Initial Plan

Project Title: Implementing an information retrieval system

Author: Aleksandar Kochev
Supervisor: Irena Spasić
Moderator: Wendy K. Ivins
Module Code: CM3203
Module Title: One Semester Individual Project
Credits: 40

Project Description

In the sphere of knee pathology, magnetic resonance imaging (MRI) can be used to visualise all structures within the knee joint, which makes it a valuable tool for increasing diagnostic accuracy and planning surgical treatments. Therefore, clinical narratives found in MRI reports convey valuable diagnostic information. [1]

KneeTex is an open–source, stand–alone application for information extraction from narrative reports that describe an MRI scan of the knee. Given an MRI report as input, the system outputs the corresponding clinical findings in the form of JavaScript Object Notation (JSON) objects. The extracted information is then mapped onto TRAK, an ontology that formally models knowledge relevant for the rehabilitation of knee conditions. This knowledge includes classification of knee conditions, detailed information about knee anatomy and an array of healthcare activities that can be used to diagnose and treat knee conditions. [1]

The goal of this project is to build an information retrieval system, in the form of a web application, that will firstly - store all the structured data produced by KneeTex, as well as the contents of the TRAK ontology, in a database and secondly – provide a user-friendly interface for querying and displaying the information located in the pre-populated database. Overall, the system should allow complex searches to be executed efficiently over the JSON objects, which represent a processed version of free text MRI reports, thus effectively supporting epidemiologic studies of knee conditions.

Project Aims and Objectives

The implementation of the system will require:

1. A database that will store structured JSON data together with the TRAK ontology
2. A user-friendly search interface over the database, including:
    a. Query formulation (*Search Page*)
    b. Presentation of the search results (*Search Results Page*)

3. Matching queries against the data to support Elasticsearch.

   Elasticsearch is an open-source search server based on Lucene. It provides a distributed, multitenant, full-text search engine with a HTTP web interface and schema-free JSON documents. [2]

   Once a search request has been submitted through the User Interface (UI), the back-end should create relevant queries, based on the submitted information, to retrieve data from Elasticsearch.

4. A Ranking algorithm for the search results. (*How should the results be ordered?*)

5. Evaluation of information retrieval. (*Are the returned search results correct?)*
   a. Create gold standard
   b. Calculate precision, recall and F-measure

6. Usability testing for the UI

Work Plan

Time Period	Tasks
Week 1 (25th January – 31st January)	Submit Initial Project Plan
Week 2 (1st February – 7th February)	1. Familiarise myself with the technologies that will need to be used to implement this system. These technologies include: • Git – for version control • Elasticsearch – for data storage and information retrieval • The Java "Play" framework – to easily create a MVC (Model-View-Controller) system architecture • Handlebars – for templating • Bootstrap – for responsive web design 2. Research what is new in the realm of web applications and look for anything that might prove useful for the implementation of this project 3. Meet supervisor 4. Start working on **Objective 1** – the creation of a database that will store all the JSON documents.
Week 3 (8th February – 14th February)	1. **Finish Objective 1.** 2. Meet supervisor 3. Start working on **Objective 2a and Objective 3** – creating a search page and a query matcher.

Week 4 (15th February – 21st February)	1. Implement a basic working version of **Objective 2a and Objective 3.** At this stage the executed searches should return data in plain JSON format, but it should be the correct data. 2. Meet supervisor 3. Start working on **Objective 2b** – the creation of a Search Results Page (SRP).
Week 5 (22nd February – 28th February)	1. Implement a basic working version of **Objective 2b** – the SRP. 2. **First review meeting with supervisor.** 3. **Finish Objective 3.** Any search request submitted on the front-end should be correctly mapped to the corresponding JSON object fields and sent as a query to Elasticsearch. The correct results should then be returned.
Week 6 (29th February – 6th March)	1. Improve the implementation of the Search Page and the Search Results Page. 2. Start working on **Objective 4 –** implement a ranking algorithm for the search results. 3. Meet supervisor
Week 7 (7th March – 13th March)	1. **Finish Objective 4.** A proper ranking algorithm should be in place. 2. Meet supervisor 3. Start working on **Objectives 5a** and **5b** – the evaluation of information retrieval.
Week 8 (14th March – 20th March)	1. **Finish Objectives 5a and 5b.** A gold standard should be created and the proper calculations should be completed. 2. Meet supervisor 3. Start working on **Objective 6 –** usability testing for the user interface.
Easter Recess (19th March – 10th April)	1. Exam preparation. 2. If needed – catch up with any outstanding work on the final year project and start forming Final Report.
Week 9 (11th April – 17th April)	1. **Finish Objectives 2a and 2b as well as Objective 6.** Any remarks about the UI after the usability testing should be corrected, hence the UI and its usability testing should be complete. 2. **Second review meeting with supervisor.**

Week 10 (18th April – 24th April)	1. Finalise the implementation of the system - Make small optimisations - Tag it as "*Version 1.0*" in GitHub 2. Meet supervisor 3. Start working on the Final Report
Week 11 (25th April – 1st May)	1. Work on Final Report 2. Meet supervisor
Week 12 (2nd May – 6th May)	1. Submit Final Report 2. Meet supervisor

Work Plan Explained:

The table above shows an optimistic work plan. The problem that needs to be solved first is the creation of the database that will store the data refined by KneeTex together with the TRAK ontology. Once a good data set is available it would be optimal to start working on search, retrieval and evaluation of results as these are the core objectives of this project. On the other hand, the UI (user interface) can be developed little by little, since it is the easiest piece of code to experiment with and improve with time.

Even though the formulation of this project is suitable for the Waterfall software development method I would still prefer to use Agile, as it will allow me to easily adapt to unexpected adjustments and unforeseen circumstances. This means that the work plan formulated above is subject to change throughout the course of this project.

References

1. Irena Spasić, Bo Zhao, Christopher B. Jones and Kate Button. 2015. KneeTex: an ontology–driven system for information extraction from MRI reports [Online]. Available at: http://www.jbiomedsem.com/content/pdf/s13326-015-0033-1.pdf [Accessed: 30-01-2016]

2. Wikipedia. September 2015. Elasticsearch [Online]. Available at: https://en.wikipedia.org/wiki/Elasticsearch [Accessed: 30-01-2016]