Aranea Go Middle East: Persicum

Vladimír Benko[1,2]

[1] Slovak Academy of Sceinces, Ľ. Štúr Institure of Linguistics
Panská 26, 811 011 Bratislava, Slovakia
`http://juls.savba.sk/~vladob`
`vladimir.benko@juls.savba.sk`
[2] Comenius University Science Park
UNESCO Chair in Plurilingual and Multicultural Commnication
Ilkovičova 8, 814 04 Bratislava, Slovakia

Abstract. Our paper introduces the creation and annotation of *Araneum Persicum*, a new Persian web-crawled corpus. Some problems encountered during the process of filtration and annotation are shown, and an ensemble approach adopted for lemmatization and morphosyntactic annotation is introduced. It is also argued that Romanization can be helpful in developing corpora for languages not based on Latin script.

Keywords: Web-crawled corpus, Persian language, Ensemble tagging

1 Introduction

The Aranea Project[3] [2] aimed at the creation and annotation of a family of web-crawled corpora for languages taught at Slovak Universities has reached the point where most "common" languages have been covered already, and their total count has approached the two-dozen landmark. For various reasons, new languages are still being added to our collection, even if there is no chance that they would ever be taught in Slovakia. The Persian language, also referred to as Farsi[4], belongs to this category as well.

Our attempt to build a Persian corpus has been initialized by our Prague colleagues working on a Persian to Czech dictionary [15] who need a reliable source of lexical evidence on contemporary Persian language, as well as our desire to make use of our experience and tools developed in the framework of our Aranea Project to process a language using a right-to-left script.

2 The Persian Language

Persian belongs to the Indo-Iranian subgroup of Indo-European languages with at least 70 million speakers[5]. If all its varieties are considered, the language

[3] http://aranea.juls.savba.sk/guest/

[4] http://www.iranian.com/Features/Dec97/Persian/

[5] https://en.wikipedia.org/wiki/Persian_language

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2022, pp. 113–121, 2022. © Tribun EU 2022

has official status in Iran, Afghanistan, and Tajikistan, as well as in several neighboring countries. Having a long history of writing, the modern Persian uses a modified Arabic script in Iran and Afghanistan, and a modified Cyrillic script in Tajikistan.

2.1 The Persian Script

The main obstacle in any attempt to grasp the Persian script is the fact that the shape of (almost) any grapheme can have as many as four different forms depending on its position within a word (initial, medial, final, and isolated, respectively). To ease this "mental burden", we decided to supplement each corpus token (word form, lemma, etc.) by its respective Romanized variant, adopting the UN 2012[6] transliteration system. The main advantage of this system for use in our environment is that all transliterated graphemes are directly accessible via Czech and Slovak keyboard, with the only exception being the "ā" character (representing the Persian "آ") that has been substituted by an "á".

In comparison to Arabic, the Persian script contains four additional graphemes representing phonemes not present in Arabic (پ, چ, ژ, گ, transliterated as "p", "č", "ž", and "g", respectively), and two graphemes (ی, ک, i.e., "y" and "k") that have slightly different shapes and their own Unicode code points – this fact can be conveniently used in secondary language filtration.

The real-world Persian texts on the web, however, also contain certain amount of words with Arabic spelling (mostly proper names and Quran-related lexical items), loanwords from other Indo-Iranian languages preserving the original orthography, nonstandard use of diacritics denoting vowels, etc., so some sort of normalization is suggested before a text can be processed by a NLP tool.

2.2 Persian Morphology

I must admit that I was only able to "plunge" into those issues in this area that generated some problems during lemmatization and PoS tagging of the corpus data.

Unlike in most other languages within the Aranea family, the basic form of a Persian verb is not represented by its infinitive, but rather by two stems (present and past, respectively). This has a rather negative influence on lemmatizers that are in such a case typically not able to guess valid lemmas for the out-of-vocabulary (OOV) lexical items.

Another peculiar feature of the Persian morphology is that certain affixes can be written either together with the stem, separated by a "half-space" ("zero-width non-joiner" character U+200c, having a special hotkey combination on a Persian keyboard), or even by a standard space. A corpus designer therefore has to make a decision about what data should be sent to the tagger (i.e., the original, half-space-normalized or even space-normalized).

[6] http://www.eki.ee/wgrs/rom1_fa.pdf

3 Persian Language Resources and NLP Tools

Even a simple "Google research" reveals lots of projects devoted to the processing of the Persian language. On the other hand, resources that would be readily available to those who would like to compile their own Persian corpora themselves are not so numerous. In general, at least tools for lemmatization and/or PoS tagging are needed. To speed up the creation of the initial (beta) version of our Persian corpus, we decided to make use of only those tools that have been already engaged in the processing of other corpora of the Aranea family.

Persian Treebanks. The obvious place to look for (syntactically) annotated corpora is the *Universal Dependencies* Portal[7]. We can find two Persian items there. The larger is the *Persian Dependency Treebank* (*PerDT*) [8] containing approximately 500 K tokens, while the considerably smaller *Seraji* [11] based on the *Upsala Persian Treebank* has 152 K tokens. Besides its size and genre coverage of the latter (it contains news only), the other issue is its "incomplete" lemmatization – lemmas for many lexical items are simply set to an "_" character. If used for training, this error is further propagated to the tagged data.

TreeTagger [10]. Despite its age, this tool is still being used by many corpus projects, including that of ours. There are several reasons for this: Firstly, it still maintained by its original developer; secondly, there are language models for many different languages; thirdly, it is stable even if applied on very large (many Gigaword) corpora; and lastly, it is very fast – especially in comparison to newer tools based on, say, a neural network.

On the other hand, the quality of its output is not as high as that of newer taggers, especially if applied to a language with rich inflectional morphology and corresponding fine-grained tagset [4]. The *TreeTagger* performs guessing of PoS tags for OOV lexical items, yet it does not attempt to guess the lemma in such a situation.

The Persian language model for *TreeTagger* has been created by means of the *PerDT* data which means that it is the tool with the largest coverage of Persian lexis.

UDPipe [12] is the main tagging tool developed within the *Universal Dependencies* Project [5]. The corresponding language model(s) can be created for all languages where a corresponding treebank exists. As the *UDPipe 3* version is still in development and therefore not released yet, and the *UDPipe 2* is available as a Python prototype (or a web service) only, i.e., not suitable for processing large-scale data [13], for users who want to run the software within their own infrastructure, the oldest *UDPipe 1* is the only option.

Although two different treebanks for Persian are available, for various reasons the only language model available has been trained on the *Seraji Treebank*

[7] https://universaldependencies.org/

resulting into a much lower coverage than that of *TreeTagger*, and also leaving many lexical items without any lemma.

CSTlemma [3]. As its name suggests, this tool does not perform a “complete” tagging, and just generates basic form for each token in the corpus. The respective language model can be trained by a (preferably large) morphological lexicon, and authors provide pretrained models for many languages. The Persian model has been created by means of the MULTEXT-East lexicon [16], and (what is its rather negative feature for our work) does not generate “compatible twin lemmas” for Persian verbs.

4 Corpus Processing

Preparation. The first step in building a new web-crawled corpus is the collection of seed URLs that are needed as one of the inputs for the *SpiderLing* [14] crawler. This used to be fairly easy to perform by the *BootCaT* [1] tool, until Microsoft stopped supporting the free Bing queries via an API some years ago. The tool itself is still operational, yet the current procedure involves a lot of manual “cut and paste” operations, which makes this option clearly rather “suboptimal”. An alternative is provided by the *WebBootCaT* functionality of the *Sketch Engine*[8] portal (if one owns an account ;-)

The procedure of “harvesting” the URLs involves providing the program with a set of “keywords” used to create n-grams that are being submitted to a search engine. The resulting lists of Internet addresses may be manually edited and used for the subsequent downloading of the actual documents. In our case, however, we did not need to let Sketch Engine to perform the downloading, and just took the list itself. This procedure can be repeated until the required number of URLs has been collected.

According to our experience, the initial list of keywords should consist of words of general semantics, such as high-frequency adverbs. The respective list for our work has been extracted from one of the Persian corpora hosted at the Sketch Engine site: the list of most 1,000 adverbs has been randomly sorted and 5 sets of 12 words have been used. The n-gram length has been set to 3 and all oth er parameters to the maximal values. The five rounds of harvesting yielded (after deduplication and removing URLs from unwanted domains, such as instagram.com and youtube.com) approximately 19,500 URLs.

Another user input needed for *SpiderLing* operation are text samples used to create language models for on-the-fly language identification and filtering during the crawling. Samples for Persian, Arabic, and English have been extracted from selected Wikipedia pages in the respective languages.

Crawling and preprocessing. The actual crawling was performed in April 2022 by *SpiderLing 2.0* in 10 parallel threads. After some 36 hours of crawling, approx.

[8] https://www.sketchengine.eu/

96 GB of raw text in a "prevertical" format have been gathered, almost 20 GB out of which have been removed during the initial deduplication targeted at 100% duplicates.

Two main filtering procedures attempted to delete texts being "insufficiently Persian" (counting frequencies of Persian characters) and "too non-Persian" (counting characters not present in the Persian alphabet, yet being based on Arabic script). The respective thresholds in both cases have been set experimentally, removing in total another 21 GB of data. A short analysis of the removed documents revealed that, besides Arabic, most of them were in fact written in the Pashto language that is also spoken both in Iran and Afghanistan.

Tokenization. In the framework of our Aranea Project, the universal tokenizer *Unitok* [6] (with custom parameter files) is used for tokenization. After initial experimentation and somewhat surprisingly, the English parameter file could be used (almost) without modification for Persian – the only change was associated with the treatment of "half-spaces" that had to be considered "letters", if non-normalized text is to be tokenized.

The tokenization procedure yielded a vertical file of 5.59 Gigatokens. The secondary deduplication procedure (performed by Onion [7]) removed more than 30% of them, retaining the 3.89 Gigatokens in approx. 4.49 M documents.

Ensemble annotation. In Computational Linguistics, the "ensemble" term is used to describe approaches where several tools are utilized to (independently) perform the same operation, assuming that aggregation of their outputs could improve the overall success rate of the whole process. In the framework of morphosyntactic annotation, we can speak about "ensemble tagging" if more than one tagger is available for a particular language – which is also the case of Persian.

If all the tools use the same tagset and they are more than two, the aggregation is usually performed by simple "voting". In our case, however, we not only do not have three "full-fledged" taggers, and the respective tagsets are not completely compatible. The component tools also do not behave in the same way with respect to OOV lexical items. The actual situation is shown in Table 1.

Table 1: Three ensemble component tools.

Word forms	TreeTagger	UDPipe	CSTlemma
Non-OOVs	PoS tag assigned	PoS tag guessed	n/a
OOVs	PoS tag guessed	PoS tag guessed	n/a
Non-OOVs	Lemma assigned	Lemma guessed	Lemma guessed
OOVs	Lemma marked as OOV	Lemma guessed	Lemma guessed

The aggregation procedure has been therefore designed as follows:

1. Only main word classes (based on the *AUT*[9] tagset) are considered for aggregation.
2. If PoS can be assigned by means of a regular expression (punctuation, digits, symbols, e-mail addresses, etc.), ignore the information from the taggers.
3. If the respective token was present in the morphological lexicon of TreeTagger, take both lemma and PoS from it.
4. If the token was OOV in TreeTagger and UDPipe guessed a lemma, take both lemma and PoS from it. If lemma guessed by CSTlemma differs, add it as an alternative. If PoS guessed by the TreeTagger differs, add it as an alternative.
5. Otherwise take the lemma from CSTlemma and PoS from TreeTagger and UDPipe (if it differs).

The result of the aggregation process is flagged in a special attribute "ztag": the respective value consists of two parts separated by a period – the left part denotes assignment of lemma, while the right that of the PoS. The uppercase letters indicate success in the morphological lexicon lookup (in case of TreeTagger), the lowercase letters indicate guessing and the exclamation mark indicates that the respective value differs from that on the left. The actual situation in a 125-Megatoken sample of *Araneum Persicum* is shown in Table 2.

Table 2: Frequency distribution of ztags.

ztag	freq	%	ztag	freq	%
`T!c.T!u`	313,397	0.25	`u!c.t!u`	734,753	0.59
`T!c.Tu`	1,744,755	1.40	`u!c.tu`	2,098,439	1.68
`T!u!c.T!u`	141,740	0.11	`uc.t!u`	2,077,455	1.66
`T!u!c.Tu`	2,082,343	1.67	`uc.tu`	3,751,262	3.00
`T!uc.T!u`	644,554	0.52	`c.t!u`	1,411,739	1.13
`T!uc.Tu`	4,477,277	3.58	`c.tu`	3,026,913	2.42
`Tc!u.T!u`	783,445	0.63	`z!`	6,629,046	5.30
`Tc!u.Tu`	1,242,515	0.99	`z#`	788,956	0.63
`Tc.T!u`	694.332	0.56	`z$`	910,190	0.73
`Tc.Tu`	3,817,999	3.05	`z@`	1,068	0.00
`Tu!c.T!u`	455,065	0.36	`zu`	5,987	0.00
`Tu!c.Tu`	13,464,342	10.77	`zv`	17,675	0.01
`Tuc.T!u`	7,834,406	6.27	`zw`	1,972	0.00
`Tuc.Tu`	65,848,758	52.68	Total	125,000,383	100.00

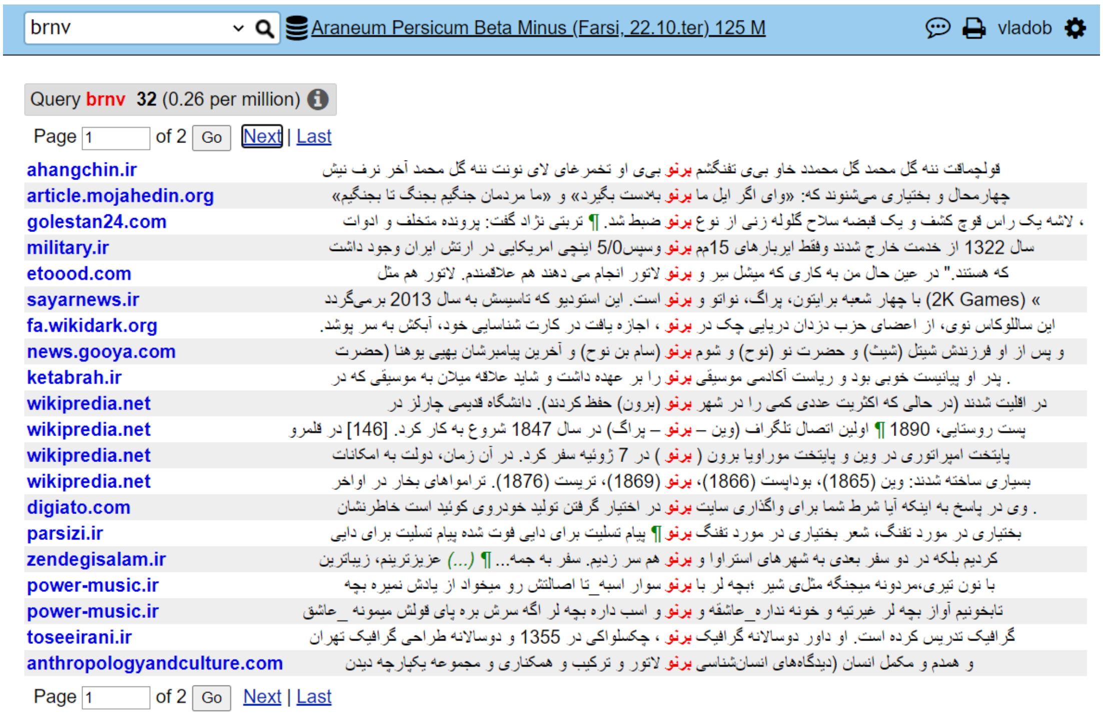
Flags starting with the "`z`" letter indicate lexical items "tagged" by regular expressions. For example, "`z!`" denotes punctuation, "`z#`" numbers, and "`z$`" symbols (special graphic characters, emoji, etc.).

As it can be seen, most items have a "`Tuc.Tu`" flag, indicating equal values assigned by all tools, followed by a "`Tu!c.Tu`", with CSTlemma assigned a differtent lemma than TreeTagger and UDPipe.

[9] http://aranea.juls.savba.sk/aranea_about/aut.html

5 Compilation by the Corpus Manager and Publication

During the development of the corpus, a small sample of the whole data was used with all parallel annotations being available for querying via the *NoSketch Engine* [9] corpus manager. This helped to identify several issues of the processing pipeline, as well as the respective tools themselves. The final beta version of the corpus containing all data, however, contains the aggregated annotations only plus the transliterated versions of both word and lemma attributes. By including these fields into the SIMPLEQUERY directive of the corpus configuration file, it is now possible to conveniently query either in Persian or transliterated versions of the respective attributes. An example of such a query is shown in Fig. 1).



brnv

Araneum Persicum Beta Minus (Farsi, 22.10.ter) 125 M

vladob

Query brnv 32 (0.26 per million)

Page 1 of 2 Go Next | Last

ahangchin.ir	قولچماقت ننه گل محمد گل محمدد خاو بیی تفنگشم برنو بیی او تخمرغای لای نونت ننه گل محمد آخر نرف نیش
article.mojahedin.org	چهارمحال و بختیاری می‌شنوند که: «وای اگر ایل ما برنو به‌دست بگیرد» و «ما مردمان جنگیم بجنگ تا بجنگیم»
golestan24.com	، لاشه یک راس قوچ کشف و یک قبضه سلاح گلوله زنی از نوع برنو ضبط شد. ¶ تربتی نژاد گفت: پرونده متخلف و ادوات
military.ir	سال 1322 از خدمت خارج شدند وفقط ایربارهای 15مم برنو وسپس5/0 اینچی امریکایی در ارتش ایران وجود داشت
etoood.com	که هستند." در عین حال من به کاری که میشل سِر و برنو لاتور انجام می دهند هم علاقمندم. لاتور هم مثل
sayarnews.ir	« (2K Games) با چهار شعبه برایتون، پراگ، نواتو و برنو است. این استودیو که تاسیسش به سال 2013 برمی‌گردد
fa.wikidark.org	این ساللوکاس نوی، از اعضای حزب دزدان دریایی چک در برنو ، اجازه یافت در کارت شناسایی خود، آبکش به سر پوشد.
news.gooya.com	و پس از او فرزندش شیتل (شیث) و حضرت نو (نوح) و شوم برنو (سام بن نوح) و آخرین پیامبرشان یهیی یوهنا (حضرت
ketabrah.ir	. پدر او پیانیست خوبی بود و ریاست آکادمی موسیقی برنو را بر عهده داشت و شاید علاقه میلان به موسیقی که در
wikipredia.net	در اقلیت شدند (در حالی که اکثریت عددی کمی را در شهر برنو (برون) حفظ کردند). دانشگاه قدیمی چارلز در
wikipredia.net	پست روستایی، 1890 ¶ اولین اتصال تلگراف (وین – برنو – پراگ) در سال 1847 شروع به کار کرد. [146] در قلمرو
wikipredia.net	پایتخت امپراتوری در وین و پایتخت موراویا برون (برنو) در 7 ژوئیه سفر کرد. در آن زمان، دولت به امکانات
wikipredia.net	بسیاری ساخته شدند: وین (1865)، بوداپست (1866)، برنو (1869)، تریست (1876). تراموا‌های بخار در اواخر
digiato.com	. وی در پاسخ به اینکه آیا شرط شما برای واگذاری سایت برنو در اختیار گرفتن تولید خودروی کوئید است خاطرنشان
parsizi.ir	بختیاری در مورد تفنگ، شعر بختیاری در مورد تفنگ برنو ¶ پیام تسلیت برای دایی فوت شده پیام تسلیت برای دایی
zendegisalam.ir	کردیم بلکه در دو سفر بعدی به شهرهای استراوا و برنو هم سر زدیم. سفر به جمه... ¶ (...) عزیزترینم، زیباترین
power-music.ir	با نون تیری،مردونه میجنگه مثل‌ی شیر ؛بچه لر با برنو سوار اسبه_تا اصالتش رو میخواد از یادش نمیره بچه
power-music.ir	تابخونیم آواز بچه لر غیرتیه و خونه نداره_عاشقه و برنو و اسب داره بچه لر اگه سرش بره پای قولش میمونه _عاشق
toseeirani.ir	گرافیک تدریس کرده است. او داور دوسالانه گرافیک برنو ، چکسلواکی در 1355 و دوسالانه طراحی گرافیک تهران
anthropologyandculture.com	و همدم و مکمل انسان (دیدگاه‌های انسان‌شناسی برنو لاتور و ترکیب و همکاری و مجموعه یکپارچه دیدن

Page 1 of 2 Go Next | Last

Fig. 1: The result of querying "brnv" ("Brno") in Araneum Persicum Minus.

The Beta versions of the Persian corpus in three sizes have been published at our Aranea Corpus portal recently.

6 Conclusion and further work

Despite the fact that the processing pipeline for Aranea corpora has been tuned and is relatively stable, any new corpus may present additional challenges, let alone in situations, when the developer(s) do not understand the language. The ensemble approach for lemmatization and tagging significantly improved the quality of annotation. The idea of providing supplementary transliterated attributes turned out to be quite successful and made the tuning of the data much easier.

For the next version of *Aranuem Persicum* we would like to add more component tools to the ensemble, and maybe also try to create own language model for UDPipe based on PerDT.

Acknowledgements This work has been, in part, supported by the Slovak VEGA Grant Agency for Science, Project No. 2/0016/21, as well as by the TAČR Technology Agency of the Czech Republic, Project No. TL0300369.

References

1. Baroni, M., Bernardini, S. BootCaT: Bootstrapping Corpora and Terms from the Web. In Proceedings of LREC 2004, pp. 1313-1316 (2004)
2. Benko, V.: Aranea: Yet Another Family of (Comparable) Web Corpora. In Petr Sojka, Aleš Horák, Ivan Kopeček and Karel Pala (Eds.): Text, Speech and Dialogue. 17th International Conference, TSD 2014, Brno, Czech Republic, September 8-12, 2014. Proceedings. LNCS 8655. Springer International Publishing Switzerland (2014)
3. Jongejan, B, and Dalianis, H.: Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. In Proceedings of the Joint Con-ference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. Suntec, Singapore : Association for Computational Linguistics, pp. 145-153 (2009)
4. Kotiurova, I., and Trenina, P.: Comparative Analysis of Automatic POS Taggers Applied to German Learner Texts. 31st Conference of Open Innovations Association (FRUCT), 2022, pp. 115-124, https://doi.org/10.23919/FRUCT54823.2022.9770886 (2022)
5. McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Z., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, 0., Bedini, C., Bertomeu Castelló, N., Lee, J. Universal Dependency Annotation for Multilingual Parsing. In Proceedings of ACL (2013)
6. Michelfeit, J., Pomikálek, J., Suchomel, V.: Text Tokenisation Using unitok. In 8th Workshop on Recent Advances in Slavonic Natural Language Processing, Brno, Tribun EU, pp. 71–75 (2014)
7. Pomikálek, Jan. Removing boilerplate and duplicate content from web corpora. PhD thesis, Masaryk university, Faculty of informatics, Brno, Czech republic (2011)
8. Rasooli, M. S., Safari, P., Moloodi, A., Nourian, A.: The Persian Dependency Treebank Made Universal. 2020 (to appear)
9. Rychlý, P.: Manatee/Bonito – A Modular Corpus Manager. In: 1st Workshop on Recent Advances in Slavonic Natural Language Processing. Brno : Masaryk University, pp. 65–70. (2007)
10. Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees. Proceedings of International Conference on New Methods in Language Processing, Manchester (1994)
11. Seraji, M., Ginter, F., Nivre, J.: Universal Dependencies for Persian. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC), pp. 2361-2365 (2016)

12. Straka, M., Hajič J., Straková J.: UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In Proceedings of the Tenth International Conference on Language Resources and Evalua-tion (LREC), Portorož, Slovenia (2016)
13. Straková, J. Personal communication (2022)
14. Suchomel, V., Pomikálek, J. Efficient Web Crawling for Large Text Corpora. In Adam Kilgarriff, Serge Sharoff. Proceedings of the seventh Web as Corpus Workshop (WAC7). Lyon, pp. 39-43 (2012)
15. Vystrčilová, D., Khademi, M., Kříhová, Z., Novák, Ľ. (2020, August 1–). Elektronická lexikální databáze indoíránských jazyků. Pilotní modul perština. Electronic Lexical Da-tabase of Indo-Iranian Languages. Pilot module: Persian. (Project No TL03000369, Technology Agency of the Czech Republic). Institute of Sociology, Czech Academy of Sciences & Faculty of Arts, Charles University. https://www.soc.cas.cz/projekt/elektronicka-lexikalni-databaze-indoiranskych-jazyku-pilotni-modul-perstina
16. Zadeh, B. Q., Rahimi, S.: Persian in MULTEXT-East Framework. In: Advances in Natural Language Processing, 5th International Conference on NLP, FinTAL 2006, Turku, Finland, August 23-25 (2006)