HAL open science

Techniques of Czech Language Lossless Text Compression

Jiří Ševčík, Jiří Dvorský

▶ To cite this version:

Jiří Ševčík, Jiří Dvorský. Techniques of Czech Language Lossless Text Compression. 15th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Sep 2016, Vilnius, Lithuania. pp.265-276, 10.1007/978-3-319-45378-1_24 . hal-01637512

HAL Id: hal-01637512

https://inria.hal.science/hal-01637512v1

Submitted on 17 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Techniques of Czech Language Lossless Text Compression

Jiří Ševčík and Jiří Dvorský

IT4Innovations, VŠB - Technical University of Ostrava,
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic
{jiri.sevcik, jiri.dvorsky}@vsb.cz

Abstract. For lossless data compression of the texts of natural language and for achieving better compression ratio we can use linguistic and grammatical properties extracted from the text analysis. This work deals with usage of word order, word categories and grammatical rules in sentences and sentence units in Czech language. Special grammatical properties of this language which are different from for example English language are used here. Further, there is an algorithm designed for searching similarities in analyzed sentence structures and its next processing to final compressed file. For analysis of the sentence units a special tool is used which allows parsing on more levels.

Keywords: lossless data compression, czech language, linguistic, morphology, graphs

1 Introduction

Within the compression we can encounter several data types. These types can be graphic data, sound data or a text in natural language. All these types have different statistic properties and dependences following from them. In case of the text compression we can point out that the character of given language has an influence on the compression itself. From this finding we can deduce that knowledge of the language structures and rules can influence the final compression ratio if we use it properly. Our work focuses on these properties. It is not a general system which processes any kind of language, but Czech language. On account of the fact that Czech language belongs to different category than English language, there will be mentioned examples in both languages for better understanding.

2 Language categories

Czech belongs to Slavic languages. These languages, similarly to English, belong to Indo-European languages. We can divide Czech language into two categories: standard Czech language which is used for official communication and common Czech language which is used in daily communication between people. In Czech

language we can find many dialects. Czech vocabulary is mainly Slavic, but it can adopt words from other languages, for example from German or Polish.

The number of language properties which are different for described categories is higher, but it is sufficient for basic understanding. Thanks to this understanding we can point out the fact that in case of inflected languages is the word arrangement looser than in the analytic languages. The word order is mainly influenced by semantic point of view of the whole sentence or by its grammatical arrangement. Knowledge of this property is one of the crucial features of our work and the whole principle of the data preprocessing for compression is built on this property. There are many branches that deal with grammatical arrangement and with natural language analysis on different levels and its importance and contribution to this work will be mentioned in following parts.

3 Previous work

We can find a large number of researches or works that deal with the compression which specializes on a particular language. The largest amount of them deals with English text or with Indo-European languages [1,2]. We can find also special compressing methods for different language categories such as Arabic [3] or Chinese [5] language. Since we are dealing with a compression of text only lossless principles will be mentioned. There exist also methods for loss compression with the informational content maintained but based on the aim of this work we will not deal with them.

In a number of these works we can encounter also preprocessing. It divides the compression into two sequential parts – preprocessing algorithm and compression algorithm. First algorithm transforms the input data for the compression based on its own compression scheme and these modified data he will deliver to the second algorithm which will, based on some standard or special methods, compress it. Decompression works just the opposite way. In the case of preprocessing algorithm we can encounter various general methods. The closest one to the topic of this work is the one on the level of words[4], symbols or sentential units. Another interesting method is the compression based on syllables [6,7].

4 Linguistics

The problem of compressing a natural language is therefore large and we can approach it with different perspectives. The aim of this work is to use some of these specific attributes and characteristics of Czech language and with the use of that attain a better compression ratio. Therefore we have to use possibilities of a science related to the language – linguistics. It represents a large group of sub-fields of study which deal with a large spectrum of language attributes from grammatical and syntactic parts to for example special usage related to the fields as sociology or psychology. We can divide it into general linguistics, applied linguistics or language linguistics. Further we can divide it based on its aims – in our case we will specialize in descriptive linguistics and theoretical linguistics. It,

based on Czech linguistic tradition [9], incorporates many subsystems because we can understand the language as a system. On theoretical level we can divide these subsystems into three categories: phonetics, grammar and syntax. Related to the aim of this work we will specialize on the second category which is divided into morphology and syntax.

Morphology we can describe as a discipline which deals with the study of the creation of words their declension and deriving of new words. Further we analyze the structure of particular words or whole word form. For the analysis of words we use morphematics. Basic morphological unit is an component which has a particular meaning – morpheme. If standing alone it may not be even a word. In the basic division morpheme may be lexical (word base) root of the word which represents action or characteristic or morpheme gramattical (*afix*) which defines grammatical categories (*sematics*) [9].

Another characteristic which morphology deals with is the study of grammatical categories which determine the meaning of particular morphemes. The most general category is a word class which is in the Czech language divided into ten types. Those can be divided based on the mentioned possibility if the *flex* (inflected words) may be used or not (inflexible words). To inflected words belong nouns, adjectives, pronouns, numerals and verbs and to Inflexible words are adverbs, prepositions, conjunctions, particles and interjection.

Morphological analysis lies on analyzing of a particular word form without taking the context into consideration. The output of this analysis are two components. One of them is the mentioned lemma and the second one is the morphological tag which comprises of 15 symbols (in special cases even 16). Each of these 15 positions are related to a particular morphological category and to every value of this category a particular symbol corresponds to [10]. In most of the cases it is a capital letter of alphabet or another alternative is a lower case letter. The last possible characteristic is "-" when used it means that for that particular word the morphological category is not applicable which is related to the rules of Czech language and grammar. A typical example is the case in verbs. The most important categories are (the number symbolizes the position in the tag) pos, gender, number, case, person, tense and grade.

The opposite procedure of analysis is the morphological synthesis. Here, in order to acquire a particular word we need to know its basic lemma form and morphological tag. There are various morphological taggers for the Czech language and each processes the text in a different way. Among the most well-known belong Free morphology tagger of Jan Hajič [13] and Ajka system[14]. Another category is the syntactic discipline. Its primary use in linguistics is to study the relationships between the words in a single clause or the relationships of individual clauses or sentences. It can be said that it is a study of the general sentential structure and its parts.

In European linguistics the dependency description is used to describe the structure of the sentence. Here, the verb is considered to be the central part of the clause or the sentence. The whole structure of the sentence is described in the form of tree where each node represents an individual constituent. The borders

between the nodes represent dependency relations and each node has only a single superordinate parent node. Here the verb is the root node. The nodes in the tree are arranged structurally according to their dependency relations and also linearly according to their sequence in the sentential word order.

5 Computer linguistics

The described linguistic disciplines are difficult and complex field and a "hand-made" analysis of the grammatical constituents or units is very demanding not only with the respect to time. Here the use of computers comes in the picture. Such processing is called Natural language processing. It is a field that links not only linguistics and informatics but also other fields such as acoustics in case of spoken word synthesis.

The corpus is the important term here. It is a structured set of texts/literature or even transcriptions of a spoken language. This set includes other information about the linguistic informations of the given text. Such information is crucial for us because they are syntactic and morphological. Various languages, including the Czech language, have many corpuses. For our task the Prague Dependency Treebank [11] is the most interesting. It has been developed since the middle of the 1990s in Institute of Formal and Applied Linguistics, Charles University in Prague and its actual version 3.0. comprises of several millions of entries. This corpus, besides other functions, includes two aforementioned levels, that is the *morphological layer* (m-level) and *analytical/syntactic layer* (a-level). A *tectogrammatical layer* is also included, however it is not crucial for our task. The actual data is stored in the structured format *PML* (Prague Markup language) which is based on the well-known marking XML and thus the computer data processing is made easier.

The resulting dependency tree includes the description of dependencies in the line descendant – parent between the individual constituents of the sentential whole. Such analytical tag is called afun and there are twenty-eight types of them. The second most important information included is the morphological part which records the lemma form to each grammatical constituent and a complete morphological tag. The last presented tool is *Treex* [12] which is capable of processing the text, creating its morphological and syntactic analysis and generate the resulting *PML* set. The tool can be configured, schemes and required resulting layers for the output adjusted and it also allows a reverse morphological synthesis. In total, twenty-three attributes are available, the majority of which is not important for our task (for example specialized linguistic categories of particular words) but we shall deal with them in the subsequent phases of development. One of them is for example *is_spaces_after* attribute which allows a detection of whitespace after given word and use of this information in the set of resulting trees.

6 Our work

The previous text proves that processing and analysis of the text can be done on the lowest levels even in the complex language such as Czech. It can be also said that despite the free word-order the clause or the grammatical constituent need to have a certain structure resulting from the given grammatical rules. The idea of our task, that deals with the use of the described analytical tool *Treex* as a part of the preprocessing in a compressive algorithm, is based on these premises. The progress of the preprocessing can summed up into several parts. The processing of the input text by analyser, finding similarities in tree structures while using the knowledge of grammatical rules and subsequent compression of the individual parts by compressive algorithms.

6.1 Processing of the text

Treex analyser is to be used for the analysis of the compressed text. The program can be used in a graphic mode via web program as well as a console application. After morphological and syntactic analysis is done, the output is transferred to the next part of the algorithm. Thus the analyser runs as the individual process and later a greater integration and interconnection with the preprocessing algorithm could be one of the subsequent parts of the development. But in the current phase the manner how the analysis runs is sufficient.

6.2 Similarities in tree structures

In informatics and in other fields of course, the graph theory offers countless of possibilities and thus there is no need, considering the length and the character of this paper, to deal with it in-depth. For our case, the searching for similarities in graphs is the most important. The search is to be done in two, mutually collaborating levels. The algorithm works in several phases:

1. **The initial phase** takes place during the actual processing of the input text in *Treex* analyser. Here we observe whether the afun *Coord* appears in the morphological analyses because here it serves as the coordinate function for grammatical constituents; Namely the symbols "*,*" and "*a*". In such case the whole tree is divisible into grammatical constituents. After the conclusion of this initializing phase the trees are divided in two sets (hereinafter referred to as A and B).
2. **The first similarity phase**: Here, and also in the following phase, the work is done by the general algorithm for similarity search in the graph. The manner and evaluation of similarities is to be discussed later. The algorithm first processes the set tagged as divisible i.e. A. In case there are more trees with necessary similarity, they are moved into resultant set C.
3. **The phase of division** – A division into sentential units is to be done for all the remaining trees from the set A. The division is made on the spot of the occurrence of the *Coord* afun. Thus two subtrees are created, to which

we assign the information about the linking coordinating function and which tree is their parent. The division into subtrees is done recursively because of the possibility of several *Coords* appearing in the single tree. After all subtrees are determined for the individual tree, they are moved into set B where all trees gained from the first phase can be found.

4. **The second similarity phase**. The same procedure as in the set A is used for the set B with the exception that all trees are to be moved into the resultant set.

As another possibility we have synthesis and modification of these phases in two which shall look as follows:

1. **The first phase** – after processing the text in *Treex*, the sentences where *Coord* afun occurs, will be divided into subtrees that will be saved into set together with all other trees that could not be divided.
2. **The second phase** – The similarity algorithm is applied on the whole set.

In the current phase of this work we cannot decidedly tell which procedure shall bear better results. It can be assumed that the processing of divided sentential units [8] will enhance the overall compression ratio but considering the small modification of preprocessing algorithm both alternatives shall be tested.

6.3 Comparing the similarity in graphs

The issue of comparing and defining the similarity of graphs is not unusual and there are many theoretical and practical works that deal with the topic. Thus there is plenty of research material and completed theoretical solutions. We are especially interested in the research group created under Czech Technical University in Prague. This group which consists of the members of well-known The Prague Stringology Club created a research group Arbology in 2008. Their work deals with the use of known algorithms from stringology for work with trees and tree-like structures[15]. The scope of their work is thus very large. In our work we can use the tree similarity on several levels which again reflect the variability of the Czech language and grammar.

Structural similarity A clause or a grammatical constituent can be regarded as a tree without further research of its attributes and dependency functions between the individual nodes.

Word-order similarity This similarity has already been mentioned – there are certain similarities to be found even in the free word-order.
Example: "Voda je studená." (Water is cold.), "Básničky se nám líbily." ("We liked a poems.") – subject "Voda" and "Básničky" and predicates "je studená" and "se nám líbily".

Morphological similarity The most frequent similarity can be found in the same grammatical case of the words.
Example: "Honzík a Anička se domluvili, že se odpoledne sejdou na hřišti."

("Honzik and Anicka agreed that they will meet on the playground on the afternoon."). Subject "Honzík" and subject "Anička" are grammatically connected and both are in nominative case. Preposition "se" and adverb "hřišti" are grammatically connected and both are in accusative case.

It can be seen that the similarity of the grammatical constituents can be found on several levels. Our goal is to create such an algorithm that will be capable of processing all input trees and create the smallest possible output set of trees for them on all similarity levels. The algorithm will be capable to determine whether it is suitable for particular clause to create a new tree if there is a similarity with the already existing output tree. Or whether an existing should be used considering that the original tree is modified and the modification is saved as the additional information. The modification will be most commonly used in addition or removal of the node in the tree or change of some morphological tags.

6.4 Tag compression

This idea of processing and subsequent compression seemed, after initial testing and several tests, very promising. However, we started encountering one significant problem with the speed of the text analysis using *Treex*. At first this did not seem as an obstacle but in case of several megabytes of input text the speed of analysis took tens of minutes. This duration was of course also caused by a non-optimized usage of the individual tool. After evaluating options and factoring that our development is at the beginning, it was decided that in the first phase of the development the morphological analysis shall be handled by a different tool called *MorphoDiTa*[18]. Another supportive argument for this choice was the fact that we needed to achieve some initial results that would allows as to verify that our input assumptions for compression based on the utilisation of knowledge of grammatical structure and rules of the Czech language are correct. This would confirm the assumption that the devised algorithm will be effectual. Therefore we decided, for the time being, to use a different tool.

Unlike *Treex*, the *MorphoDiTa* executes the text analysis only on the morphological level, ignoring the syntactic or tectogrammatical layer. Due to this focus, the tool is much faster and it can be also used for the reverse generating of the original word, if the lemma and morphological tag are known. There are more tools suitable for the operation, however *MorphoDiTa* was chosen for its simple integration into our current task. There is also a possibility to work the tool into our algorithm because it can be used as a standalone tool or a library, which was another reason for the use. Thanks to the output information that had been provided by *MorphoDiTa*, our subsequent research focused on the possible compression forms of the morphological tag (see the previous chapters). This seems sufficient because these tags are utilised very often in the complex algorithm which we already devised, and their analysis will make the following work easier.

As already described, the tag itself consists of 15 marks. Because the marks (to make things easier we shall use only this designation as the detailed description can be found here [13,10]) correspond to grammatical categories and advanced rules of the Czech language. From this knowledge we can extrapolate the fact that some marks can be used only if other signs occur or in case when we cannot use any mark in combination with the group of other – symbol "-".

The most important mark in the tag are the first (*POS*) and the second (*SUBPOS*) marks according to which it can be clearly said what combinations of marks will appear in the subsequent positions and which positions will remain empty. After applying this knowledge, it is possible to employ compression into the following form. All this on condition that in case of decompression we will be able to assemble the original tag.

Example - original tag : N N I P 1 - - - - - A - - - - -
Example - no gaps: N N I P 1 A

Another possibility is the elimination of the first mark – *POS* based on the condition of its reverse deducibility from the sign *SUBPOS*.

Example - *POS* elimination: N I P 1 A

These pieces of information allow us to effectively compress the whole mark. Another problem we have dealt with is the manner of how the tag is saved which went through two phases. During the course of the research we arrived to several possibilities. As the first we could name these: – Saving the whole sign as the text without compression. – Saving the mark as the text after the previous two-stage compression. This possibility seems as a very ineffective one. Given the assumption that in the each position only a limited and finite number of symbol can appear, the further research was lead in this direction. The first possibility lies in the assignment of the numerical indication of the individual signs within its category only.

Example - numeric: 34 3 2 1 1

As evident, a greater efficiency can be achieved. However this state was not final and thanks to the extracted knowledge of the structure of the sign, it is possible to make the numerical indication even more efficient by very simple means. We know that the certain category of the signs can be used only in the case of use of the certain *POS*/*SUBPOS*, therefore we can divide the signs into other categories and use the numbering only within a frame of these smaller categories. This way we reduce the highest used numbers.

Example - numeric, reduced: 34 2 1 1 1

In the following part of the research, our concern proceeded towards the possibilities of saving the sequences of numbers that represent tags. In the current phase of the development we work with several possible variants:

– saving the number as *float*/*int*,

– encryption of the number using the existing coding algorithms,
– binary representation of the number,
– combination of several possibilities.

The first possibility was quickly dismissed for the inefficiency which mainly lied in the compression of small numbers. There the second option as utilized why using elias-gamma coding [16]. This manner of encoding is further used in other parts of this task and the final values are recorded using this method. Another possibility – the binary representation of the number – is currently in the phase of development but it is possible to say that this method seems applicable. The principle lies in finding of the highest possible number which needs to be encoded, then in the determination of the number of bites that represent it, thus also determining the number of bites upon which all other numbers will be represented. The last possibility related to the combination of various techniques offers a huge space for experimenting. It is for example expected that the algorithm alone will determine the most suitable methods (or the their combinations) of encoding for the given initial text.

6.5 Coding of lemmas

The current version of the algorithm uses LZ algorithms [17] for the compression of lemmas as it currently seems as sufficient enough. However further ahead we expect to use other tests for different compression algorithms which are primarily intended for the compression of text.

6.6 Organisation of saving

After analysis and text processing, all unique lemmas and tags are arranged according to their frequency. They are kept in the list for the final processing. Therefore it is important for us to use the optimal encoding of sequences of numbers also due to saving of position and indexes. The resulting compressed file is sorted into three parts:

1. The list of all lemmas found in the text
2. The list of all signs found in the text
3. For the every word the original text here is represented by a pair of indexes that are referring to the previous two lists.

Here, the usage of various methods of encoding will be advantageous according to the total size of lists of lemmas/tags. As already stated, in the current phase we use only one method of the numerical coding.

6.7 The use of external corpuses

One of the possible extensions for achieving of the best possible compression is the use of external dictionaries, both for lemmas and tags. In case of tags the use

is simpler because the number of possible combinations can be unambiguously determined. According to it, all possible variations are generated. This method is rather more difficult with lemma dictionaries. As the language that is still being used, the Czech constantly develops and thus the new words and lemmas are being created. On the other hand, some words become archaic and less used. In this case, the generating of all lemmas is inefficient and insufficient. There is another factor to consider; that in various types of texts (fiction, spoken word, scientific texts) the frequency of words is vastly different. Another drawback is the non-existence of the word/lemma of the original text in the given corpus. A typical example are for example the additional postfixes of lemmas which are added by analyser for better accuracy.

The corpora are available in several categories at UCNKP [19]. Organised corpora of words and their lemmas for different categories of text and period of time can be acquired from there. On closer examination one can see that for different types of text the frequencies do differ and thus there is no universal list that could be considered as the best for all types. The initial thought of using the corpus lied in the method that the compressed file will not include the first two lists with lemmas and tags and that the pair of indexes, representing the initial text, will refer to positions in external corpora. This method should ensure the smaller size of the resulting file, however for the reverse decompression it will be necessary to use the external file with corpus which might not be really optimal.

In the initial experiment of our work, the general corpus *SYN2010* was used. The corpus includes more than 30,000 lemmas. During the tests, we compared the intersection of lemmas included in this corpus with lemmas appearing from the set of literature of fiction (which is used in our other tests). After evaluation, the corpus was found insufficient for our testing set because of the great number of missing lemmas. However, despite these initial findings we plan to use these corpora at least for the specific types of initial tests. Recently, the corpus *SYN2015* has been available which, besides morphological analysis, also includes the syntactic analysis. For the next parts of our work we shall use this corpus as it cancels the necessity to use the external tools *MorphoDiTa* or *Treex* for the creation of input text analysis during the testing.

6.8 Results

Brief characteristic of files used in the test is given in Table 1. Due to the aforementioned reasons, our tests dealt with a single method of compression which is:

- the compression of the lemma forms using LZ,
- encoding of tags using the numerical encoding,
- representation of indexes of the initial text using elias-gamma code.

After the evaluation of the results, see Table 2, we arrived at several conclusions which will assist us in our further work. The most substantial part, which influences the final size of file the most, is the encoding of indexes. The method

Table 1. File characteristics

File	File size (megabytes)	Number of words	Number of unique lemmas	Number of unique tags
A	1	359,614	14,854	574
B	5	1,764,258	34,286	718
C	10	3,578,666	41,466	733
D	50	17,874,992	110,254	911
E	100	35,635,693	162,935	981

Table 2. Compression results

File	Lemma (bits)	Lemma index (bits)	Tags (bits)	Tag index (bits)	Compression ratio (≈%)
A	452,896	3,867,038	7,155	1,896,490	74
B	1,015,200	19,518,472	9,148	9,394,852	71.2
C	1,237,792	39,005,856	9,304	18,913,870	70.5
D	3,301,856	197,755,574	12,132	94,951,596	70.6
E	4,945,152	395,358,961	13,123	189,143,515	70.3

of encoding we chose does not seem as the most suitable in case of truly huge numbers which obviously results from this type of elias-code. Another crucial factor is the size of occurrence of the original tags in the text. It can be seen that if the initial set of words increases hundredfold then the number of the original tags increases only twice as much. This fact underlines the attributes of the Czech language and supports our assumptions that despite the complexity of the rules of language, the regularities in structural composition can be found. The similar ratio can be seen even in the number of the original lemmas but here the size ratio is not so prominent. But this is the result (besides other things) of the selection of the input text that was analysed and compressed.

7 Conclusion and future works

It can be said that the first phase of our tests met its purpose and confirmed our initial assumptions while setting the direction of further research. The results alone are not the best possible (due to the chosen methods of encoding), however the improvement of the methods and their optimization will make the main content of our work. After we finish this phase, we shall focus on the possibility of at least partial integration of external dictionaries and then on the search of similarities at several levels.

Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project “IT4Innovations

excellence in science - LQ1602" and from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center – LM2015070".

References

1. Yuret D.: *Discovery of Linguistic Relations using Lexical Attraction*, PhD Thesis, Massachusetts Institute of Technology, 1998
2. Bach J., Witten H.: *Lexical Attraction for Text Compression*, Proceedings of Data Compression Conference, pp. 516-516, 1999.
3. Awajan A.: *Multilayer model for Arabic text compression*, Int. Arab J. Inf. Technol., vol. 8, no. 2, pp. 188–196, 2011.
4. Moffat A.: *Word-Based Text Compression*, Software, Practice and Experience, pp. 185-198, 1989
5. Chang K. Y., Yang G. T.: *A Data Compression System for Chinese Fonts and Binary Images Using Classification Techniques*, Pr. Exp., vol. 22, no. 12, , 1992.
6. Lanský J., Zemlicka M.: *Compression of Small Text Files Using Syllables*, Technical Report of Department of Software Engineering No. 2006- 1, 2006
7. Akman I., Bayindir H., Ozleme S., Akin Z., Misra S.: *Lossless Text Compression Technique Using Syllable Based Morphology*, vol. 8, no. 1, pp. 66–74, 2011.
8. Kazik O.: *Lingvistická komprese textu*, Diploma Thesis, Charles University in Prague, 2009
9. Hajicova E., Panevova J, Sgall P.: *Úvod do teoretické a počítačové lingvistiky, I. svazek - Teoretická lingvistika*, Nakladatelství Karolinum, Praha, 2002.
10. J. Hana, D. Zeman : Manual for Morphological Annotation, Revision for the Prague Dependency Treebank 2.0., 2005
11. Honetschläger V. a kol.: *The Prague Dependency Treebank 3.0*, https://ufal.mff.cuni.cz/pdt3.0
12. *Treex Highly Modular NLP Framework*, http://ufal.mff.cuni.cz/treex
13. Hajič, J.: *Disambiguation of Rich Inflection (Computational Morphology of Czech)*, Karolinum, Praha, 2004
14. Sedlacek R.: *Morfologicky analyzator cestiny*, Diploma thesis, Masaryk university Brno, 1999
15. Melichar B., Janousek J., Flouri T.: *Introduction to Arbology*, Czech Technical University in Prague, 2008
16. Elias, P.: *Universal codeword sets and representations of the integers*, Information Theory, IEEE Transactions on , vol.21, no.2, pp.194-203, Mar
17. Ziv J., Lempel A. *A universal algorithm for sequential data compression*, IEEE Transactions on Information Theory 23 (1977), 337–343
18. Strakova J., Straka M. Hajic J.: *Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition*, Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, June 2014
19. Ceský narodni korpus: *Srovnavaci frekvencni seznamy*, Ustav Ceského narodniho korpusu FF UK, Praha 2010. Available on: http://ucnk.ff.cuni.cz/srovnani10.php