



7-14-2022

Working with VanderBot to Add Multilingual Content (in English and Arabic) to Wikidata

Anchalee Panigabutra-Roberts
University of Tennessee, Knoxville, apanigab@utk.edu

Steve Baskauf
Vanderbilt University, steve.baskauf@vanderbilt.edu

Iman Dagher
University of California, Los Angeles, idagher@library.ucla.edu

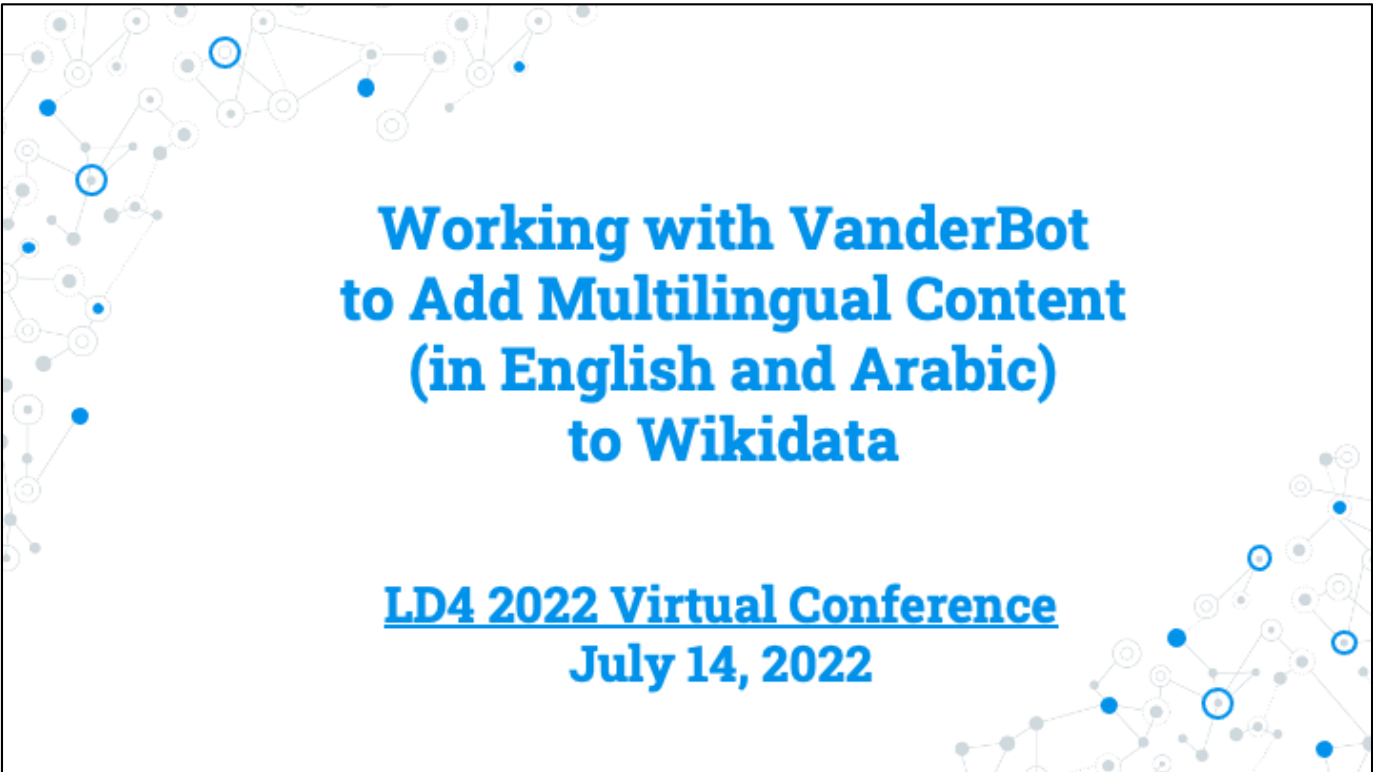
Follow this and additional works at: https://trace.tennessee.edu/utk_libfac

 Part of the [Cataloging and Metadata Commons](#)

Recommended Citation

Panigabutra-Roberts, Anchalee; Baskauf, Steve; and Dagher, Iman, "Working with VanderBot to Add Multilingual Content (in English and Arabic) to Wikidata" (2022). *UT Libraries Faculty: Other Publications and Presentations*.
https://trace.tennessee.edu/utk_libfac/12

This Presentation is brought to you for free and open access by the University Libraries at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in UT Libraries Faculty: Other Publications and Presentations by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.



Working with VanderBot to Add Multilingual Content (in English and Arabic) to Wikidata

LD4 2022 Virtual Conference
July 14, 2022

Hello. My name is Anchalee Panigabutra-Roberts with the Thai nickname Joy. I am the Head of Cataloging at the University of Tennessee Libraries in Knoxville, Tennessee, United States. I have a privilege to work with two of my senior colleagues to work on this presentation on “Working with VanderBot to Add Multilingual Content in English and Arabic to Wikidata.”



Steve Baskauf


Data Science and Data Curation Specialist
Jean & Alexander Heard Libraries
Vanderbilt University
Nashville, Tennessee, U.S.A.
steve.baskauf@Vanderbilt.Edu

Iman Dagher

Arabic & Islamic Studies Metadata
Librarian
University of California, Los Angeles
Los Angeles, California, U.S.A
idagher@library.ucla.edu

Anchalee Panigabutra-Roberts

Head of Cataloging
The University of Tennessee Libraries
Knoxville, Tennessee, U.S.A.
apanigab@utk.edu



Steve Baskauf, Data Science and Data Curation Specialist at Vanderbilt University and Iman Dagher, Arabic & Islamic Studies Metadata Librarian at the University of California, Los Angeles and I collaborated on this project to apply VanderBot for works of translations in English from the original in Arabic.

Steve and I will go through the presentation and we will be happy to answer questions from Q&As at the end of our presentation.

I also like to acknowledge Iman who also attends this session.

Outline

- The project's background and goals
- VanderBot and workflow
- Object-relationship data modeling for works of translation
- Case examples of English translations of the original works in Arabic and the workflow
- Reflection on required workload, quality control issues and required skill sets
- Conclusion & additional Work for VanderBot

Background

- ❖ Joy's and Iman's mutual interest in Wikidata and in Medieval Islamic science and technology
- ❖ Joy's project to learn how to apply VanderBot (from Steve's [blog](#)) to work with multilingual content in Arabic and English, as a fellow at the [IDEA Institute on AI 2021](#)

4

My interest in Wikidata came from my general interest in linked data and how it can transform our practices in describing information objects and how we can build the relationships among the objects to trace scholarly communication of ideas. I also got interested in Islamic technology from my past job working for the American University in Cairo Libraries where I first encountered the work of al-Jazari, the Muslim inventor of the elephant clock as a precursor of the modern robotics.

Working in Cairo, Egypt led me to get connected with Iman who shares the interest in Medieval Islamic technology and Wikidata.

I'm thankful for Iman's support for my work and for her contribution to add Arabic information into this and other Wikidata projects.

Steve has also been supportive of my learning to apply VanderBot and very generous to share his time and expertise with us throughout this process.

I heard about VanderBot through LD4 Wikidata Affinity Group and PCC Wikidata Pilot earlier in 2021. But I didn't have a chance to learn how to use it until July 2021 when I attended the IDEA Institute on AI at UTK. I dedicated my time at the institute to learn how VanderBot worked in the Wikidata sandbox. It's not until this year that I could get back into working with Steve and Iman to apply VanderBot to create Wikidata items with the real data and decided as a group to submit a conference proposal to this conference. Thank you again for this opportunity to share with you our initial analysis from this experiment and work-in-progress with VanderBot with works of English translations of the original works in Arabic.

The Goals

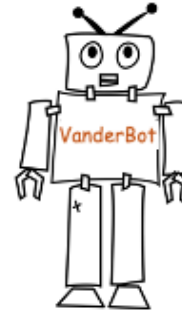
- ❖ To explore tools to add Wikidata items for English translations of original works in Arabic language.
- ❖ To reduce the barrier for Arabic-language speakers to contribute to Wikidata by using spreadsheets to input data into Wikidata.
- ❖ To enrich some underrepresented topics in Wikidata such as Medieval Islamic technology with multilingual data/labels.

Our initial goals are (three goals in the slide).

After we applied VanderBot with our datasets on translations, we learned many lessons from this initial test and will share our lessons learned with you in this presentation.

Next, Steve will present on his work with VanderBot in this project.

VanderBot

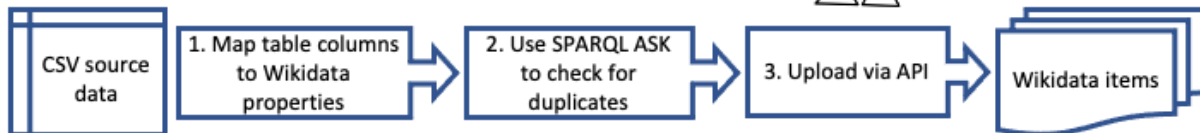
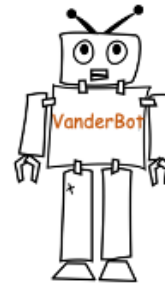


The Python script nicknamed “VanderBot” is not really a bot in the typical sense because it doesn’t operate on its own. Rather, it is a tool that works alongside a human to reduce the amount of time and work required to create items, statements, and references.

Upload workflow in Python script

```
def upload_csv():  
    "upload_csv.py"  
    "outfile": "  
    {  
        "manage_descriptions": True,  
        "label_description_language_list": [  
            "en", "ar"  
        ]  
    },  
    "output_file_name": "bananas3.csv",  
    "prop_list": [  
        {  
            "pid": "P50",  
            "variable": "author",  
            "value_type": "item",  
            "qual": [],  
            "ref": []  
        },  
        {  
            "pid": "P1476",  
            "variable": "description",  
            "value_type": "string",  
            "qual": [{"qualifier": "P1476"}],  
            "ref": []  
        }  
    ]  
}
```

```
query_string = '''ask where {  
    ?entity rdfs:label ?label_string .  
}'''  
  
response = requests.post(endpoint, data=dict(query=  
data = response.json()) # NOTE: the conversion from  
return data['boolean']  
  
label_string = "كتاب الجول (ترجمة: بطورية, 1978)"  
language = "ar"  
exists = ask_query(label_string, language)
```



The source of the data is a CSV spreadsheet and the end product is Wikidata items whose claims correspond to the data in the CSV.

There are three stages of the process that are relevant to the topic of this talk: mapping the CSV columns to Wikidata properties, checking to make sure that there are no existing items with the same labels and descriptions, and actually uploading the data using the Wikidata application programming interface, or API.

1. Map table columns

Specify languages to be used for labels and descriptions

```
{
  "data_path": "",
  "item_source_csv": "",
  "item_pattern_file": "",
  "outfiles": [
    {
      "manage_descriptions": true,
      "label_description_language_list": [
        "en", "ar"
      ],
      "output_file_name": "banumusai.csv",
      "prop_list": [
        {
          "pid": "P50",
          "variable": "author",
          "value_type": "item",
          "qual": [],
          "ref": []
        },
        {
          "pid": "P1476",
          "variable": "title",
          "value_type": "monolingual_text",
          "language": "en",
          "qual": [],
          "ref": []
        },
        {
          "pid": "P855",
          "variable": "translator",
          "value_type": "text",
          "qual": [],
          "ref": []
        }
      ]
    }
  ]
}
```

This simplified configuration file is used to generate a W3C standard mapping file used by VanderBot.

	A	B	C	D	E	F	G	H	I
1	qid	label_en	label_ar	description_en	description_ar	author_uid	author	title_uid	title
2		The Book of Ingenious Devices (1978)	(1978) الترجمة الإنجليزية: كتاب الجول Translation of: Khāb Al-Hiyāl	الترجمة الإنجليزية لكتاب الجول	الترجمة الإنجليزية لكتاب الجول		Q423210		The Book of Ingenious Devices (Khāb Al-Hiyāl)

A simplified JSON configuration file is used to map the labels that are used for the column headers to Wikidata PIDs for the properties whose values are listed in that column. For each property, the type of value, such as item, string, date, etc. is indicated. There is also an opportunity to associate columns containing qualifier data and reference information with that property.

The configuration file is also used to indicate a list of languages in which the labels and descriptions will be provided. In the case of our project, we are providing both English and Arabic, so the language codes “en” and “ar” are listed.

Although the JSON file looks complicated, it isn't that hard to use if you start with an existing file, and copy and paste the appropriate information for your model. Once the configuration is set, this file is used to create both a blank spreadsheet with the appropriate headers and a more complicated standardized file that is used by the VanderBot script to do the actual column mapping.


2. Check for duplicates

```
1 import requests
2
3 def ask_query(label_string, language):
4     endpoint = 'https://query.wikidata.org/sparql'
5     request_header = {
6         'Accept': 'application/json',
7         'Content-Type': 'application/x-www-form-urlencoded',
8         'User-Agent': 'vanderbot/1.91'
9     }
10    query_string = '''ask where {
11        ?entity rdfs:label ""'+ label_string + '"""@' + language + '''
12    }'''
13
14    response = requests.post(endpoint, data=dict(query=query_string), headers=request_header)
15    data = response.json() # NOTE: the conversion from JSON to Python data structure turns JSON true
16    return data['boolean']
17
18 label_string = '(1978، ترجمة إنجليزية، كتاب الحيل'
19 language = 'ar'
20 exists = ask_query(label_string, language)
21 print(exists)
```

Using this format is necessary for successful queries with non-Latin characters.

ASK SPARQL query form determines whether a pattern exists.

Scan for blog post on this topic.



Wikidata items must have unique label/description combinations in each language.

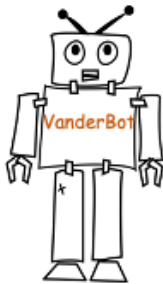
Wikidata only allows one item to have a particular combination of label and description in a language. If you try to create a new item that duplicates the label and description of an existing item, the API will throw an error. For this reason, before attempting to create each new item, the VanderBot script sends SPARQL queries to the Wikidata Query Service to find out if the label/description combination in any of the target languages already exists. Most people who use the Query Service are familiar with SELECT queries, which retrieve information. In this case, we use a less common query form called ASK. An ASK query simply asks whether a particular pattern exists in the Wikidata graph. The response from the Query Service is either True or False.

If the Query Service is asked whether the label/description combination in a row in the spreadsheet already exists and the response is True, VanderBot will skip trying to upload that row and log a warning.

In the process of working with non-Latin character sets, we made several interesting observations. One is that the method used to send the query to the Query Service is important. If raw UTF characters are sent, they sometimes fail to match identical strings in the Wikidata graph. Instead, the non-Latin labels and descriptions must be sent as URL-encoded strings. The other interesting discovery is that two identical-looking strings written in Arabic script may not match if one is encoded using the Arabic character set and the other is encoded using the Farsi character set. This can happen if a Persian keyboard is used to enter an Arabic-tagged string. For more details about querying Wikidata using Python, scan the QR code to read a blog post on the subject.

3. Upload via API

	A	B	C	D	E	F	G	H	I
1	qid	label_en	label_ar	description_en	description_ar	author_uid	author	title_uid	title
2		The Book of Ingenious Devices (1978)	(1978. ترجمة إنجليزية) كتاب الخيل Translation of: Khāb Al-Hiyāl	الترجمة الإنجليزية لكتاب الخيل		Q423210			The Book of ingenious Devices (Kitāb Al-Hiyāl)



vanderbi.lt/vanderbot

```
[[{"qid": "Q423210", "label_en": "The Book of Ingenious Devices (1978)", "label_ar": "(1978. ترجمة إنجليزية) كتاب الخيل", "description_en": "Translation of: Khāb Al-Hiyāl", "description_ar": "الترجمة الإنجليزية لكتاب الخيل", "author_uid": "Q423210", "author": "The Book of ingenious Devices (Kitāb Al-Hiyāl)", "title_uid": "Q423210", "title": "The Book of ingenious Devices (Kitāb Al-Hiyāl)"}]]
```

W3C standard mapping file

CSV spreadsheet

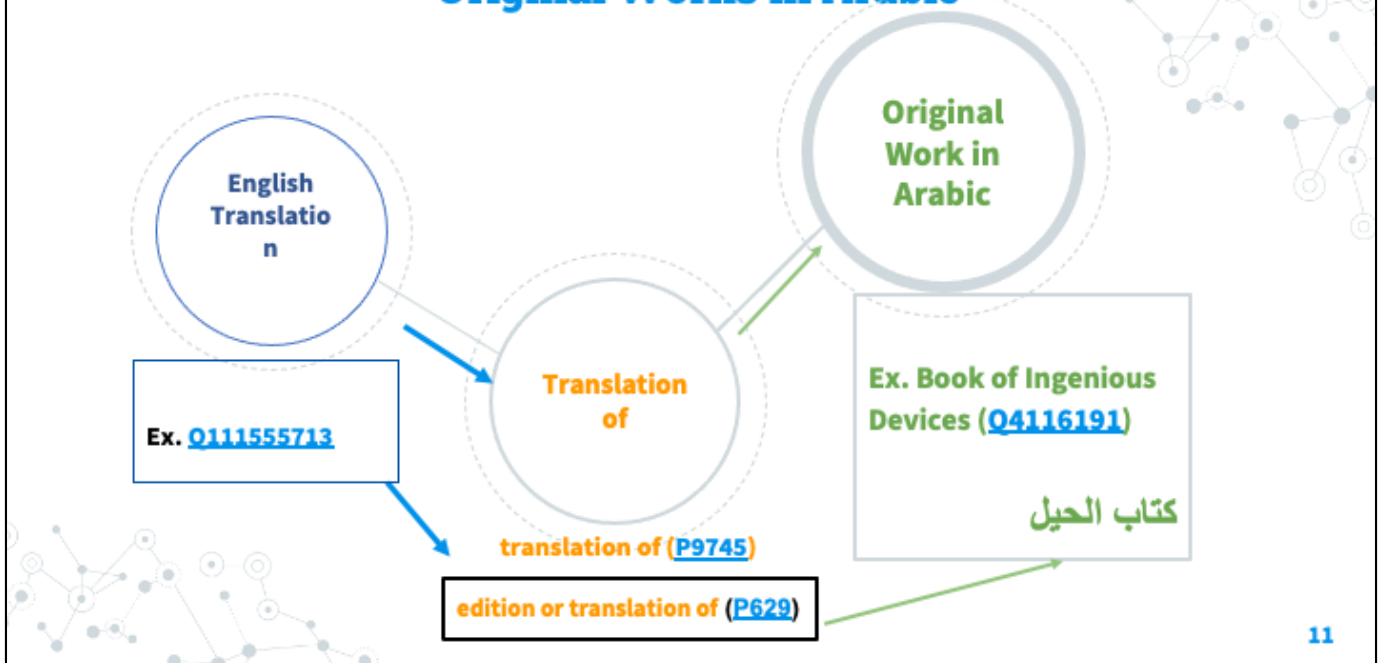
Wikidata API



Wikidata item

Once the check for duplicates is completed, the VanderBot script sends the data to the Wikidata API in the correct format and the new item is created. The script also captures the claim and reference identifiers that are sent back from the API and records them in the CSV so that they can be used in the future if that particular information needs to be changed or deleted.

The Object-Relationship Model for English Translations of Original Works in Arabic



11

Thank you so much, Steve, for your explanation for the technical aspects of working with VanderBot in this project.

The first lesson I learned in this project is that it is important to model the datasets to design the spreadsheets and to decide on the Wikidata requirements for the types of work you are working with.

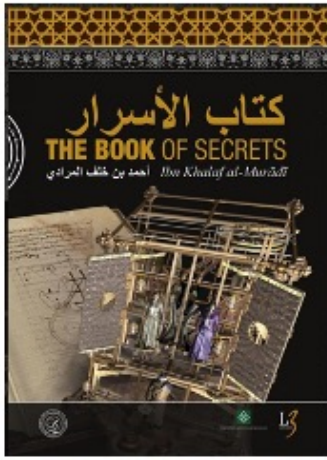
In this case, I chose to work with works of translations from Arabic to English.

This data model illustrates the relationships between the English translations in the triples of English translations as Q Wikidata items, with translation of property P975 or edition or translation of property P629 to connect with the Q item for the original work in Arabic.

And I'd recommend using **edition or translation of (P629)** instead of **translation of (P9745)**, since the inverse relationship **has edition or translation (P747)** works better with **edition or translation of (P629)**

There is the inverse label: **translated as (Q107567880)**, but it is not a property and the case example in Wikidata does not even include this label/item.

Pushing the Limit of Our Data Modeling for VanderBot & Other Problems



[The Book of Secrets in the Results of Ideas: Incredible Machines from 1000 Years Ago \(Q112831378\)](#)

This title includes the English translation and interpretation, the Arabic transcription and interpretation, the facsimile of the Arabic manuscript, and an interactive DVD-ROM (3 volumes and 1 DVD-ROM).

It would require adding the following data elements:

- ⊙ number of parts of this work ([P2635](#))
- ⊙ has part or parts ([P527](#))
 - English translation
 - Arabic transcription
 - facsimile ([Q194070](#)) of
 - Original work (Q item. Ex. [Q3831834](#))
 - DVD-ROM ([Q2144513](#))

An example to base our new/extended data model on:

The album amicorum of Jacob Heyblocq : introduction, transcriptions, paraphrases & notes to the facsimile ([Q82260552](#))

12

Steve and I tweaked the spreadsheet based on an example as shown in Steve's presentation. Then he used the data Iman and I supplied to run VanderBot for additional two items for this project. Then I verified the results and analyzed our work based on the three results, so far. And here are some of my observations. For this title, I found that it is more complex than our current data model for a single translation from Arabic to English can be used to represent it. The set has three books; the English translation and interpretation, the Arabic transcription and interpretation, the facsimile of the Arabic manuscript, and an interactive DVD-ROM.

To completely represent this title's set, I would need to add the following properties and data:

- number of parts of this work ([P2635](#))
- has part or parts ([P527](#))
- English translation
- Arabic transcription
- facsimile ([Q194070](#)) of Original work (Q item. Ex. [Q3831834](#))
- DVD-ROM ([Q2144513](#))

I found this example in Wikidata to learn about these additional properties from the Wikidata item **Q82260552**.

We also need to add another value for: **Instance of: book (Q571)** to all of our Wikidata items.

Not All Required Data Nor Desired Properties Are Readily Available

[The Book of Secrets in the Results of Ideas: Incredible Machines from 1000 Years Ago \(Q112831378\)](#)

The manuscript's information from **Dr. Silvia Scipioni**, Director of Biblioteca Medicea Laurenziana, Florence, Italy.

The manuscript has the shelfmark '**Orientali 152**' with this bibliographic information:

<http://opac.bmlonline.it/Bibliografia.htm?idlist=0&record=445712426399>

Two translators are not in Wikidata:

Dr. Ahmed Ragab, Associate Professor, Institute of the History of Medicine, the Johns Hopkins University has his LC Name Authority: <http://id.loc.gov/authorities/names/no2016003240>

Email confirmation that he is the translator of this book along with **Dr. Soha Bayoumi**: https://scholar.harvard.edu/files/sbayoumi/files/soha_bayoumi_cv_0.pdf

Dr. Soha Bayoumi is listed in Wikidata as the *author name string* in

<https://www.wikidata.org/wiki/Q39042839>

*There is no equivalent property for: *translator name string*.

For additional research for this English translation of al-Muradi's work and its original text in Arabic, I contacted the Biblioteca Medicea Laurenziana of Florence, Italy to ask for the manuscript's information and received the manuscript's information from Dr. Silvia Scipioni, the director, that the manuscript has the shelfmark Orientali 152 and included the bibliographic information in her email:

<http://opac.bmlonline.it/Bibliografia.htm?idlist=0&record=445712426399>

I also emailed Dr. Ahmed Ragab to confirm that he is the translator of this work with Dr. Soha Bayoumi, but they both don't have Wikidata items we could readily use.

Dr. Ragab has his LC name authority that can be used to create a Wikidata item.

Dr. Soha Bayoumi has her name in Wikidata in the author name string, so she also needs to have a Wikidata item created for her.

It would be good if the Wikidata property for '**translator name string**' can be created for future use.

The Book of Instruction in the Elements of the Art of Astrology (1998)
([Q112831309](#))

Originally published:

London : Luzac, 1934.
(OCLC no. [3109131](#))

Publisher not in Wikidata:

*Institut für Geschichte der
Arabisch-Islamischen
Wissenschaften (Frankfurt am
Main, Germany):*
[http://id.loc.gov/authorities/names
/n86023255](http://id.loc.gov/authorities/names/n86023255)

Translator: *Robert Ramsay Wright*,
zoologist ([Q7349157](#))

[Wikipedia](#)'s reference to his life and works
in the *Dictionary of Canadian Biography*:
[http://www.biographi.ca/en/bio/wright_robe
rt_ramsay_16E.html](http://www.biographi.ca/en/bio/wright_robert_ramsay_16E.html)



Credit for the book cover image:
Goodreads: [tiny.utk.edu/yod1S](https://www.goodreads.com/user/show/10411111-tiny)

14

For the ***Book of Instruction in the Elements of the Art of Astrology (1998)***, it is a reprint of the original translation in 1934.

In Wikidata, I revised the title to add (1998) to distinguish this reprint edition from its original publication in 1934.

I could not add the publisher due to the lack of Wikidata item for the publisher that I need to create later.

I did not add the translator in VanderBot due to missing his information on this publication, when I found his Wikipedia page including his biographical information, but with no mentioning of his background on this subject nor his works of translations.

It was not until I explored further through the references listed in his Wikipedia that I would find this translation. It was mentioned in his [Wikipedia](#)'s reference, in the *Dictionary of Canadian Biography*:

http://www.biographi.ca/en/bio/wright_robert_ramsay_16E.html The original translation in 1934 is listed on his list of publications at the end of his biography. The biography itself does not include his expertise in Arabic language nor anything else related to this work. This discovery led me to his Wikidata item [Q7349157](#) to add him as the translator to this translation's Wikidata item after we already created it with VanderBot.

Adding More Values to Existing Properties

Example: Adding more main subjects to the newly created Wikidata item

<https://www.wikidata.org/wiki/Q111555713>

qid	label_en	mainSubject_uuid	mainSubject
Q111555713	The Book of Ingenious Devices (1978)	8873FAB0-1B40-4620-9965-F63865A67B87	Q101333
Q111555713	The Book of Ingenious Devices (1978)		Q184199
Q111555713	The Book of Ingenious Devices (1978)		Q43302
Q111555713	The Book of Ingenious Devices (1978)		Q170196

15

Another known issue is that VanderBot won't work with multiple values for the same property.

The workaround is to add additional values to the same property, in this case, the main subject, AFTER we completed the initial upload by VanderBot to create the item.

This example shows how we created the second spreadsheet to put the Q item information for the newly created item in the CSV file to add more main subjects.

****Adding the existing main subject in Wikidata is to prevent Wikidata from creating a new item from the additional values alone.**

Reflection on Required Workload and Quality Control Issues

- ◎ The data are mostly Q item numbers; missing visual clues to do quality control based on the textual description and metadata.
- ◎ Quality control can be done after VanderBot processed the data from the spreadsheet.
- ◎ Not all required information are readily available as Wikidata items and may need to be created manually.
- ◎ Required elements for some complex item description may not be included in the data model and need to be manually input into Wikidata.

It still requires some extensive research.

Required Skill Sets to Work with VanderBot



Required skills to input the data into the spreadsheets

Basic knowledge of Wikidata.

Recommendation

Learn how to create Wikidata items before working with the spreadsheets to input data to create Wikidata items from the batchload.

Working as a team

In a team setting, it maybe a good idea to include a team member who has the basic knowledge of Python and Wikidata to process the data with VanderBot.

Conclusion

- © For works of translations, VanderBot can work with the basic data requirements.
- © For more complex translation works (e.g. works including facsimiles of the original texts), the model can be modified to add more Wikidata properties, or to manually input additional data elements.
- © Working with VanderBot with spreadsheets for translations still requires a basic understanding of Wikidata and familiarity with the best metadata practices for translations, their original works, and also for manuscripts in our cases.
- © For multilingual descriptions, it also requires expertise in the language and culture of the original works.

Additional Work to Complete the Data Model for Translations



[Manuscript Properties](#)

Per Iman's recommendation, we will consult the manuscript properties from the Wikidata:WikiProject Books to form our data modeling for VanderBot for the manuscript of the original work.

References



Baskauf, Steven J. *LD4 VanderBot Tutorial*, <https://heardlibrary.github.io/digital-scholarship/script/wikidata/ld4/>

Baskauf, Steven J. and Jessica K. Baskauf, "Using the W3C *Generating RDF from Tabular Data on the Web* Recommendation to Manage Small Wikidata Datasets," *Semantic Web*, September 7, 2021, <https://content.iospress.com/articles/semantic-web/sw210443>

British Library, "Arabic scientific manuscripts go live in Qatar Digital Library," *Asian and African Studies Blog*, November 3, 2014, <https://blogs.bl.uk/asian-and-african/2014/11/arabic-scientific-manuscripts-go-live-in-qatar-digital-library.html>

Court, John P. M., "WRIGHT, ROBERT RAMSAY, zoologist, university professor and administrator, and author," in *Dictionary of Canadian Biography*, vol. 16, University of Toronto/Université Laval, 2003–, http://www.biographi.ca/en/bio/wright_robert_ramsay_16E.html



Websites accessed on: July 14, 2022

References (2)

IDEA Institute on Artificial Intelligence (2021), *Panigabutra-Roberts was a fellow funded by the Institute of Museum and Library Services grant: RE-246419-OLS-20*, <https://idea.infosci.utk.edu/institute2021/fellows2021/>

Unicode converter: <https://www.branah.com/unicode-converter>

University of Tennessee, Knoxville, "Anchalee Panigabutra-Roberts," *Discover Our Faculty website*, <https://faculty.utk.edu/Anchalee.Panigabutra-Roberts/publications>

W3C, "1.1. Graph-based Data Model," *RDF 1.1 Concepts and Abstract Syntax: W3C Recommendation 25 February 2014*, <https://www.w3.org/TR/rdf11-concepts/#data-model>

Wikimedia Foundation, "Manuscript properties," *Wikidata:WikiProject Books*, https://www.wikidata.org/wiki/Wikidata:WikiProject_Books#Manuscript_properties

Wikimedia Foundation, *Wikidata:Introduction*, <https://www.wikidata.org/wiki/Wikidata:Introduction>

Websites accessed on: July 14, 2022



Thank you!

Any questions?



Credits

Special thanks to all the people who made and released these awesome resources for free:

- ◎ Presentation template by [SlidesCarnival](#)
- ◎ Photographs by [Unsplash](#)