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	2016/2017
Semester Start Date	01/22/2017
Semester End Date	05/18/2017
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. 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 this year

Normalization through approximate matching

Named Entity Recognition (NER) is an essential step in many text-mining pipelines. In a nutshell, it represents the process of identifying concepts of interest in the text, in order to index them against the source for further processing (further processing includes various information extraction tasks such as: concept enrichment, concept pairing for building biological networks, relationship extraction, etc). There are many approaches to NER, including the use of machine learning, (e.g. tmVar for the prediction of mutation mentions), as well as the use of pre-compiled dictionaries (e.g. using EntrezGene nomenclature to locate gene mentions in the text). Compared to prevalent machine learning models for NER, the dictionary based approach is more practical and generic (because one needs specialized machine-learning models for each type of concept e.g. a model for identifying chemicals would be very different from those for disease identification: the features of interest are different, therefore the data-modeling might be different, the algorithms used, etc). However, the dictionary-based approach also suffers from its lack of flexibility, (because terms are usually mentioned with a slight difference to the pre-compiled nomenclature, the nomenclature is not always exhaustive, and terms can sometimes be simply misspelled), which significantly affects recall. Creating custom rules to allow more flexibility (e.g ignoring case, white space, transforming punctuation, etc) can enhance recall slightly, but going any further than that can be cumbersome, and can sometimes lead to adverse effects (false positives). Approximate matching can be computationally expensive for large text, however, we propose using Word2Vec (with lexical- instead of semantic-embedding) for approximate match retrieval (other types of lexical hashing can be used if the students wish to). This hybrid approach takes advantage of available pre-compiled dictionaries, but overcomes the recall problem mentioned above, and most importantly can be used to process large amounts of text efficiently.

Chemical name identification – (Bonus: can be generalized to general entity recognition)

Chemicals are of significant importance in biology, and they represent a very special type of textual entity, because they look more 'different' than most other biological context, especially in their long name form:

1. › "10,11-Dihydro-10,11-dihydroxy-5H-dibenzazepine-5-carboxamide"
2. › "10,11-Dihydro-10-oxo-5H-dibenz(b,f)azepine-5-carboxamide"
3. › "10,11-Dihydro-5-(3-(methylamino)propyl)-5H-dibenz(b,f)azepine hydrochloride"
4. › "10,11-Dihydro-5-(3-(methylamino)propyl)-5H-dibenz(b,f)azepine monohydrochloride"
5. › "10,11-dihydro-5-(gamma-dimethylaminopropylidene)-5H-dibenzo(a,d)cycloheptene"
6. › "10,11-dihydro-N-methyl-5H-dibenzo[a,d]cycloheptene-Delta(5,gamma)-propylamine"

They can be punctuation rich, they usually have numbers embedded in the name, but most importantly they share a lot of basic building units in a manner that actual chemicals share constituent molecules. This should make them easy to identify in text without relying on neighboring context because they have very distinct features in themselves. (so why not just use a dictionary-based approach: because the slightest deviation form the dictionary name, such as replacing a bracket with a square bracket, or a hyphen with an underscore, results in a false negative). Variations of HMM models, which can be though of as state transition machines with probabilities assigned to state transitions, are usually used for the identification and normalization of chemicals.

Considering the example chemical list above, “10,11-[D/d]ihydro-” represents the entry state for all of them (with P=1) then 2 out of them (P=2/6 = 0.33) move to “5-(3-(methylamino)propyl)-5H-dibenz(b,f)azepine”, etc. Note that the building blocks themselves can have their own state transitions (e.g. considering more examples “methylamino” could be tokenized into “methyl->amin->o with probabilities assigned to each transtion). The probabilities are based on the positional frequencies of the building tokens. The project aims at developing an alternative to HMMs for chemicals NER, which is based on the term morphology (i.e does not necessarily need neighboring context). It should be able to identify the longest phrase referring to the chemical (e.g. if “dihydro N-methyl” has a space in it how do you identify the delimiters of the chemical entity). The methodology be generic enough to be used for other types of entities which do not necessarily have as distinct a feature set.

Relationship/Event Extraction/Modeling

When analyzing text, it is much easier to assert that two concepts are associated (based on their co-occurence frequencies), than to assert what type of relationship/event they are participating in. This is because the former is mainly based on 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], but 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. Can these 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. **Data Structures for indexing text.** Extracting relational information from text results in substantial data sets (for each n concepts n.(n-1)/2 potential relationships, so for 1k concepts 999*500 ~ 0.5M 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 need to show significant performance increase, compared to a conventional relational database such mySQL or PostgresSQL. **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 a 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. 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. Students get assigned one project and they work either alone or in groups of two/three. Students in the interactive discussions with the whole class and the instructors solve the project problems. Students regularly present their progress and defend their approach and results in front of the whole class. 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, 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**% - Active participation **20.00**% - Written report **50.00**% - Research Project **10.00**% - Presentation **10.00**% - Oral presentation

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 3 times without justifiable reason will lead to failing the course.
Additional Information	

Tentative Course Schedule

(Time, topic/emphasis & resources)

Week	Lectures	Topic
1	Thu 01/26/2017	Introduction
2	Thu 02/02/2017	Students get assigned to the projects. Projects explanations.
3	Thu 02/09/2017	Presentations of progress on individual reports and discussions with the whole class
4	Thu 02/16/2017	Presentations of progress on individual reports and discussions with the whole class
5	Thu 02/23/2017	Presentations of progress on individual reports and discussions with the whole class
6	Thu 03/02/2017	Presentations of progress on individual reports and discussions with the whole class
7	Thu 03/09/2017	Presentations of progress on individual reports and discussions with the whole class
8	Thu 03/16/2017	Presentations of progress on individual reports and discussions with the whole class
9	Thu 03/23/2017	Mid-term report
10	Thu 03/30/2017	Presentations of progress on individual reports and discussions with the whole class
11	Thu 04/06/2017	Presentations of progress on individual reports and discussions with the whole class
12	Thu 04/13/2017	Presentations of progress on individual reports and discussions with the whole class
13	Thu 04/20/2017	Presentations of progress on individual reports and discussions with the whole class
14	Thu 04/27/2017	Presentations of progress on individual reports and discussions with the whole class
15	Thu 05/04/2017	Presentations of progress on individual reports and discussions with the whole class
16	Thu 05/11/2017	Presentations of progress on individual reports and discussions with the whole class
17	Thu 05/18/2017	Final report
18		

Note

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