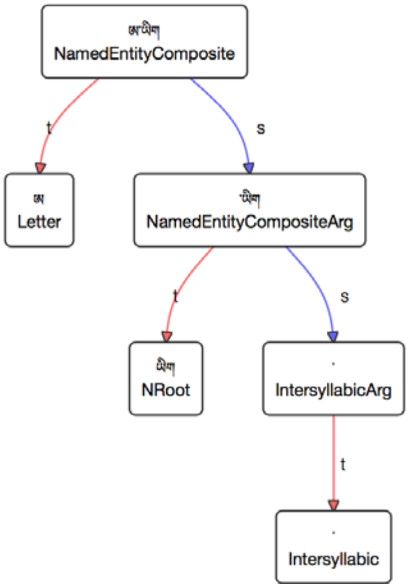

Fig. 5: Letter as a constituent in a named entity composite (14) 'a yig' (ཨ་ཡིག)

Foreign nouns (NForeign), foreign personal names (PersNameForeign), Tibetan toponyms (PlaceName) and foreign toponyms (PlaceNameForeign) were allowed to be heads of "PersonNP" – a nominal phrase with *-pa-*/*-ba-* suffix denoting a person by its relationship with an object. This solution made it possible to parse combinations like, e.g., (17) (someone who relates to Tsandra, cf. Fig. 6)

(17) ཙནྡྲ་པ

tsan+d+ra-pa

Tsandra-DER

'follower of Tsandra'

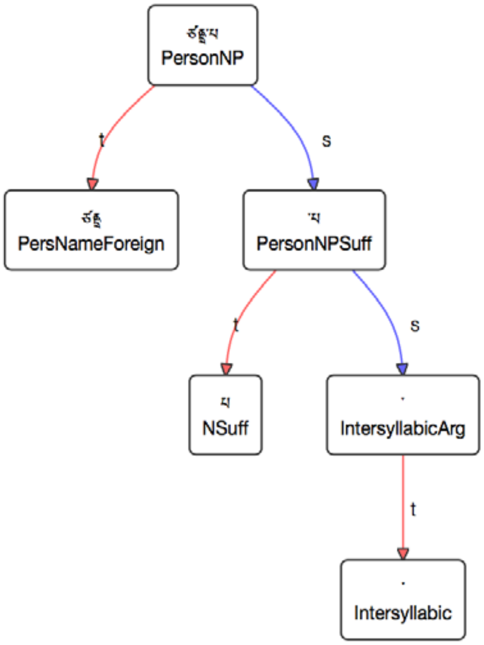

Fig. 6: Foreign personal name as a constituent in a person NP (17)'tsan+d+ra pa' (ཙནྡྲ་པ)

The CIC «ComparativeGroup» (comparative group that consists of nominal group and terminative comparative group) was created in the grammar for the general use of comparative groups. In order to parse special cases like (18), where (19) is a nonsense word, and (20) is a comparative marker plus a terminative case marker (meaning 'like kakhi'),

(18) ཀཁི་ལྟར

kakhi lta r

kakhi COMP TERM

'like *kakhi*'

(19) ཀཁི་

kakhi

'*kakhi*'

(20) ལྟར

lta r

COMP TERM

'like'

«NonsenseArg» CIC (nonsense word as an argument) was also created and added as a possible argument of «ComparativeGroup». Ellipsis is prohibited for the both components and the linear order of the specifier relatively to the head is left.

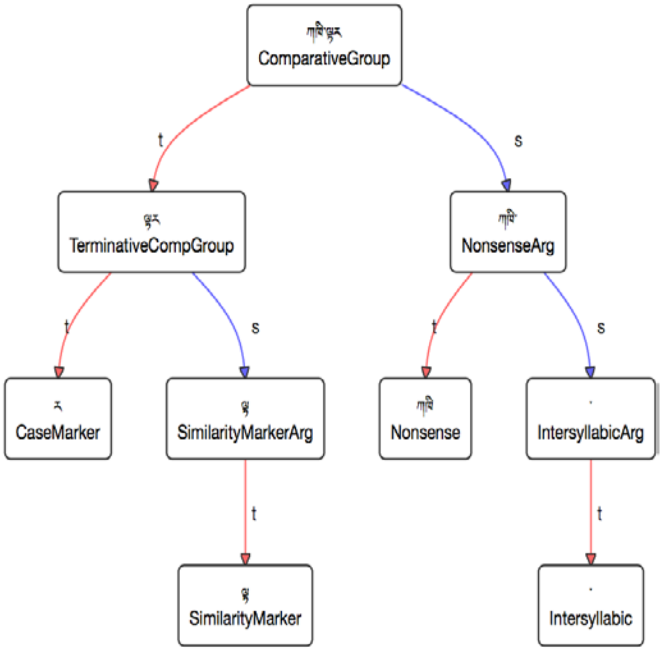

Fig. 7: Nonsense atom as a constituent in a comparative group (18) 'kakhi ltar'(ཀཁི་ལྟར)

6. CURRENT CORPUS ANNOTATION STATISTICS

The corpus of Tibetan Indigenous Grammatical Treatises within the scope of the units previously treated as tokens is now completely covered by the automatic morphosyntactic annotation generated with AIIRE natural language processor with use of developed grammar and dictionaries. The total amount of units, previously treated as tokens, is 48185 excluding *tshegs* between tokens. Only 33102 of these units proved to have 'atomic' interpretations; as the annotation contains different versions with different segmentations for ambiguous constructions, it is rather difficult to evaluate the amount of real atomic units in the corpus, but the first versions of token parsing, which have the least amount of atomic units, have 65030 atomic units for the corpus (again, excluding intersyllabic delimiters).

Table 2. Coverage of non-atomic constituents

CIC	weight sum	coverage >= 1%
ErgativeNPNoSep	3078.71	16.51
NPGenComposite	2446.80	13.12
Participle	1530.56	8.21
InstanceNPWithAdjunct	1342.46	7.20
Masdar	1027.01	5.51
PersonNP	974.54	5.23
TerminativeNPNoSep	718.75	3.85
CompositeClassNP	531.09	2.85
MasdarNoTenseNoMood	524.13	2.81
CompositeAttrGroup	490.63	2.63
Adjective	488.08	2.62
CompositeAtomicVP	435.00	2.33
Converb	424.03	2.27
TerminativeNP	401.67	2.15
OrdinalNumeral	376.55	2.02
CompositeAtomicVPNoTenseNoMood	374.40	2.01
AgentNP	222.43	1.19
TerminativeVPNoSep	221.50	1.19
NominalizedQuot	219.00	1.17
CompositeTransitiveVP	207.59	1.11
NamedEntityComposite	206.91	1.11

In order to count statistics on the CICs usage in the versions of morphosyntactic analysis in the annotation, the following approach was adopted. Distinct CICs are counted for distinct constituent locations (document identifier, left position, and right position), so that for each CIC, the weight of this CIC can be calculated per location as the ratio of 1.0 to the total amount of distinct CIC versions for this location. It means that, e.g, if there is only one CIC version for a specific location, then its weight equals 1.0 for this location, and if there are 4 versions of CIC for one location, then this CIC has a weight of 0.25 for this location. Thus, a sum of all weights of a CIC can be counted for all possible locations in the corpus, and the coverage (the ratio of the amount of atoms covered by parse trees to the total amount of atoms) of each CIC can be counted as the percentage of its weight sum from the total sum. The tables below show statistics on usages of classes of non-atomic and atomic constituents, respectively (only trees that cover the tokens completely are taken into account); frequencies below 1 percent are not shown here.

As it is shown in Table 2, the controversial and nonobvious CICs mentioned above are not relatively frequent: e.g., Named Entity Composites (compounds with exponents or letters) have only 1.11% of coverage.

However, as it is shown in Table 3, intersyllabic delimiters, although their interpretation was originally nonobvious, are the most frequent atomic constituents and cover 51.07% of all atomic constituent usages; exponents and letters have 3.82 and 1.52 percents, respectively, but are more frequent, than, e.g. numeral roots (1.22%) or indicative pronoun roots (1.02%). Personal names and loanwords (foreign nouns, names, foreign phrases) have below 1 percent of total corpus coverage, but they also caused a reasonable grammar refactoring.

Table 3. Coverage of atomic constituents

CIC	weight sum	coverage >= 1%
Intersyllabic	34544.00	51.07
NRoot	8215.88	12.15
Punct	5246.00	7.76
VRoot	4456.02	6.59
CaseMarker	3612.97	5.34
Exponent	2584.60	3.82
VSuff	1569.30	2.32
Letter	1029.13	1.52
Topicalizer	855.00	1.26
NumRoot	824.75	1.22
PIndRoot	692.17	1.02

7. CONCLUSION AND FURTHER WORK

Computational linguists dealing with Tibetan language data face new kinds of challenges which are characteristic of this language combining isolation with agglutination. It turns out that many traditional techniques and concepts are not directly applicable for this data, and new ways of text processing should be developed. The current result of this study is not only fully annotated (within the scope of the units previously treated as tokens) corpus of Tibetan Indigenous Grammatical Treatises, published at http://corpora.spbu.ru/corman/, but also a free open-source Tibetan formal grammar and electronic dictionary (http://svn.aiire.org/repos/tibet/trunk/aiire/lang) that completely cover morphosyntactic structures within the given corpus and work as a module of the AIIRE natural language processor.

Computer modeling of rare and controversial linguistic constructions turned out to be no less important for obtaining this result than previously published results relating to the modeling of the main, most frequent phenomena of the Tibetan language.

Further work will be aimed at the syntactic annotation of units that are larger than those that were previously treated as tokens: phrases, sentences and whole texts in the corpus, and at the simultaneous refinement of the formal grammar in order to ensure the completeness of this annotation.

As this study shows, however, when analyzing Tibetan sentences, the level of ambiguity is much higher, and, in order to work with them, due to the problem of combinatorial explosion, the introduction of semantic restrictions will be required, in connection with which it is planned to develop a module of syntactic semantics similar to the corresponding module already developed for Russian, and a computer ontology, modeling the concepts behind the meanings of the Tibetan morphemes and their idiomatic combinations, that reflects the Tibetan linguistic worldview.

8. ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research, Grant No. 16-06-00578 *Morphosyntactycal analyser of texts in the Tibetan language*.

9. REFERENCES

[1] Aho, A.V., and Corasick, M.J. 1975. Efficient string matching: An aid to bibliographic search. *Communications of the ACM*. 18, 6, 333–340.

[2] Atserias J. et al. 1998. Morphosyntactic analysis and parsing of unrestricted Spanish text. In *Proceedings of LREC'98*. Granada, Spain.

[3] Beyer, S. 1992. *The classical Tibetan language*. State University of New York, New York.

[4] Boudlal, A., Lakhouaja, A., Mazroui, A., and Meziane, A. 2011. Alkhalil Morpho Sys1: A Morphosyntactic analysis system for Arabic texts, In *ACIT'2010 Proceedings* (Riyadh, Saudi Arabia).

[5] Dobrov, A.V. 2014. *Automatic classification of news by means of syntactic semantics* [Avtomaticheskaja rubrikacija novostnyh soobshhenij sredstvami sintaksicheskoj semantiki], Doctoral Thesis. Saint-Petersburg State University,

[6] Dobrov, A., Dobrova, A., Grokhovskiy, P., Soms, N., and Zakharov, V. 2016. Morphosyntactic analyzer for the Tibetan language: aspects of structural ambiguity. In: *Lecture notes in computer science*, 9924, 215-222. DOI: 10.1007/978-3-319-45510-5_25

[7] Gladkii, A.V. 1985. Syntactic structures of natural language in automated communication systems [Sintaksicheskie struktury estestvennogo jazyka v avtomatizirovannyh sistemah obshhenija]. Nauka, Moscow.

[8] Grokhovskii, P.L., Zakharov, V.P., Smirnova, M.O, and Khokhlova, M.V. 2014. The corpus of works of the Tibetan grammatical tradition. *Automatic documentation and mathematical linguistics*, 49, 5, 182—191.

[9] Gui-Xian Xu, Chang-Zhi Wang, Li-Hui Wang, Yu-Hong Zhou, Wei-Kang Li, Hao Xu, Qing Huang. 2017. Semantic classification method for network Tibetan corpus. *Cluster computing*, 20, 1, March 2017, 155-165.

[10] Haspelmath, M. 2011. The indeterminacy of word segmentation and the nature of morphology and syntax. *Folia linguistica*. 45, 1, 31–80.

[11] Kayne, R.S. *The antisymmetry of syntax* — Cambridge, Mass: The MIT Press, 1994

[12] Johannessen, J.B. *Coordination* — Oxford, New York: Oxford University Press, 1998

[13] Liu H., Nuo M., Wu J., and He Y. 2012. Building large scale text corpus for Tibetan natural language processing by extracting text from web pages. In *Proceedings of the 10th Workshop on Asian language resources*, 11–20, COLING, Mumbai, Dec. 2012.

[14] Prószéky, G., and Kis, B. 1999. A unification-based approach to morpho-syntactic parsing of agglutinative and other (highly) inflectional languages. *In Proceedings of the 37th ACL, Association for computational linguistics*, 261–268.

[15] Qiu L., Long C., and Zhao X. 2012. A joint approach for building a large Tibetan corpus with syntactic parsing and semantic role labeling. 2012. In: *2012 Fifth International conference on intelligent networks and intelligent systems*, 1-3 Nov. 2012, 232-235.

[16] Rybka, R., Sboev, A., Moloshnikov, I. and Gudovskikh, D. 2015. Morpho-syntactic parsing based on neural networks and corpus data. In: *Artificial intelligence and natural language and information extraction, social media and web search FRUCT Conference* (AINL-ISMW FRUCT). IEEE, St. Petersburg, 89-95.

[17] Seara, I.C., Pacheco, F.S., and Kafka, S.G., et al. 2010. Morphosyntactic parser for Brazilian Portuguese: methodology for development and assessment. In*: 9th International conference on computational processing of Portuguese language (PROPOR 2010)*. 1–6.

[18] Tseitin G.S. 1985. *Programming in associative networks* [Programmirovanie na associativnyh setjah], Computers in designing and manufacturing [EVM v proektirovanii i proizvodstve] (2). Mashinostroenie, Leningrad. P. 16-48.

[19] Van den Bosch, A., Busser, B., Canisius, S., and Daelemans, W. 2007. An efficient memory-based morpho-syntactic tagger and parser for Dutch. In P. Dirix, I. Schuurman, V. Vandeghinste, & F. Van Eynde (Eds.), Computational linguistics in the Netherlands: Selected papers from the Seventeenth CLIN Meeting. Leuven. P. 99-114.