How GDS used data to improve content and user journeys on GOV.UK

Summary

Project timing – May – August 2019

Team – Data science team and developers from the Government Digital Service (GDS)

Tools – machine learning, network analysis and creating semantic vectors using Google's universal sentence encoder

Objective

The GDS team working on this project wanted to improve navigation on GOV.UK so it was easier for users to find what they needed. The team aimed to:

- reduce duplicate or overlapping pieces of guidance
- group related content with links

To do this the team used data to:

- identify similar or duplicate GOV.UK content
- use an automated process to generate related content links for most GOV.UK pages

Background

GOV.UK is the main website for government. It enables users to interact with government and find services, guidance and news from all government departments. There are over 400,000 unique pieces of content on GOV.UK and statistics showed that over 6 months, 700 pieces of content were published per week.

This amount of content, and high publishing rate, leads to duplicate content and makes it hard for users to find what they need.

GOV.UK provides different types of navigational help, including:

- a search function
- breadcrumb trails that show a list of nested web pages
- related links, which are shown to the right-hand side of content on a page

Before this project, only around 2% of GOV.UK's content (about 8,000 pages) had related links.

Investigating and trying solutions

To improve users' experience, we investigated:

- semantic vectors to group content and find similar or duplicate entries
- network analysis to align how GOV.UK pages are connected with hyperlinks (the structural network) with the way users travel between pages (the user generated functional network)
- machine learning to auto generate links on GOV.UK pages

Semantic vectors

To reduce the amount of similar or duplicate content on GOV.UK we first needed to find this content. We wrote a blog post on how we tested and created semantic vectors to plot the content with similarities and duplications.

After testing various tools we chose Google's universal sentence encoder to turn content into semantic vectors. By using the universal sentence encoder we could tell publishers where their content overlapped with pre-existing content. Publishers could use the information at the publication stage to prevent duplicate content going live, or retrospectively to clean up existing duplicate content.

We could also use semantic vectors to make decisions on changes to GOV.UK's taxonomy. The taxonomy is structured into topics and publishers tag content with the most appropriate topics.

Network analysis

We wrote a blog post on how we used network analysis to discover and assess:

- whether all content areas of GOV.UK are accessible and have links
- individual pages' position and significance within the overall network

The team identified hub pages which have more connections than others. To investigate these, we calculated network properties, including:

- network density
- connectedness
- link distribution
- centrality measures

The team created the GOV.UK Network Data Pipeline using Python to automatically extract user journeys from BigQuery – the database that stores our Google Analytics data. We aggregated these journeys for a specific time period to learn how users interact with the site. The result of these journeys was a functional network, or map of connected content.

We compared the functional network to how GOV.UK was structured, and used the results to inform changes to that structure and improve navigation for users.

Machine learning and A/B testing

When publishers create a new piece of content they add links to other related pages. Navigational links that facilitate browsing are automatically linked to the new content item. This creates GOV.UK's structural network which consists of approximately 350,000 links. We wrote a blog post about use machine learning and A/B testing to check the structure of GOV.UK content and automatically generate related content links.

After considering different algorithms, we decided to implement the node2vec algorithm to create links and display them on the right-hand side of GOV.UK pages. These would help people find the content they needed more easily.

We tested the user journeys through the links using 2 null hypotheses:

1. There is no significant difference in the proportion of journeys using at least one related link. If this is true, then people are not clicking our links, probably because they do not find them useful or interesting.

2. There is no significant difference in the proportion of journeys that use internal search. If this is true, then we did not provide people with relevant links to click on so they still use internal search to navigate.

We used an A/B test to randomly assign users to one of two possible versions of the site. Half of the users were directed to the control site (A) where only a small percentage of the pages had links which were added by the publishers. The other half of the users were directed to the version of the site (B) with the algorithmically generated related links. To speed up our analysis we wrote a software package that allowed us to run routine analysis, as soon as each experiment completed.

The results showed that both of our null hypotheses were false. The first implied that users found the related links interesting and/or relevant so they clicked them. User journeys showed an improvement for more than 10,000 users per day. The second result implied that there's a potential reduction in internal search by about 20% to 40%.

Applying our solutions

We wrote a blog post on how we put our findings into production. We used Jupyter notebooks for rapid development and prototyping which allowed us to code, document and publish data in a single place.

To automate the process of generating related links, we decided to modularise the code from the Jupyter notebooks and write it in object-oriented Python. By using a combination of user journey information and the structure of GOV.UK we could keep adapting the generated links.

To maintain a high quality of related links on GOV.UK, we added a number of

steps to our process, including:

- excluding a number of pages that should not have links coming from or going to them
- applying a confidence threshold to the links generated by node2vec which only allowed through the most relevant links
- generating a spreadsheet of the top 200 most popular pages so that our content designers could check the generated links and make changes where necessary

Results

We've been running the process to generate links every 3 weeks. These links are automatically displayed on GOV.UK when publishers have not set their own links, which ensures that users have a way to find the content they're looking for.

We're continuously iterating and refining the related links process, and monitoring results to make sure users' experiences are improving. In the future, we aim to bring the link generation online and use it as part of the publishing process. All of our code is available to view on GitHub.

Automatically generated links will not replace hand-curated links as they will not have the same considered context as those created by a subject matter expert.

Prior to this work, the vast majority of GOV.UK pages did not have curated recommended links. We automated that process, improving thousands of user journeys per day.