Office for National Statistics

Review of UK Big Data EssNet WP2 SGA1 work

WP2 face-to-face meeting, 4/10/17

Outline

- Ethical/legal issues
- Website identification
  - Using registry information
  - Using scraped data
- E-commerce
- Job vacancy
- Outstanding issues/further work

Ethical/legal issues

- Originally – ONS could not scrape without checking terms and conditions – causes challenges
- Changed this by developing **ONS web-scraping policy** with sign-off from senior management (will be published soon)
- Now we can scrape without checking terms and conditions most of the time
- Also developed ‘netiquette’ with NL

Ethical/legal issues

Extract from web-scraping policy discussing our legal approach:

- We will cease scraping whenever we are asked to do so by the website owner.
- We will carry out scraping in a manner which does not cause financial detriment to any website owner.
- We will abide by the Data Protection Act and other data sharing legislation, as outlined in section 6.3. This includes ensuring that personal data is not revealed in any published statistics or research.
- We will check and abide by the terms and conditions of websites wherever it is practical for us to do so, and contact website owners in the event of any uncertainty regarding the terms and conditions.
- **Where it is not practical for us to check the terms and conditions of a website, for example where we are scraping large numbers of websites, we may scrape where we can justify that it is ethical and in the public good for us to do so. This decision will be made with reference to the balance between risk & negative consequence and efficacy & public benefit.**
- We will continue to monitor the legal situation as it evolves and amend our approach accordingly.

Website identification

- Work took place in two stages:
  - When we could not scrape data without checking T&Cs, we tried using *registry information* instead
  - Now that we can, we have re-done our research using scraped data

Website identification – registry information

- When any individual or organisation registers to hold a domain name, their name and address must be provided to a registrar. These data are then usually made publically available and can be accessed via a ‘whois’ lookup. We tried using this data for website identification.
- Process -
  - We used manual identification to form training and test sets of businesses matched to websites.
  - For each business - queried the Bing API with the business name, first 10 returned URLs stored as candidate domains.
  - Obtained the address information from the registry for these candidate domains and tried to match against the address on the business register.

Website identification – registry information - Findings

- Utilising registry information for the top search responses and performing a simple match with information on the business register – recall of 37%, precision of 100%.
- This is insufficient by itself but may be useful when combined with other methods.
- It is important to consider bias in any website identification method. We found it easier to find websites which conduct ecommerce (47% found) than to find websites which did not conduct e-commerce (33% found)
- **Challenges –** bias in websites identified, insufficiently powerful classifier, insufficient size of training set

Website identification – using scraped data

- Obtained larger training and test set by scraping an online registry
- Queried the Bing API with the business name, obtained the first 10 results as candidate websites
- Trained a random forests model to recover business websites using features –
  - Presence of full/partial business name
  - Presence full/partial business name in URL
  - Presence of correct business postcode
  - Presence of correct business address
  - Website structure - page-counter, number of internal links
  - Page rank in results

Website identification – using scraped data - Findings

- Most informative features – search rank, whether postcode present on website, whether name present on website
- Recall of 52%, precision of 99.3%
- **Challenges –** bias in training set (more likely to contain easy-to-find websites), restrictions on Bing API, cannot raise recall by much even by reducing decision threshold

E-commerce

Process:

- Created training set of 500 businesses by running results from our IT survey through automatic website identification.
- Scrape each website, extract text
- Train Naive Bayes classifier with features based on occurrence of all words in corpus on each website (‘bag of words’)

- Doesn’t work very well yet – recall is only 18% – but we are still developing
- Technologies – scrapy, NLTK in python – worked pretty well
- **Challenges –** computational challenges (sensitive data, don’t have suitable environment). Model does not perform well yet. Could use more complex NLP techniques instead of ‘bag of words’.

Job advertisement

- UK ICT survey does not capture information on whether a business advertises job vacancies online – no good source of training data
- Obtained ‘positive’ (websites with job vacancies) and ‘negative’ (websites without job vacancies) samples separately:

  Positives: scraped from an online listings page containing business websites which offer jobs vacancies

  Negatives : sampled businesses from the UK ICT survey which report having a website, identify a website for each of these businesses, and verified that this website does not contain job vacancies.
- Sample contained 400 total business websites
- Extract text, train classifier in same fashion as e-commerce
- Reasonable recall, poor precision - but can’t evaluate properly due to insufficiently large training set
- **Challenges –** Insufficiently large training set. Problem with False Positives – websites being identified as containing websites when they actually do not. Could be resolved by using more complex NLP. Some of the most informative features were words like ‘opportunities’, which may appear on most business websites.

Outstanding Issues/Further work

- Need larger training sets! Currently working on this – identifying more websites automatically – but will not help with job advertisement use-case
- Need to worry about bias in training set/website identification – for example, training set matching businesses to websites scraped from online registries may be biased towards easier-to-find websites which might be more likely to conduct e-commerce. This is hard to fix.
- Could utilise proper NLP rather than bag-of-words, investigate alternative classification algorithms
- Takes a long time to get good performance from ML – need to do more feature engineering.
- Other challenges – reliance on Bing API, difficulty running classifiers on secure data

WP1 interdependencies

- What WP1 want WP2 to be able to do – for a list of enterprise names from a survey, they want us to be able to identify the subset which have a website and the subset which identify jobs on that website
- We have already piloted this, but our results may not be good enough yet – **should we develop this use-case further in SGA2?**
- Relevant WP1 workstream currently doing this step manually, and then automatically extracting the total number of vacancies. Timing issues mean we probably couldn't feed automatically-generated data into WP1, but could feed into each other's final reports (eg – WP1 final report could include our analysis on doing the automatic step manually)
- WP1 need to find the page on that website which contains the job vacancies – again, they are currently doing this manually. **Could we do it automatically for them?** Could pick this up in the UK.
- WP1 don't want final face-to-face joint meeting – they want to meet with CEDEFOP in Milan instead