

Investigating relationships between successive URLs

Predictable relationships between consecutive URLs

The hypothesis is that there is no relationship between the sequences of government GOVUK website URLs visited by a user. Clicks are essentially randomly distributed, according to their availability. There is a natural degree of accuracy, of a naive prediction, just due the layout of the government website.

In summary:

- there is a [65%] chance that the next URL will be found in the next [10] recommended URLs
- the Mean Reciprocal Rank (MRR) for the predictions is [0.43] on a holdout test set

These results should not be regarded in terms of good or bad, rather they just establish a reference score. This serves as a benchmark to be beaten in future tailored predictions, which incorporate additional explanatory data regarding the user. This enables meaningful discussion and development of future models. These measures should be seen as the naive or random prediction threshold. Any future models should be better than this measure.

Data used

The data used in the analysis is pulled from a single hour of usage (midday) of the GovUK website on 2020-11-13. This contains 844k unique records. As can be seen beneath, for a single `fullVisitorId`, there can be multiple values for `visitId`. It relates to hits which are classified as `PAGE`.

Illustration of data

```
visit_data[visit_data['fullVisitorId'] == 3250250007886851499]
```

	date	fullVisitorId	visitId	hour	minute	hitNumber	pagePath	pageTitle
185908	20201113	3250250007886851499	1605268476	12	2	10	/government/news/be-passport-aware-check-passp...	Be Passport Aware: Check passport expiry befor...
36	20201113	3250250007886851499	1605269124	12	5	1	/government/organisations/hm-revenue-customs/c...	Income Tax: general enquiries - GOV.UK
37	20201113	3250250007886851499	1605269124	12	7	2	/government/organisations/hm-revenue-customs	HM Revenue & Customs - GOV.UK
38	20201113	3250250007886851499	1605269124	12	12	3	/search/all?keywords=personal+tax+account&orde...	personal tax account - Search - GOV.UK
39	20201113	3250250007886851499	1605269124	12	12	6	/personal-tax-account	Personal tax account: sign in or set up - GOV.UK
40	20201113	3250250007886851499	1605269124	12	12	7	/personal-tax-account/sign-in/prove-identity	Prove your identity to continue - GOV.UK
322076	20201113	3250250007886851499	1605270057	12	20	1	/government/organisations/hm-revenue-customs	HM Revenue & Customs - GOV.UK
322077	20201113	3250250007886851499	1605270057	12	21	2	/log-in-register-hmrc-online-services	HMRC services: sign in or register - GOV.UK

As seen, `pagePath` represents the URL visited.

Rationale for considering only the preceding website

Typically in time series processes, the most important data point is the current one, in determining the next data point. In a memoryless process, the only data point which is relevant is the current one and the

following data point will be a random jump to one of the connected points in the space. In the context of the GovUK website, it is frequently the case that a visitor can only visit the next URL via the current URL.

Conditioning on time

As we are focussing on the single hour of data, we have implicitly conditioned on time. Different predictions would be made for different slices of time. This would mean that this approach could incorporate non-stationary effects and the impact of current events. As will be discussed beneath, the model could be run in real time, as it is effectively relying on aggregations and dictionary data access which is inherently fast.

Conditioning on location

Location is not considered here, but would be possible given that there is information about geolocation of the user. This simply requires consideration of aggregation weights as a function of distance, discounting information from users further away.

Preparing the lagged data

The data requires a lagged URL to be created in order to see the link between current and past data. We are trying to predict the URL to be visited, by referencing the lagged URL.

Of note, a full 53% of the users on the government website only visit one link. This makes it impossible to create a lagged URL. These users data are removed from the dataset, as we cannot build a prediction conditioned on the past, as there is no history for the user.

Removing login, search and done activity

Any URLs for search or login are removed. Search activity only accounts for [4%] of the remaining data, which when considered in combination with the number of users who only visit one URL, is indicative of external search engines.

Login activity, [6%] of the data, was removed as its a deterministic intermediate link to certain URLs, ie the probability of $P(a \rightarrow \text{login} \rightarrow b)$ is the same as $P(a \rightarrow b)$, as $P(\text{login} \rightarrow b)$ is 100%. Which in turn would mean that it obscures the relationship $P(a \rightarrow b)$. This admittedly ignores the losses from people who forget their login details.

Done activity was also removed, approximately a further [1%] of the data. The rationale for removing done activity was that it just represents the completion of an activity or form and is not helping the user access information.

Circular journeys

Where the next URL is the same as the preceding URL, the data is removed.

Aggregated data - top sites

```
pd.DataFrame(visit_data_reduced['pagePath_abbreviated_lag1'].value_counts()).head(20)
```

	pagePath_abbreviated_lag1
companies-house	9175
coronavirus	6387
find-local-council	2903
sold-bought-vehicle	2802
sold-bought-vehicle/y	2780
new-national-restrictions-from-5-november	2695
hm-revenue-customs	2574
transition	2356
vehicle-tax	2342
sold-bought-vehicle/no	1829
universal-credit	1785

Aggregated data - some low count sites

```
pd.DataFrame(visit_data_reduced['pagePath_abbreviated_lag1'].value_counts()).tail(10)
```

	pagePath_abbreviated_lag1
hmrc-internal-manuals/bim45650	1
transition-check/c[]=employ-eu-citizens&c[]=owns-operates-business-organisation-uk&c[]=owns-operates-business-organisation&c[]=visiting-driving&c[]=visiting-eu&c[]=visiting-row&c[]=travel-eu-business&c[]=working-uk&c[]=living-uk&c[]=nationality-eu&page=14	1
hm-revenue-customs/keywords=sickness+benefit	1
world/canada	1
treasury-direction-made-under-sections-71-and-76-of-the-coronavirus-act-2020/treasury-direction-made-under-sections-71-and-76-of-the-coronavirus-act-2020-in-relation-to-the-job-retention-bonus	1
cma-cases/liberty-global-plc-telefonica-s-a-merger-inquiry	1
disability-living-allowance-is-ending-leaflet	1
transition-check/c[]=do-not-ip&c[]=sell-public-sector&c[]=sell-public-sector-contracts&c[]=do-not-eu-uk-funding&c[]=personal-eu-org&c[]=personal-eu-org-use&c[]=do-not-employ-eu-citizens&c[]=owns-operates-business-organisation-uk&c[]=owns-operates-business-organisation&c[]=move-to-eu-no&c[]=travel-eu-business-no&c[]=working-uk&c[]=living-uk&c[]=nationality-uk&page=18	1
report-of-the-mid-staffordshire-nhs-foundation-trust-public-inquiry	1
calculate-your-holiday-entitlement/7.6	1

As seen, some sites have only one single visit in the data. There are 34k unique URLs in the data, in the end.

Model build

Train test split

In order to build the model, the data is split using stratified sampling, so that 70% of users are in the train data, and the remainder in the test. No consideration was made for the actual time. This means that higher frequency effects than a one hour period are being ignored.

```
train_data[train_data['fullVisitorId'] == 3250250007886851499]
```

	date	fullVisitorId	visitId	hour	minute	pagePath_abbreviated	pagePath_abbreviated_lag1
129484	20201113	3250250007886851499	1605269124	12	5	hm-revenue-customs/income-tax-enquiries-for-in...	be-passport-aware-check-passport-expiry-before...
129485	20201113	3250250007886851499	1605269124	12	7	hm-revenue-customs	hm-revenue-customs/income-tax-enquiries-for-in...
129486	20201113	3250250007886851499	1605269124	12	12	personal-tax-account	hm-revenue-customs
129487	20201113	3250250007886851499	1605270057	12	20	hm-revenue-customs	personal-tax-account

The model is based on the aggregation of data in the train data set. Using dictionary data structures, and then ranking the likely next URL based on the counts in the dictionary. The dictionary key could be considered to be a tuple of (current_URL, next_URL), although a nested dictionary structure is used, so that we have `{current_URL: {next_URL: count}}`. Although simple, this kind of cross tabulation captures interactions between fields and could be used recursively.

A particular advantage of this structure is that it easily handles the high cardinality of the data (many different potential values for the target). It can also be built online.

Scoring test data

In order to get the rank of an observation in the test data, the tuple from the test data looks up the rank in the train data.

In practice, if this solution was deployed, the first part of the tuple key would be used (ie the current page), and the top [10] predictions would be returned as a list of `key: value` pairs.

Illustration of train dictionary using Companies House

The most popular URL is for Companies House. This is a view of the dictionary of links for Companies House. Due to the dictionary structure, the entries are not sorted by value. The values represent the count of frequency of the second part of the key in the train dataset. In this example, the most popular link is to `get-information-about-a-company`.

```
pagePath_predictor_unflatten['companies-house'] # most folks are going to 'get information about a company'
{'get-information-about-a-company': 5295,
 'file-your-confirmation-statement-with-companies-house': 484,
 'register-for-email-reminders-from-companies-house': 6,
 'companies-house/about-our-services': 33,
 'file-your-company-annual-accounts': 112,
 'companies-house-guidance-for-limited-companies-partnerships-and-other-company-types': 10,
 'births-deaths-marriages': 1,
 'javid-sajid-chancellor-of-the-exchequer-hm-treasury-acoba-advice': 1,
 'company-registration-filing/forms': 30,
 'file-changes-to-a-company-with-companies-house': 205,
 'company-registration-filing/starting-company': 87,
 'make-changes-to-your-limited-company/share-structure': 2,
 'limited-company-formation/register-your-company': 6,
 'claim-for-wages-through-the-coronavirus-job-retention-scheme': 2,
 'hm-revenue-customs': 12,
 'companies-house-urges-all-companies-to-sign-up-for-email-reminders': 10,
```

Illustration of results by parent URL

In the test set, we can aggregate by the parent URL, and find the mean reciprocal rank overall. As can be seen the results vary widely by the parent URL. The mean reciprocal rank for the companies house URL is very high, nearly 1, which reflects the dominance shown above in the category `get-information-about-a-company`.

In the same way someone should not be impressed with 50% accuracy in coin toss predictions, we should not be impressed with these results. We have just quantified the natural ease of predicting the next link on GOVUK.

```
pagePath_abbreviated_lag1_test_summary.head()
```

	pagePath_abbreviated_lag1	count	mean_reciprocal_rank
0	companies-house	2615	0.875255
1	coronavirus	1868	0.302533
2	vehicle-tax	906	0.450450
3	find-local-council	868	0.044449
4	sold-bought-vehicle/y	843	0.785823

In contrast to the results for the `companies-house` URL, we see that the predictive performance of 'find-local-council' is poor. This reflects the high cardinality of the potential child URL. This would be partially remedied by the inclusion of location in the explanatory data.

Final words

The predictive power of the model should be seen as establishing an initial benchmark for the mean reciprocal rank (MRR). Other data should be included, most likely location and then ideally user specific information such as other URL activity over preceding sessions. In turn, the URL activity from previous sessions could be expressed by translating the related content into document vectors, for example using the average of the sentence embeddings, and including dwell times to weight the content. The benefit of additional data needs to be weighed up against the cost of acquiring the data and managing the computational environment necessary to process the data.

Additional questions

Q. Is this not a simple tabulation of what has happened previously, to predict the future, with no regard to the user?

A. Yes, this is simply aggregating results and ordering URLs by the measure $P(\text{new_URL} | \text{current_URL}, \text{current_time})$. We can make this more nuanced by using a tighter definition of the data we are conditioning on. For example, location and device are a good start, as these are immediately available.

The impact of personalisation, via addition of user specific data, should be measured against this benchmark. This then enables a cost benefit analysis (CBA) of the effort to record, store, recall and process user data, in the name of personalisation.