# Assignment 5 – CIR

**Submission Date: November 13, Monday** *Report(by 10pm)*
*Demo(During week starting November 13. Schedule to be posted near to the time)*

**Type: Team(2 Pairs) work** **Weightage: 30%**

## I. Introduction

The purpose of the assignment is to:

a) Practice iterative development by building on Assignments 3 and 4

b) Integrate different pieces, worked on by different pairs of the team, to design and build a functional software.

c) Practice Requirements elicitation, analysis, and specification.

d) Design and construct a functional application in the domain of text processing with data analysis and visualization focus.

Please contact us at cs3219.cir@gmail.com in case you have any query about requirements of this assignment.

## II. Your Task

Simon, a Research Engineer at an NLP Research Lab, wants your team to implement an information retrieval application(web-based or desktop utility) which helps him to query and visualize conference publication data. **Your task is to analyze requirements, design and implement it.** Following is a set of requirements provided by Simon (R*equirements from the user are expected to be ambiguous, incomplete or inconsistent. You could clarify by writing a mail to Simon at* cs3219.cir@gmail.com *. You could make reasonable assumptions, and identify constraints, based on your earlier work in Assignments 3 and 4, while taking decisions in specifying requirements, designing, and building the application.)*

1. The overall aim of CIR is to provide Simon with a usable interface which accepts his intuitive queries and provides him with outputs/visualizations to help him analyze various aspects of research publications. In particular, Simon is interested in trend-based queries.

   An example of a trend-based query is:

   "For the conference D$ give number of cited documents published in each of the years 2000 to 201X."

   E.g., if we chose $ to be 12 and 13 and represent them on the common scale, we are effectively analyzing how much the conference D in (13) compared to previous year (12) has changed in terms of focus on the research happened in recent times (since the year 2000). Here, a conference citing recent papers is considered more up to date if it cites more papers from recent years. Hence, if we find D13 citing more recent papers as compared to D12, we can conclude that D13 is more focused on recent research. In fact, this can be done for any two years E.g., D12 vs. D02, or in an incremental way for D9 to D10 to…. so on and so forth. It really boils down to using the previously built modular components and mix and match to generate insights.

2. As a minimum viable product, CIR should provide for
   (a) **At least 3 trend-based queries and respective visualizations**. Here are some examples of trend based queries:

   **Trend 1:Transition over time**

   X<year_1>, X<year_2>, ….., X<year_n> i.e. trend of transition over time for a particular conference X.

   Required: For the conference X<year_i> vs X<year_j> (or many year_j s) generate insights for number of cited documents published in each of the years <custom_citation_start> to <custom_citation_end>.

   Required: Repeat the above step for conferences <custom_list_of_confrences> (instead of years) for the conference X<year_i> vs X<year_j> (or many year_j s).

   **Trend 2:Contemporary comparison**

   A<year>, B<year>, ….., Z<year> i.e. compare different conferences over same time <year>.

   Required: For the conference A<year> vs X<year> (or for many years) generate insights for number of cited documents published in each of the years <custom_citation_start> to <custom_citation_end>.

   Required: Repeat the above step for conferences <custom_list_of_conferences> (instead of years) for the conference A<year> vs X<year> (or conferences Xs).

   **Trend 3:Top N X of Y**

   Where N is a number; X can be authors, conference, citations, venue, booktitle, base papers, etc., and; Y can be conference, author, cited authors, citing authors, etc.(wherever attributes makes sense) you can pick a few of these types or implement for abstract X and Y.

   Example: Top 3 authors for A12

   Example: Top 5 citations for B.

   Example: Top 10 citations for Joshua Bengio.

   (b) **In addition, any 2 other queries of your choice and corresponding visualizations** involving analysis of citations e.g. a citation network.

   (c) **<u>Overall , 3 or more different types</u> of visualizations to achieve all the tasks given in a) and b) above**.

   (d) Querying of **at least one of the data sets** used in Assignments 3 or 4.

   (e) Design of the application that clearly identifies **at least one non-functional requirement** and incorporates it. Non-functional requirements could include for example, usability, scalability(e.g. providing for multiple data sets, handling issues of

big datasets and multiple users by using cloud storage and services), uniqueness(e.g. in terms of visualizations or queries/trends, adding a chatbot feature), robust or scalable architecture, maintainability(e.g. through excellent design /code quality)

3. ***Challenge yourselves***

Bonus marks(upto 5 marks) could be awarded to teams for providing any one or more of the following:

(a) Application design convincingly offers 2 or more non-functional requirements.

(b) Think of providing a service, for example, a widget which can be plugged on a web page, e.g., on ACL anthology EMNLP conference W96 page http://www.aclweb.org/anthology/W/W96/#0200/, to visualize trends of that conference.

*Write to us or meet us if you are accepting the this challenge to clarify on the requirements.*

## III. Submission Guidelines

1. What to submit: TeamNumber_Report_CIR.docx (or pdf). A report template is given at the end of this document. Report is expected to be about 5-15 pages. Exceeding the page guideline of 5-15 pages does not invite any penalty.

2. Demonstration:

When: **November 14-17, 2017.** Timeslots for demo will be notified near to the week.

What to present:

i. Design

ii. Demo of working product ie integrated CIR - querying and visualization components for the tasks described in section II above.

## IV. Important Information

a) Read this document carefully.
b) Use any tool or language to implement the requirements.
**c)** If you have any query about this Assignment, send email to **cs3219.cir@gmail.com**
**d)** If you are using Assignment 4 dataset, you can use first 200,000 lines of (full) dataset http://labs.semanticscholar.org/corpus/ for this assignment as well.
**e)** In case you are unable to provide functionality for the listed queries and visualization(s), write your reasons in the report and attempt to replace it with an alternative meaningful query and visualization.
f) In addition to requirements listed in this document, you are encouraged to offer more visualizations, more kinds of queries, summarizing information or visualization as part of the user interaction. As development team, you could follow a specific development process, use a specific design notation, a specific architecture, design pattern(s), use tools for your development and testing, or use any specific services, e.g., AWS services, a coding language which is not usual or worked with before. We would like to acknowledge and give credit any of your extra work. You should include a mention of your additional effort in Report(add a separate section) and highlight during the demo.

g) Here is an inspirational work in the domain of academic literature search http://academic.research.microsoft.com/

7. **Marking guide:**

*CS3219 is about software engineering, and design principles and patterns. Report and code will, therefore, be assessed for clear requirements specification, design documentation (including diagrams and decisions) and implementation (including code, output, and demonstration of functionality).*

*The Report will be marked for clarity, correctness, and completeness.*

*While marking demo and report, evaluators will also take note of development and documentation practices followed, for example, requirements specification, design notation, the usability of accepting input, the layout of output, error handling, etc.*

*These practices are taught in various pre-requisite modules as well as in CS3219 and are expected from year 2/3 of CS students. You can select relevant subtopics from lectures on development process, requirements specification, architecture design and design patterns and use the concepts covered in your work.*

*Marking will, however, accommodate and acknowledge creativity and different approaches taken by teams.*

Here is a mark weightage for various components.

Report documentation & SE practices – 15-18 marks

Data extraction and Visualization functionality (code & demo)– 12-15 marks

Challenge functionality (code & demo) – upto 5 marks

**--- Report Template ---**

(Report should be written for technical developers. Write with a perspective such that your peers could easily use your documentation to create a similar product)

**Cover page** stating name of the product (Choose a suitable name or use CIR.)

Link of your GIT repo (keep it private while developing the software & make it public on the day of submission).

Link **if** you are providing a web-based service.

| | | | | |
|---|---|---|---|---|
| **Student Name** | | | | |
| **Matriculation Number** | | | | |

**Introduction**

1-2 paragraph(s) including objective of assignment in your own words; individual contribution of each member in doing this assignment )

**Requirement Specification:**

Provide a **requirement specification** for the application you designed and built. Prefer to give it in a tabular format. It should be based on your team's understanding and analysis, unambiguous, complete, categorized (Refer concepts covered in lecture on Requirements specification e.g. functional and non functional requirements, data, design, implementation requirements, constraints etc. etc.), prioritized, labelled, identified for development iterations etc.

**Design and Implementation**

Provide architecture/component level design (diagrams + descriptions), API specification of components, your decisions, and implementation specific technical information (E.g. coding language, framework(s), development tools) of your product. For API specification you can follow a tabular format providing component wise API and its client components OR any other format e.g. CRC format (see lecture slides on architecture) and provide API in place of responsibilities.

Provide well-labelled screenshot(s) of visualizations for requirements listed in 2(a) and 2(b) in Section II above.

For any one of the visualizations :

- Explain its Purpose including what does the visualization show.
- List Step wise method you followed in creating the visualization. Be precise and succinct.

**Additional Section (optional)**
Add a section on any additional feature or challenge you take to design and implement. There is no fixed format or strict requirements for this section. Use text or diagrams to explain. Keep it simple and concise.
Add any other comments or information you may have.

---