Methodological and Technical Solutions for Creating and Forming a Knowledge Base by Integrating the Mathematica System and the Nevod Package

Valery B. Taranchuk
Department of Computer Applications and Systems
Belarusian State University
Minsk, Republic of Belarus
taranchuk@bsu.by

Vladislav A. Savionok
Department of Software for Information Technologies
Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
v.savenok@bsuir.by

***Abstract*—This paper presents an example of integration of a local intelligent computer system based on the Nevod library with the knowledge base of Wolfram Mathematica computer algebra system, which can be interpreted as an analogue of knowledge base localization actions. Examples of the use of tools to analyze the local knowledge base, its transfer from virtual to real status are presented and explained.**

***Index Terms*—semantic analysis, Wolfram Mathematica, Wolfram Knowledgebase, entity, temporal markers, Nevod**

I. Introduction

Based on the evaluation of the current state of artificial intelligence (AI) in [1], there is ongoing development in various directions including formal ontologies, artificial neural networks, machine learning, and multi-agent systems. However, these advancements do not result in a cumulative increase in the intelligence level of modern intelligent computer systems (ICS).

Modern frameworks for AI design support primarily focus on creating specialized solutions that function as standalone components within an ICS. To ensure compatibility across all developed components, it is necessary to unify these tools into a single technology that supports the entire ICS lifecycle. Unification and convergence of new generation ICS and their components is essential for guaranteed compatibility. In [1], the text outlines the primary activities proposed to address the fundamental methodological issues causing the current state of AI. It is worth noting that these problems have already been tackled in the design, development, content updating, and functionality expansion of computer mathematics systems [2], [3].

In this paper, methodological and technical solutions for the integration of IKS knowledge are presented, software tools within the Wolfram Mathematica (WM) computer algebra system using Wolfram Language (WL) are identified and described, and their application to the Wolfram Knowledgebase (WKB) is discussed. The WKB is the world's largest and most comprehensive repository of computable data, containing specialized knowledge from thousands of fields and a wide variety of computational algorithms. The goal is to present the data in a clear and concise manner, using objective language and following conventional academic formats and structures. Examples are presented to demonstrate how various WM tools and independent Nevod library [4] can be used together along with how local applications can be kept current through cooperative use. The integration with WKB is based on a universal approach, emphasizing the requirement for unification of new generations of ICS, the adopted method can be applied to the integration of WM with other knowledge systems.

II. Temporal Markers Analysis

One of the directions in the field of text processing is the extraction of its semantic component – semantic analysis. In this direction such tasks as document search in local and global networks, automatic annotation and abstracting [5], document classification and clustering, text synthesis [6] and machine translation, text tone analysis, fact extraction (publications are mentioned in [7]) are solved. An integral component of the task of fact extraction and determination of relations between objects is the localization in time of the event corresponding to the fact. The information that allows to localize the event on the timeline is conveyed by means of text expressions – temporal markers, that are diverse in form and content. The final result of extracting temporal markers from the text is their representation and interpretation within the framework of the formal model defined in the process of semantic analysis [8]. To solve the task of extracting temporal markers from text, the toolkit of Microsoft.Recognizers.Text [9], one of the leaders in the field of entity recognition, is widely used.

III. Preparing Test Dataset

A. Initial Test Dataset

The MS Recognizers Text (MRT) library provides the ability to recognize entities in texts of various languages and is widely used in Microsoft products, for example: in pre-defined

templates for LUIS (Language Understanding Intelligent Service), in the platform for creating dialog bots Power Virtual Agents [10], in cognitive language services in Azure cloud infrastructure – NER (Named Entity Recognition). The library is distributed under an open source and free software license from MIT; along with the source code in the repository on GitHub [9] test dataset for different languages are published.

The Microsoft.Recognizers.Text.DateTime module, and in particular its BaseDateExtractor component, is used in MRT to search for temporal markers in the text. This component corresponds to a test dataset represented in JSON format – the DateExtractor.json file [11]. The dataset contains 143 elements that include absolute and relative dates in different forms, as well as metainformation, which is used to check the correctness of the extraction results. A search context, a reference date that indicates the point in time used to translate relative temporal markers into absolute ones, can be attached to the dataset element. An example of a test dataset element with comments is shown in Fig. 1.

```
{
  "Input": "i will leave in 3 weeks", // - input text for search
  // search context:
  "Context": { "ReferenceDateTime": "2018-06-20T00:00:00" },
  "NotSupported": "python,javascript",
  "Results": [ // list of expected results:
    // each result includes text, type, start position and length
    { "Text": "in 3 weeks", "Type": "date", "Start": 13, "Length": 10 }
  ]
},
```

Fig. 1. Example of a test dataset element.

B. Motivation For Refining the Test Dataset

In [7], by comparing the capabilities with MRT in solving the problem of temporal marker extraction, the functional completeness of the Nevod library [4], which realizes the method of searching in text [12], is confirmed. For this purpose two program modules have been developed: $mMRT$ and $mNevod$, providing search and extraction of temporal markers from the text. Comparative testing of the program modules was carried out on the described test set using WM tools in the analysis of the results.

In order to compare the functionality of $mNevod$ and $mMRT$ modules in solving the problem of extraction of temporal markers in text using not only the DateExtractor test dataset but also other representative datasets, the $mDataWM$ service application was developed with Wolfram Mathematica functions. The software contains tools that allow you to separate the dataset for processing from the meta-information, compare the quality of the results from processing a modified dataset using $mMRT$ and $mNevod$ modules, manipulate any dataset, and test the libraries' efficiency.

When checking and tuning fact extraction tools, in particular temporal markers, an important position for evaluation is the focus on recognition rather than unambiguous identification of entities in the text. The initial test set DateExtractor of the MRT library does not allow us to fully analyze the functionality of the corresponding tools of this type – it does not take into account the possibility of distortion of the input text. It seems reasonable to compose a new test dataset that would take this aspect into account. The methodology of forming a representative test set is described in [7].

C. Using Wolfram Mathematica to Refine Initial Test Dataset

The $mDataWM$ application enables the creation and analysis of test datasets in any natural language. The tools of the $mDataWM$ module implement the following functions:

- distort initial dataset and form a modified one;
- import/export to interface Mathematica with the $mMRT$ and $mNevod$ modules (handling files and separating data from meta-information);
- evaluate the quality of the results of dataset processing.

The following Mathematica kernel functions are used in $mDataWM$:

- $Import[source]$ – imports data from $source$, returning a Wolfram Language representation of it.
- $Export[dest, expr, "format"]$ – exports $data$ in the specified format "$format$".
- $Map[f, expr]$ or $f/@expr$ – applies f to each element on the first level in $expr$. $Map[f, expr, levelspec]$ – applies f to parts of $expr$ specified by $levelspec$. $Map[f]$ – represents an operator form of Map that can be applied to an expression.
- $MapIndexed[f, expr]$ – applies f to the elements of $expr$, giving the part specification of each element as a second argument to f.
- $Association[key1- > val_1, key_2- > val_2, ...]$ or $< |key_1- > val_1, key_2- > val_2, ...| >$ – represents an association between keys and values.
- $AssociateTo[a, key- > val]$ – changes the association a by adding the key-value pair $key- > val$.
- $SortBy[list, f]$ – sorts the elements of $list$ in the order defined by applying f to each of them. $SortBy[list, f_1, f_2, ...]$ – breaks ties by successively using the values obtained from the f_i.
- $KeyMemberQ[assoc, form]$ – yields True if a key in the association $assoc$ matches $form$, and False otherwise. $KeyMemberQ[form]$ – represents an operator form of $KeyMemberQ$ that can be applied to an expression.
- $KeyDrop[assoc, \{key_1, key_2, ...\}]$ – yields an association from which elements with keys key_i have been dropped. $KeyDrop[\{key_1, key_2, ...\}]$ – represents an operator form of $KeyDrop$ that can be applied to an expression.
- $KeyTake[assoc, \{key_1, key_2, ...\}]$ – yields an association containing only the elements with keys key_i. $KeyTake[\{key_1, key_2, ...\}]$ – represents an operator form of $KeyTake$ that can be applied to an expression.
- $RandomSample[\{e_1, e_2, ...\}, n]$ – gives a pseudorandom sample of n of the e_i. $RandomSample[\{e_1, e_2, ...\}]$ – gives a pseudorandom permutation of the e_i.

- $Select[list, crit]$ – picks out all elements e_i of list for which $crit[e_i]$ is True. $Select[crit]$ – represents an operator form of $Select$ that can be applied to an expression.
- $Delete[expr, n]$ – deletes the element at position n in $expr$. If n is negative, the position is counted from the end.
- $StringReplace[" string", s- > sp]$ – replaces the string expression s by sp wherever it appears in "$string$". $StringReplace[s- > sp]$ – represents an operator form of $StringReplace$ that can be applied to an expression.
- $Count[list, pattern]$ – gives the number of elements in $list$ that match $pattern$.

Orienting on the tools for extracting temporal markers in the text, using fragments from DateExtractor, a new test set was prepared. For correct comparison, including with MRT, we excluded elements from DateExtractor that give 1.4% of situations unrecognized by the Nevod library mentioned in [7]. In the resulting set of 141 elements, the distortions (errors) most typical for manual typing were introduced in such a way that they affect the text fragments representing the target for extraction. Due to the extensibility of the patterns in the Nevod package, rules for leveling the corresponding error situations were added, resulting in identical results of 91.4% when processing the set with the $mNevod$ and $mMRT$ software modules. Further updates of the basic Nevod pattern package allowed to achieve a 100% coverage of the test set, solving the remaining 8.6% cases [7].

IV. Using WDR to Verify the Functional Completeness of Temporal Markers Extraction Tools

In the following example, the DateExtractor source test set is hosted, used and modified in the Wolfram Data Repository [13], [14] (WDR, a WKB integration mechanism), and is used to test the correctness of temporal marker extraction, pattern-based target search tools: $mMRT$ based on MS Recognizers Text and $mNevod$ based on Nevod (test results can also be uploaded to WDR).

A. Creating a WDR Thematic Block

Creation of a new WDR thematic block is provided by the $CreateDatabin$ function. When creating a new WDR block, it is possible to specify in advance the semantics of the data to be contained in this block [15]. Data upload in WDR is carried out by $DatabinUpload$ function. It is a common practice to upload data in small batches in parallel to speed up the block creation operation. The $Take$ function can be used to split the initial data array into processable segments *($Take[list, n]$ gives the first n elements of list; $Take[list, -n]$ gives the last n elements of list; $Take[list, \{m, n\}]$ gives elements m through n of list).*

B. Extracting Data from WDR

Data extraction from the WDR is performed using the functions $Databin$ *(represents a databin in the Wolfram Data Drop)* [16] and $Normal[expr]$ *(converts expr to a normal expression from a variety of special forms)*. An example of getting the full content of a thematic block is shown in Fig. 2. Examples of obtaining part of the content with a given element extraction step are shown in Fig. 3 and Fig. 4.

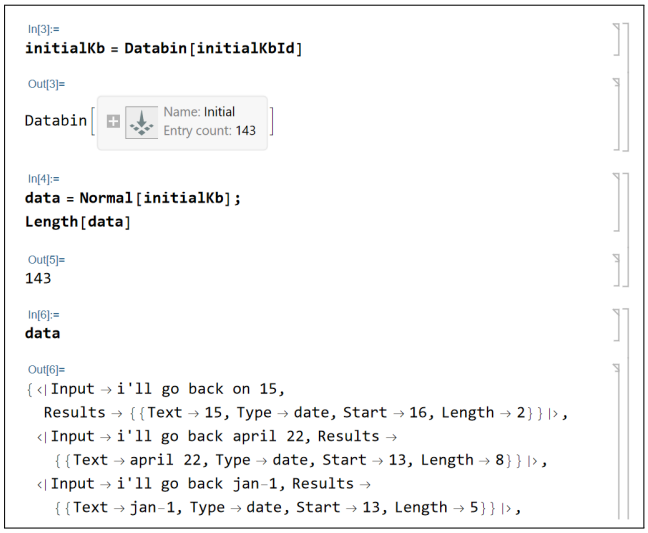

Fig. 2. Extracting all the data from the thematic block.

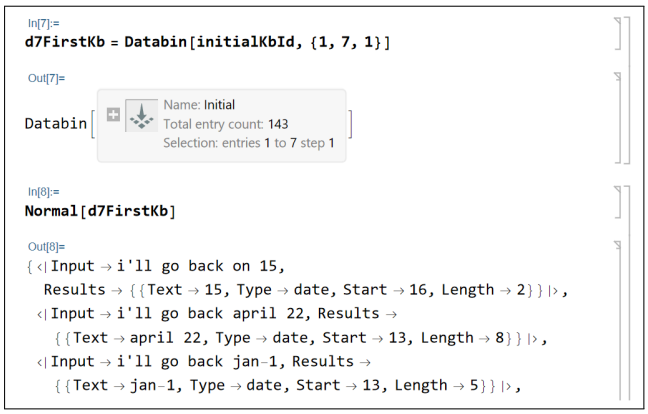

Fig. 3. Extracting elements 1 through 7 from the thematic block.

C. Verifying Functional Completeness of Temporal Marker Extraction Tools

The main steps of checking the functional completeness of the tool for extracting temporal markers from text [7] when integrated with WDR:

1) Save the test set from the thematic block;
2) Run the tested tool (e.g., $mNevod$, $mMRT$);
3) Read the obtained extraction results;
4) Compare with the expected results from the meta-information of the test set;
5) Load the obtained results into WDR.

An example of the result of $mNevod$ is shown in Fig. 5. The form of presentation is the same as that of the $mMRT$ module: for each $Input$ string, the module lists the extracted temporal markers in the $Results$ list in text and numeric form.

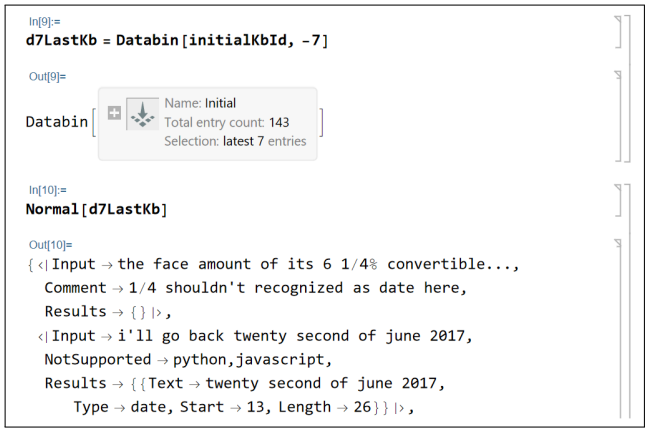

Fig. 4. Extracting the last 7 elements from the thematic block.

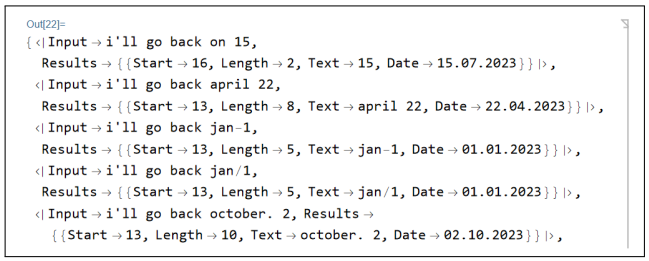

Fig. 5. Example of extraction result by $mNevod$ module.

Note that the Nevod library, due to its structure, provides an additional possibility of using WDR. Nevod is a multi-purpose library designed for searching text for matches with patterns. Patterns are defined independently of the library in a special language of their description, this allows flexible customization of search and extraction of entities from text [17]. In the initial version, to solve the task of extracting temporal markers from text, the standard set for date retrieval from the Nevod library of basic patterns was used. When checking the functional completeness, the shortcomings of this set of patterns were revealed, it was supplemented and included as a component of the $mNevod$ module. Taking into account the independence of patterns from the library, it seems reasonable to place the obtained augmented set of patterns in WDR, which will make publicly available the current version of the set, and at the same time simplify the task of its correction by users.

Conclusion

The example of solving the problem of testing the functional completeness of temporal marker extraction tools shows the integration of a local knowledge base with one of the largest repositories of computable knowledge, Wolfram Knowledgebase, via Wolfram Data Repository (WDR). Data preparation, test set generation based on MS Recognizers DateExtractor is done with Wolfram Mathematica tools. Creation of a new WDR thematic block was described, and the possibility of placing not only the test set but also the configuration of individual time index extraction tools in the WDR.

References

[1] “Tekhnologiya kompleksnoi podderzhki zhiznennogo tsikla semantich-eski sovmestimykh intellektual’nykh komp’yuternykh sistem novogo pokoleniya”, V. Golenkov, Ed., Minsk: Bestprint, 2023, 1064 p.
[2] “Wolfram Mathematica: Modern Technical Computing”. Wolfram. https://www.wolfram.com/mathematica/ (accessed August 14, 2023).
[3] V.B. Taranchuk “Integration of computer algebra tools into OSTIS applications” in Open Semantic Technologies for Intelligent Systems (OSTIS-2022), Research Papers Collection, issue 6, pp. 369–374, 2022.
[4] “Nevod is a language and technology for pattern-based text search”. GitHub. https://github.com/nezaboodka/nevod (accessed Aug 2, 2022).
[5] T.V. Batura, A.M. Bakiev “Metody i sistemy avtomaticheskogo referirovaniya tekstov”, Novosibirsk, A.P. Ershov Institute of Informatics Systems, 2019, 110 p.
[6] S.F. Lipnitsky “Mathematical Model Of The Synthesis Of Texts Based On Merging Of Communicative Fragments”. Problems of Physics, Mathematics and Technics, no 4, Dec. 2018, pp. 106–110.
[7] V.A. Savenok and V.B. Taranchuk, “Features and tools of the Nevod library in solving problems of extracting temporal markers in the text,” Problems of Physics, Mathematics and Technics, no 4, pp. 84–92, Dec. 2022, doi: 10.54341/20778708_2022_4_53_84.
[8] E.A. Suleimanova, “Semantic analysis of contextual dates. Program systems: theory and applications”, 2015, vol. 6, no 4, pp. 367–399.
[9] “Microsoft Recognizers Text Overview”. Microsoft GitHub. https://github.com/microsoft/Recognizers-Text (accessed July 25, 2023).
[10] “Intelligent Virtual Agents and Bots”. Microsoft Power Virtual Agents. https://powervirtualagents.microsoft.com/en-us/ (accessed August 20, 2023).
[11] “Recognizers-Text/Specs/DateTime/English/DateExtractor.json at master”. Microsoft GitHub. https://github.com/microsoft/Recognizers-Text/blob/master/Specs/DateTime/English/DateExtractor.json (accessed August 20, 2023).
[12] Method of Searching for Matches with Patterns in Text, by D.A. Surkov, K.A. Surkov, Y.M. Chetyrko, I.V. Shimko, and V.A. Savionok (2020, March 31) Eurasian patent 037156 [Online]. Available: https://old.eapo.org/ru/publications/publicat/viewpubl.php?id=037156
[13] “Wolfram Data Repository: Computable Access to Curated Data”. Wolfram Data Repository. https://datarepository.wolframcloud.com (accessed September 3, 2023).
[14] “Launching the Wolfram Data Repository: Data Publishing that Really Works”. Stephen Wolfram Writings. https://writings.stephenwolfram.com/2017/04/launching-the-wolfram-data-repository-data-publishing-that-really-works/ (accessed September 3, 2023).
[15] “Data Semantics”. Wolfram Datadrop Quick Reference. https://www.wolfram.com/datadrop/quick-reference/data-semantics/ (accessed July 14, 2023).
[16] “Databin”. Wolfram Language Documentation. https://reference.wolfram.com/language/ref/Databin.html (accessed July 14, 2023).
[17] “Nevod Basic Patterns”. Nezaboodka GitHub. https://github.com/nezaboodka/nevod-patterns (accessed September 5, 2023).