Linking external information about entities for sensitivity detection

Lewis Ponsonby

2464980P

Proposal

Motivation

With the plethora of counties that have implemented freedom of information laws in light of the 'Information Age', it is becoming increasingly common that government documents are being made readily accessible to the public after a certain period of time. For example, the government of the United Kingdom release government documents when they are 20 years old - although these documents are released to the public, they may still contain sensitive information. When government documents are released, they must be 'sensitivity reviewed' to highlight potentially sensitive details, with said details being redacted or the entire document itself being restricted. With the increasing amount of data stored digitally, it is clear that sensitivity reviewers are in need of tools able to aid them in sensitivity detection. This can come in the form of linking external information as described in the project title.

Aims

The project should be a web-based application that allows 'sensitivity reviewers' to upload batches of documents and have them stored and processed; this should produce additional information displayed in a user-friendly format in order to assist the sensitivity reviewing process. Entities should be identified and accompanied by additional descriptions as well as the likelihood to which the entity is sensitive. Analysis of entities and documents can be presented to the user such as entity frequency, similar documents and topic modelling.

Progress

- Programming language and web-based development tool selected, the project will be a Python-Django web-application
- User stories and MoSCoW requirements created for web-app plan
- Natural language processing and entity tagging implemented using Spacy & DBpedia Spotlight pipeline
- Entity abstracts retrieved using BeautifulSoup via DBpedia URL in JSON dump

- Database architecture produced outlining tables, primary keys, attributes & types as well as foreign key relations
- Document upload, storage and deletion functioning for the Django web-app
- Text pipeline implemented into web-app, uploaded documents can be viewed with entities highlighted and their abstracts available as overlays
- Existing government documents and sensitivity truths used to produce a classification model that predicts the likelihood of a model being sensitive
- Sensitivity predictions used in document view to provide an intuitive colour scheme signifying the potential sensitivity of entities in a document

Problems and risks

Problems

- Finding compatible versions for Python, Django, Spacy and DBpedia took some experimentation to function
- Document uploads were time consuming, introducing caching of repeated entities and abstracts sped up this process
- Struggled to flesh out project idea with additional features initially

Risks

- Some entity predictions may be inaccurate. **Mitigation:** Use DBpedia confidence scores to restrict less confident entity predictions.
- Document topic modelling may be a time-consuming process to repeat. **Mitigation:** Store topic model locally and reproduce it in larger time intervals than every time the page is reloaded.
- Users during the user study may struggle to identify and evaluate sensitive documents. **Mitigation:** Provide users additional tasks that do not involve sensitivity reviewing but provides valuable feedback on the project.

Plan

- Week 1: Implement LDA topic modelling feature.
    - **Deliverable:** Web-app with implemented LDA topic modelling for uploaded documents with visual display for topics and their sensitivities.
- Week 2-3: Finalize initial project implementation.
    - **Deliverable:** Web-app with all implemented features that passes tests from the testing suite and has polished site-flow & UI-UX details.
- Week 4: Plan user study.
    - **Deliverable:** Detailed user-study plan with feature combinations that will be removed/kept evaluating how well users complete given tasks, list of tasks to be evaluated, participant numbers and participant information sheet.
- Week 5: Create separate project implementations for relevant user study sections.
    - **Deliverable:** Several project implementations corresponding to the feature combinations defined from Week 4. Each implementation will have certain a given combination of features removed or kept.
- Week 6: Run user study.
    - **Deliverable:** Quantitative measures of user performance for given feature combinations and qualitative feedback on project.
- Week 7: Act on user study feedback.
    - **Deliverable:** Final project implementation having acted on the qualitative feedback from user study.
- Week 8-10: Write up.
    - **Deliverable:** First dissertation draft submitted to supervisor two weeks before final deadline.