FIT5149 S2 2024 Assessment 2: Probabilistic Graphical Models

18 September 2023

Marks	25% of all marks for the unit
Due Date	11:55 PM Monday, 14 October 2024 (Week 12)
Extension	An extension could be granted for circumstances. Please refer to the university webpage on special consideration. Please note that ALL special form of extension is now to be submitted centrally. All students MUST submit an online special consideration form via Special Consideration Application. Students will no longer seek extensions from the teaching team/ Chief Examiner (CE). All extensions will be via the special consideration form process and extensions will be approved by SEBS (not the CE).
Lateness	For all assessment items handed in after the official due date. Without an agreed extension, a 5% penalty applies to the student's mark for each day after the due date (including weekends and public holidays) for up to 7 days. Assessment items handed in after 7 days will not be considered/marked.
Individual Assessment	This assessment is an **individual assignment**, and you need to complete this assessment by yourself.
Authorship	The final submission must be **identifiable your own work**. Breaches of this requirement will result in an assignment not being accepted for assessment and may result in disciplinary actions.
Submission	Everyone is required to submit one ZIP file containing a prediction file, the implementation for Task 1 and Task 2 respectively. The Zip file must be submitted via Moodle. A draft submission won't be marked.
Programming language	Either R or Python

1 Task 1: Bayesian Networks

In the provided zip file, you'll find the A2_task1_model.bif, which contains a Bayesian network model in Bayesian Interchange Format (BIF). Please load the model using the provided Jupyter notebook and answer all the questions outlined in the A2_Task1_questions.pdf included in the zip file. Make sure to include all intermediate steps in the Jupyter notebook for each question. In the corresponding markdown cells, explain your chosen methods and why you choose them. This Jupyter notebook should be included as part of your Assignment 2 submission.

Hints:

- When computing the probabilities, explain which inference method do you choose and why you choose that inference method.
- For independence related questions, explain why you think two variables are d-separated by listing all relevant blocked and unblocked paths. Additionally, explain why each of those paths is blocked or unblocked.
- There might be more than one smallest sets of random variables that d-separates two given variables for Q2 d). It is sufficient to select one of them.
- When computing expectations of variables, explain which inference method do you choose and why you choose that inference method.
- The CPDs of a Bayesian network are automatically generated. It is fine to ignore the warnings, such as "WARNING:pgmpy:Probability values don't exactly sum to 1....", as long as the number is close to 1.

2 Task 2: Conditional Random Fields in Practice

Task Description: This task aims to practice your ability to apply the Conditional Random Fields (CRF) models to semantic role labeling (SRL) Màrquez et al. [2008], which is a sequence labeling task. You are required to use either sklearn-crfsuite, pgmpy or the R wrapper for CRFsuite to complete this task.

Sentence-level semantic analysis is concerned with the characterization of events, such as "who" did "what" to "whom" at "when". The primary task of SRL is to indicate which semantic relations hold among a predicate and its associated arguments. The relation labels are drawn from a predefined list of possible semantic roles for a given predicate or a class of predicates.

[President Bush]$_{ARG0}$ [nominated]$_{V}$ [two individuals]$_{ARG1}$ [on Tuesday]$_{ARGM-TMP}$ [to replace retiring jurists]$_{ARGM-PRP}$

In this example:
- The semantic role *ARG0* refers to the agent or the doer of the action (President Bush).
- *V* indicates the predicate (the action word, in this case, "nominated").
- *ARG1* refers to the patient or the entity affected by the action (two individuals).
- *ARGM-TMP* indicates a temporal adjunct (on Tuesday).
- *ARGM-PRP* indicates a purpose adjunct (to replace retiring jurists).

The labels include both core arguments (such as 'ARG0', 'ARG1', 'ARG2'), adjuncts (such as 'ARGM-TMP', 'ARGM-MNR', 'ARGM-LOC'), as well as additional labels, such as:

- **ARGM-CAU**: Causal adjunct, indicating the reason or cause of the event.
- **R-ARGM-TMP**: Reference to a temporal adjunct.
- **C-ARG1**: Continuation of the argument labeled as ARG1.
- **ARGM-MOD**: Modal adjunct, indicating modality (e.g., might, could).

- **ARGM-NEG**: Negation marker (e.g., ”not”).
- _: default label for the remaining cases.

The labels of semantic roles are located in the last column of the dataset (in the CoNLL-U format). *Your predictive task is to build a CRF model to estimate the labels in the last column.*

2.1 Subtasks

This task involves designing features, training CRF models with appropriate hyperparameters, evaluating model performance, and interpreting the results. You will work individually to complete the following subtasks:

2.1.1 Prediction Task

- Develop a model with CRF to semantic role labelling.
- Select and submit the best-performing model for final submission.
- Apply appropriate data preprocessing methods, feature selection, or feature designing techniques.
- Tune hyperparameters of the models to optimize performance.

2.1.2 Analysis Task

- Design features and identify the key features that significantly influence the predictive outcomes.
- Provide statistical evidence to support your findings.
- Explain which features (including newly created ones) are particularly useful for semantic role labelling and why.

2.1.3 Documentation

In the Jupyter notebook, using Markdown cells to document your approach and results, including:

- Feature Design: Explain how features were chosen and designed, including word-level and context-level features.
- Methodology: Document the steps taken from preprocessing the dataset, building the model, to performance evaluation.
- Hyperparameter Tuning: Include details on the model’s hyperparameters and how tuning was performed to optimize performance.
- Evidence-based discussion and analysis of all experiments and experimental results.

You could utilise the .ipynb (Python) or .rmd(R) file to document the above information together with your code/model implementation. You do not need a separate pdf file to write a report.

2.2 Datasets and Features

You need to explore the dataset to understand the relationship between tokens and their roles, using this understanding to design meaningful features for CRF.

Datasets

- Training set (en_ewt-up-train.conllu): 12543 sentences labeled with semantic roles and synthetic information.
- Development set (en_ewt-up-dev.conllu): 2002 sentences labeled with semantic roles and synthetic information.
- test set (en_ewt-up-test-no-labels.conllu): 2077 sentences labeled with synthetic information. All semantic roles are set to the default label underscore (_).

You are allowed to train your final CRF model on both en_ewt-up-train.conllu and en_ewt-up-dev.conllu to predict labels on en_ewt-up-test-no-labels.conllu.

Understanding CoNLL Data CoNLL (Conference on Computational Natural Language Learning) format is widely used for tasks like syntactic parsing and semantic role labeling (SRL). Each line in a CoNLL-U formatted file represents a word or token and contains several columns with linguistic information. Blank lines separate sentences. The typical columns in CoNLL-U format include:

- **ID**: The position of the word or token in the sentence.
- **Form**: The word or punctuation symbol.
- **Lemma**: The base or dictionary form of the word.
- **UPOS**: Universal part-of-speech (POS) tag, such as `NOUN`, `VERB`, etc.
- **XPOS**: Language-specific part-of-speech tag (e.g., `NNP` for proper nouns in English).
- **Features**: Additional grammatical features such as number, case, tense, etc., represented as key-value pairs (e.g., `Number=Sing` for singular nouns).
- **Head**: The ID of the syntactic head of the current word (which word this token depends on syntactically).
- **Deprel**: The type of syntactic dependency relation that connects a word to its head (e.g., `nsubj` for nominal subject, `obj` for object).
- **Deps**: Enhanced dependency graph, which may include additional syntactic relations (often same as the `deprel` column).
- **Misc**: Miscellaneous information, such as whether the token has a trailing space (e.g., `SpaceAfter=No`).
- **Predicate**: If the word is a verb, this column specifies its predicate sense (e.g., `morph.01`), otherwise, it is marked as an underscore (_).
- **Argument Role**: The semantic role assigned to the word with respect to the predicate (e.g., `ARG0` for the agent, `ARG1` for the patient).

Example of CoNLL Data Below is a shorter example sentence in CoNLL-U format for SRL:

```
# sent_id = weblog-blogspot.com_zentelligence_20040423000200_ENG_20040423_000200-0001
# text = What if Google Morphed Into GoogleOS?
1    What      what      PRON   WP   PronType=Int      0    root    0:root    _    _        _
2    if        if        SCONJ  IN   _                 4    mark    4:mark    _    _        _
3    Google    Google    PROPN  NNP  Number=Sing       4    nsubj   4:nsubj   _    _
     ARG1
4    Morphed   morph     VERB   VBD  Mood=Ind|Tense=Past 1  advcl   1:advcl:if _ morph.01 V
5    Into      into      ADP    IN   _                 6    case    6:case    _    _        _
6    GoogleOS  GoogleOS  PROPN  NNP  Number=Sing       4    obl     4:obl:into SpaceAfter=
     No _ ARG2
7    ?         ?         PUNCT  .    _                 4    punct   4:punct   _    _        _
```

In this example:

- **Token 4** (`Morphed`) is a predicate, as indicated by `morph.01`, and its role label is `V`.
- **Token 3** (`Google`) is labeled with the semantic role `ARG1`, meaning it is the patient or subject of the predicate `Morphed`.
- **Token 6** (`GoogleOS`) is labeled as `ARG2`, meaning it is another argument of the predicate `Morphed`, often representing the object or result.

By understanding this structure, students can extract relevant features such as the word itself, POS tag, dependency relation, and predicate information. Contextual features such as previous and next tokens are also essential for accurately predicting semantic roles in CRF-based models.

2.3 Evaluation Metrics

To assess the performance of the Conditional Random Fields (CRF) model for Semantic Role Labeling (SRL), the following standard sequence labeling metrics are used. For all those metrics, we consider the label set excluding the default label (_). The developed model is supposed to predict a label for each word in a sentence.

Precision: Precision measures the accuracy of the positive predictions made by the model. It is the ratio of true positive predictions to the total number of positive predictions (i.e., the sum of true positives and false positives):

$$Precision = \frac{TruePositives(TP)}{TruePositives(TP) + FalsePositives(FP)}$$

Recall: Recall measures the model's ability to correctly identify all actual positive instances. It is the ratio of true positive predictions to the total number of actual positives (i.e., the sum of true positives and false negatives):

$$Recall = \frac{TruePositives(TP)}{TruePositives(TP) + FalseNegatives(FN)}$$

F1-Score: The F1-score is the harmonic mean of precision and recall. It provides a balance between the two, especially when there is an uneven distribution of class labels:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Support: Support refers to the number of true instances of each class (label) in the dataset, providing insight into the distribution of labels. It does not have a formula but is simply the count of occurrences for each label.

In SRL, the goal is to predict a label (such as `ARG0`, `ARG1`, `ARGM-TMP`, etc.) for each word in a sentence. To calculate overall performance metrics, the predicted labels for all sentences are concatenated and flattened into a single sequence, as are the true labels. The overall precision, recall, and F1-score are then computed across all predicted and true labels.

To compute these metrics, you can use the `classification_report` function from the `sklearn` package, and the `flatten` function from `sklearn_crfsuite.utils` to concatenate and flatten the labels across sentences.

Students should provide a detailed evaluation using these metrics and interpret the model's performance. The key metric for this task is macro-average F1 score.

2.4 Suggested Steps

1. Load the SRL dataset with the provided sample code, inspect its structure and understand the provided features. Modify the dataset loading code to add or remove features of interest.

2. Feature Engineering:

    - Extract useful features at both word and sentence levels:
        - Word-level: Current word, POS tag, capitalization, word prefixes/suffixes, whether the word is a predicate.
        - Sentence-level: Previous and next word, previous and next POS tags, dependency relations, syntactic chunk information, and other contextual features.
    - Incorporate morphological and syntactic features, such as lemma, POS and dependency relations.
    - If available, use external resources such as deep learning-based feature representations (e.g., pre-trained embeddings) to enhance the feature set (e.g., word embeddings, character embeddings).

3. CRF Model Development:

    - Use *sklearn-crfsuite*, pgmpy, or the R version of *Crfsuite* to implement the CRF model, initializing it with the extracted features and labels.
    - Tune hyperparameters like regularization coefficients (c1, c2) to avoid overfitting and ensure a balance between model complexity and performance.
    - Perform k-fold cross-validation on the training data to select the optimal hyperparameters and prevent overfitting.

4. Model Evaluation:

    - Evaluate the model using standard metrics such as precision, recall, F1-score, and accuracy for each semantic role.
    - Generate a confusion matrix to identify patterns of common misclassifications and assess which roles are often confused with others.
    - Conduct an error analysis to understand where the model struggles (e.g., ambiguous predicates, rare or complex roles, sentences with unusual syntactic structures).
    - Compare the performance across different sentence lengths, predicates, and argument types to identify performance bottlenecks.

5. Result Analysis:

    - Summarize the key findings from the model's performance, highlighting the best-performing and worst-performing semantic roles.
    - Discuss the role of each feature in contributing to the model's accuracy, identifying which features had the most significant impact on the results.
    - Reflect on the effect of feature engineering, including any improvements made by adding syntactic or morphological features.
    - Suggest potential improvements to the model, such as incorporating deep learning-based feature representations or adjusting the training data to address label imbalance.

3 In-class Interview (Mandatory)

You need to demonstrate the code in Task 1 and 2 to your TA in an in-class in-person interview. **It will be conducted during the week 12 applied session you allocated to.** Fail to explain how your model is designed with your code will result in **ZERO marks** for the tasks you cannot explain.

Details:

- **Time/Date**: Week 12, during your allocated Applied sessions
- **Duration**: Approximate 5 minutes per student
- **Location**: Normal classroom/location of allocated applied session
- **Content**: Please explain your code of A2 and answer questions if any

Fail to attend the mandatory interview will result in ZERO marks for A2 regardless of your submission of work on Moodle.

***Note: this interview cannot be postponed even you have been granted with an approved extension. You need to show up in week 12 applied sessions with your code (at current progress), then run and explain it to your TA.**

4 Submission

To finish this data analysis challenge, all the groups are required to submit the following files:

- Task 1: "Assignment_2_Task_1.ipynb", which includes the answers, intermediate steps, explanations and justification to all questions.
- Task 2: The submission for Task 2 consists of two files.
  - **"A2_Task_2_pred_labels.csv"**, where the label prediction on the test set is stored.
    - In your "A2_Task_2_pred_labels.csv", there will be two columns: the first column is the **id** column from A2_test.csv. The second column include predicted labels for semantic roles.
    - The "A2_Task_2_pred_labels.csv" must be reproducible by the assessor with your submitted R/Python code.
  - **"Assignment_2_Task_2.ipynb"**, which is the **R/Python implementation** of your model with documentation. The output of your implementation must include the label prediction, which generates the file **"A2_Task_2_pred_labels.csv"**. If you use R, the R version should be either R 3.64, R 4.2 or above. If you use Python, Python 3 and above is recommended. The use of Jupyter notebook or R Markdown is **required**. All the files, except the provided datasets, which are required for running your implementation must be compressed into a **zip** file, named as "**{student-id}_ass2_impl.zip**". Please note that the unnecessary code must be excluded in your final submission. **The feature design, model evaluation and result analysis should be included in your Jupyter notebook or R Markdown files.** *However, you should keep a copy of the implementation used for the CRF model just in case of the interview.*

4.1 Some Hints

There are some hints that we summarise based on the past submissions made by previous cohorts.

- Avoid using the absolute path in your Jupyter notebook. Instead, use a relative path (.e.g, "./train_data_withlabels.csv") and place the data file in the same folder where your Jupyter notebook is.
- Avoid just plainly showing the results without meaningful interpretation/discussion. For example, if you use any plot, you will need to clearly discuss the information delivered by the plots in the context of the task.
- Choose the appropriate plots or statistics to show the right information.

- While developing the model, make clear, for example, how the optimal parameters are chosen if there is any, etc.
- Be precise in the use of various tools and the corresponding discussion.
- Avoid submitting an extremely long Jupyter notebook or an Rmd file from RStudio, which could result in a lot of redundant information, easily losing the focus of your work.
- Make sure the logic (or the methodology) you used to develop the model is properly documented.
- Write your discussion using the markdown cells and avoid putting it in the code cell as we will use this to gauge your reasoning skills.
- Make use of the discussion forum and consultations to clear any doubts that you may have regarding the tasks you want to accomplish.
- Before making the final submission, you must make sure that your Jupyter Notebook or Rmd file runs without any errors. A possible step is to click "Kernel → Restart & Run All".

4.2 How to submit the files?

The Moodle setup allows you to upload a zip file "**{student-id}_ass2_impl.zip**"', which includes:

- 'Assignment_2_Task_1.ipynb".
- "Assignment_2_Task_2.ipynb".
- "A2_Task_2_pred_labels.csv", where the label prediction on the test set is stored.

While submitting your assignment, you can ignore the Turnitin warning message generated for the ZIP file.

5 Academic integrity

Please be aware of University's policy on academic integrity. Monash University takes academic misconduct[1] very seriously. You can learn from the above materials and understand the principle of how the analysis was done. However, you must finish this assessment with your own work.

References

L. Màrquez, X. Carreras, K. C. Litkowski, and S. Stevenson. Semantic role labeling: an introduction to the special issue, 2008.

[1] https://www.monash.edu/students/study-support/academic-integrity