Wikipedia Tools for Google Spreadsheets

Thomas Steiner
Google Germany GmbH
ABC Str. 19, 20354 Hamburg, Germany
tomac@google.com

ABSTRACT

In this paper, we introduce the *Wikipedia Tools for Google Spreadsheets*. Google Spreadsheets is part of a free, Web-based software office suite offered by Google within its Google Docs service. It allows users to create and edit spreadsheets online, while collaborating with other users in real-time. Wikipedia is a free-access, free-content Internet encyclopedia, whose content and data is available, among other means, through an API. With the *Wikipedia Tools for Google Spreadsheets*, we have created a toolkit that facilitates working with Wikipedia data from within a spreadsheet context. We make these tools available as open-source on GitHub,[1] released under the permissive Apache 2.0 license.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Web-based services

Keywords

Wikipedia, Wikidata, Google Spreadsheets, Google Sheets

1. INTRODUCTION

In the world of Computer Science, *spreadsheet* applications serve for the organization, analysis, and storage of data in tabular form. Spreadsheets are the computerized simulation of paper accounting worksheets, and operate on data represented as *cells of an array*, organized in rows and columns. Cells can contain numeric or textual data, or the results of *formulas* that automatically calculate and display a value based on the contents of other cells. With the *Wikipedia Tools for Google Spreadsheets*, we introduce a toolkit of such formulas, tailored to the universe of Wikipedia, that enables a wide range of potential use cases starting from marketing, to search engine optimization, to business analysis. Especially through the *chaining* of formulas, the true power and ease of spreadsheet applications can be unleashed.

[1] *Wikipedia Tools for Google Spreadsheets*: https://github.com/tomayac/wikipedia-tools-for-google-spreadsheets

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2891112

1.1 Wikipedia and Wikidata

Wikipedia's content and data is available through the Wikipedia API (https://{language}.wikipedia.org/w/api.php), where `{language}` represents one of the currently 291 supported Wikipedia languages,[2] for example, `en` for English, `de` for German, or `zu` for Zulu. Wikidata is a collaboratively edited knowledge base and intended to provide a common source of structured data which can be used by projects such as Wikipedia. Its content and data is available through the Wikidata API (https://www.wikidata.org/w/api.php). Both the Wikipedia and the Wikidata APIs' data is available as XML or JSON, among other formats. Wikipedia pageviews data, *i.e.*, the number of times within a given period of time that a given Wikipedia article has been viewed can be obtained using the Pageviews API (https://wikimedia.org/api/rest_v1/?doc). The data is available in JSON format.

1.2 Google Spreadsheets and Apps Scripts

Google Spreadsheets can be extended with custom functions (or formulas) using Google Apps Scripts[3] that are written in standard JavaScript.[4] To illustrate this, a trivial function is defined in Listing 1 that can then be used from within a spreadsheet as outlined in Listing 2. Custom functions can access external resources on the Web by fetching URLs with the `UrlFetchApp`, one of the scripting services available in Google Apps Script. Fetched data can either be in XML or JSON format and parsed with convenience functions.

```
function DOUBLE(input) {
  return input * 2;
}
```

Listing 1: Custom Google Sheets function called `DOUBLE`.

```
=DOUBLE(A1)
```

Listing 2: Usage of the custom `DOUBLE` function from Listing 1 in a cell with the value of cell `A1` as a parameter.

2. LIST OF DEVELOPED FUNCTIONS

In our *Wikipedia Tools for Google Spreadsheets*, we provide eleven functions that—in traditional spreadsheets style—follow an all-uppercase naming convention and start with

[2] List of Wikipedias: https://meta.wikimedia.org/wiki/List_of_Wikipedias
[3] Google Apps Script: https://developers.google.com/apps-script/
[4] Custom functions in Google Sheets: https://developers.google.com/apps-script/guides/sheets/functions

a WIKI prefix. These functions are wrappers around the particular Wikipedia or Wikidata API calls, or the Pageviews API respectively. Figure 1 shows exemplary output for the English Wikipedia article https://en.wikipedia.org/wiki/Berlin and the English Wikipedia category https://en.wikipedia.org/wiki/Category:Berlin. The functions are listed below.

WIKITRANSLATE Returns Wikipedia translations (language links) for a Wikipedia article.

WIKISYNONYMS Returns Wikipedia synonyms (redirects) for a Wikipedia article.

WIKIEXPAND Returns Wikipedia translations (language links) and synonyms (redirects) for a Wikipedia article.

WIKICATEGORYMEMBERS Returns Wikipedia category members for a Wikipedia category.

WIKISUBCATEGORIES Returns Wikipedia subcategories for a Wikipedia category.

WIKIINBOUNDLINKS Returns Wikipedia inbound links for a Wikipedia article.

WIKIOUTBOUNDLINKS Returns Wikipedia outbound links for a Wikipedia article.

WIKIMUTUALLINKS Returns Wikipedia mutual links, i.e, the intersection of inbound and outbound links for a Wikipedia article.

WIKIGEOCOORDINATES Returns Wikipedia geocoordinates for a Wikipedia article.

WIKIDATAFACTS Returns Wikidata facts for a Wikipedia article.

WIKIPAGEVIEWS Returns Wikipedia pageviews statistics for a Wikipedia article.

WIKIPAGEEDITS Returns Wikipedia pageedits statistics for a Wikipedia article.

Most functions directly wrap native API calls, with three exceptions: *(i)* the functionality of the WIKISYNONYMS and the WIKITRANSLATE functions is combined in the WIKIEXPAND function, both the WIKITRANSLATE and the WIKIEXPAND function accept an optional target languages parameter that allows for limiting the output to just a subset of all available Wikipedia languages; *(ii)* the function WIKIMUTUALLINKS is the intersection of the two functions WIKIINBOUNDLINKS and WIKIOUTBOUNDLINKS; and *(iii)* the function WIKIDATAFACTS provides a list of claims [11] (or facts), enriched with entity and property labels for improved readability, limited to single-value objects, and simplified using an adapted version of Maxime Lathuilière's simplifyClaims function[5] from his Wikidata SDK [6]. This allows us to return two columns—in RDF [2] terms "predicate" and "object" pairs—with one unique object, for example, the predicate ISO 3166-2 code with the object DE-BE, and deliberately discarding multi-value claims, for example, predicate head of government with objects Michael Müller and Klaus Wowereit, among many others. While in the concrete example the ordering is clear (temporal), this is not true in the general case, for example, with predicate instance of. As a result, in WIKIDATAFACTS, we prefer indisputability of claims over their completeness. Listing 3 exemplarily shows the complete implementation of the WIKISYNONYMS function.

[5]Wikidata SDK simplifyClaims function: https://github.com/maxlath/wikidata-sdk#simplify-claims-results

```
/**
 * Returns Wikipedia synonyms
 * @param {string} article The Wikipedia article
 * @return {Array<string>} The list of synonyms
 */
function WIKISYNONYMS(article) {
  'use strict';
  if (!article) {
    return '';
  }
  var results = [];
  try {
    var language = article.split(/:(.+)?/)[0];
    var title = article.split(/:(.+)?/)[1];
    if (!title) {
      return '';
    }
    title = title.replace(/\s/g, '_');
    var url = 'https://' + language +
        '.wikipedia.org/w/api.php' +
        '?action=query' +
        '&blnamespace=0' +
        '&list=backlinks' +
        '&blfilterredir=redirects' +
        '&bllimit=max' +
        '&format=xml' +
        '&bltitle=' +
        encodeURIComponent(title);
    var xml = UrlFetchApp.fetch(url)
        .getContentText();
    var document = XmlService.parse(xml);
    var entries = document.getRootElement()
        .getChild('query').getChild('backlinks')
        .getChildren('bl');
    for (var i = 0; i < entries.length; i++) {
      var text = entries[i].getAttribute('title')
          .getValue();
      results[i] = text;
    }
  } catch (e) {
    // no-op
  }
  return results.length > 0 ? results : '';
}
```

Listing 3: Implementation of WIKISYNONYMS.

3. USAGE SCENARIOS

We have tested the *Wikipedia Tools for Google Spreadsheets* with different usage scenarios in mind. These include, but are not limited to, the ones listed in the following.

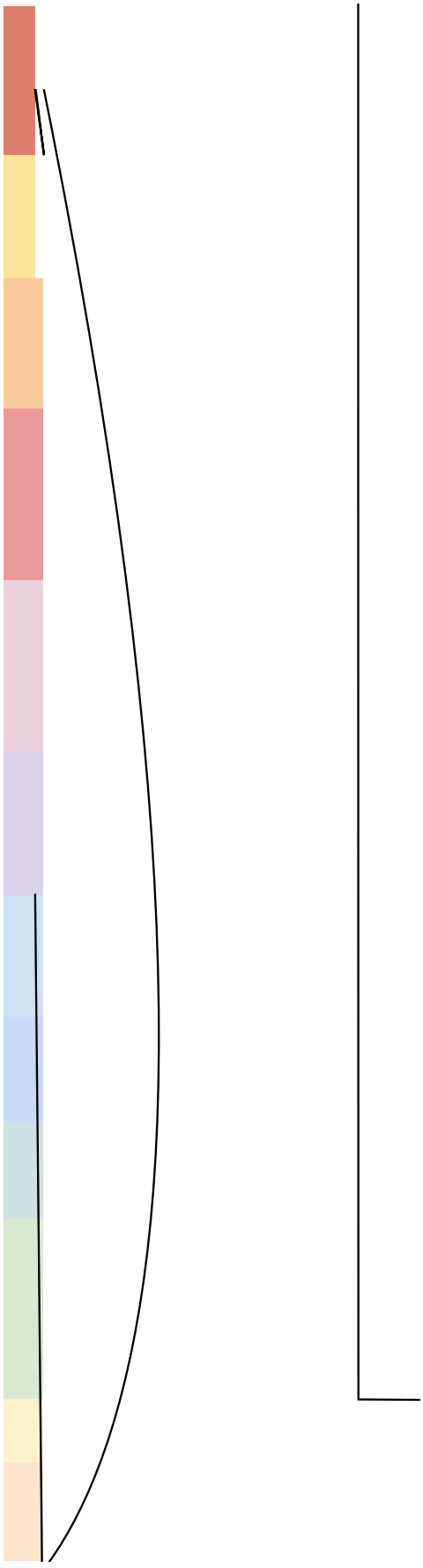

Figure 1: Example output for each function in the *Wikipedia Tools for Google Spreadsheets* (cropped). Live spreadsheet: https://goo.gl/yvbmex.

of an image carousel can be seen in Google's Knowledge Graph [10] Web search results pages when searching for "visitor attractions in montreal" (demo https://goo.gl/Ugt0je).

3.2 Usage Scenario II: Search Ads

Search advertisers can greatly profit from the information that is contained in Wikipedia and Wikidata. For example, if we imagine a hotel booking site, it may be desirable to advertise based on points of interest (POIs) and create advertisements automatically featuring known facts of such POIs. Figure 3 shows an example where skyscrapers listed in the category *skyscrapers over 350 meter*[7] are first obtained via `WIKICATEGORYMEMBERS` and then checked for their "height" fact via `WIKIDATAFACTS`, which is then used in two templates to create ads. Search keywords are generated by calling `WIKISYNONYMS` and combined with terms like "hotel".

3.3 Usage Scenario III: Marketing Campaigns

On January 13, 2016, Google Maps added Street View imagery for the model railway *Miniatur Wunderland*.[8] Taking global Wikipedia pageviews as a popularity indicator, we can examine if the marketing campaign has had any impact on the attraction, assuming that more pageviews translate to increased visitor interest. Therefore, we first obtain the *Miniatur Wunderland* article in all available languages via `WIKITRANSLATE` and then retrieve pageviews via `WIKIPAGEVIEWS`. Figure 4 shows indeed an international uptake of pageviews starting January 13 after an earlier linear curve progression (except for the German article, which had a peak on January 8, a long weekend after a public holiday).

4. RELATED WORK

In his book *Google Apps Script for Beginners* [4], Gabet gives an introduction to extending Google Spreadsheets with custom functions. A similar introduction is given in Ferreira's *Google Apps Script: Web Application Development Essentials* [3]. In [5], Han *et al.* describe their approach *RDF123* to translate spreadsheets data to RDF, the inverse of what we do in `WIKIDATAFACTS`. Olsen and Moser show in [8] how Web APIs can be taught with spreadsheets. The process of calling Web APIs via spreadsheets is further described in [9]. Further, in [1], Abramson *et al.* describe how they enabled spreadsheets to have "super-computing" powers through parallelized custom functions. An open-source toolkit for mining Wikipedia—not bound to spreadsheets, but designed for general use with the Java programming language—is described by Milne *et al.* in [7].

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced the *Wikipedia Tools for Google Spreadsheets*. First, we have introduced the data sources Wikipedia and Wikidata and their different APIs. Second, we have shown how Google Spreadsheets can be extended through custom functions that can then be used from within a cell context as if they were native functions. In the following, we have listed the implemented functions, and explained where they extend the functionality of the underly-

[7]Skyscrapers over 350 meter: https://en.wikipedia.org/wiki/Category:Skyscrapers_over_350_meters

[8]Miniatur Wunderland on Google Street View: https://www.google.com/maps/about/behind-the-scenes/streetview/treks/miniatur-wunderland/

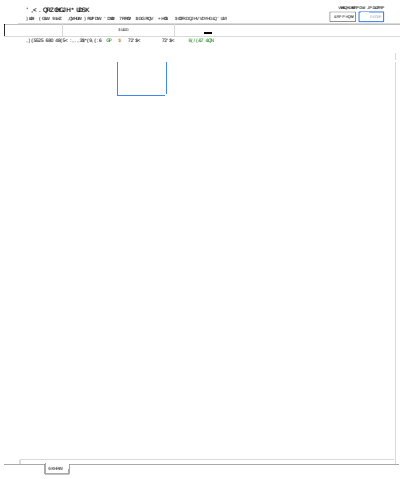

Figure 2: Usage scenario I: *Wikipedia Tools for Google Spreadsheets* used to create an ordered category panel based on Wikipedia category memberships and accumulated Wikipedia pageviews for popularity ranking (here: the top-10 visitor attractions in Montreal). Live spreadsheet: https://goo.gl/Njvt1T.

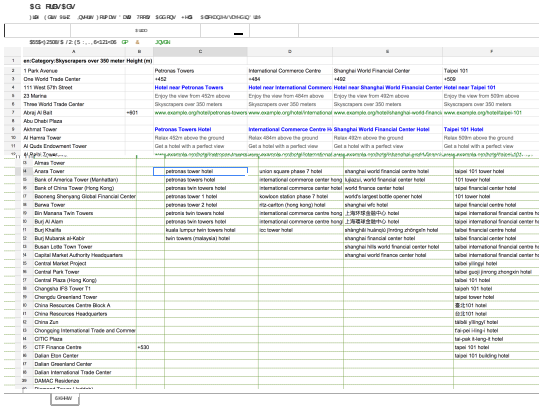

Figure 3: Usage scenario II: *Wikipedia Tools for Google Spreadsheets* used to create textual search ads based on Wikidata facts (here: skyscraper heights) and Wikipedia synonyms as keywords combined with the term "hotel". Live spreadsheet: https://goo.gl/np1Is8.

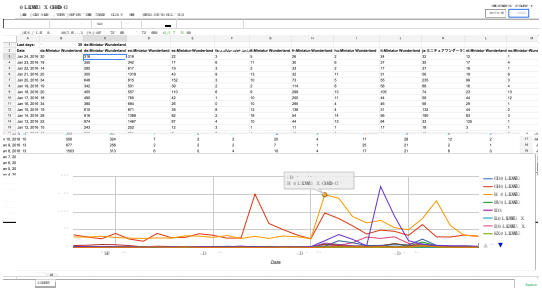

Figure 4: Usage scenario III: *Wikipedia Tools for Google Spreadsheets* used to evaluate the impact of a marketing campaign (here: model railway *Miniatur Wunderland* being featured on Google Street View since January 13, 2016). Live spreadsheet: https://goo.gl/q1yhuV.

ing wrapped API functions. We have then focused on three different usage scenarios that illustrate how to work with the *Wikipedia Tools for Google Spreadsheets* and finally have provided an overlook on related work in the area.

Future work will focus on adding more functions as need be and potentially making the functions more parameterizable. In the current iteration, we have favored simplicity and ease of use over customizability, essentially making the most common use case the only option. Possibly, in upcoming releases, we will add an advanced mode that allows experienced users to fine-tune the functions' results, for example, to implicitly include bot traffic in `WIKIPAGEVIEWS` that we have currently excluded on purpose.

Concluding, we were positively surprised by the increased productivity and short turnaround time enabled by the *Wikipedia Tools for Google Spreadsheets* for the rapid prototyping of ideas, especially in combination with the fill-down and fill-right features in spreadsheets and the charting capabilities. We look forward to making the tools even more powerful and hope to attract collaborators for the open source project available on GitHub at https://github.com/tomayac/wikipedia-tools-for-google-spreadsheets. As a positive side effect, the tools can even help improve Wikipedia and Wikidata when authors add missing data, for example, we added an image to one of the visitor attractions of Montreal, as this fact was initially missing in Wikidata (and thus in Figure 2).

6. REFERENCES

[1] D. Abramson, L. Kotler, D. Mather, and P. Roe. ActiveSheets: Super-Computing with Spreadsheets. In U. Seattle, editor, *Proceedings of the High Performance Computing Symposium { HPC 2001*, pages 110–115, San Diego, USA, 2001.

[2] R. Cyganiak, D. Wood, and M. Lanthaler. RDF 1.1 Concepts and Abstract Syntax. Recommendation, W3C, Feb. 2014.

[3] J. Ferreira. *Google Apps Script: Web Application Development Essentials*. O'Reilly Media, 2014.

[4] S. Gabet. *Google Apps Script for Beginners*. Packt Publishing, 2014.

[5] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi. RDF123: From Spreadsheets to RDF. In *The Semantic Web { ISWC 2008*, volume 5318 of *LNCS*, pages 451–466. Springer, 2008.

[6] M. Lathuilière. Wikidata SDK, 2016. https://github.com/maxlath/wikidata-sdk (2016-02-08).

[7] D. Milne and I. H. Witten. An Open-Source Toolkit for Mining Wikipedia. *Artificial Intelligence*, 194:222–239, Jan. 2013.

[8] T. Olsen and K. Moser. Teaching Web APIs in Introductory and Programming Classes: Why and How. Paper 16, SIGED: IAIM Conference, Feb. 2013.

[9] K. Patel, S. Prish, S. Sadhu, L. Bizek, and X. Pan. Spreadsheet Functions to Call REST API Sources, May 15 2014. US Patent App. 13/672,704.

[10] A. Singhal. "Introducing the Knowledge Graph: things, not strings", Official Google Blog, May 2012. http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html.

[11] D. Vrandečić and M. Krötzsch. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM*, 57(10):78–85, Sept. 2014.