1 Objective

In this assignment, you will 1) crawl a text corpus of your interest, 2) build a search engine to query over the corpus, and 3) perform text classification and clustering. Your final score will depend not only on how you developed your system but also on its novelty and your creativity: in other words, to get a high score you do not only need to implement a system that works, but also a system that is useful and user-friendly.

If you are not sure whether your idea for the assignment is good, you can discuss it with the lecturers. For hands-on help (e.g., installation of programs or suggestions on which library or IDE to use), please contact either Sandro Cavallari (sandro001@e.ntu.edu.sg) or Haiyun Peng (peng0065@e.ntu.edu.sg).

2 Deadline

The assignment constitutes about 35% of your total grade of the course. Assignments are to be submitted via Blackboard (do not email your submission). You can send a progress report (not compulsory) by 7th March to get an initial feedback (not graded) about your work. After recess week, there will be one more dedicated session (attendance is not compulsory) where you will get the chance to present and discuss your current progress and get a feedback (not graded) about it.

Deadline	11 April 2018 (Wednesday), 11:59pm SGT
Late submissions	5% deducted for each rounded-off day after the deadline

3 Group

The assignment shall be done in groups (to be signed up via Blackboard). Each group may have up to five students and minimum two. Please note that the bigger the group the more we will expect from you (in terms of magnitude of the developed system, content, analysis, etc.). If you like, you can specify who did what in your final report and each member will be graded accordingly. If this information is not specified, a unique grade will be given to the whole project and this will be shared among all members of the group.

Some overlap between projects from different groups is allowed but beware that, if we find out that a project has more than 30% overlap with another project from this year or past years, your group will be disqualified (and get zero points as final grade for the assignment). Hence, it is OK to share the general idea of your assignment with other groups but not implementation details. Solo assignments are not allowed: if you really cannot find a group, contact Danyuan Ho (dyho@ntu.edu.sg) for a mini-project on knowledge engineering for Singlish sentiment analysis. If you are not a Singlish speaker, you can contact Soujanya Poria (sporia@ntu.edu.sg) for a similar project in English.

4 Details (100 points)

4.1 Crawling (20 points)

Crawl text data from any sources which you are interested in and permitted to access, e.g., Twitter API, Facebook API, or Amazon API. The crawled corpus should have at least 10,000 records and at least 100,000 words. Before crawling any data, carefully consider the questions in this material, e.g., check whether the data have enough details to answer the questions. You can use any third party libraries for the crawling task. If you find it difficult to crawl such big data, you can approach the lecturers for help, but skipping the crawling task might result in some penalty.

Question 1: Explain and provide the following:

1. How you crawled the corpus (e.g., source, keywords, API, library) and stored them (e.g., whether a record corresponds to a file or a line, meta information like publication date, author name, record ID)
2. What kind of information users might like to retrieve from your crawled corpus (i.e., applications), with example queries
3. The numbers of records, words, and types (i.e., unique words) in the corpus

4.2 Indexing and querying (40 points)

Indexing: Solr+Lucene+Jetty in Java are recommended to index the crawled data. Solr is written in Java and runs as a standalone full-text search server within a servlet container such as Jetty. Solr uses the Lucene Java search library at its core for full-text indexing and search, and has REST-like HTTP/XML and JSON APIs that make it easy to use from virtually any programming language. Solr's powerful external configuration allows it to be tailored to almost any type of application without Java coding. Useful documentations include:

- Solr project: http://lucene.apache.org/solr
- Solr wiki: http://wiki.apache.org/solr/FrontPage
- Lucene tutorial: http://lucene.apache.org/core/quickstart.html
- Solr with Jetty: http://wiki.apache.org/solr/SolrJetty
- Jetty tutorial: http://wiki.eclipse.org/Jetty/Tutorial

You can also choose other inverted-index text search engine open projects, e.g., Sphinx, Nutch, and Lemur. However, you should NOT simply adopt SQL-based solutions for text search (for example, you CANNOT solve text search simply using Microsoft Sqlserver, MySql, and the likes of them).

Querying: You need to provide a simple but friendly User-Interface (UI) for querying. It could be either a web-based or mobile app based UI. You could use JSP in JAVA or Django in Python to develop your UI website. Since Solr provides REST-like APIs to access indexes, one extra JSon or RESTFul library would be enough. Otherwise, you may use any third party library.

The UI must be kept simple. A too complex UI is not necessary nor encouraged. Detailed information besides text is allowed to be shown for the query results, e.g., product images on Amazon, ratings on Amazon, and pictures on Flickr. The details should be designed to solve specific problems.

Question 2: Perform the following tasks:

- Build a simple Web interface for the search engine (e.g., Google)
- A simple UI for crawling and incremental indexing of new data would be a bonus (but not compulsory)
- Write five queries, get their results, and measure the speed of the querying

Question 3: Explore some innovations for enhancing the indexing and ranking. Explain why they are important to solve specific problems, illustrated with examples. Possible innovations include (but are not limited to) the following:

- Interactive search (e.g., refine search results based on users' relevance feedback)
- Improve search results by integrating machine learning or data mining techniques (e.g., classification or cluster techniques)
- Go beyond text-based search (e.g., implement image retrieval or multimedia retrieval)
- Exploit geo-spatial data (i.e., map information) to refine query results/improve presentation/visualization
- Others (brainstorm with your group members!!)

4.3 Classification (40 points)

Define a set of categories (minimum three) the collected data could belong to and perform automatic classification on them. For example, you could crawl news articles and classify them into specific topics, e.g., sport, finance, and politics (auto-categorization). Another option is to classify data into positive, negative or neutral (sentiment analysis). Different classification approaches can be applied, including:

- knowledge based, e.g., semantic networks and ontologies
- rule based, e.g., linguistic patterns and POS tagging
- machine learning based, e.g., SVM and ANN
- hybrid

You can tap into any resource or toolkit you like, as long as you motivate your choices and you are able to critically analyze obtained results. Some possible choices include:

- Weka: http://cs.waikato.ac.nz/ml/weka
- Hadoop: http://hadoop.apache.org
- Pylearn2: http://deeplearning.net/software/pylearn2
- SciKit: http://scikit-learn.org
- NLTK: http://nltk.org
- DBPedia: http://dbpedia.org
- Theano: http://github.com/Theano
- Keras: http://github.com/fchollet/keras
- Tensorflow: http://github.com/tensorflow/tensorflow
- WordNet: http://wordnet.princeton.edu
- SenticNet: http://sentic.net

Question 4: Perform the following tasks:

- Motivate the choice of your classification approach in relation with the state of the art
- Discuss whether you had to preprocess data and why
- Build an evaluation dataset by manually labeling 10% of the collected data (at least 1,000 records) with an inter-annotator agreement of at least 80%
- Provide evaluation metrics such as precision, recall, and F-measure and discuss results
- Discuss performance metrics, e.g., records classified per second, and scalability of the system
- A simple UI for visualizing classified data would be a bonus (but not compulsory)

Question 5: Explore some innovations for enhancing classification. Explain why they are important to solve specific problems, illustrated with examples. Possible innovations include (but are not limited to) the following:

- Ensemble classification (e.g., leverage on multiple classification approaches)
- Cognitive classification (e.g., use brain-inspired algorithms)
- Multi-faceted classification (e.g., take into account multiple aspects of data)

5 Submission

Submission has to be done via Blackboard (do not email your report). The submission shall consist of one single PDF file containing the following five key items:

1. The names of the group members
2. Your answers for all the above questions
3. A YouTube link to a video presentation of up to 5 minutes: in the video, introduce your group members and their roles, explain the applications and the impact of your work and highlight, if any, the creative parts of your work (note that you do not have to give all the answers in the video presentation)
4. A Dropbox (or similar) link to a compressed (e.g., zip) file with crawled text data, queries and their results, manual classifications, automatic classification results, and any other data for Questions 3 and 5
5. A Dropbox (or similar) link to a compressed (e.g., zip) file with all your source codes and libraries, with a readme file that explains how to compile and run the source codes