جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Course Syllabus: Application of AI in Bioinfomatics - CS 321

Division	Computer, Electrical and Mathematical Sciences & Engineering
Course Number	CS 321
Course Title	Application of AI in Bioinfomatics
Academic Semester	Spring
Academic Year	2017/2018
Semester Start Date	01/28/2018
Semester End Date	05/24/2018
Class Schedule (Days & Time)	09:00 AM - 12:00 PM | Thu

Instructor(s)

Name	Email	Phone	Office Location	Office Hours
Vladimir Bajic	vladimir.bajic@kaust.edu.sa	+966128082386	4219, 3, Ibn Sina (bldg. 3)	Wednesday 10.00 AM - 12:00 PM, Building 3, Floor 4, Room 4219 Phone: +966 (0)12 8087318

Teaching Assistant(s)

Name	Email
Dr Adil Salhi	adil.salhi@kaust.edu.sa

Course Information

Comprehensive Course Description	**Summary** Course consists of selected projects. The projects may change each year. These projects cover application of artificial intelligence (AI) to some of the relevant problems of analysis of large biological data and generally deal with complex information. Each year, the targeted problems change. Students get assigned one project and they work either alone or in groups of two. Students, in the interactive discussions with the whole class and the instructor, solve the project problems. Students regularly present their progress and defend their approach and results in front of the whole class. During one semester, several types of topics are dealt with (e.g., data integration; knowledge-, text- and data-mining of big biomedical data). Students get direct experience in research methodology, report writing, presentations, and, most importantly, different ways of approaching solving AI applications for different bioinformatics problems. **Projects for 2018 year** **Topic: Applications of word2vec type methods for molecular function prediction** Data in many types of problems are described by very large number of descriptors, so called features. This causes the problem in the downstream analysis of the data. There are many ways how to compress information that encodes the data. One of latest approaches is based on the word2vec and similar types of methods. These methods are capable to compress information coming from several thousands of several tens of thousands of descriptors into several hundred new descriptors captured in the new feature vectors that describe original data items. This allows for significantly more efficient analysis of the original data that can be subjected further to various machine learning and AI processing. **Project 1. Apply above to analysis of function of different transcripts.** **Project 2. Apply above to the analysis of different data describing cancers.** **Project 3: Relationship/Event Extraction/Modeling** When analyzing text, it is much easier to assert that two concepts are associated (based on their co-occurrence frequencies), than to assert what type of relationship/event they are participating in. This is because the former is mainly based on named entity recognition (NER), however, the latter requires a deeper type of analysis. Even if relationship terms are identified within the text, it is much more challenging to assert which concepts they involve. e.g., consider the following sentence: “A lexer is generally combined with a parser, which together analyze the syntax of text.” One can extract: [lexer] <- is generally combined with -> [parser], but how do these relate to [the syntax of text], the relationship is [analyze]. This needs the model to figure out that “which together” refers to [lexer] and [parser]. This is a very simple example, and sentences can get much more complicated than this. To complicate things further, two concepts can co-occur multiple times, in different contexts, describing different relationships. These can be used collectively to build some type of model for the association. The students may feel free to explore different aspects of this challenging problem. **Project 4. Data Structures for indexing text.** Extracting relational information from text results in substantial data sets (for each n concepts [n x (n-1) / 2] potential relationships could be extracted, so for 1,000 concepts 999 x 500 ~ 500,000 potential associations). If these associations are saved into a repository, the indexing process becomes IO bound (the process spends most of its time writing to disk), and fetching results for queries against this substantial set is affected by the size of the index. Another way to extract relations is to save only the concepts index (no pairing), then serve pairing queries based on the concepts index. This optimizes the indexing process, but pairing queries become JOIN dependent, and consequently, potentially still expensive. Depending on the query and the size of the data set, and especially if the pairing involves more than one layer (leading to nested JOINs e.g., diseases associated to genes involved in pathways) this type of querying becomes almost prohibitive. This project aims at building efficient data structures for storing the created indexes in a manner that allows near to real-time query response. Using hashtables (c implementation) or dictionaries (python) one can build very fast indexes that can serve queries from memory. A server process can be used to load these indexes into memory, and serve queries against them (e.g., using named pipes). Students are free to explore other approaches, such as a Spark-based index for example, but they hopefully would be able to show significant performance increase, compared to a conventional relational database such mySQL or PostgresSQL. **Project 5. Topic Specific Knowledgebase.** This is a data integration project, in which students are tasked with creating a comprehensive source of information (in the form of a database repository, preferably with a web-interface) regarding a particular topic (e.g., a disease such as Alzheimer's Disease). The repository should be based on core information extracted from text which we will provide in the form of pre-computed indexes from PubMed/PMC relevant to the chosen topic, but the students should use other sources of structured data to complement the text-mining. The students must first identify which complementary data is relevant and important to the topic, but is potentially missing through text-mining alone, then create a schema for the imported data consistent with provided indexes. The importing and integration of data must be automated as much as possible.
Course Description from Program Guide	These projects cover application of AI to some of the relevant problems of analysis of large biological data and generally deal with complex information. Each year problems change. Students get assigned one (1) project and they work either alone or in groups of 2. Students in the interactive discussions with the whole class and the instructor solve the project problems. Students regularly present their progress and defend their approach and results in front of the whole class. During one (1) semester several types of topics are dealt with. Students get direct experience in research methodology, report writing, presentations and, most importantly, different ways of approaching solving AI problems

Goals and Objectives	Course consists of selected projects. The projects may change each year. These projects cover application of artificial intelligence (AI) to some of the relevant problems of analysis of large biological data and generally deal with complex information. Each year, the targeted problems change. Students get assigned one project and they work either alone or in groups of two. Students, in the interactive discussions with the whole class and the instructor, solve the project problems. Students regularly present their progress and defend their approach and results in front of the whole class. During one semester, several types of topics are dealt with (e.g., data integration; knowledge-, text- and data-mining of big biomedical data). Students get direct experience in research methodology, report writing, presentations, and, most importantly, different ways of approaching solving AI applications for different bioinformatics problems.
Required Knowledge	C/C++, Java, Python, HPC (parallel computing) programming experience
Reference Texts	1. **Entity linking (entity normalization/disambiguation)** https://en.wikipedia.org/wiki/Entity_linking 2. **Information extraction** https://en.wikipedia.org/wiki/Information_extraction 3. **Controlled Vocabulary/Dictionary https://en.wikipedia.org/wiki/Controlled_vocabulary** 4. **Named Entity Recognition https://en.wikipedia.org/wiki/Named-entity_recognition** 5. **Relation Extraction http://stanford.edu/class/cs124/lec/rel.pdf** 6. **NCBI Text Mining Tools (including tmVar)** **https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/** 7. **Approximate matching** **https://courses.cs.washington.edu/courses/cse427/16au/slides/approximate_matching.pdf** 8. **Word2Vec http://www-personal.umich.edu/~ronxin/pdf/w2vexp.pdf** 9. **HMMs (Hidden Markov models) https://en.wikipedia.org/wiki/Hidden_Markov_model** 10. **Chemical named entities recognition: a review on approaches and applications** **https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4022577/**
Method of evaluation	**10.00**% - Written report **50.00**% - Research Project **20.00**% - Presentation **10.00**% - Oral presentation **10.00**% - Attendance and Participation
Nature of the assignments	Research project as defiend in the course description complemented by presentation of results, discussions on the methods of solution, written mid-term and final report.
Course Policies	Student absence of more than three times without justifiable reason will lead to failing the course.
Additional Information	Assessment of students is continuous.

Tentative Course Schedule *(Time, topic/emphasis & resources)*		
Week	**Lectures**	**Topic**
1	Thu 02/01/2018	Introduction
2	Thu 02/08/2018	Students get assigned to the projects. Projects explanations.
3	Thu 02/15/2018	Presentations of progress on individual reports and discussions with the whole class
4	Thu 02/22/2018	Presentations of progress on individual reports and discussions with the whole class
5	Thu 03/01/2018	Presentations of progress on individual reports and discussions with the whole class
6	Thu 03/08/2018	Presentations of progress on individual reports and discussions with the whole class
7	Thu 03/15/2018	Presentations of progress on individual reports and discussions with the whole class
8	Thu 03/22/2018	Presentations of progress on individual reports and discussions with the whole class
9	Thu 03/29/2018	Mid-term report
10	Thu 04/05/2018	Presentations of progress on individual reports and discussions with the whole class
11	Thu 04/12/2018	Presentations of progress on individual reports and discussions with the whole class
12	Thu 04/19/2018	Presentations of progress on individual reports and discussions with the whole class
13	Thu 04/26/2018	Presentations of progress on individual reports and discussions with the whole class
14	Thu 05/03/2018	Presentations of progress on individual reports and discussions with the whole class
15	Thu 05/10/2018	Presentations of progress on individual reports and discussions with the whole class
16	Thu 05/17/2018	Presentations of progress on individual reports and discussions with the whole class
17	Thu 05/24/2018	Final report
18		

Note

The instructor reserves the right to make changes to this syllabus as necessary.