

# Knowledge Representation and Reasoning

**Topic:** knowledge representation in AI and Natural Language processing

**Assignment marked out of:** 100% **Weight:**

40% of overall grade

**Number of Words:** 5000-7000

**Deliverables:**

- Soft copy via your account at Blackboard. Only one student submits if it is a group project
  - Share a folder on either Google Drive or Dropbox, including report, presentation and five significant papers. The folder name should be in the following format:
    - Individual: *Student ID-Student First Name*
    - Group: *Group-Student ID-Student First Name, Student ID-Student First Name*
- 

## 1. Overview

In this project assignment, you will demonstrate your understanding of knowledge representation in a Webbased practical AI programming project. **Choose only one of the following assignments. It might be a group assignment with maximum of two students; a division of work must be clarified in the report.** You might use a Java servlet (preferably using NetBeans 7 with Java EE and TomCat “can be downloaded for free from <http://netbeans.org/downloads/>”) for making your assignment we-based.

**Assignment 1:** *Statistical parsing of Arabic for web user interface. [1] -level basic*

**Assignment 2:** *Preprocessing of Arabic text: tokenization & POS tagging. [2] – level basic*

**Assignment 3:** *Developing an Arabic Named Entity Recognition System. [3, 10, 11]-level basic*

**Assignment 4:** *Processing Arabic Questions using open Source tools [5][6]-level intermediate-to-advanced*

**Assignment 5:** *Implementing Recommender systems using data mining and knowledge discovery tools. [7]-level intermediate-to-advanced*

**Assignment 6:** *Automatic document summarization. [8]- level intermediate-to-advanced*

**Assignment 7:** *Using text genres analysis to verify financial institution annual reports against authoritative guidance-level intermediate-to-advanced*

**Assignment 8:** *Analysis of Modality in Text Utterances-level basic*

**Assignment 9:** *Deep Learning for Arabic Natural Language Processing [12]-level intermediate-toadvanced*

<p><b>Proposed Assignment:</b> You could also propose your own project that is related to the Knowledge Representation and Reasoning module, but it needs approval from me. Your proposal should follow that same format used in this assignment brief.</p>
---

You will present your project and demonstrate your work in a lab session. You will also hand in a report, which includes the following:

1. An introduction with a general description of the problem domain, and the aspects you focus on.
2. A description of your solution, including a description of the algorithm you defined, any clever ideas you came up with or borrowed, and so on.
3. A discussion of the performance of the system, the problems encountered, error analysis, etc.
4. Conclusion, including suggestions for future enhancements.

The report should be in PDF format and between 12 and 22 pages in size, excluding references. You are required to use ACL (2012) style (available for LaTeX and Word) in producing the PDF document. These templates are available at: <http://acl2012.org/call/sub01.asp>

[You might use https://www.sharelatex.com/](https://www.sharelatex.com/) for Latex documents

## 2. Requirements

Students are expected to implement a java web application (preferably using NetBeans 7 with Java EE and TomCat “can be downloaded for free from <http://netbeans.org/downloads/>”), if possible, responsible for complex knowledge representation in AI application.

Pls. see the **Marking Scheme Section** which will give you an idea about the criteria for marking and their weights.

*Note that the code will need to be extended and revised by other developers, so make sure to include full and clear comments and documentation.*

### Assignment Milestones

#### **Milestone 1: Preparation**

1. You have acquired all the required training and testing data.
2. You have installed the necessary software (Netbeans, Weka, Bikel Parser, among others)
3. You have run the application on a small sample of data, or created a small “Hello world” application.

#### **Milestone 2: Development**

1. You have developed the application with all functions and features.
2. All various components, functions, features and classes are integrated together in one single application.
3. The program accepts all instances of the training data as input and gives the expected output.

#### **Milestone 3: Testing and Evaluation Deadline:**

1. Gold standard is created or acquired.
2. Continuous cycle of testing-development-testing until satisfactory results are gained. Error analysis of results achieved will guide you to the points of improvements. You can refer to your trial and the mythology you followed.
3. Testing results in terms of standard evaluation metrics are reported with error analysis. Try to compare with state of the art research.

### **2.1 Assignment 1: Statistical parsing of Arabic for web user interface**

This task consists of training a statistical parser for Arabic and porting it on a web interface allowing it to accept user input and provide parse results.

1. **Training a statistical parser for Arabic:** Use the Bikel parser which is already tuned for Arabic. The parser can be downloaded from <http://www.cis.upenn.edu/~dbikel/software.html> and the (Arabic Treebank) training data from Software and Resources folder. You might use/compare with another parser (e.g. Stanford Arabic Parser, <http://nlp.stanford.edu/software/lex-parser.shtml>)
2. **Port the parser to the web using a java servlet:** From a web server (using NetBeans and TomCat) you should be able to send input to the parser and get output from it to be displayed back in the server.
3. **Take user input and give parse output:** user input is a free Arabic script text not tokenized, transliterated or formatted in any way. See how you can format the raw text to get a successful parse from the parser. Provide means for presenting the output sentence graphically.

#### 4. Test and Evaluate

### 2.2 Assignment 2: Preprocessing of Arabic text: tokenization & POS tagging

This task consists of using SVM machine learning in order to pre-process raw Arabic text and produced tokenized and part-of-speech (POS) tagged analysis. You are recommended to use WEKA or RapidMiner for this task. Refer to the paper *Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks* By Mona Diab, Kadri Hacioglu and Daniel Jurafsky, Published in *HLT-NAACL 04* or recent work by the first author. You can download the tool described in this paper from Software and Tools folder:

1. **Design classifiers for tokenization of Arabic text:** Arabic words consists of clitics that need to be separated in the tokenization task.
2. **Design classifiers for POS-tagging of Arabic text:** Each word should be assigned the right POS category
3. **Train on the Arabic Treebank:** The model will be trained on the Arabic Treebank from Software and Resources folder
4. **Test and evaluate.**

A demonstration of a similar system can be seen here: <http://nlp.ldeo.columbia.edu/amira/>

#### Tokenization Sample Input sentence

ملو بستحى الحكم المجري رودناس لوب فلكر ءازج تحيحص اثر فلقرع بكسيه لخاد المنطقة نم لبق اليساندرو اتسن  
wlm yHtsb AlHkm Almjry sAndwr bwl rklp jzA' SHyHp Avr Erqlp hyskY dAxl AlmnTqp mn qbl  
AlysAndrw nstA.

Tokenization Sample Output sentence w lm yHtsb Al Hkm Al mjry sAndwr bwl rklp jzA' SHyHp Avr Erqlp  
hyskY dAxl Al mnTqp mn qbl AlysAndrw nstA .

POS-Tagging Sample Input sentence w lm yHtsb Al Hkm Al mjry sAndwr bwl rklp jzA' SHyHp Avr Erqlp  
hyskY dAxl Al mnTqp mn qbl AlysAndrw nstA .

POS-Tagging Sample Output sentence w/CC lm/RP yHtsb/VBP AlHkm/NN Almjry/JJ  
sAndwr/NO\_FUNC bwl/NNP rklp/NN jzA'/NN SHyHp/JJ Avr/IN Erqlp/NN hyskY/NO\_FUNC dAxl/IN  
AlmnTqp/NN mn/IN qbl/NN AlysAndrw/NNP nstA/NN ./PUNC

### 2.3 Assignment 3: Developing an Arabic Named Entity Recognition System.

In this assignment, you have to build a rule-based Named Entity Recognition system (RBNER) for Arabic, which is capable of identifying one or more of ENAMEX categories (i.e. Person, Location and Organization NEs), using GATE tool [9]. A RBNER system consists basically of a set of linguistic rules (i.e. grammars) and a set of gazetteers (i.e. dictionaries/keyword lists). A linguistic rule may utilize NE Gazetteer(s) in its structure to support and implement the rule efficiently. Then, you will need to evaluate the performance of the rule-based NER system when applied on a standard dataset/corpus (i.e. *ANERcorp*<sup>1</sup> dataset). It is recommended that you have a look at the following papers: [3, 10, 11].

- ❖ **The system environment:** *GATE* platform [9], which allows you to implement linguistic rules, create/add gazetteers and evaluate the produced system.
- ❖ **The NE gazetteers:** You need to consider NE gazetteers in the structure of the new linguistic rules. An example of gazetteers to be considered is ANERGazet<sup>2</sup>

<sup>1,2</sup> Available to download on <http://www1.ccls.columbia.edu/~ybenajiba/downloads.html>

- ❖ **The linguistic rules:** The rules need to be implemented in JAPE language. Read the GATE user manual to learn about JAPE. Also, reading [3, 10, 11] might help too.

- 
- ❖ **System evaluation:** The performance of the system, when applied on ANERcorp dataset, can be evaluated using GATE built-in evaluation tool, so-called *AnnotationDiff*. The results should be in terms of precision, recall and f-measure.

---

## 2.4 Assignment 4: *Processing Arabic Questions using open Source tools*

In this assignment you will use any open source tool such as QANUS (can be downloaded from <http://www.qanus.com/download/>) or OpenEphyra (can be downloaded from <http://sourceforge.net/projects/openephyra/>) or any other tool to develop Question Answering System. Refer to the papers in reference [5] and [6] to know more about Question Answering Systems and related tasks and tools. Then, you can use some standard set of questions (both English and Arabic) from TREC(<http://www.emi.ac.ma/bouzoubaa/download/>) or CLEF (<http://www1.ccls.columbia.edu/~ybenajiba/downloads.html>).

1. **Processing of questions:** Process the English questions using the open source tool and predict the classes of the question. Processing of question involves word segmentation and POS tagging.
2. **Modification of the source code:** Modify the source code of the tool to process Arabic questions
3. **Test and evaluate:** Compare the performance of the tool for both English and Arabic question sets.

## 2.5 Assignment 5: *Implementing Recommender systems using data mining and knowledge discovery tools*

Recommender Systems are software tools and techniques providing suggestions for items to such as what items to buy, what music to listen to, or what online news to read. A recommender system normally focuses on to generate the recommendations of a specific type of item based on some recommendation technique. You can find more about recommender system in [7]. In this assignment, you are required to perform the following task for recommender system:

1. **Recommender Algorithms:** Compile and compare the at least four recommender algorithms.
2. **Data set and Tools:** Identify the data set for recommender system. You can use your Facebook/ LinkenIn/ Instagram friend list or list of books on Amazon, YouTube video lectures, online music store or any other data of your choice. Select any data mining tool useful for recommender system such as RecommendeLab, RapidMiner, KNIME, Weka .
3. **Implementation of Recommender Algorithm:** Implement the best algorithm described in Task 1 using tools of your choice from step 1. A sample implementation can be found in [[https://bib.irb.hr/datoteka/596976.rcomm2012\\_recommenders.pdf](https://bib.irb.hr/datoteka/596976.rcomm2012_recommenders.pdf)].
4. **Results:** Present the results and interesting patterns.

## 2.6 Assignment 6: *Automatic document summarization*

Document summarization is the technique of identifying and extracting important information from text documents. The output of the document summarization is usually significantly smaller than original document and is not longer than half of the original document under any circumstances. In this assignment you are required to do the following task:

1. **Summarization Algorithms:** Discuss at least three document summarization technique.

2. **Implementation of summarization algorithm:** Implement one of the document summarization techniques using Perl, Java or Python. Optionally you can use automatic summarization tool such as Mead [<http://www.summarization.com/mead/>]
3. **Results:** Rate the summarization of text produced by program/tool. Present the summarization results.

## 2.7 Assignment 7: Using text genres analysis to verify financial institution annual reports against authoritative guidance

Recently, the government of the United Arab Emirates has made progress to encourage corporates to include sustainability efforts within their annual reports. These annual reports should communicate the corporate efforts on sustainability, environment, and social issues. These reports are made public, but it can take a lot of efforts to do manual review by current and/or prospect shareholders. Automating the task of verifying annual reports against authoritative guidance issued by the government can help investors task corporate efforts on sustainability. An example government guidance is given here:

<https://www.google.com/url?sa=t&source=web&rct=j&url=https://adxservices.adx.ae/WebServices/DataServices/contentDownload.aspx%3Fdoc%3D1704806&ved=2ahUKEwjQ7Yem1-PmAhVgGDQIHbIgAPkQFjAIegQIBRAB&usq=AOvVaw2Oor9oMHRh9ZBTEG0CVhw>

The goal of this project would be to analyze annual reports of the top 10 corporations in Abu Dhabi Stock Exchange and through NLP and text genres analysis, students are asked to find text pieces within these 10 reports that match/confirm the authoritative guidance given in the above link. Example corporate bank:

<https://www.adcb.com/about/sstanrprt/sustainability.aspx>

Student will need to define a text bi-clustering method to align text segments of corporate annual reports on sustainability to text segments within government guidance. Ad hoc text analysis may also be applied to find text patterns in annual reports. For example, students can run regular expressions rules to find tokens/phrases that are over-represented/under-represented in annual reports.

## 2.8 Assignment 8: Analysis of Modality in Text Utterances

Modality is a system for enabling speakers to express intentions and beliefs through text. Modality can be manifested in different forms including modal auxiliary verbs (must, should, can ...). See this resource for introduction: [https://en.wikipedia.org/wiki/Linguistic\\_modality](https://en.wikipedia.org/wiki/Linguistic_modality)

Modality has a number of categories (epistemic, deontic, and circumstantial). Epistemic is related to knowledge of the speaker about necessity or possibility of something. Deontic is related to obligations, while circumstantial is related to context. Expression of modality can expose a great deal of unclarity regarding the underlying category. For instance, consider these uses of the modal verb 'must':

- (1) Agatha *must* be the murderer. (expressing epistemic modality)
- (2) Agatha *must* go to jail. (expressing deontic modality)
- (3) Agatha *must* sneeze. (expressing circumstantial modality)

Source: [https://en.wikipedia.org/wiki/Linguistic\\_modality](https://en.wikipedia.org/wiki/Linguistic_modality)

The goal of this project is to analyze text corpora, identify modal expressions, and then classify them to categories (e.g. epistemic, deontic ...etc.). This method to realize this goal is open-ended (you can use machine learning or develop your own rule-based system).

Please note that this project does not focus on one verb (must)! You need to cover a substantial amount of modal expressions.

Bonus: It would be great if you develop a system of analyzing modality in Arabic literature.

## 2.9 Assignment 9: Deep learning for Arabic NLP

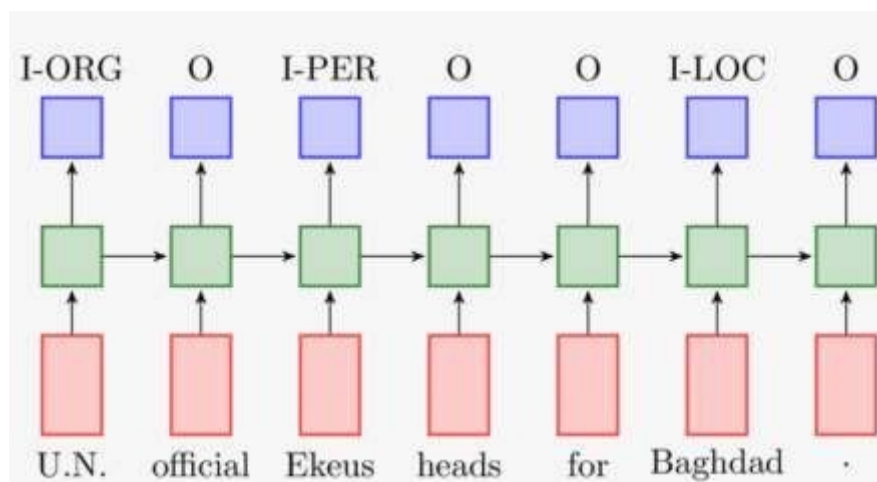
Deep learning technology such as, RNNs, GRUs, CNNs and LSTMs, are now hot topics for NLP. The focus will be on machine learning and specifically ‘deep’ neural network approaches to the automated analysis of natural language text. Topics will typically include representation learning for words (and possibly larger linguistic units). Neural networks are one of the most powerful classes of Arabic Natural language processing model, achieving state-of-the-art results on a wide range of benchmarks. A key aspect behind their success is the ability to discover representations that can capture relevant underlying structure in the training data. Any of the following topics can be used in this assignment:

- Named Entity Recognition
- Sentiment Analysis
- Machine Translation
- Language Generation and Multi-Document Summarization
- Text Classification and Categorization
- Word Sense Disambiguation using various types of Recurrent Neural Networks
- Part-of-Speech Tagging
- Semantic Parsing and Question Answering
- Paraphrase Detection
- Character Recognition
- Spell Checking
- Word Embedding
- Stemming

I am going to specify the first three tasks and you need only to choose one of them. You can do the same if you decided to choose another task.

### 2.7.1 Named Entity Recognition

Recognizing proper names in a piece of given text is very important for many tasks, including information retrieval, information extraction, summarization, and translation. In this task, you will build a deep learningbased named entity recognizer. You should use a bi-directional LSTM (check the resources below). This task is a *sequence labeling* task, where your model should label each token as either, Person, Location, Organization or Other. See the image below. For example, “Mr. Samy is on his way to Cairo”. Your model should label “Samy” as Person (PER) named entity, and label “Cairo” as Location (LOC) named entity. All other tokens should be labeled as Other (O).



**Web link to the Dataset •**  
[CONLL2003 dataset.](#)

## Web links to Resources

- [Named entity recognition with bidirectional LSTM-CNNs](#)
- [Neural architectures for named entity recognition](#)
- <https://www.sciencedirect.com/science/article/pii/S0885230818301657>
- [https://medium.com/@rohit.sharma\\_7010/a-complete-tutorial-for-named-entity-recognition-and-extraction-in-natural-language-processing-71322b6fb090](https://medium.com/@rohit.sharma_7010/a-complete-tutorial-for-named-entity-recognition-and-extraction-in-natural-language-processing-71322b6fb090)

### 2.7.2 Sentiment Analysis

The sentiment of a piece of text is the tone or the impression of its author. For instance, “The weather is cool!” has a positive sentiment, while “This movie is a waste of time!” has a negative sentiment. In this task, you will build a deep learning-based sentiment analyzer. You should use an architecture suitable for handling sequences (RNN, LSTM, etc.). Your model should take a sentence as input and classify it to either positive or negative sentiment.

## Web link to the Dataset

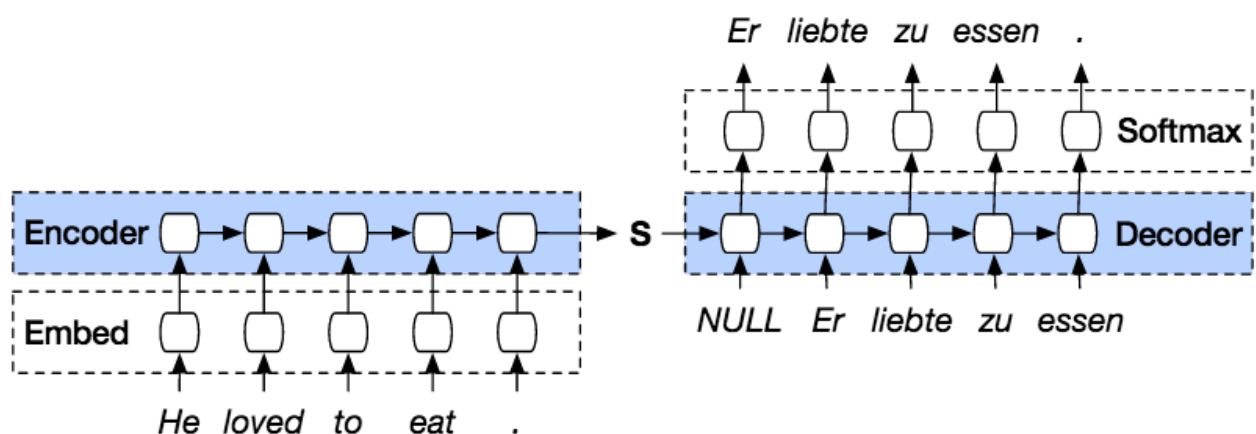
- [IMDB sentiment dataset](#). It contains ~25K reviews with their labels.

## Web links to Resources

- <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- [An LSTM approach to short text sentiment classification with word embeddings](#)
- <https://towardsdatascience.com/sentiment-analysis-using-lstm-step-by-step-50d074f09948>

### 2.7.3 Machine Translation

Machine Translation is the task of translating a piece of text in one language to its equivalent in another language. Your task is to build an Arabic-English machine translation model. Your model should be based on the Sequence-to-Sequence architectures (See resources below). Your model should have an encoder and a decoder network based on LSTM. **Bonus:** Use attention mechanism (see resources below).



## Web link to the Dataset

- [OpenSubtitles 2018 Arabic- English dataset](#). (You can use other datasets for other languages but ArabicEnglish is preferred).

## Web links to Resources

- [Sequence to sequence learning with neural networks](#)

- [Effective approaches to attention-based neural machine translation](#)
- [Neural machine translation by jointly learning to align and translate](#)
- [https://pytorch.org/tutorials/intermediate/seq2seq\\_translation\\_tutorial.html](https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html)
- <https://machinelearningmastery.com/define-encoder-decoder-sequence-sequence-modelneuralmachine-translation-keras/>

In this assignment I suggest to use the following tools and technologies:

1. **Anaconda** — Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications, that aims to simplify package management and deployment. You can download it from the link below according to your system <https://www.anaconda.com/download/>
2. **Spyder** — Spyder is an open source cross-platform IDE for scientific programming in the Python language. It comes installed with anaconda. If not, install it using anaconda navigator.
3. **Tensorflow** — TensorFlow is an open-source software library for dataflow programming across a range of tasks. Download link — [https://www.tensorflow.org/install/install\\_windows](https://www.tensorflow.org/install/install_windows)
4. **Keras** — Keras is an open source neural network library written in Python. Activate Tensorflow env and install keras using ‘pip install keras’

### 3. Guidelines for Report

Below are guidelines on how to write-up your report for the final project (no more than **7,000 words limit**). Of course, for a short class project, all of the sections may not be relevant. However, you may use it as a general guide in structuring your final report.

A "standard" experimental AI paper consists of the following sections:

#### 1. Introduction

*Motivate and abstractly describe the problem you are addressing and how you are addressing it. What is the problem? Why is it important? What is your basic approach? A short discussion of how it fits into related work in the area is also desirable. Summarize the basic results and conclusions that you will present.*

#### 2. Problem Definition and Algorithm

##### 2.1 Task Definition

*Precisely define the problem you are addressing (i.e. formally specify the inputs and outputs). Elaborate on why this is an interesting and important problem. Include a simple specific example, providing the I/O showing how the output is related to the input specifying the desired/achieved properties of the output illustrating the basic terms used.*

##### 2.2 Algorithm Definition

*Describe in reasonable detail the algorithm (rules) you are using to address this problem. A pseudo-code description of the algorithm you are using is frequently useful. Trace through a concrete example, showing how your algorithm processes this example. The example should be complex enough to illustrate all of the important aspects of the problem but simple enough to be easily understood. If possible, an intuitively meaningful example is better than one with meaningless symbols.*

#### 3. Experimental Evaluation

##### 3.1 Methodology

*What are criteria you are using to evaluate your method? What specific hypotheses does your experiment test? Describe the experimental methodology that you used. What are the dependent and independent variables? What is the training/test data that was used, and why is it realistic or interesting? Exactly what performance data did you collect and how are you presenting and analyzing it? Comparisons to competing methods that address the same problem are particularly useful.*

##### 3.2 Results



*Present the quantitative results of your experiments. Graphical data presentation such as graphs and histograms are frequently better than tables. What are the basic differences revealed in the data? Are they statistically significant?*

### 3.3 Discussion

*Is your hypothesis supported? What conclusions do the results support about the strengths and weaknesses of your method compared to other methods? How can the results, be explained in terms of the underlying properties of the algorithm and/or the data.*

### 4. Related Work

*Answer the following questions for each piece of related work that addresses the same or a similar problem. What is their problem and method? How is your problem and method different? Why is your problem and method better?*

### 5. Future Work

*What are the major shortcomings of your current method? For each shortcoming, propose additions or enhancements that would help overcome it.*

### 6. Conclusion

*Briefly summarize the important results and conclusions presented in the paper. What are the most important points illustrated by your work? How will your results improve future research and applications in the area?*

### Bibliography & Citations

*Be sure to include a standard, well-formatted, comprehensive bibliography with citations from the text referring to previously published papers in the scientific literature that you utilized or are related to your work. Always use a consistent citation style for your references. The standard style used around the university is the Harvard Style. However, I will accept any other standard style (e.g. APA style) as long as it is used consistently.*

Try to make your report EASY to read.

- Be sure to include an overview in the beginning, which outlines what the report will be describing, in a section-by-section fashion.
- Include simple examples (or better, a single simple example throughout), to help illustrate the ideas.
- A picture is worth (at least) a thousand words. Use figures, flow-charts, graphs, whenever appropriate.
- The material should be structured, and flow. It should NOT be a core-dump of everything you happened to read when you were looking at things related to X. Readers (read "the people who will assign your grade!") get annoyed by having to wade through irrelevant material.
- If you are giving a high-level description of an algorithm, be sure to explicitly state its input and output.
- Many algorithms have a flow of information, from one subroutine to another. Provide one or more figures, to make the ideas clear.
- Also, proof-read your report. As a grader, I find it *very irritating* to read a report that has pages of easytofix typos, illegible figures, missing citations, etc. And you really don't want to irritate the person who is assigning your grade...
- If you are describing a precise algorithm, you should give the actual formulas, using terms that are welldefined, in the report.
- Your report should be self-contained. You are allowed to copy figures from other sources (if they are properly credited). But if you do, be sure to define the terms that appear in that figure!
- Save trees - hand in a 2-sided version. And use section numbers, and page numbers!

The submission must accompany a CD containing your code; also include a tutorial, and a user manual which will help the user to run the agent based system. **An agreed dataset should be provided.** Use your creativity to make the submission better .

## 4. Demonstration

The demonstration times for individual teams will be posted later in the semester. It is planned that the demonstrations will take place around the submission deadline. Pls. make your appointment.

## 5. Academic Integrity

Copying or paraphrasing someone's work (code included), or permitting your own work to be copied or paraphrased, even if only in part, is not allowed, and would result in a disciplinary action according to the university policy. Any resources or ideas borrowed from other sources should be explicitly referenced in text and bibliographies.

## 6. Marking Scheme:

The grading will be broken down based on the following criteria:

Deliverable	Criterion			Max	Actual
	Total for Software				
Report	based on quality of report	Introduction and Literature Review	<ul style="list-style-type: none"> <li>• Articulation of research issue/problem</li> <li>• Coherence of the research aim(s) and objectives</li> <li>• Relevance and importance of the research issue</li> <li>• Criteria for the proposed solution</li> <li>• Explanation of constraints</li> <li>•</li> </ul>	10%	
			<ul style="list-style-type: none"> <li>• Organisation and logical sequence of the contents of the dissertation</li> <li>• Comprehensive and correct citation of references and/or bibliography</li> <li>• Appropriate written style and use of language</li> <li>• General quality of presentation</li> <li>• Supporting documents are provided</li> <li>•</li> </ul>	10%	
			<ul style="list-style-type: none"> <li>• Comprehensive, rigorous and critical review of the literature</li> <li>• Appropriateness of theoretical concepts employed</li> </ul>	10%	
			Subtotal	30%	

		Research Methodology, and data Analysis, interpretation and discussion	<ul style="list-style-type: none"> <li>• Relevant and effective research methodology</li> <li>• Rigour of application of the methods of investigation</li> <li>• Identification of a solution and exploration of alternatives</li> <li>• Development of an application</li> <li>•</li> </ul>	25%		
			<ul style="list-style-type: none"> <li>• Quality and depth of research</li> <li>• Appropriateness of methods of analysis</li> <li>• Rigour of application of the methods of analysis</li> <li>• Reproducibility of results</li> </ul>	25%		
		Subtotal		50%		
		Creativity, and Conclusion(s)	<ul style="list-style-type: none"> <li>• Producing results close to or exceeding those in published research. If there is no relevant published research, then this score will be used for accuracy and coverage sufficient to support interpretation and conclusion. Testing in multiple conditions</li> </ul>	10%		
			<ul style="list-style-type: none"> <li>• Design and implementation that demonstrates software engineering skills and completeness</li> </ul>			
			<ul style="list-style-type: none"> <li>• Consistency of conclusion(s) with research objectives</li> <li>• Consistency of conclusion(s) with findings and discussion</li> <li>• Comprehensive of the implications of the conclusion(s)</li> <li>• Appropriateness of recommendations on the basis of the conclusion(s)</li> <li>• Value of the research and makes a contribution to knowledge and /or practice</li> </ul>	10%		
		Subtotal		20%		
		Total		100%		

## References

- [1] Daniel M. Bikel. 2004. A Distributional Analysis of a Lexicalized Statistical Parsing Model. In the proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)
- [2] Mona Diab, Kadri Hacioglu and Daniel Jurafsky. 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In HLT-NAACL 04.
- [3] Abdallah, S., Shaalan, K., Shoaib, M., Integrating Rule-based System with Classification for Arabic Named Entity Recognition, Lecture Notes in Computer Science, Computational Linguistics and Intelligent Text Processing, 7181: 311-322, 2012.
- [4] Mohammed Attia, Antonio Toral, Lamia Tounsi, Monica Monachini and Josef van Genabith. 2010. 'An automatically built Named Entity lexicon for Arabic'. LREC 2010. Valletta, Malta.
- [5] J-P Ng and M-Y Kan, "QANUS: AN Open Source Question-Answering Platform", 2010, <http://wing.comp.nus.edu.sg/~junping/docs/qanus.pdf>
- [6] Nico Schlaefler, "A Semantic Approach to Question Answering". VDM Verlag Dr. Mueller, ISBN 3836450739, 2007.
- [7] Recommender Systems, <http://www.cc.uah.es/drg/courses/datamining/IntroRecSys.pdf>
- [8] Dipanjan Das and André F.T. Martins, "A Survey on Automatic Text Summarization", Literature Survey for the Language and Statistics II course at Carnegie Mellon University, 2007
- [9] Cunningham H, Maynard D, Bontcheva K, Tablan V, Aswani N, Roberts I et al. Text Processing with GATE (Version 6). University of Sheffield Department of Computer Science, 2011.
- [10] Oudah, M. and Shaalan, K. Person Name Recognition Using the Hybrid Approach. Lecture Notes in Computer Science, Natural Language Processing and Information Systems, Springer Berlin Heidelberg, vol. 7934, pages 237–248, 2013.
- [11] Shaalan, K., Oudah, M., A Hybrid Approach to Arabic Named Entity Recognition, Journal of Information Science (JIS), 40(1): 67-87, SAGE Publications Ltd, UK, 2014.
- [12] Al-Ayyoub, Nuseir, Alsmearat, Jararweh, Gupta, M. Al-Ayyoub, A. Nuseir, K. Alsmearat, Y. Jararweh, B. Gupta. Deep learning for arabic NLP: A survey, Journal of Computational Science, 26 (2018), pp. 522-531, 2018.