FIT5212 Data analysis for semi-structured data - S1 2023

Assignment 1

Name: Tsz Yan CHUNG

Student ID: 32973381

Part 1 – Text Classification

1. Introduction

This report aims to present the findings of a text classification task performed on articles on the academic website arXiv.org with three content tags: “Information Theory”, “ComputationalLinguistics”, and “ComputerVision” which can occur in any combination.

The classification algorithm applied is SVC and RNN, both using various input sizes (first 1000 cases and all cases), input (“title” or “abstract” column), and text preprocessing methods (stemming and stopwords removal). This report will go through all 16 combinations of configurations for each content tag and investigate the impact of various parameter settings.

2. Methodology

- Algorithm: Two algorithms were utilized in this text classification task: Statistical classifier Support Vector Classification SVC, and Recurrent Neural Network.
- Input: a model will be trained on either the “Abstracts” or the “Title” label as two different configurations.
- Text preprocessing methods: two methods were applied separately :
  1. Stopwords removal: Remove stopwords from the label field by using spaCy’s stopwords library.
  2. Stemming: Apply PorterStemmer to tokens to reduce all words to their most basic grammar form.

  Text preprocessing methods were applied separately, each model will either be processed in method 1 (P1) or method 2 (P2)

- Data size: All training data is derived from the training data provided: “train.csv”, and models will be trained on:
  1. 90% of “train.csv”, and the remaining 10% were used as a validation dataset.
  2. First 900 records of “train.csv”, and 100 records were applied as a validation dataset

3. Results

SVC: Overall, SVC provides satisfactory results across all configuration settings. The algorithm achieved an average of 92.0%, 93.7%, and 91.5% across all models for the three content tags: “Information Theory”, “ComputationalLinguistics”, and “ComputerVision” respectively. All models achieved more than 83% accuracy, with the lowest Macro F1 score of 0.7758, indicating satisfactory performance across different configurations.

For the label fields, it appeared that models applied on the “abstract” field perform better in most cases, regardless of the tokenized applied. This is likely due to the richer contents in the training and testing dataset, and hence it can provide more information, which leads to more accurate results.

Another parameter that shows a significant impact on models’ performance is training sizes. Models using larger datasets generally perform better in terms of all performance metrics.

Out of the three content tags, it appeared that the model’s prediction on the “Computational Linguistics” tag is the least accurate, especially for those using the small training dataset.

It cannot be observed that different text-preprocessing methods pose a significant impact on models’ performance.

RNN: The accuracy of RNN models is significantly lower than that of SVC models, with a high of 89.7% and a low of 36.9%. Discovery in Marco F1 score demonstrates the same pattern, with a high of 0.8966, a low of 0.3179, and a mean value of 0.5590, indicating that the model is only slightly better than random guessing.

RNN models generally perform better on the “title” target field, which is surprising at first sight. This is likely due to the computation limitation of a student’s laptop, which is unable to handle complex RNN model architecture that tackles long sequences in the label better, such as the “abstract” column.

Similarly, RNN models perform significantly better on larger training datasets in terms of all evaluation metrics, which is a relatively common finding in machine learning tasks. There is no evidence suggesting one text-preprocessing method is better than the other.

RNN models perform poorly across all content tags, with the “ComputerVision” being the most underperforming tag, with the worst model for that tag having a Macro F1 score of 0.3179, indicating poor prediction results.

Among all combinations of configuration, there is a specific setting that shows satisfactory results across all content tags. The model that was applied to the “title” field, using the stopwords removal method performed exceptionally well with an average Macro F1 score of 0.xx, and an accuracy of xxxx

Overall, the simple RNN built in for this text-classification task is not able to capture meaningful information from the training data.

Four precision-recall graphs were generated, corresponding to RNN models trained on “title”, RNN models trained on “abstract”, SVC models trained on “title”, and SVC models trained on “abstract”.

For RNN models trained on the “abstract” column, all models fluctuate at low precision levels (0.2 to 0.6) when the recall value is low, and started to extend horizontally. This indicates that these models are not able to capture valuable information from the training process.

For RNN models trained on the “title” column, some improvements were observed. Models trained on the large dataset are demonstrating a concave line performance, meaning that they are achieving relatively high precision and high recall level, and only started to drop off as the recall level reached over 0.8.

As for the SVC algorithm, most of the models are performing on par, or better than those of RNN. SVC models trained on the “title” field all experience that concaves line shape, with those trained on the small dataset dropping off earlier than those trained on the large dataset. Models like the P2_ALL_T_IT and P1_ALL_T_IT performed even better, the precision level only dropping off when they reach the level where precision and recall trade-off happens.

Those trained on the “abstract” field further outperformed those trained on the “title” field, due to the significantly more information fed to the models allowing them to fully process the context, and hence, resulted in better prediction performance.

4. Conclusion

To conclude, for this task, SVC algorithm performs significantly better in terms of most of the evaluation metrics when compared to those using the RNN algorithm. This is largely due to the computational limitations of the hardware, leading to the RNN not being able to build upon its complexity and process the information. There is not enough evidence to show that one text-preprocessing method is better than the other, while the “abstract” field provides more information to the SVC models, resulting in overall better performance for models training on it. In contrast, RNN models generally perform better when trained on “title” field, due to their inability to capture complex and long information from the other field.

Part 2 – Topic Modelling

1. Introduction

This part of the report aims to tackle an unsupervised topic modeling task using the Latent Dirichlet Allocation (LDA) algorithm. The objective is to evaluate the impact of different parameters set up in LDA on the groupings among articles. Combinations of parameters include two different text-preprocessing methods: stopwords removal and stemming; as well as training model on two different sizes of the dataset, 1000 articles and 20,000 articles. This task will include bi-grams, and it will train and test on the "abstract" column of the training and testing dataset respectively.

2. Methodology

The training dataset will be pre-processed with either one of the following methods:

- Stopwords removal (P1): remove stopwords from articles' abstract using spaCy's stopwords library.
- Stemming (P2): return tokens within articles to their most basic grammar form.

The dataset will then be processed by the LDA model, looping through different values of K, where K represents the total number of topics in the final classification result.

Models will be named according to their configuration: P1_1000, P2_1000, P1_ALL, P2_ALL

3. Results

In general, the two LDA models trained on the larger dataset perform significantly better than those trained on the small dataset. Terms in the better performing models have a high frequency within the selected topic compared to the overall frequency in the whole corpus, indicating the modes' ability to capture distinct tokens among groupings, and hence, groupings are relatively intuitive and comprehensible.

P2_ALL: In P2_ALL, heavily clustered topics 2, 3, and 8 involved the following top keywords respectively: "object, video, method, dataset, 3d", "imag, method, learn, model, train, text, sentenc", and "model, language, train, data, the label". While topic 2 seems to deviate more from the other two topics, the other two groupings demonstrated similarities and relation to various semi-structured data concepts.

Group 8 has a high proportion of distinct words like "word, language, English", indicating it could be related to the ComputationalLinguistics tag; while topic 3 potentially focuses more on ComputerVision, with distinctive words like "imag, resolution, color" have high relative frequency to the corpus.

As for the groups on the opposite side of the spectrum, topics 5 and 10 endowed words that possess different characteristics, such as "problem, function, user, the system", with distinct words like "linear, 'entropi, algebra, matrix" and "energi, interfere, relay, antenna" respectively. By looking at their unique words, it seems like group 2 focuses more on mathematical content, while Group 10 is more about the computer architecture side of content.

Group 4 was isolated from all other groups. It contains the following top words "segment, imag, result", with distinct words like "clinic, medic, medic_imag, diagnosi". It potentially is an information technology-related article group, however, from the context, it is very likely to be focusing on the healthcare sector and related technology. This could be the reason that group 4 and group 1 are in the diagonal of top right and bottom left, with topic 1 having high amount of distinct words like “code, decod, messag”, which

P2_1000: Compared the evaluated result in the P2_ALL model to the P2_1000 model, which applied the same text-preprocessing method but with a significantly smaller dataset. The topic groupings appeared to be more ambiguous.

In P2_1000’s graph, group 5 was isolated with the distinctive word “image, attribute quality”. While “imag” also appeared quite frequently in Group 2 and Group 8, which group 5 is relatively distanced to.

Groups among the main cluster on the bottom left failed to demonstrate distinctive characteristics to separate one from the others. The top keywords in Group 8 are “object, method, model, learn”, which group 3 shares a lot of the common words. This provides a general idea that they could be machine learning-related topics, but there is not enough evidence to distinguish one from the other.

Group 6 in P2_1000 is an exception that is able to differentiate itself properly, with unique top words such as “word, language, natural language”, indicating that it is likely to be a group of NLP-related articles.

P1 models: P1_1000 suffered from similar issues to P2_1000. Groups in P1_1000 are generally incomprehensible, with low relative term frequency to the corpus, and unclear terms definition like “n, m, k, non” due to the application of different text preprocessing methods. The model is not able to obtain valuable information across articles.

While both were trained on the same dataset, P1_ALL has more scattered distribution compared to P2_ALL. Terms across all groups demonstrate moderate relative frequency to the corpus, potentially indicating poor performance in groupings. Incomprehensible terms were observed in some groups.

In group 7, incomprehensible terms like “n, k, o, p, linear, algorithm” were the most relevant terms. Speculation on this observation could be the group is heavily related to mathematics topics, where those relevant terms are all math variables. However, this hypothesis could not be validated without further context.

At lower relevance metrics, groupings are better defined in P1_1000. For example, terms like “video, face, temporal, tracking, detection” in group 4 indicate the group’s focus on visual footage capturing; “Language, text, words, languages” in group 2 shows that the group is likely to be related to NLP articles; “Computational, neural, network, memory, efficient” in group 10 indicates the group could be focusing on deep learning topics.

4. Conclusion

To conclude, this task discovered that LDA training on large dataset generally results in better performance, regardless of the text-preprocessing method applied. Top relevant terms at lower relevance metric values for groups in these models are highly comprehensible and distinctive, indicating the models’ ability to capture insight and information from training data.

In contrast, models trained with a small dataset are more likely to suffer from ambiguous groupings. Top relevant terms across relevance metrics failed to deliver the group’s focus.

Text-preprocessing methods pose no significant impact on models’ performance, while the ones applied the stopwords removal method may result in terms with duplicate meanings, due to prefix, suffix, or other grammatical rules issues.