

Beta

 Try the new code view

🔗 main



CITS4012\_2023 / README.md



adlnlp Update README.md



3 contributors



188 lines (114 sloc) | 18.1 KB



# CITS4012 Assignment Specification - Wiki QA

Lecture XX or Lab XX = Lecture/Lab Reference

**(Justify your decision)** = You are required to justify your final decision/selection with the empirical evidence by testing the techniques in the section 3. Model Testing. Must write in the report (Section 4. Documentation)

**(Explain the performance)** = You are required to explain the trend of performance, and the reason (or your opinion) why the trends show as they do. Must write in the report (Section 4. Documentation)

**Templates:** ipynb template and documentation template can be found in the Assignment Submission Method

## Submission Due: 20 May 2023

This assignment can be done individually or in teams of two or three(max). We strongly encourage healthy collaboration. See the [University of Western Australia Working in Groups Guide](#). If your team member does not engage or collaborate, please contact the unit coordinator ([Dr. Caren Han](#)) with describing the contributions of each collaborator. We strongly recommend to start working early so that you will have ample time to discover stumbling blocks and ask questions.

In this assignment, you are to propose and implement a Wiki QA (Question Answering) framework using the Sequence model and different NLP features. The QA framework should have the ability to read documents/texts and answer questions about them. The detailed information for the implementation and submission step are specified in the following sections. Note that lecture note and lab exercises would be a good starting point and baseline for the assignment.

Note that we specified which lectures and labs are highly related!

## Table of Contents

- [0. Important Dates \(Very Important!!\)](#)
- [1. DataSet](#)
- [2. QA Model Implementation](#)
- [3. Model Testing](#)
- [4. Documentation](#)
- [Assignment Submission Method](#)
- [FAQ](#)



## 0.Important Dates

---

The Important date for the Assignment can be summarised as follows:

- **Assignment Specification Release:** 20 April 2023
- **Assignment Group EOI Due:** 23 April 2023
- **Assignment Group Release:** 24 April 2023
- **Assignment Group Revision Due:** 26 April 2023
- **Assignment Submission Due:** 20 May 2023

All deadlines are **11:59 PM (AWST)**.

NOTE: If you want to do individual or already have group members in mind, Please Submit **Group EOI** by EOI Due (23 April 2023 11:59PM) Otherwise, your group members will be selected by our teaching team.

**After the Group Release date (24 April 2023 11:59 PM): YOU MUST CONTACT YOUR GROUP MEMBER BY THE GROUP REVISION DUE (26 April 2023 11:59 PM).** If you want to change your group to individual after EOI due, you can revise it by the Assignment Group Revision Due. However, in this case, you should get your team member's agreement. Please email the Unit Coordinator ([Dr. Caren Han](#)) with your group member's agreement.

If you DO NOT contact or respond your group member by the Group Revision Due (26 April 2023 11:59 PM), and your group member REQUESTs for working as an individual, you must work the Assignment individually.



## 1. DataSet [4 marks]



You need to put the code that you conduct all actions for this section to the [ipynb template](#)

In this assignment, you are asked to use Microsoft Research WikiQA Corpus. The WikiQA corpus includes a set of question and sentence pairs, which is collected and annotated for research on open-domain question answering. The question sources were derived from Bing query logs, and each question is linked to a Wikipedia page that potentially has the answer. More detail on this data can be found in the paper, [WikiQA: A Challenge Dataset for Open-Domain Question Answering](#).

- **Download datasets:** You will be provided two datasets, including the [training dataset](#) and the [testing dataset](#). Both datasets contain the following attributes: QuestionID, Question, DocumentID, DocumentTitle, SentenceID, Sentence, and Label (answer sentence, if label=1). If you want to explore or use the full dataset, you can download it via the [Link](#). However, the training and testing split should be followed by the one we provided.
- **Data Wrangling (Justify your decision):** You need to first wrangle the dataset. As you can see in the following Figure 1, each row is based on each sentence of the document. You need to construct three different types of data for training the model: Question, Document and Answer. To construct the document data, you should concatenate (with space or full-stop and space) each sentence that has the same DocumentID. For the answer data, use the sentence that has Label as 1.

**Tips for the Data Wrangling:** For example, you can label each token of the document into three token types (Start Token of the Answer, End Token of the Answer, and None) or four token types (Start Token of the Answer, Inner Token of the Answer, End Token of the Answer, and None).

Please check how we have the dataset for N-to-N Task in [Lab6](#).

Note: 1) Some questions may not have answers. (All labels are 0) 2) Some documents may have multiple questions.

QuestionID	Question	DocumentID	DocumentTitle	SentenceID	Document	Sentence	Label
0	Q1	D1	Glacier cave	D1-0	A partly submerged glacier cave on Perito More...		0
1	Q1	D1	Glacier cave	D1-1	The ice facade is approximately 60 m high		0
2	Q1	D1	Glacier cave	D1-2	Ice formations in the Titlis glacier cave		0
3	Q1	D1	Glacier cave	D1-3	A glacier cave is a cave formed within the ice		1
4	Q1	D1	Glacier cave	D1-4	Glacier caves are often called ice caves, but...		0
5	Q2	D2	Circular motion	D2-0	In physics, circular motion is a movement of ...		0
6	Q2	D2	Circular motion	D2-1	It can be uniform, with constant angular rate ...		0
7	Q3	D3	Circular motion	D3-0	The rotation around a fixed axis of a thesauri		0

Figure 1. WikiQA: Raw Data - Sample View



## 2.QA Model Implementation [10 marks]



You need to put the code that you conduct all actions for this section to the [ipynb template](#)

You are to propose and implement the open-ended QA framework using word embedding, different types of feature combinations, and deep learning models. The following architecture describes an overview of QA framework architecture. (Click [This Link](#) to view the high-resolution image.)

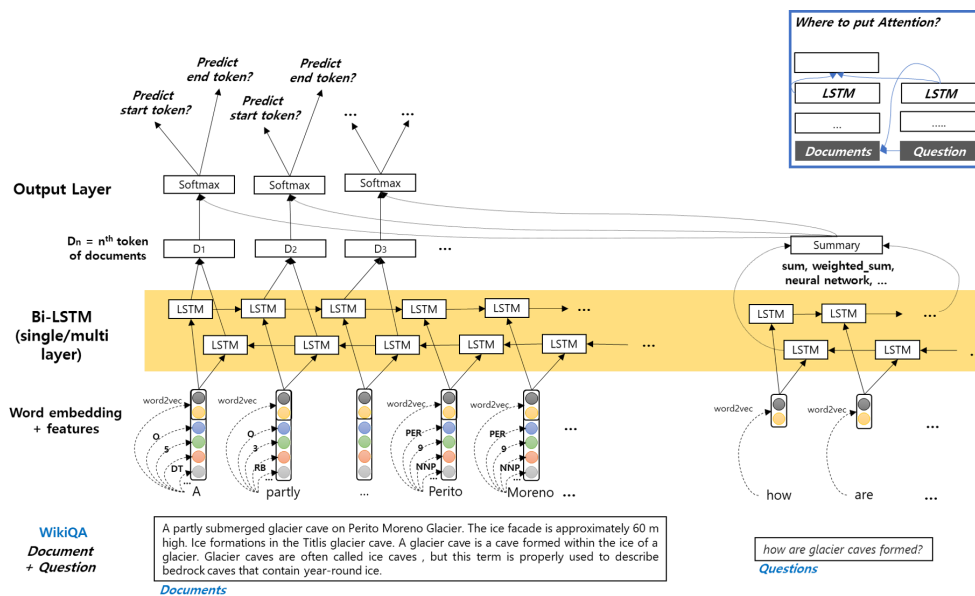


Figure 2. Overview of the Architecture of the WikiQA

### 1) Input Embedding [5 marks]

You are asked to generate the word vector by using the word embedding model and different types of features. For example, in Figure 2, the word 'Perito' is converted to the vector [word2vec, PER, 9, NNP, ...].

- **Word embedding (Justify your decision)** - refer to [Lab2](#), [Lab6](#). You are to apply a pre-trained word embedding model, including word2vec CBOW or skip-gram, fastText, or gloVe. For example, you can find various types of pre-trained word embedding models from the following link: <https://github.com/RaRe-Technologies/gensim-data#datasets>
- **Feature extraction (Justify your decision)** Different types of features should be extracted in order to improve the performance and evaluate different combinations of model specification (will discuss more in the 4. Documentation section - Evaluation). You are asked to extract at least **three types** of features from the following list:
  1. PoS Tags -refer to [Lab6](#)
  2. TF-IDF -refer to [Lab1](#)
  3. Named Entity Tags -refer to [Lab6](#)
  4. Word Match Feature: check whether the word appears in the question by using decapitalisation or lemmatization -refer to Lecture 5 and [Lab5](#)
  5. \*Others: Any one embedding type can be included if you have a logical reason why it requires to the wiki QA.

## 2) Sequence QA Model (Recurrent Model with Attention) [5 marks] -refer to Lecture 7, [Lab7(Will be released on 24 April 2023 9AM)], [Lab4 -RNN/Bi-RNN](#), [Lab6 - Bi-LSTM](#)

In this QA framework, you are to implement a sequence model using RNN (such as RNN, LSTM, Bi-LSTM, GRU, Bi-GRU, etc.) with attention.

Figure 2 describes the Bi-LSTM-based model as an example. As can be seen in Figure 2, your sequence model should have two input layers (one for the Question and another for Document) and one output layer (Answer). The output layer should predict the answer (whether the each token is the answer of start token/end token or not). Also, your sequence model is required to include an attention layer to get better performance (which needs to be presented in your architecture). The positions to add the Attention layer are recommended in the blue box (top-right) of Figure 2.

For the sequence model, you can use single or multiple layers but you should provide the optimal number of layers. **(Justify your decision)**

The detailed architecture of your sequence model needs to be drawn and described in the report (refer to section 4 - Documentation). You need to justify (in the report) why you apply the specific type of RNN and put the attention layer in the specific position. **(Justify your decision)**

The final trained model should be submitted in your Python package.



## 3. Model Testing [9 marks]



You need to put the code that you conduct all actions for this section to the **ipynb template**

You need to justify your decision and explain the pattern by testing the performance of your implementation. The testing result should include precision, recall and F-value -refer to [Lab4\(Exercise\)](#).

You need to write a manual (readme) for the assessor. Your manual should guide how to test your program and also includes a list of packages (with version) that you used. If you work on Google Colab or Jupyter Notebook (.ipynb), your manual should guide the assessor on where to upload the required files (trained model, dataset, etc.). Note the assessor will use Google Colab to open your ipynb file. Unless you have a function that downloads required files from URL or Google Drive.

The following model testings should be conducted. For each testing, you MUST include and visualise the table/graph in the ipynb file (your code) and your documentation (section 4). Note that you MUST make the final model and conduct the ablation studies with that final model as follows:

- **Input Embedding Ablation Study[3 marks]: (Explain the performance)**  
Test at least three types of input embedding variants (e.g. word2vec only, word2vec+POS tag embedding, word2vec+all 3 features embeddings) for your model, and visualise a table/graph with the performance (exact value of precision, recall, and f1) of all 3 variants.
- **Attention Ablation Study[3 marks]: (Explain the performance)**  
Test at least three types of attention variants (attention calculation variants or attention alignment variants) for your model, and visualise a table/graph with the performance (exact value of precision, recall, and f1) of all 3 variants. e.g. try three different attention calculations, including dot-product, scaled dot-product, and cosine attention calculation.
- **Hyper Parameter Testing[3 marks]: (Explain the performance)**  
Test at least different 5 hyperparameter variants (5 different epoch values or 5 learning rates) for your model, and visualise a table/graph with the performance (exact value of precision, recall, and f1) of all 5 variants.



## 4. Documentation [7 marks]



Please use the provided documentation template (**overleaf** or **word**)

You should submit pdf version of the assignment report (8 pages Maximum - excluding reference and appendix). The detailed documentation requirement for each section can be found above section 1. Dataset (2 marks), 2. QA Model Implementation (2 marks), and 3. Model Testing (3 marks). Please check the requirement items with the tag of (Justify your decision) or (Explain the performance). You MUST write those items to the report.

\*\*The justification MUST be based on the previous literature reference (incl. international conference or journal publication) or empirical evaluation. (Check the definition of 'empirical evaluation' at the following FAQ Section). \*\*

**Note that We DO NOT MARK the Documentation if it is not implemented as described in the report.**



## Assignment Submission Method

---

**Due Date:** 11:59 PM, Saturday 20 May 2023

**Submission:** LMS Assignment Submission Box (The submission box will be opened on 1 May 2023)

**You Must Submit Three Files:**

- pdf file: a report (documentation) with the given template ([overleaf](#) or [word](#)) (file name: cits4012\_your\_groupid.pdf)
- ipynb file or python packages (.py files): An ipynb file or python package that includes all your implementation (the implementation is described in the following sections - dataset, qa model, model testing). You MUST use this [ipynb template](#) (file name: cits4012\_your\_groupid.ipynb)
- zip file: A zip file that contains trained models, dataset, readme - if necessary, and all other required files that you used for your program. (file name: cits4012\_your\_groupid.zip)



## FAQ

---

**Question: My Model performs really bad (Low F1). What did I do wrong?**

Answer: Please don't worry about the low performance as our training dataset is very small and your model is very basic deep learning model. Of course, there is something wrong with your code if it comes out below 10% (0.1).



**Question: How can we make POS Tag or NER Tag to the embedding?**

Answer: You can either use it as a categorial embedding (e.g. putting the POS tag index- like Noun-->0, Verb -->1, etc) or applied word embedding (e.g. Extracting the Pretrained Word2Vec embedding for the term 'Noun' or 'Verb'). If you have any other approach to apply, please proceed! (Note you need to have a justification - why you apply it by using theoretial justification or empirical evaluation)

**Question: Do I need to save the word embedding model or Recurrent models?**

Answer: We highly recommend you to save your best word embedding model, and load it when you use it in your code. Otherwise, it is sometimes removed and overwrite all your code so that you need to re-run the whole code again.

**Question: Is there any other marking scheme/marking criteria available for the assignment?**

Answer: The assignment specification is extremely detailed and the partial mark details are already given. The marking will be directly conducted based on the specification. It means you will get the full mark if you fulfill all the requirement that we mentioned in the specification.

**Question: What do I need to write in the justification or explanation? How much do I need to articulate?**

Answer: We recommend conducting empirical evaluation. Empirical evaluation refers to the appraisal of a theory by observation in experiments. The key to good empirical evaluation is the proper design and execution of the experiments so that the particular factors to be tested can be easily separated from other confounding factors. Hence, visualizing the comparison of different testing results is a good way to justify your decision. Explain the trends based on your understanding. You can find another way (other than comparing different models) as well - like citing the peer-reviewed publications.

**Question: Can I use other Sequence Model(like Transformer)?**

Answer: Yes, you can but please make sure all requirements in other sections should be successfully followed. For example, make sure you have all required dataset settings, input embeddings and model testings.

[Give feedback](#)