CS Honours Projects 2017

Contents

1. Project : attendance as result predictor

PROPOSER: ***Sonia Berman***

ABBREVIATION: MustWeGo

Brief Description: It is a widely held view that students would perform better if lecture attendance was higher, yet students are not altering their habits. This project will investigate the use of data mining techniques as a means of providing evidence that lecture attendance improves performance. This may in turn help to incentivise more students to go to lectures and improve university throughput.

Each student will look at using a different data mining technique on the same (anonymised) data set. They will first build what is essentially an Admissions tool, that will look at data on university applicants (school subject marks, age, local/not, etc.) and predict student performance in a first year course based on this. They will then add to this data the lecture attendance of these students in one of their courses and use the new data set to predict their final mark for that course. Ideally the second model will be far more accurate. The first model in itself should be a useful tool for Admissions decisions.

Computer Science Content: Databases, software development.

Specific Learning Outcomes: Data storage, data exploration, data mining. Students will be expected to use new technology such as Hadoop/Spark tools, in addition to conventional data mining tools, in their work.

Skills Required by Team as a Whole: You should do the BIGDB module. You should enjoy playing with data. You need to be a clear and critical thinker, to make sure that you understand the data mining results and don't simply report outputs.

Theory: Data mining methods, big data storage.

Implementation: Skills required will be those covered in the BIGDB database module. Project difficulty: average.

Data: UCT data will be used. Based on past experience, UCT is extremely efficient and helpful in this regard, so no problems anticipated.

Facilities needed: Only free software will be needed.

Supervision: Weekly meetings with supervisor.

NUMBER OF STUDENTS: 2 (OR 3)

2. Project : targeted student interventions

PROPOSER: ***Sonia Berman***

ABBREVIATION: WillIDop

Brief Description: Universities have developed many interventions over the years to assist students in need of extra academic support. However it is often difficult to know which students are in need of such help. This project will investigate whether data mining of temporal data, in the form of marks over time in a course, can identify reasonably accurately, during a course, which students are likely to fail and should be targeted for intervention programs.

Aside: “dop” is the Afrikaans word for fail (pronounce the “o” as in “more”).

This project involves data mining of an (anonymised) data set comprising many years of Vula gradebook data, along with the academic history of each student in the form of final marks for school subjects and university courses already completed. One of the students on this project will use a temporal data mining technique, and the other will use a conventional (non-temporal) one. The latter will also be responsible for creating the user interface and ETL (extract, transform, load) components of the system, so that convenors of courses in a variety of subjects, at different levels, and with very different gradebooks, can tailor it to their needs. It would be especially useful if output includes an indication of why specific students have been suggested by the product. If 3 people take on this project, the study would need to include an investigation into the usefulness of Vula activity logs in performance prediction, and the associated extension to the user interface and ETL processes. For example it may be the case that a drop in Vula activity indicates a student is starting to struggle, or that late submission of assignments is the best such indicator.

Computer Science Content: Databases, software development.

Specific Learning Outcomes: Data storage, data exploration, data mining. Students will be expected to either use new algorithms (if taking on the temporal modelling) or to use new technology such as Hadoop/Spark tools (non-temporal case), in addition to conventional data mining tools, in their work.

Skills Required by Team as a Whole: You should do the BIGDB module. You should enjoy playing with data. You need to be a clear and critical thinker, to make sure that you understand the data mining results and don't simply report outputs.

Theory: Data mining methods, big data storage, temporal data mining.

Implementation: Skills required will be those covered in the BIGDB database module. Project difficulty: average.

Data: UCT data will be used. Based on past experience, both UCT Planning and UCT Vula staff are extremely efficient and helpful in this regard, so no problems are anticipated.

Facilities needed: Only free software will be needed.

Supervision: Weekly meetings with supervisor.

NUMBER OF STUDENTS: 2 (OR 3)

3. PROJECT : RELATIONAL TO NOSQL MAPPING

PROPOSER: ***Sonia Berman***

ABBREVIATION: NOSQLKIT

Brief Description: More and more companies and organisations are entering the world of big data analytics, so there is a growing need for people and processes to move from the relational database environment into the NoSQL one. This project will build a toolkit to facilitate this increasingly common requirement.

The project will include tools to automatically map from relational schemas and/or ER models into each of the 4 types of NoSQL database, and to automatically map SQL queries into corresponding queries for at least 3 types of NoSQL database. As part of this project, students will build a system for generating data for relational database schemas in order to be able to test their work; such a product is in itself useful for educational purposes. One student will work on the schema mapping and test data generator, and the other on the query mapping.

Computer Science Content: Databases, software development.

Specific Learning Outcomes: NoSQL, data modelling and relational database skills, automatic code generation.

Skills Required by Team as a Whole: You should do the BIGDB module. You should enjoy playing with data.

Theory: key-value, document-oriented, column-oriented and graph data storage systems.

Implementation: Skills required will be those covered in the BIGDB database module. Project difficulty: average.

Data: Any data set, e.g. from kaggle, can be used. You will also generate test data yourself later on.

Facilities needed: Only free software will be needed.

Supervision: Weekly meetings with supervisor.

NUMBER OF STUDENTS: 2

4. PROJECT : CRAFTWARE SALES

PROPOSER: *Edwin Blake*

CO-SUPERVISOR/EXTERNAL ADVISOR: *Meryl Glaser*

ABBREVIATION: CRAFTS

Brief Description: We require a number of projects ranging from rather technical backend systems to co-designed user-interfaces for the ongoing work we are doing with our partners the Deaf Community of Cape Town (DCCT) (www.dcct.org.za).

DCCT produces a number of craft objects for sale (www.dcct.org.za/?q=craftware-catalog). The purpose of this project is to assist DCCT in marketing the products. The craftware business is run from the DCCT centre in Heathfield. The business also has a store of pre-made stock.

They produce a wide range of products listed in the catalogue. All products are hand-made and customisable — that is, you can have your name and a motif of your choosing printed on your chosen product. Your system needs to cater for this customization.

In general you cannot use preconceived ideas and you will need to design to suit the business process of the Craftware ladies. You will therefore adopt a method we have called community-based co-design.

“Community-Based” conveys the fact that we deal with groups of people rather than individuals. We have develop ways of entering into design conversations with people who do not have technical skills but who are knowledgeable on their own needs and especially how their own communities operate. “Co-design” derives from the application of the action research paradigm in a design setting: both the computer experts and the community members are designers on an equal footing and work cooperatively.

Please see www.cs.uct.ac.za/~edwin/honsProj.html for more details.

Computer Science Content: ICT4D, Community-Based Co-Design (including Action Research, participatory design, user centred design), online sales systems..

Specific Learning Outcomes: In doing this project you will acquire knowledge of Community-Based Co-Design, the equivalent of the honours module that was not given this yea. It won’t be hard but it will require a flexible mindset. This project will require regular visits to DCCT in Heathfield.

Skills Required by Team as a Whole: Commitment to user-centred design, knowledge of . and web development skills, it will be useful for one person to have an interest in business processes and marketing. Students have to have a professional and respectful attitude to working with clients and co-designers.

Theory: Willingness to learn about the more advanced aspects of user-centred design. Agile software engineering.

Implementation: Solid system implementation skills across a broad range of topics as outlined above.

Facilities needed: Product catalogue, Drupal system, Sign language interpreter.

Supervision: The project is run with the NGO DCCT (Deaf Community of Cape Town) and with Meryl Glaser and their craftware section. Regular project meetings will be held and students will be expected to undertake regular visits to DCCT in Heathfield

NUMBER OF STUDENTS: 4 (OR FEWER)

5. PROJECT : COMPUTER LITERACY LESSONS FOR DEAF ADULTS

PROPOSER: EDWIN BLAKE

CO-SUPERVISOR/EXTERNAL ADVISOR: MERYL GLASER

ABBREVIATION: SIGNS

Brief Description:

We require a number of projects for the ongoing work we are doing with our partners the grassroots NGO "Deaf Community of Cape Town (DCCT)" (www.dcct.org.za). We are working to empower people in this community.

We have already developed a system to train Deaf learners in computer literacy using mobile devices. We would now like to create an extended version of the system for the computers on which the Deaf learners practice and integrate it.

We have developed a mobile prototype that supports teaching computer literacy skills to Deaf people, using South African Sign Language (SASL) as the medium of instruction. We support Deaf people learning computer literacy skills using the International Computer Driving License (ICDL www.icdl.org.za) approved curriculum and e-learner developed by Computers 4 Kids (www.computers4kids.co.za). We now have a XML specification that is used to structure lesson content and is generated by the content authoring tool. The XML specification was an abstraction of the hierarchical structure of the e-learner manual. The current mobile client uses the XML specification and displays the e-learner contents as SASL videos and images.

Once the assets (SASL videos, images, lesson texts etc.) are created the system will manage them. For SASL videos and images to be meaningful, they need to be organized in a logical manner that reflects the e-learner lesson structure.

Your task will be to extend and integrate the mobile client as a better PC-based application.

Please see www.cs.uct.ac.za/~edwin/honsProj.html for more details.

Computer Science Content: ICT4D, multimedia databases, user-centred design, scripting

Specific Learning Outcomes: This project combines user-centred design and development of a robust frontend system. You will learn about Co-Design of tool which are needed for online learning. The project is firmly embedded in an ICT4D context

Skills Required by Team as a Whole: The team as a whole will have to have strong software development skills. An interest helping real users is also a requirement. The skills include multimedia presentation, pedagogy, multimedia database and xml.

Theory: Some background and willingness to learn about user-centred design; multimedia databases.

Implementation: Solid system implementation skills. User testing.

Data: collection of sign language videos, images and computer literacy lessons.

Facilities needed: PC development, combined with regular vists to the user group.

Supervision: The project is run with the NGO DCCT (Deaf Community of Cape Town) and with Meryl Glaser who will represent them. Regular project meetings will be held and students will be expected to meet demands set by Ms Glaser as the client.

NUMBER OF STUDENTS: 3 (OR POSSIBLY 2)

6. PROJECT : VIDEO EDITING AND UPLOAD TOOL

PROPOSER: ***Edwin Blake***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Meryl Glaser***

ABBREVIATION: VIDEOUP

Brief Description:

We require a number of projects ranging from rather technical backend systems to co-designed user-interfaces for the ongoing work we are doing with our partners the Deaf Community of Cape Town (DCCT) (www.dcct.org.za).

DCCT captures many of their events on video using a high quality video camera. There is a requirement to archive the videos with suitable metadata and also to extract and edit clips and possibly add subtitles (for those of us who are not fluent in SASL), as well as transcoding them to compressed web friendly formats. The recorded SASL videos can be edited to remove the audio channel to reduce video file size. The resulting videos can be encoded using the H.264 video codec with a frame size of 640 x 480 pixels and a frame rate of 25 frames per second as per the ITU requirements. These clips should be added to the existing website. You can see that the current website largely has static images (www.dcct.org.za/?q=gallery-page).

Please see www.cs.uct.ac.za/~edwin/honsProj.html for more details.

Computer Science Content: Web development, video coding, user centred interface design.

Specific Learning Outcomes: ICT4D, video coding and web development, user-centred design.

Skills Required by Team as a Whole: video editing and web development skills (or be willing to acquire them by July)

Theory: Some background and willingness to learn about user-centred design; web development on a CMS (content management platform), video coding.

Implementation: Video transcoding will require some delving into codecs, for the web interface you will be scripting Drupal.

Facilities needed: Videos to experiment on.

Supervision: The project is run with the NGO DCCT (Deaf Community of Cape Town) and with Meryl Glaser and their system administrator. Regular project meetings will be held and students will be expected to meet demands set by Ms Glaser as the client.

NUMBER OF STUDENTS: 2

7. PROJECT : DATA CAPTURE FOR GROOTE SCHUUR NEONATAL OUTCOMES

PROPOSER: ***Melissa Densmore***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Dr. Lloyd Tooke, Groote Schuur***

ABBREVIATION: VON

Brief Description:

This project involves implementing and testing a data capture and report management system for the Groote Schuur Neonatal Unit. Project members will co-design, implement and evaluate a system that will capture patient identifiers for premature infants born under 1.5kg, which will be uploaded to the Vermont Oxford Network (VON), a system set up for tracking neonatal outcomes globally. Based on the information provided, the system will also produce a discharge sheet that will be kept with the patient files. The current process is entirely manual and involves a paper-based form followed by manual entry, which is done by Dr. Tooke.

Computer Science Content: Human-Computer Interaction, Networked Communications, Software Engineering

Specific Learning Outcomes: Students will gain real-world experience developing a system that will improve workflow in the Groote Schuur Neonatal ICU. This project will involve 1) working with doctors and hospital staff to develop and evaluate potential designs, 2) implementation of a data entry user interface, 3) integration with the VON API, 4) feature support for the discharge form.

Skills Required by Team as a Whole: Students will require user-centred design skills, including interviewing and interacting with the hospital staff that will use the system. They are expected to have sensitivity to private health information, and to do mobile/tablet application development.

Theory: User-Centered Design, Networking

Implementation: Android Development,

Data: Students will interact with the staff of Groote Schuur Neonatalogy through a user-centered design process. We will provide the documentation from VON for the upload API as well as a paper copy of the existing form.

Facilities needed: Computers for software development, Test mobile devices for development to be provided. Students will need to travel to Groote Schuur Hospital to interact with the staff.

Supervision: Regular meetings with internal and external supervisors, weekly progress reports expected. Students will do a final presentation for the participating staff.

Bursaries: R3,000 per student

NUMBER OF STUDENTS: 3 (OR 2, 4)

8. PROJECT : DEAF FRIENDS MATERNAL HEALTH CAMPAIGN

PROPOSER: ***Melissa Densmore***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Marion Heap***

ABBREVIATION: DEAFMOMS

Brief Description: For this project you will work with Deaf mothers to design a mobile app to support a public health campaign targeted towards improving maternal health outcomes in the Deaf community. The Deaf (with a capital D) community includes both hearing and non-hearing persons who primarily communicate by sign language, in this case South African Sign Language. Marion Heap will work with video producers to develop the videos. The app will support viewing of the videos in order, and when linked with the Mom Connect government campaign (http://www.health.gov.za/index.php/about-mom-connect) will unlock the previously downloaded video matching that week's message on the phone. The app also should support peer to peer sharing of the videos, supporting content, and community chat about the videos.

Computer Science Content: User-Centered Co-Design methods, Android development, Peer-to-peer and delay tolerant communications

Specific Learning Outcomes: Students will gain experience with human computer interaction methods, peer to peer file sharing and communication, and android development. They will gain specific expertise in maternal health and working with the Deaf community.

Skills Required by Team as a Whole: Students must be willing to listen to and be inspired by the research participants, and sensitive to the topic of maternal health and the needs of the Deaf community. They must also be comfortable preparing interactive prototypes and ultimately a mobile app to support the project.

Theory: HCI, Networking

Implementation: Android User interface and backend communications development, usability evaluations

Data: Marion Heap to facilitate access to Deaf community

Facilities needed: Students will need to travel to research site for evaluations and design elicitations

Supervision: Regular meetings with internal and external supervisors, weekly progress reports expected. Students will do a final presentation for the participating staff.

NUMBER OF STUDENTS: 3 (OR 2)

9. PROJECT : FAKE NEWS MONITOR

PROPOSER: ***Melissa Densmore***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Kyle Findlay***

ABBREVIATION: FAKENEWS

Brief Description: A website to collate and annotate fake or dubious news sources in South Africa. A place where users can go to find collated feeds of fake news, along with ways of safely sharing that content (e.g. generated URLs with “dubious warning” overlay), and analysing the agenda behind the various sources e.g. content analysis such as topic modelling, Bayesian surprise, network maps, and various other forms of Natural Language Processing. They will be expected to evaluate the website with real users.

Computer Science Content: Kyle Findlay

Specific Learning Outcomes: Content analysis components e.g. Bayesian statistics, word2vec, Latent Dirichlet Allocation, User Centered Design

Skills Required by Team as a Whole: Applied data science techniques for social and political research

Theory: Information Retrieval and Classification, User-Centered Design, Visualization

Implementation: R, Python, machine learning, Web Development, Web crawling/scraping

Data: List of websites to be provided by Kyle Findlay

Facilities needed: None

Supervision: Regular meetings with internal and external supervisors, weekly progress reports expected. Students will do a final presentation for Kyle and other researchers on the project from the Centre for Film and Media.

NUMBER OF STUDENTS: 2 (OR 3)

10. Project : Machine Learning for Health

Proposer: **Brian DeRenzi**

Co-supervisor/External advisor: **Deshen Moodley**

Abbreviation: ml4h

Brief Description: This project will focus on applying state-of-the-art machine learning and AI techniques to public health treatment data. There are two main phases to the project:

(1) *Context and cleaning*. In the first phase of the projects, students will work together to meet with the owners of the data to understand contextually-relevant (clinically important) questions that should be answered by the data. They will then work to clean the existing data in a non-destructive, repeatable way.

(2) *Developing adaptive models*. The students will then work independently to apply various AI and machine learning techniques to deal with temporal health records and dynamic model update to answer the set of questions identified during phase 1 of the project.

Computer Science Content: Artificial intelligence, machine learning, eHealth

Specific Learning Outcomes: Students should end with a strong grasp of the application of machine learning algorithms and the process of applied research in the health domain.

Skills Required by Team as a Whole: Solid understanding of machine learning and AI theory. Different students will focus on different streaming techniques for model updating.

Theory: See above.

Implementation: Strong Python and/or Java skills. Students will benefit from experience with pandas, scikit-learn and the like.

Data: The primary data source will be HIV treatment data from a large treatment centre in a country in southeast Africa. We are also in discussions with the UCT groups working with the Western Cape government (among others) to augment the dataset.

Facilities needed: N/A

Supervision: Once the project starts, students can expect a weekly 30 minute meeting with the primary and secondary supervisors and additional time as necessary.

Bursaries: TBD – there is potential for a bursary for an outstanding student who will focus on the data from the Western Cape.

Number of Students: 2 (or 3)

11. Project : Chef Registration System

Proposer: **Brian DeRenzi**

Co-supervisor/External advisor: ***Infinity Culinary Training***

Abbreviation: chefreg

Brief Description: The Infinity Culinary Training organization is a Cape Town-based non-profit, located in town, that trains disadvantaged South African women and men to work in the hospitality industry. The school offers a tuition-free program to teach not only the basic cooking skills and professional tools necessary for immediate employment, but the all-important life skills required to turn a job into a career.

A large part of this project will be following a standard design methodology to work with the organization to understand their needs and build out and evaluate a set of products to meet those needs. Below are two potential projects that have come out of initial discussions, though they will evolve over time.

The organization currently uses a collection of well-designed spreadsheets to keep track of students and maintain a "CV" for each one with up-to-date contact information, previous employment, etc. The goal of this project would be to build out and evaluate a web-based student registration platform to keep track of students during their time with the program and after. This includes a dashboard for generating visualizations of employment rates, number of graduates and any other information the school requires.

Another student will be focused on building out a mobile application for the students to use to allow them to automatically generate a CV that can be sent to prospective employers. The CV should be available in plain text, PDF and as a web link, as well as offer various templates to allow some level of customization.

Computer Science Content: HCI, ICT4D, tech4chefs!

Specific Learning Outcomes: Understanding of the design process, working with external clients, building out rapid prototypes in an iterative manner.

Skills Required by Team as a Whole: The teams will need to understand different design methodologies to be able to pick, and follow, a suitable methodology for the project. Students should also have basic Android development and web development skills.

Theory: Design methodologies and software engineering.

Implementation: Python and Java are the two most likely candidates for this project, though there is some flexibility depending on student strength.

Data: Test data will need to be generated during the first phase of the project.

Facilities needed: N/A

Supervision: Once the project starts, students can expect a weekly 30 minute meeting with the primary supervisor and additional time as necessary. Students will be invited to the .

Bursaries: While there is no bursary available, students can expect the occasional free lunch when meeting with students and the organization.

NUMBER OF STUDENTS: 2 (OR 3)

12. PROJECT : MOBILE EMR

PROPOSER: ***Brian DeRenzi***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Jembi Health Systems***

ABBREVIATION: MEMR

Brief Description: Quality health delivery relies on accurate and up-to-date medical records being available. Patient migration (for family, job, or infrastructure reasons) mean that these records need to be available nationally. Infrastructure challenges and data costs make this a difficult problem to solve centrally with a web-based system.

This project will investigate two different approaches to solving this problem. The first is a decentralized medical record system that will ensure that up-to-date time stamped patient interaction data is available when needed.

The second will explore algorithms for detecting duplicate registrations as well as selective synchronization of patient data from a centralized repository based on patient movement. Both projects will be released publically with a permissive open source license from the start of the project.

Computer Science Content: Algorithms, HCI, mHealth

Specific Learning Outcomes: Real-world experience developing distributed systems.

Skills Required by Team as a Whole: Strong foundation in algorithms and data structures. The student working on the decentralized medical record system will need to fully understand how these systems work, including their benefit and challenges. The registration and synchronization student will have a solid understanding of algorithm development and tree-based data structures.

Theory: Algorithms and data structures

Implementation: Strong background in Java programming.

Data: Test data will need to be generated during the first phase of the project.

Facilities needed: N/A

Supervision: Once the project starts, students can expect a weekly 30 minute meeting with the primary supervisor and additional time as necessary. Students will also have the opportunity to hot desk from the Jembi offices in Tokai (which are very nice).

NUMBER OF STUDENTS: 2

13. PROJECT : AN SMS-BASED RUNYANKORE PERSONALIZED DRUG PRESCRIPTION APPLICATION

PROPOSER: ***Joan Byamugisha***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Brian DeRenzi***

ABBREVIATION: SMSDP

Brief Description: Research has shown that providing patients with personalized medical information about their diagnosis and treatment results in their improved compliance to the prescribed treatments, which further results in better patient outcomes and reduced healthcare costs [1, 2, 4]. It has thus been recommended that patients be provided with information customized to their situation, in a language and format they can understand [3].

This project focuses specifically on personalized drug prescription explanations, and further on the production of such explanations in Runyankore, a Bantu language indigenous to Uganda. Runyankore was selected to localize the drug explanations because the use of English in medical information exacerbates literacy difficulties already prevalent in situations of health [3]. This is part of a larger project in which Runyankore sentences have been generated through ontology verbalization, and this project is aimed at using the Runyankore NLG system to generate and deliver personalized drug explanations via SMS.

Aims:

a. to develop an interface through which the different attributes associated with drug prescription explanations can be customized;

b. to develop an application which links the personalization interface to the Runyankore NLG system, retrieves the generated explanations, and sends them via SMS to a designated number; and

c. to deploy and evaluate the developed application.

[1] Cawsey et al., (2000). The Evaluation of a Personalized Health Information System for Patients with Cancer. User Modeling and User-Adapted Interaction Journal, Vol. 10, No. 1.

[2] DiMarcko et al., (2007). The Development of a Natural Language Generation System for Personalized e-Health Information. In proceedings of the 12th International Health (Medical) Informatics Congress (Medinfo 2007, Brisbane, Australia.

[3] DiMarcko et al, (2009). Self-Managed Access to Personalized Healthcare through Automated Generation of Tailored Health Educational Materials. In proceedings of the American Association for Artificial Intelligence (AAAI) Fall Symposium on Virtual Health, Washington D. C., USA.

[4] Wilcox et al, (2011). Characterizing Patient-Friendly Micro-Explanations of Medical Events. In proceedings

Computer Science Content: HCI, ICT4D

Specific Learning Outcomes: By the end of the project, the student should be able to:

a. design an appropriate user interface, which allows for the easy update and customization of drug prescription attributes according to patient;

b. design and implement an SMS-based application; and

c. deploy and evaluate the performance of the application

Skills Required by Team as a Whole: Python, UX design, understanding of design and ICT4D principles

Theory: HCI, ICT4D

Implementation: Python is the preferred language of choice, though Java could work as well.

Data: Test data will be generated during the course of the project

Facilities needed: N/A; existing computers, phones, and open source software will be utilized.

Supervision: I will available to:

a. provide any background explanations, if required,;

b. provide the Runyankore NLG system to interface with;

c. explain/translate any information presented in Runyankore;

d. recruit Runyankore-speaking study participants;

e. work with Dr. DeRenzi on assessing the implementation, deployment, and testing; and

f. avail access to the Deployments reading group, if required.

NUMBER OF STUDENTS: 2

14. PROJECT : HAND GESTURE RECOGNITION

PROPOSER: *James Gain*

CO-SUPERVISOR/EXTERNAL ADVISOR: *Deshen Moodley*

ABBREVIATION: HANDGR

Brief Description: Sign languages are every bit as complex and varied as spoken languages. Much like English is widely regarded as an international norm and standard despite having many thousands of varieties and dialects, American Sign Language (ASL) dominates the representation of sign language. South Africa has many different sign languages, but documenting and developing resources for teaching and learning them is a time-consuming and difficult exercise. As part of the effort to increase the visibility of our sign languages, we would like to develop the tooling to ease these processes. Hand Gesture Recognition (HGR) using camera-based devices such as the Leap Motion Controller has been thoroughly researched and offers a lot of insight into adapting such technologies for sign languages. Additional technologies such as EMG-based HGR devices like the Myo offer new paths of investigation. This project will look at using these HGR technologies to help students learn sign language alphabets using the Leap Motion Controller and the Myo, individually and combined. A variety of machine learning techniques can be employed in developing these systems. This also has implications more generally for control of computer systems through gesture-recognition, which is particularly pertinant to situations where keyboards, or touch-screens are less appropriate, such as in Virtual Reality.

Computer Science Content: Computer Vision, Machine Learning, Human Computer Interaction

Specific Learning Outcomes: Selection, tailoring and application of appropriate machine learning techniques.

Skills Required by Team as a Whole: Roles involve: Development of gesture recognition systems for: (a) Kinnect, (b) Myo, (c) Joint recognition over both streams.

Theory: Moderate – concepts in Computer Vision and Machine Learning . Requires some mathematics.

Implementation: Moderate – extension and implementation of computer vision and machine learning algorithms, likely in C++.

Data: A set of sample gestures to be used for training.

Facilities needed: Xbox Kinnect, Myo, Standard PC.

Supervision: Weekly meetings with supervisor and occasional attendance and presentation at larger postgraduate research group meetings.

NUMBER OF STUDENTS: 3

15. PROJECT : INTERLOCKING PRINTABLE THREE-DIMENSIONAL PUZZLES

PROPOSER: *James Gain*

ABBREVIATION: PUZLOCK

Brief Description: Interlocking puzzles are a type of three-dimensional puzzle where solid pieces lock together to form a final stable shape. Usually, they must be assembled in a specific order and this provides the puzzle element. Finding a valid interlocking puzzle given an enclosing volume for the final target shape is an interesting problem in computational geometry that has been solved by Song et al. (see ref). However, their solution requires a blocky outer surface aligned on a regular grid and can be very slow (up to 10 hours). This project will involve speeding up the computation using multiple CPUs in a cluster and also by exploiting GPU co-processors. A second goal is to generalize the algorithm to allow a detailed non-voxelised outer surface that more closely resembles the target shape (e.g., a bunny, teapot, etc). Finally, the design must be physically realised on a 3D printer and this requires a certain amount of geometric post-processing.

Reference:

Peng Song, Chi-Wing Fu, and Daniel Cohen-Or. “Recursive interlocking puzzles”. ACM Transactions on Graphics, 31, 6, Article 128 (November 2012).

Computer Science Content: Computer Graphics, Computational Geometry, 3D Printing, Parallel Computing

Specific Learning Outcomes: Exposure to optimisation techniques and algorithm restructuring for CPU and GPU.

Skills Required by Team as a Whole: research-based development, optimisation. Roles involve: (a) optimising for CPU, (b) optimising for GPU, (c) geometric restructuring for 3D printing.

Theory: Moderate – well defined concepts in Computer Graphics and Computational Geometry. Requires some mathematics.

Implementation: Challenging – extension and implementation of computational geometry algorithms, optimisation on multi-core CPU and GPU, geometric restructuring for 3D printing.

Data: Simple geometric models easily sourcable from the internet.

Facilities needed: Multicore CPU with reasonable GPU. Access to Hex Cluster and 3D Printer and Material will be provided.

Supervision: Weekly meetings with supervisor and occasional attendance and presentation at larger postgraduate research group meetings.

NUMBER OF STUDENTS: 3 (OR 2)

16. PROJECT : INTERFACES FOR GEOMETRIC MODEL PLACEMENT

PROPOSER: *James Gain*

CO-SUPERVISOR/EXTERNAL ADVISOR: *Brian DeRenzi*

ABBREVIATION: GEOPLACE

Brief Description: Many 3D animation and modelling packages rely heavily on 3D user interfaces for authoring and displaying shapes and scenes. The main challenge is to overcome the limitations of a 2D display (LCD screen) and 2D input device (mouse). A key task in such modelling is the placement of geometric objects in a 3D scene by a combination of translation and rotation. While it is typically possible to perform the corresponding task in the physical world in under a second, this usually takes up to ten seconds in modelling packages. This suggests that improvements can be made to existing software by adopting and simulating characteristics from the real world.

Each team member in this project will develop an alternative method for placing 3D objects in a scene and compare it to the current best-practice standard using an extensive user experiment. While this project does require 3D user interface development, there will necessarily be a strong focus on formal quantitative experiment design.

Reference:

Steven Rybicki, Brian DeRenzi, and James Gain. “Usability and Performance of Mouse-based Rotation Controllers”, Graphics Interface 2016, 1-3 June, Victoria, Canada.

Computer Science Content: Computer Graphics, 3D User Interfaces

Specific Learning Outcomes: Iterative development of 3D user interfaces; practice with user experiment design and analysis

Skills Required by Team as a Whole: A prior background in psychology or statistics will be beneficial. Team members will develop independent interfaces and compare against a best-practice existing approach.

Theory: Moderate – effective design and analysis of large-scale quantitative user experiments. Some basic affine transformation concepts from introductory computer graphics.

Implementation: Moderate - Designing and developing 3D user interfaces, likely in Unity.

Data: Nothing that is not easily sourced from the internet.

Facilities needed: Standard PC

Supervision: Weekly meetings with supervisor and occasional attendance and presentation at larger postgraduate research group meetings.

NUMBER OF STUDENTS: 2

17. PROJECT : CROWD PATCHES WITH VARYING POPULATIONS

PROPOSER: *Ulysse Vimont*

Co-supervisor/External advisor: ***James Gain***

Abbreviation: crwpop

Brief Description: Crowd patches are an efficient tool for populating large virtual environments with virtual characters at a low computational cost through pre-computated tileable assets [YMPT09]. However, crowds represented using such structures are notoriously periodic in time, resulting in a constant population and concentration. Recently, Jordao et al. [JPCC14] proposed a method for intuitively creating such crowds and adding some style variations while ensuring animation time-continuity.

This project aims at pushing further the limitations of the patch representation for animated virtual crowds by introducing population-varying patches (e.g. patches representing a portion of animated crowd where the number of individuals changes from *n* to *n+1* or *n-1*). These patches should allow the representation of complex crowd patterns like the variation of crowd density during the day.

References:

[YMPT09] Yersirepn, B., Maïm, J., Pettré, J., & Thalmann, D. (2009, February). Crowd patches: populating large-scale virtual environments for real-time applications. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (pp. 207-214). ACM.

[JPCC14] Jordao, K., Pettré, J., Christie, M., & Cani, M. P. (2014, May). Crowd sculpting: A space-time sculpting method for populating virtual environments. In *Computer Graphics Forum* (Vol. 33, No. 2, pp. 351-360).

Computer Science Content: Computer Graphics, Computational animation, Virtual crowds

Specific Learning Outcomes: Understanding of Animation techniques

Skills Required by Team as a Whole: Roles involve: (a) creating population-varying crowd patches, (b) creating population-varying crowd scenarios, (c) developing the algorithm for laying out patches in space and time for adapting to an input scenario.

Theory: High – need to define new animation representations, structures and algorithms.

Implementation: Moderate – CPU-only implementation, mostly 2D+time interface.

Data: To be created.

Facilities needed: Desktop computer.

Supervision: Weekly meetings with supervisor and occasional attendance and presentation at larger postgraduate research group meetings.

Number of Students: 3 (or 2)

18. Project : Optimization of Test-Driven Development of ontologies

Proposer: ***Maria Keet***

Abbreviation: oton

Brief Description: Test-driven development is a well-known process in software engineering, which contrasts with waterfall methodologies: one defines a function or feature requirement, tests the software doesn't have it, implements the function, and tests that it works according to specification. First comprehensive preliminary results have been obtained to 'port' this TDD approach to ontology development, using a modeller's desired axiom as a feature requirement. Performance is promising but seems to have room for optimisation, refactoring and regression testing is to be worked out, the interface of the current plugin is functional but without consideration of the modeller, and it has not been evaluated with modellers to see whether this TDD methodology is in any way better than other ontology development approaches and methodologies. Your tasks are to solve one or more of these problems, which is mostly research-oriented. Some background information of TDD for ontologies is accessible at:

Keet, C.M., Lawrynowicz, A. Test-Driven Development of Ontologies. 13th Extended Semantic Web Conference (ESWC'16). Springer LNCS vol. 9678, 642-657. 29 May - 2 June, 2016, Crete, Greece. http://www.meteck.org/files/tddOntoESWC16.pdf

Computer Science Content: Artificial Intelligence, Knowledge representation and reasoning, ontologies, Semantic Web, software engineering

Specific Learning Outcomes: Become competent in Semantic Web technologies and ontology engineering.

Skills Required by Team as a Whole: Basics of OWL, Semantic Web technologies, then it depends on the subtopics chosen: programming skills, basic statistics for the reasoner performance experiments, and HCI for usability of the tool.

Theory: OWL and related Semantic Web technologies, automated reasoners, benchmarking, methodology development, HCI and user evaluations

Implementation: Java (most likely), OWL API and/or OWLink to interface with the automated reasoner, Protégé plugins. Extensions for implementing the refactoring requires good programming skills, the others less so.

Data: Users are needed for the HCI/methodology part to collect data from (if that part is chosen). The current TDDonto2 plugin and the other software is freely available.

Facilities needed: PC with usual development environment, some of the Semantic Web.

Supervision: There will be regular project meetings with the supervisor. Background material will be provided. There may also be interaction with the Polish PI in the ARISTOTELES project, Dr. Lawrynowicz.

NUMBER OF STUDENTS: 3 (OR 2)

19. PROJECT : SPELLCHECKERS FOR ISIZULU AND ISIXHOSA

PROPOSER: *Maria Keet*

EXTERNAL ADVISOR: *Dr. Khumalo (isiZulu), Dr. Motinyane-Masoko (isiXhosa)*

ABBREVIATION: ALSPEL

Brief Description: First successes have been obtained with a data-driven spellchecker for isiZulu that uses a statistical language model for error detection. It is currently the only end-user usable working isiZulu spellchecker. While its accuracy can be improved and users would like it also as a Chrome plugin, the most desired new feature for the isiZulu spellchecker is error correction. One project member's task is to realise this error correction, which has not been tried yet for any of the Nguni languages (so this is very much a research project). Some members of the African Languages section at UCT would like a spellchecker for isiXhosa. The second component of this project is expected to be a software engineering project to develop the isiXhosa spellchecker. It has the option to be morphed into a research project, focusing on bootstrapping and language (dis)similarity rather than usability of the tool.

The current spellchecker, code, and some more information about it can be found at: https://keet.wordpress.com/2016/11/11/launch-of-the-isizulu-spellchecker/

Computer Science Content: Natual Language Processing, Computational Linguistics

Specific Learning Outcomes: Design, implement, and evaluate NLP/CL systems, research.

Skills Required by Team as a Whole: Basic mathematical and statistical modelling, programming, software development methodologies. For the error correction, some knowledge of the chosen language may be helpful. Knowledge of isiXhosa is not required.

Theory: Basic rule-based approaches and statistical modelling as applied in computing.

Implementation: a programming language of choice

Data: Datasets (isiXhosa text documents) will be provided, and, if isiZulu is chosen, some of the grammar will be provided (in computer processable format) if deemed needed for the error correction (nouns and verbs, and some phonological conditioning rules).

Facilities needed: PC with usual development environment.

Supervision: There will be regular project meetings with the supervisor. For isiZulu, a linguist (Dr. Khumalo, UKZN) may be involved, depending on the language skills of the student. For the isiXhosa spellchecker, linguists of the African Languages Section will be involved (Dr. Motinyane-Masoko, UCT).

NUMBER OF STUDENTS: 2 (OR 3)

20. PROJECT : MODEL-BASED SOFTWARE INTEROPERABILITY

PROPOSER: *Maria Keet*

ABBREVIATION: INTEROP

Brief Description: New large software systems are becoming more complex, and existing software has to be integrated due to collaborations, analysis of scientific data, company mergers and acquisitions, or for better service delivery management in public administration. Typically, this requires design of the system in different languages, such as an EER diagram for a database backend and one or more UML diagrams for the application front-end, which have to link up seamlessly. Currently, there is no software support for this, and the workarounds are time-consuming and error-prone. Steps to solve this problem have been made, notably on a unifying metamodel of UML, EER, and ORM, its formalization, an approach for model transformation and interoperability, and a small set of rules for the main entities to achieve that, as proof of principle. There are several theoretical and software engineering aspects remaining to realise the interoperability in praxis, and your project will cover one or more of the following, depending on the team's skills: algorithm design and implementation for model interoperability that interacts with the formalised metamodel, for finding possible alignment patterns and declaring links, novel principles for pattern-based refactoring of model fragments, model (module) management. It also entails a '21st century design' for the old proof-of-concept logic-based modelling tool Icom that currently can handle only multiple models in one custom graphical modelling language (rather than commonly known notations), uses the ALCQI fragment of OWL2 only, and has the old Racer reasoner behind the graphics in the back-end. Language feature coverage considerations may be either breadth (OWL 2 DL-ish) or the so-called 'core fragment' that is exceedingly suitable for Ontology-Based Data Access. Some background information can be found at http://www.meteck.org/SAAR.html.

Computer Science Content: Artificial Intelligence, Knowledge representation and reasoning, rule specifications/languages in theory and practice, conceptual modelling languages, software engineering.

Specific Learning Outcomes: rule-based systems, modelling and CASE tools, formal foundations of software systems, software development methodologies

Skills Required by Team as a Whole:

Theory: basic knowledge of UML/EER/ORM and some logics (DL, OWL), algorithm design, good programming skills for the implementation.

Implementation: mainly Java, code repositories, XML, ATL, OWLink

Data: current Icom code will be provided (which can be used either as inspiration or to modify and extend)

Facilities needed: PC with development environment; Background material will be provided.

Supervision: There will be regular project meetings. If necessary, there may be contact also with collaborators in this multi-year project: Prof Fillottrani (Universidad Nacional del Sur) regarding the software, and Mrs Khan (CS PhD student at UCT) regarding module generation and management.

NUMBER OF STUDENTS: 2-4

21. PROJECT : AUTOMATED MARKING OF LANGUAGE LEARNING EXERCISES

PROPOSER: *Maria Keet*

ABBREVIATION: ALLEx

Brief Description: If you have learned another natural language at school or university, you will be familiar with those typical limited paper-based exercises, such as 'complete the sentence', 'write down the plural', 'conjugate this', and so on, which are then manually marked or the answer has to be found by looking it up in the end of the textbook. This is time- and resource-consuming and somewhat inflexible. Some work has been done on computer-based language learning exercises, such as with root questions rather than fixed ones and trying to identify levels of difficulty (e.g., asking for plurals of regular nouns only cf. irregular ones as well). Good exercises have variation so as to limit the rote-learning effect, which can be achieved with a natural language generation (NLG) component. Several core theories, techniques, and tools for natural language generation are available, but they may not be readily usable for the tasks at hand (e.g., SimpleNLG and the (functional programming-based) Grammatical Framework). Your task is to develop an NLG-based language learning platform with automated question generation and automated marking of the answers a learner provides.

While isiZulu or isiXhosa would be the much preferred natural language for the exercises, another language may be considered (Spanish, Italian, German, French, Dutch, or Afrikaans), which involves more or less statistical or knowledge representation-based NLG design, depending on the chosen language. More information on IsiZulu knowledge-to-text NLG can be found on the GeNI project page at http://www.meteck.org/files/geni/.

Computer Science Content: Natural Language Processing/Computational Linguistics, natural language generation, educational technologies, computer-assisted language learning.

Specific Learning Outcomes: at the end of the project, you will have a working knowledge of natural language generation systems (one part of the project) and computer-assisted language learning technologies (the other part of the project). This is expected to be a software development project that includes learning new technologies.

Skills Required by Team as a Whole: natural language generation systems, e-learning, software engineering methodologies. Knowledge of a language other than English is preferred, but not essential.

Theory: natural language generation components (no prior knowledge of that is required), systematics of online questions, Grammatical Framework

Implementation: programming ability in an appropriate programming language to develop the system, including functional programming if Grammatical Framework is used.

Data: none required; samples of typical language exercises can be provided for some of the target languages. If isiZulu is chosen, in-house developed NLG algorithms and code can be reused.

Facilities needed: PC with development environment, the departmental automarker may be provided if that technology is chosen as one of the components.

Supervision: There will be regular project meetings. Background material will be provided.

Bursaries: no, but a paid vac job (programmer) will be available if isiZulu or isiXhosa is chosen.

NUMBER OF STUDENTS: 2 (OR 3)

22. PROJECT : CROWDSOURCING FOR COLLABORATIVE TERMINOLOGY DEVELOPMENT

PROPOSER: ***Maria Keet***

ABBREVIATION: COMC

Brief Description: Terminology development in the sense of the process of inventing and standardising new terms done in the 'standard' way is exclusionary: typically small workshops with mainly terminologists and linguists are used, rather than domain experts or even asking the public on which term they prefer. It has been shown experimentally that this can lead to suboptimal vocabularies for various reasons (Keet and Barbour, 2014), especially when the terms are not stable and accepted in the particular natural language. The idea is to democratise the process by involving 'the masses' in online games through tailor-made crowdsourcing software with 1 and 2-player games. Your task is to design and implement the crowdsourcing tool that can meet the full set of requirements with its peculiar data storage needs, run at least one experiment with the data collection (computer science terms, optionally also another discipline), and analyse (mine and recompute) the results. On the latter, e.g., different payoffs for term proposals and voting may result in a different final preferred vocabulary, which the software should be able to compute as a different scenario. Regarding the language(s) for term harvesting, one (or more) of the official languages of South Africa is preferred, but not a requirement, and knowledge of the chosen language is not required.
Keet, C.M., Barbour, G. Limitations of Regular Terminology Development practices: the case of the isiZulu Computing Terminology. *Alter*nation, 2014, 12: 13-48.

Computer Science Content: Advanced database technologies, data analytics, software development, quantitative experiments, crowdsourcing, optionally some interface design.

Specific Learning Outcomes: crowdsourcing technologies, working knowledge of some advanced database features, experiment design and execution, software development methodologies.

Skills Required by Team as a Whole:

Theory: data analysis, algorithm design, some concurrency (for the 2-player game)

Implementation: databases/data store, web programming.

Data: to be collected as part of the project.

Facilities needed: PC with usual development environment and RDBMS software.

Supervision: There will be regular project meetings. Background material will be provided.

NUMBER OF STUDENTS: 2

23. PROJECT : DEFEASIBLE REASONING FOR ONTOLOGIES IN PROTÉGÉ

PROPOSER: ***Tommie Meyer***

ABBREVIATION: DROP

Brief Description: Protégé is a free and open-source ontology editor. Using Protégé it is possible to employ classical logical reasoning to build high-quality ontologies and to query them, thereby completing the first step in building intelligent systems. Its plugin architecture allows for extending the basic capabilities built into the core Protégé system. The goal of this project is to extend the capabilities of Protégé with a form of non-classical reasoning, known as defeasible reasoning.

Defeasible reasoning is a logical system with an existing implementation that is able to deal with exceptions to general rules in a systematic and well-defined manner. Because defeasible reasoning differs from classical reasoning, the core Protégé system is not able to deal with it.

It is also possible to develop an environment for defeasible reasoning from scratch, instead of adding it on top of Protégé. These two approaches---building on top of an existing system vs. developing an environment from scratch---both have their advantages and disadvantages. In this project we forego the flexibility provided by developing such an environment from scratch, and opt for the convenience of using the existing Protégé system.

Different team members will be tasked with extending different components of Protégé, enabling the system to represent and reason with defeasible ontologies in different ways.

Computer Science Content: Artificial Intelligence, Knowledge Representation and Reasoning.

Specific Learning Outcomes: Exposure to the theoretical and the implementational aspects of logic-based reasoning.

Skills Required by Team as a Whole:

Theory: Defeasible reasoning, logic-based reasoning, the basis of which is being provided in the LAI honours course.

Implementation: The implementation, once the theory is well-understood, is of average difficulty. Access will be provided to an existing defeasible reasoner.

Data: Access to defeasible ontologies (to be provided).

Facilities needed: A good computer; access to Protégé, and the OWL API (which is freely available).

Supervision: Regular meetings with the supervisor (roughly, every two weeks).

NUMBER OF STUDENTS: 2 (OR 3)

24. PROJECT : PREFERENTIAL REASONING FOR ONTOLOGIES IN PROTÉGÉ

PROPOSER: *Tommie Meyer*

ABBREVIATION: PROP

Brief Description: Protégé is a free and open-source ontology editor. Using Protégé, it is possible to employ classical logical reasoning to build high-quality ontologies and to query them, thereby completing the first step in building intelligent systems. Its plugin architecture allows for extending the basic capabilities built into the core Protégé system. The goal of this project is to extend the capabilities of Protégé with a form of reasoning, known as preferential reasoning. Preferential reasoning is a logical system, based on classical reasoning, that is able to deal with inconsistent ontologies in a systematic and well-defined manner.

It is also possible to develop an environment for preferential reasoning from scratch, instead of adding it on top of Protégé. These two approaches---building on top of an existing system vs. developing an environment from scratch---both have their advantages and disadvantages. In this project we forego the the flexibility provided by developing such an environment from scratch, and opt for the convenience of using the existing Protégé system.

This project has a theoretical component, and one (or possibly two) implementational components. One team member will be tasked with the theoretical component. The other(s) will be tasked with extending different aspects of Protégé to be able to deal with preferential reasoning.

Computer Science Content: Artificial Intelligence, Knowledge Representation and Reasoning.

Specific Learning Outcomes: Exposure to the theoretical and the implementational aspects of logic-based reasoning.

Skills Required by Team as a Whole :

Theory: Preferential reasoning, logic-based reasoning, the basis of which is being provided in the LAI honours course.

Implementation: The implementation, once the theory is well-understood, is of average difficulty. Access will be provided to an existing classical reasoner.

Data: Reasonably-sized ontologies (which are freely available).

Facilities needed: A good computer; access to Protégé, and the OWL API (which is freely available).

Supervision: Regular meetings with the supervisor (roughly, every two weeks).

NUMBER OF STUDENTS: 3 (OR 2)

25. PROJECT : DEFEASIBLE REASONING FOR ONTOLOGIES

PROPOSER: ***Tommie Meyer***

ABBREVIATION: DRO

Brief Description: The goal of this project is to implement a basic ontology editing environment for a form of non-classical reasoning, known as defeasible reasoning. Defeasible reasoning is a logical system, with an existing application, that is able to deal with exceptions to general rules in a systematic and well-defined manner. The system will be used to construct high-quality defeasible ontologies and to query them, thereby completing the first basic steps in building intelligent systems that are able to deal with defeasibility.

It is also possible to provide such an environment for defeasible reasoning within existing systems such as the Protégé ontology editing environment. These two approaches---building on top of an existing system vs. developing an environment from scratch---both have their advantages and disadvantages. In this project we opt for the flexibility provided by developing such an environment from scratch, and forego the convenience of building it on top of the existing Protégé system.

Different team members will be tasked with implementing different aspects of the environment, to be able to represent and reason with defeasible ontologies in different ways.

Computer Science Content: Artificial Intelligence, Knowledge Representation and Reasoning.

Specific Learning Outcomes: Exposure to the theoretical and the implementational aspects of logic-based reasoning.

Skills Required by Team as a Whole:

Theory: Defeasible reasoning, logic-based reasoning, the basis of which is being provided in the LAI honours course.

Implementation: The implementation, once the theory is well-understood, is of average difficulty. Access will be provided to an existing defeasible reasoner.

Data: Access to defeasible ontologies (to be provided).

Facilities needed: A good computer and access to the OWL API (which is freely available).

Supervision: Regular meetings with the supervisor (roughly, every two weeks).

NUMBER OF STUDENTS: 2 (OR 3)

26. PROJECT : PREFERENTIAL REASONING FOR ONTOLOGIES

PROPOSER: ***Tommie Meyer***

ABBREVIATION: PRO

Brief Description: The goal of this project is to implement an ontology editing environment for a form of non-classical reasoning known as preferential reasoning. Preferential reasoning is a logical system, based on classical reasoning, that is able to deal with inconsistent ontologies in a systematic and well-defined manner. The system will be used to construct high-quality preferential ontologies and to query them, thereby completing the first basic steps in building intelligent systems that are able to deal with preference.

It is also possible to provide such an environment for preferential reasoning within existing systems such as the Protégé ontology editing environment. These two approaches---building on top of an existing system vs. developing an environment from scratch---both have their advantages and disadvantages. In this project we opt for the flexibility provided by developing such an environment from scratch, and forego the convenience of building it on top of the existing Protégé system.

This project has a theoretical component, and one (or possibly two) implementational components. One team member will be tasked with the theoretical component. The other(s) will be tasked with extending different aspects of Protégé to be able to deal with preferential reasoning.

Computer Science Content: Artificial Intelligence, Knowledge Representation and Reasoning.

Specific Learning Outcomes: Exposure to the theoretical and the implementational aspects of logic-based reasoning.

Skills Required by Team as a Whole :

Theory: Preferential reasoning, logic-based reasoning, the basis of which is being provided in the LAI honours course.

Implementation: The implementation, once the theory is well-understood, is of average difficulty. Access will be provided to an existing classical reasoner.

Data: Reasonably-sized ontologies (which are freely available).

Facilities needed: A good computer, and access to the OWL API (which is freely available).

Supervision: Regular meetings with the supervisor (roughly, every two weeks).

NUMBER OF STUDENTS: 3 (OR 2)

27. PROJECT : A COMPUTER VISION SYSTEM FOR GALAXY DETECTION AND CLASSIFICATION FROM OPTICAL RADIO ASTRONOMY IMAGES

PROPOSER: ***Deshen Moodley***

ABBREVIATION: ASTCVS

Brief Description: This project aims to incorporate both traditional machine learning, probabilistic graphical models (Bayesian Networks) and knowledge based systems to develop components of a cognitive vision system for detecting and classifying galaxies and other phenomena from optical radio astronomy images.

The project will involve three distinct roles to create different executable components of a generalised computer vision system for optical astronomy imagery

1. Image processing and feature extraction: This will involve implementing and evaluating various image processing techniques for pre-processing, feature extraction, and dimensionality reduction of the extracted features.
2. Statistical machine learning: Various machine learning techniques will be implemented and tested to classify and identify phenomena, e.g. galaxy type, from different combinations of features. This task will also explore automating hyper parameter optimisation, model selection and model configuration.
3. Knowledge modelling: This will involve the development of one or more Bayesian Networks and an ontology to encode probabilistic causal chains, i.e. which combination of features/observations must be present to increase the likelihood of the presence of a complex phenomenon, e.g. galaxy type, dark matter etc.

Each component will expose a web service interface so that it can be executed with different options/parameters.

Computer Science Content:

Image processing and computer vision, machine learning, ontology engineering, Bayesian Artificial Intelligence, distributed web services

Specific Learning Outcomes: designing and implementing a cognitive vision system for astronomy, machine learning, knowledge based systems, image processing

Skills Required by Team as a Whole:

Theory: Machine learning, image processing, ontologies, Bayesian Artificial Intelligence, basic astronomy would be an advantage, but is not essential

Implementation: Each component must be implemented and wrapped as a generic executable service. Configure and conduct experiments using a machine learning framework: Apache MLib/Spark, and OpenCV. Implement a tool to inference over a Bayesian Network and an ontology

Data: The Galaxy Zoo data set from Kaggle and potentially other data sets from the SKA

Facilities needed: A good computer with an appropriate development environment

Supervision: Regular meetings with supervisor, roughly every two weeks

NUMBER OF STUDENTS: 3 (4 WITH SOME EXPANSION)

28. PROJECT : AUTONOMOUS SELF-LEARNING AGENTS IN 3D VIRTUAL WORLDS

PROPOSER: *Deshen Moodley*

CO-SUPERVISOR/EXTERNAL ADVISOR: *Tommie Meyer*

ABBREVIATION: SLA3DW

Brief Description:

Cognitive software agents are programs that use artificial intelligence techniques to autonomously learn, act and adapt to a changing environment. There are numerous application domains for cognitive agents, including autonomous robots, computer games, personal recommender agents and adaptive computer security systems. However, designing and evaluating these agents are often challenging. This project involves extending and testing an existing 3D virtual world simulation platform developed within the Adaptive and Cognitive Systems Lab. The platform uses the Unity 3D game engine and introduces a simulation harness and test bed for designing and evaluating learning and adaptation mechanisms in different 3D virtual world scenarios. The current agent architecture, QCog, uses reinforcement learning to learn a strategy to enable a humanoid to adapt and survive against various predators in the world. The system has a perception component which transforms sensory input from the world into situations of interest or perceptions, a production memory component which uses a rules engine to rank and propose actions based on current perceptions, and a Q-Learning module which uses reinforcement learning to learn and adapt an optimal strategy to achieve a set of goals. The current system has been tested on an application case study involving a predator-prey scenario, i.e. the humanoid learning to survive against encounters with dangerous predators in the world.

Team members will select one of the following tasks:

- Develop and evaluate a new scenario in the 3D world using the existing learning mechanism. This will involve a research component to investigate and identify core metrics for measuring adaption and self-learning. The implementation component will involve designing and implementing one or more reference virtual world scenario in Unity/QCog to evaluate the performance of various learning and adaption strategies based on these metrics.
- Implement and test a planning component using Partially Observable Markov Decision Processes. (This has already been implemented in another project)
- Develop and test a richer rules based engine for the perception component using Description Logics. The current engine uses simple production rules implemented in the Java based Jess Engine.

Computer Science Content: 3D Graphics, Cognitive Agents, Reinforcement learning, Description Logics, Partially Observable Markov Decision Processes

Specific Learning Outcomes: To design, implement and evaluate a cognitive agent system that self-learns and adapts in dynamic worlds with continuous observations.

Skills Required by Team as a Whole:

Theory: 3D Graphics, Cognitive Agents, Reinforcement learning, Description Logics, Partially Observable Markov Decision Processes

Implementation: Unity 3D, QCog, C#, Java

Data: The in-house QCog/Unity simulation platform will be made available to students who take this project.

Facilities needed: A good computer with an appropriate development environment, and a good graphics card

Supervision: Regular meetings with supervisor, roughly every two weeks

NUMBER OF STUDENTS: 3

29. PROJECT : STREAMING LEARNING

PROPOSER: ***Deshen Moodley***

ABBREVIATION: STRMLN

This project aims to incorporate both traditional machine learning, probabilistic graphical models (Bayesian Networks) and knowledge based systems to develop generic components of a cognitive system for learning from continuous data streams. These components will be evaluated on two data sets/application domains, i.e. JSE stock market data and global daily weather data.

The project will involve three distinct roles to create three executable components of a generalised stream learning system. Each component will provide a web service interface so that it can be invoked by other components.

1. Perception component: This will involve implementing a rules engine, various aggregation functions, temporal structures, e.g. sliding windows, to view and analyse data at different temporal scales. This layer will transform quantitative measurements from incoming streaming data to qualitative states or situations of interest. This component will be evaluated and tested for data analysis, trend detection, and change detection
2. Predictive component: Various machine learning techniques will be implemented and tested to predict future situations based on time series data. The module must be able to predict values for subsequent time steps as well as future trend changes.
3. Causal modelling component: This component will involve using a Bayesian Network to model a cause and effect network to explain which events most likely lead to an observed situation.

Each component will expose a web service interface so that it can be executed with different options/parameters.

Computer Science Content: Machine learning, Bayesian Networks, Knowledge Based Systems

Specific Learning Outcomes: Stream learning, machine learning for prediction, temporal aggregation and reasoning

Skills Required by Team as a Whole:

Theory: Machine learning, Bayesian Networks, rules based systems, time series analysis

Implementation: Machine learning or analytics platform: scipy-learn, Mlib

Data: JSE stock market data and global daily weather data is available

Facilities needed: A good computer with an appropriate development environment

Supervision: Regular meetings with supervisor, approximately every two weeks

NUMBER OF STUDENTS: 3

30. PROJECT : ANALYSING AND PREDICTING UNDERGRADUATE COMPUTER SCIENCE PERFORMANCE USING MACHINE LEARNING

PROPOSER: ***Deshen Moodley***

ABBREVIATION: STDPRF

Brief Description:

The aim of the project is to use machine learning and Bayesian Networks to analyse and model characteristics of South African students registering for Computer Science at UCT. The project will explore the impact of various factors on student progression and performance. These factors will include the matric points and NBT scores, the characteristics of the school that a student attended, including the quintile of the School, whether it is urban or rural and the province in which it is located. The resultant machine learning system will take as input the characteristics of a specific student and will output the most likely expected undergraduate performance of the student. The project will explore both traditional machine learning techniques and Bayesian Networks. Such a model can be used to facilitate selection and intervention to increase student throughput in undergraduate Computer Science programs.

Computer Science Content: Machine learning, Bayesian Networks

Specific Learning Outcomes: Learn how to use machine learning and Bayesian Networks for prediction

Skills Required by Team as a Whole:

Theory: Machine learning, Bayesian Networks

Implementation: A machine learning framework, Scipy-learn, Apache MLib/Spark or Weka. Implement a tool to inference over a Bayesian Network.

Data: UCT undergraduate Computer Science student data and the South African master schools list is available

Facilities needed: A good computer with an appropriate development environment

Supervision: Regular meetings with supervisor, approximately every two weeks

NUMBER OF STUDENTS: 2

31. PROJECT : FARMAID

PROPOSER: ***Aslam Safla***

ABBREVIATION: FARMAID

Brief Description: This project involves the creation of a mobile app to provides crop-specific information to illiterate farmers with the help of various audio visual tools. The app will provide them with high tech solutions like analytics and could work even without an internet connection once it is transferred to a farmer's Android smart phone. The app should break the language barrier by providing information in regional languages.

Computer Science Content: Human-Computer Interaction for Development, interface design, mobile application development.

Specific Learning Outcomes: Scientific Students will learn about the complexities of real-world deployments.

Skills Required by Team as a Whole:

Theory: User-Centred Design, Mobile Development, Human-Computer Interaction for Development

Implementation: The project will require the development of a mobile application with a GUI interface so that even illiterate farmers can use the application**Data:** Radio astronomy data from VLA, GMRT and MeerKAT.

Data: Nothing that is not easily sourced from the internet.

Facilities needed: Computers for software development, Test phones for development

Supervision: Regular meetings with supervisors.

NUMBER OF STUDENTS: 3

32. PROJECT : EXTENDING PARAVIEW FOR 3D RADIO ASTRONOMY VISUALISATION

PROPOSER: ***Rob Simmonds***

ABBREVIATION: 3DRA

Brief Description: The aim of this project is to add Radio Astronomy visualisation capabilities to the Paraview data analysis and visualization application (http://www.paraview.org/). Radio Astronomy visualisation is generally done in 2D, viewing slices from multi-dimensional cubes. In this project you will create readers for radio astronomy data held in FITS and HDF5 formats.

Assess how to process the data to reveal 3D structures within the data and user Paraview to display these. You will also assess the capabilities of Paraview to perform visual analytics similar to those performed in the cyberSKA viewer, to expose other features. This project will contribute to other work considering ways of analysing data from MeerKAT and the future SKA telescopes, which will produce data cubes that are too large to be explored in existing viewers.

Computer Science Content: Image processing, visualisation, data anaysis

Specific Learning Outcomes: Scientific data manipulation and anslysis.

Skills Required by Team as a Whole: C++, image processesing, visualisation.

Implementation: Using Paraview C++ API.

Data: Radio astronomy data from VLA, GMRT and MeerKAT.

Facilities needed: African Research Cloud

Supervision: In callaboration with IDIA visualiation team.

NUMBER OF STUDENTS: 2 (OR 3)

33. PROJECT : ROLE BASED AUTHORIZATION IN FEDERATED CLOUDS

PROPOSER: ***Rob Simmonds***

ABBREVIATION: RBAUTHZ

Brief Description: The aim of this project is to assess authorization tools that are currently available and document how they could be used to support role based authorization in the African Research Cloud. This is a federated cloud system that currently has instances at UCT and NWU, is being extended to include other South African universities, and in the future to include sites in other African countries. The aim is to provide authorisation services that act both at the Open Stack Infrastructure as a Service level and also for higher level services such as Scientific Gateways. This should enable group leads to add and remove users, and allow site administrators to enable and disable access to local services for whole groups rather than individual users.

For this project, you should consider how to use of tools such as Grouper, COmanage, Perun and VOMs, to achieve the project objectives.

Computer Science Content: Security, Networking.

Specific Learning Outcomes: Cloud interoperability, Authenticaion and Authorization techniques.

Skills Required by Team as a Whole: Security concepts.

Theory: Federated Identity Management, Authiroatiion techniquies.

Implementation: intergrating and extending existing networked services.

Data: None.

Facilities needed: Simple servers. These can be hosed on the African Research Cloud.

NUMBER OF STUDENTS: 2 (OR 3)

34. Project : Honours Lab Locker Controller

Proposer: *Gary Stewart*

Co-supervisor/External advisor: *Sam Chetty and Craig Balfour*

Abbreviation: LockIT

Brief Description: The new Honours lab lockers currently have no locking mechanism, which is not ideal. The intention is build an integrated system which includes the following components:

- An "embedded" device component controlled by a microcontroller like an Arduino or single board computer like a Raspberry Pi which manages the locking mechanism(s), also with a digital display
- A mobile application which allows users to lock and unlock a locker, amongst other things
- A web based application for users and administration of the system

Computer Science Content: Embedded devices, Internet of Things, Mobile Development, Web Development

Specific Learning Outcomes: Developing an integrated embedded device, mobile and web based system.

Skills Required by Team as a Whole: Python, Mobile Development, Web Development

Theory: Embedded devices, Internet of Things, Mobile Development, Web Development

Implementation: Developing an integrated embedded device, mobile and web based system.

Data: None

Facilities needed: Arduino or Raspberry Pi, Mobile device(s), Locking Mechanism

Supervision: Regular meetings with supervisor, approximately every two weeks.

Number of Students: 3 (or 2)

35. Project : UCT Computer Science Community App

Proposer: *Gary Stewart*

Co-supervisor/External advisor: *Stephan Jamieson*

Abbreviation: CS@UCT

Brief Description: We would like to make UCT Computer Science one of the most vibrant departments on campus. Also to make it more inclusive and able to better engage students who may feel left out, e.g. first year students, female students and inexperienced students who are new to computing. So although these groups may not be specifically targeted, it is hoped that creating a system which better engages Computer Science students at large will create a better sense of belonging overall.

UCT CS already has fairly good mechanisms for student engagement – class representative system, first year mentoring and student societies.

What is envisaged is a mobile application and web-based application, possibly integrated with Vula, which features some of the existing student engagement mechanisms and provides other ways of better interacting with students and creating a greater sense of belonging.

Computer Science Content: Software Engineering, Mobile Development, Web Development

Specific Learning Outcomes: Developing a typical real world integrated mobile and web based system.

Skills Required by Team as a Whole: Mobile Development and Web Development technologies

Theory: Software Engineering, Mobile Development, Web Development

Implementation: Developing an integrated mobile and web based system.

Data: None

Facilities needed: Mobile device(s)

Supervision: Regular meetings with supervisor, approximately every two weeks.

NUMBER OF STUDENTS: 2

36. PROJECT : RASPBERRY PI SENSE HAT EDUCATIONAL FRAMEWORK

PROPOSER: ***Raspberry Pi Sense Hat Educational Framework***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Dale Taylor (Physics Lecturer)***

ABBREVIATION: RASPISE

Brief Description: The Raspberry Pi single board computer is increasingly used in education, particularly Computer Science education, to make learning more exciting. The Sense Hat is an add-on board with a LED grid digital display, sensors for measuring motion and temperature, and a simple joystick.

The proposed solution would involve creating educational frameworks which would engage students to learn programming. Two ideas come to mind:

- using the LED grid and joystick to for programming some old-style arcade based game(s)
- using the motion sensors, e.g. accelerometer and gyroscope, to reinforce programming language constructs and Physics concepts

Computer Science Content: Single Board Computers, Computer Science Education

Specific Learning Outcomes: Creating a set of resources based on a single board computer hardware device to facilitate more engaging learning.

Skills Required by Team as a Whole: Python

Theory: Single Board Computers, Computer Science Education

Implementation: Creating a set of resources based on a single board computer hardware device to facilitate more engaging learning.

Data: None

Facilities needed: Raspberry Pis and Sense Hats

Supervision: Regular meetings with supervisor, approximately every two weeks.

NUMBER OF STUDENTS: 2

37. PROJECT : PERSONAL EMAIL SEARCH

PROPOSER: ***Hussein Suleman***

ABBREVIATION: FINDMAIL

Brief Description: Hussein has a lot of email and he is not alone. Email scalability was never a problem in the past. Things have changed. Now people glibly sent out emails with very large attachments. Mailing lists and spammers send out large amounts to text to millions of people everyday.

In the working world, there is often a need to search and browse through very large collections of emails for auditing purposes, to track down individuals, to verify decisions, etc. However, for purposes of sanity, most users will either delete or archive email after it has been handled.

The goal of this project is to create an email search system for an individual. The system should be simple to use, able to ingest emails from archives, and should provide fast and accurate search and browse functions over email. There could also be non-conventional visualisations of the email, such as graph-based views of email connections, or plug-ins for integration with popular email clients like Thunderbird.

The tools to be produced should work online or off-line, and must be declared open source.

Computer Science Content: Digital Libraries, ICT4D, information retrieval

Specific Learning Outcomes: Experimental research, software development

Skills Required by Team as a Whole: Web application development

Theory: Information retrieval - information will be provided

Implementation: Web applications, IR algorithms

Data: Datasets will be provided by supervisor and collaborators and/or are available from public repositories or harvestable from the Web.

Facilities needed: All resources, such as server access, will be provided.

Supervision: Project will be located within the Digital Libraries Laboratory (dl.cs.uct.ac.za), with regular

meetings and presentations with supervisors and group. External input will come from collaborators at UCT and elsewhere if and when needed.

Bursaries: Possibility for NRF funding – depends on many factors.

NUMBER OF STUDENTS: 2 OR 3

38. PROJECT : SEARCHING FOR DEVELOPMENT

PROPOSER: ***Hussein Suleman***

ABBREVIATION: SEARCH4D

Brief Description: Search is king, if you believe the search engines, that is.

This project is about the development of Information Retrieval (IR) systems to support human and socio-economic development in African countries, especially South Africa. Development is defined in the National Development Plan (NDP) to include areas such as job creation, education and the establishment of identity and dignity. The goal of this particular project is to develop core search technologies that can be used in local environments, and potentially be applicable to a large number of development contexts.

The essence of the project will be to create a document archival system, with a focus on information retrieval, such that users can create accessible collections in the areas of education, medicine, language, etc., with minimal effort and operational in low-resource environments.

The tools to be produced should work online or off-line, and must be declared open source.

Computer Science Content: Digital Libraries, ICT4D, information retrieval

Specific Learning Outcomes: Experimental research, software development

Skills Required by Team as a Whole: Web application development

Theory: Information retrieval - information will be provided

Implementation: Web applications, IR algorithms

Data: Datasets will be provided by collaborators and/or are available from public repositories or harvestable from the Web.

Facilities needed: All resources, such as server access, will be provided.

Supervision: Project will be located within the Digital Libraries Laboratory (dl.cs.uct.ac.za), with regular meetings and presentations with supervisors and group. External input will come from collaborators at UCT and elsewhere if and when needed.

Bursaries: Possibility for NRF funding – depends on many factors.

NUMBER OF STUDENTS: 2 OR 3

39. PROJECT : LEARN.CS.UCT.AC.ZA

PROPOSER: ***Lighton Phiri***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Hussein Suleman***

ABBREVIATION: LEARN

Brief Description: The Department of Computer Science has had a research publication archive (pubs.cs.uct.ac.za) for many years but does not have an active archive of teaching material. This project is about the creation of an advanced repository for the long term storage and dissemination of teaching-related material, such as course notes, slides, tests, memos, past assignments and past exam papers.

There are existing archival tools, such as Eprints and Dspace, that can serve as a basic repository. However, teaching needs are very specific and students may want to obtain all exam papers on a particular topic, or all tests in a particular course. Meeting such needs will require configuration and extension of basic archival software.

Extensions that will enable effective use include: active integration with vula; desktop-based ingest of educational material, using a virtual WebDAV; upstream integration with OpenUCT; a fine-grained permissions system to allow selective access to students/public; and integration with UCT's authentication system. These extensions will form the original contribution of the project.

Computer Science Content: Digital Libraries, ICT4D, information retrieval

Specific Learning Outcomes: Experimental research, software development

Skills Required by Team as a Whole: Web application development

Theory: Information retrieval - information will be provided

Implementation: Web applications, IR algorithms

Data: Datasets will be provided by collaborators and/or are available from public repositories or harvestable from the Web.

Facilities needed: All resources, such as server access, will be provided.

Supervision: Project will be located within the Digital Libraries Laboratory (dl.cs.uct.ac.za), with regular meetings and presentations with supervisors and group. External input will come from collaborators at UCT and elsewhere if and when needed.

Bursaries: Possibility for NRF funding – depends on many factors.

NUMBER OF STUDENTS: 2 OR 3

40. PROJECT : RANKING RESULTS BY LANGUAGE SIMILARITY

PROPOSER: ***Catherine Chavula***

CO-SUPERVISOR/EXTERNAL ADVISOR: ***Hussein Suleman***

ABBREVIATION: SIMRANK

Brief Description: Many African languages, especially those spoken by minority communities, lack digital documents. For example, common information in African languages, such as local newspapers, are generally published in one or two major languages of the country. Finding information written in African Languages on the Web or other multilingual repositories with other dominant languages such as English is very hard, as relevant documents in small languages are buried in results of dominant languages. Although many people are multilingual, they are not proficient in all languages or major languages such as English.

If a person searches for a phrase in isiZulu and gets results in isiXhosa, the results may still be useful and readable. Thus, those results should appear higher up on the list than, say, the results in Italian.

The aim of the project is to develop a ranking function that ranks documents by language similarity and to investigate whether presenting search results (in multiple languages) based on language similarity is useful to the user. The project will have two components: a search engine, which captures query logs, e.g., information about user interaction, and has a Web-based interface for searching and displaying multilingual results; and a ranking algorithm that incorporates the language similarity feature.

Computer Science Content: Information Retrieval (IR), Machine Learning

Specific Learning Outcomes: Experimental research, software development, IR algorithm

Skills Required by Team as a Whole: Programming in Java and/or Python.

Theory: As above. Knowledge of African languages is not a requirement.

Implementation: Web application, IR algorithms and machine learning. No prior background in machine learning and IR is required - these skills will be learnt in the project.

Data: Datasets will be provided by collaborators or will be harvested from the Web. General background reading material will also be provided for each component.

Facilities needed: Standard computing facilities. Open source tools and libraries will be used.

Supervision: Project will be located within the Digital Libraries Laboratory (dl.cs.uct.ac.za), with regular meetings and presentations with supervisors and group. Primary supervision will be by Catherine Chavula, a PhD student.

Bursaries: Possibility for NRF funding – depends on many factors.

Number of Students: 2

41. Project : Popular twitter topics on South African social issues: fake or reality?

Proposer: Selvas Mwanza

Co-supervisor/External advisor: Hussein Suleman

Abbreviation: SASITWIT

Brief Description: Social media refers to the wide range of Internet-based and mobile services that allow real time users to participate in online exchanges, contribute user-created content, or join online communities. With the ease of publishing content, social media has been adopted by many in South Africa as a medium for capturing and sharing real-time information on various topics.

Social media has given rise to a new generation of activism. Social media has not only been at the core of major social stories in South Africa, but drove some of biggest recent events, like #FeesMustFall. However, people are always suspicious of social media. Is there really a widespread movement on a specific issue? Or are there fake or spam accounts being used to influence social media? Recent media reports, even in South Africa, have suggested that social media is not immune from sophisticated forms of spamming.

The goal of this project is to investigate if popular twitter topics on South African social issues represent the majority of South Africans or are orchestrated by spoof users on Twitter. The project work is divided into four tasks, namely:

1. Data collection: Data will be downloaded from Twitter using the Twitter Web API.
2. Data cleaning: Social media data is informal. It contains special tokens, slang, misspellings and grammatical errors. Data cleaning will involve removing special tokens from the data, expanding slang words and correcting misspellings.
3. Graph analysis: A social graph will be created using follower-followee relationships, retweets (RT) and mentions (@username). Graph analysis will involve finding patterns of the sub-graphs induced by the initial adopters of topics.
4. Implementation and visualization: Finally, an algorithm will be implemented using patterns identified in task 3 to identify and visualize topics that represent the majority of South Africans and topics orchestrated by spoof Twitter accounts.

Computer Science Content: Text Mining, Social Media Mining

Specific Learning Outcomes: Experimental research, Software development

Skills Required by Team as a Whole: Programming in any language

Theory: Basic knowledge of graph theory

Implementation: Desktop/Web application, Text mining and social media mining algorithms

Data: Data collection and preparation will be part of the work of the project.

Facilities needed: All resources will be provided.

Supervision: Project will be done in the Digital Libraries Laboratory (dl.cs.uct.ac.za), with regular meetings and presentations with supervisors and group. Primary supervision will be by Selvas Mwanza, a PhD student.

Bursaries: Possibility for NRF funding – depends on many factors.

Number of Students: 2 or 3

42. Project : Ranking Results by Time and Topic

Proposer: *Jivashi Nagar*

Co-supervisor/External advisor: *Hussein Suleman*

Abbreviation: timerank

Brief Description: Search engines have become an important tool in our daily lives.

Users use them to find information on any topic. Users rely so much on the search engines that only after submitting few keyword(s) as query, they expect the search engine to disambiguate the query and retrieve the relevant documents from its huge dataset. This is not always possible, unless the search engines use implicit or explicit information.

The aim of this study is to investigate the use of time-sensitive topics as a form of auxiliary information that can supplement queries. Users would supply their queries as a point in time, and the most popular topics searched for at that time would be used as additional implicit information to guide the search engine. The search engine would then need to rank documents differently, based on the additional implicit information.

The project will include aspects to: develop a test collection of queries and documents; analyse collections of queries to determine topics; develop a ranking algorithm that includes context; and perform experiments to test the effectiveness of the algorithms.

Computer Science Content: Information Retrieval (IR), Machine Learning

Specific Learning Outcomes: Experimental research, software development, IR algorithm

Skills Required by Team as a Whole: Programming in Java and/or Python.

Theory: As above.

Implementation: Web application, IR algorithms and machine learning. No prior background in machine learning and IR is required - these skills will be learnt in the project.

Data: Datasets will be provided by collaborators or will be harvested from the Web. General background reading material will also be provided for each component.

Facilities needed: Standard computing facilities. Open source tools and libraries will be used.

Supervision: Project will be located within the Digital Libraries Laboratory (dl.cs.uct.ac.za), with regular meetings and presentations with supervisors and group. Primary supervision will be by Jivashi Nagar, a PhD student.

Bursaries: Possibility for NRF funding – depends on many factors.

Number of Students: 2