FIT5212-DATA ANALYSIS FOR SEMI-STRUCTURED DATA

(ASSIGNMENT - 1)

TEXT CLASSIFICATION & TOPIC MODELLING

Keerthana
Muralitharan
kmur0015@student.monash.edu
Student ID: 30159474

Contents

1. Text Classification

Text Classification is to be done on the content gathered from the popular academic website arXiv.org for articles tagged under computer science content which are also present under mathematics and physics categories.We are provided with a training set whose data is from 1990-2014 and the testing set data consists of data from 2015 and a bit from 2016 as well.

InfoTheory, CompVis and Math are the three classes which are occurring in the text in any combination whether present in all three classes or any two,one or not even in any of the three classes.Our assignment task is to build three text classifiers that predict these three classes only by using the Abstract field using the modelling statistical text classifiers and neural network classifiers and decide the efficiency among the all 6 models.

1. 1 Pre-processing & Understanding the code

Section in code – `Text Processing & Analysis`

The data for the train data and test data read using ***pandas*** dataframe using ***read_csv*** function from the train data and test data file. The dimensions of the train and test data is ***54731 and 19679 rows*** and ***8 columns*** respectively

```
The dimesions of the training dataframe:  (54731, 8)
The dimesions of the test dataframe:  (19679, 8)
```

Fig 1.dimensions

The dataframes are then checked for null values. The train file did not have any null values while the ***test file had one row of null value***.

```
[ ] # check for the NULL values in the training dataframe
    df_train.isna().sum()

ID            0
URL           0
Date          0
Title         0
InfoTheory    0
CompVis       0
Math          0
Abstract      0
dtype: int64
```

```
# check for Null values in the test dataframe
df_test.isna().sum()

ID            0
URL           0
Date          0
Title         0
InfoTheory    1
CompVis       1
Math          1
Abstract      1
dtype: int64
```

Fig 2.Checking Null Values

These null values are removed and then processed before building text classifier models.

2.Neural Network Method:

Section in code – `Part 1: Neural Network Method`

In this Method the three text classifiers for *InfoThoery,CompVis,Math* classes are implemented by building a simple Recurrent Neural Networks using TorchText and Pytorch. Then confusion matrix for each of text classifier models is being analysed.

2.1 Preprocessing of the data

Sections in code – `2.1 Preprocessing of the data`

The data is being pre-processed to build the models by initializing the Output and Input from the dataset using TorchText to set the ***Text***(Abstract-Input) and ***Label***(InfoTheory/CompVis/Math-Output) fields.

- Since the test dataset has a *null* value, we cannot proceed to build the text classifiers so the empty lines are being replaced by special characters(Null) for the *Abstract* field and *(0) for* all the other 3 output classes *{InfoTheory,CompVis,Math}*
- The Input data from the *Abstract* is being tokenized using the ***spacy*** tokenizer

- The stopwords and punctuation are also removed from the tokens
- The tokens are all converted to lowercase
- duplicate tokens are removed and a unique set of tokens is only sent for building the dataset

The Seed value is set as **200** like torch.manual_seed(seed), and it will set the seed of the random number generator to a fixed value,so that the results will be reproducible.

The TabularDataset class is used to read the input data, since it handles to read the data present in csv format.It process the data using the ***Fields*** we have previously defined.

The RNN model with different layers of Input, Embedding, Hidden and the Output layer dimensions is set up using a class function.

2.2 Extracting the fields and setting up train functions

Sections in code – `Info Theory Model, Comp Vis Model, Math Model`

The input for these 3 text classifier is obtained from the preprocessed tokens from the *Tabular Dataset*.The total number of input lines in the three text classifiers is listed in the figure below.

```
print(f'Number of Info_Theory training lines: {len(train_data_Info)}')
print(f'Number of  Comp_Vis training lines: {len(train_data_Comp)}')
print(f'Number of Math training lines: {len(train_data_math)}')
print(f'Number of testing lines: {len(test_data_Info)}')
```

```
Number of Info_Theory training lines: 54731
Number of  Comp_Vis training lines: 54731
Number of Math training lines: 54731
Number of testing lines: 19679
```

Fig 3. Data size

After tokenization, we could see that there are a total of ***91885*** tokens in the input dataset.

```
TEXT.build_vocab(train_data_Info,)
print("The text length of the vocabularis : ",len(TEXT.vocab))
```

```
The text length of the vocabularis :  91885
```

Fig 4.Vocabulary length

2.3 Building the model

Sections in code – `Info Theory Model, Comp Vis Model, Math Model`

The dimension for the different layers is being set-up in the RNN and the optimiser and the criterion for the model is setup.The Batch Function which is being used to create the iterators out of the train and test data in the model respective to the model being built is defined with the **batch size of 32.** The **vocab size** will be **5002** because of the ***<unk> and <pad>*** token.

```
print(f"Unique tokens in TEXT vocabulary: {len(TEXT.vocab)}")
```

```
Unique tokens in TEXT vocabulary: 5002
```

Fig 5.Unique Tokens

2.4 Model Analysis

Sections in code – `Confusion Matrix of the models`

The initial run for the model was done with the dimensions of the Input layer to be **total vocabulary length**, Embedding Dimensions as **100**, Hidden dimensions as **256** and Output dimensions as 1 with batch size of 32 and learning rate 1e-3, but it did not yield the best accuracy.

Based on the results got for the first iteration, by altering the **Input dimensions**, **Batch sizes, Learning Rates,criterion(loss function)**-*binary cross entropy with logits* and **the criterion** -*is a function which is used to calculate the error* of the model.The trained model iterates and processes with *one batch* at a given time.Based on all the factors we could get the most accurate model which was tested and the final model with the below features and the confusion matrix which was yielded by these dimensions is being listed below when compared with the other models. The confusion matrices for other parameters is affixed in the *Appendix.*

Table 1 : Confusion Matrix for RNN

Input layer = 5000 Embedding Layer = 100, Hidden layer = 256 Output layer = 1 batch size of 32 learning rate 1e-3.	Full confusion matrix for method on InfoTheory: [[15676 387] [3534 82]] Full confusion matrix for method on CompVis: [[17218 309] [2131 21]] Full confusion matrix for method on Math: [[13574 175] [5839 91]]

2. Machine Learning Method:

Using the traditional machine learning algorithms like Logistic Regression,Linear SVC,Random Forest,Bernoulli's Naïve Bayes we have developed three best accurate text classifiers for Infotheory,Compvis and Math classes

2.1 Pre-processing

Sections in code – `Pre-processing`

While preprocessing the data we have to follow the first step where the empty elements have been identified and it is **dropped from the test dataset** using a function from Pandas library function called *dropna()*. The second step involves the data being converted to a list for processing where one list for Input text column is denoting to the Abstract Column and three other lists for each label for the output classes which are InfoTheory,CompVis,Math columns from the datasets.The length of the training dataset lists is ***54731.***

2.2 Tokenization & Lemmatization

Sections in code – `Tokenization & Lemmatization`

The TF_IDF vectorizer is used to convert the input text into vectors. This vectorizer is then used to transform the input text of the training and testing data into vectors with parameters set with minimum df, maximum df, token pattern and remove the stopwords and also the punctuations is filtered.

The tokenizer which is used here is **Regex tokenizer** that filters other characters and keeps only the words and whitespaces. The train data is being fit with this transformation and they are converted to features. The total length of the pre-processed vectorizer features is **20335** which is then used to build the models to find the best classifier**.**

2.3 Model Building

Sections in code – `Model Evaluation & Model Building-InfoTheory, CompVis, Math Model`

Four different traditional models are built and tested using Cross-validation for 5 folds to identify the best fit model for the classification task. Methods selected are **Logistic Regression, Bernoulli's Naïve Bayes, Linear SVC** and **Random Forest.** These models are built for all of the three classes and one of the model which has higher accuracy is selected as the best Classifier model. A boxplot with jitter points is plotted across the accuracy and the most accurate model will be chosen.

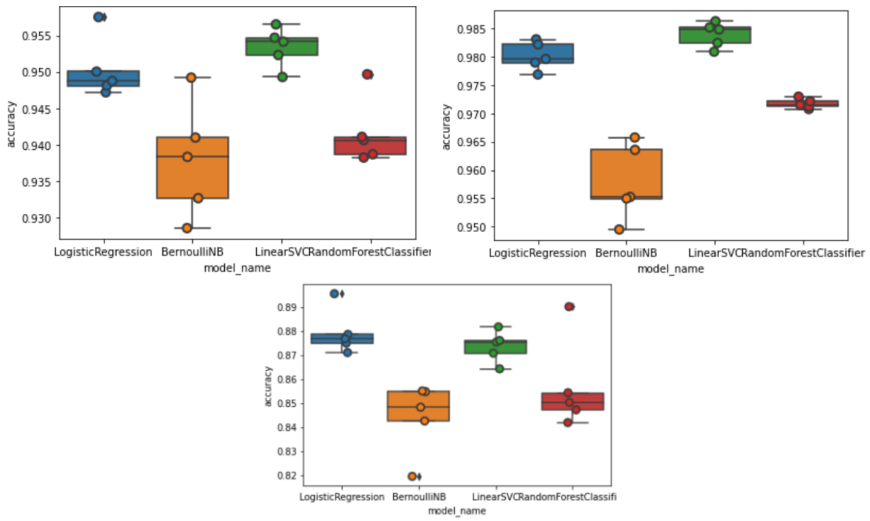

Fig 6 – Box plot for all 3 classes

From the above graphs for the three classes amongst these Four algorithms, **Linear SVC** has the high accuracy, therefore it is best method with the highest accuracy in the trainset with least variations among five-fold cross- validation . The final three text classifiers is built using the Linear SVC method and confusion matrix is being generated.

2.4 Results

Sections in code – `Results`

Based on the all the other techniques we could infer that the Linear SVC is the most accurate model with less distortion, therefore it is used to build the text classifiers for all the three classes with the final confusion matrices:

Table 2. Confusion Matrix for Linear SVC

Linear SVC C=1 Kernel : Linear	Confusion Matrix for InfoTheory : [[15777 285] [665 2951]] Confusion Matrix for CompVis : [[17426 100] [503 1649]] Confusion Matrix for Math : [[12731 1017] [1482 4448]]

3 Inferences – Text Classification

Based on the above analysis, with two different techniques one being RNN and Traditional Machine learning techniques using different algorithms in each case, below are the inference and results comparing the two methods.

3.1 Neural Network Method

Recurrent Neural Network is used in building the text classifier for the three classes. This method have few parameters that can be changed to evaluate its performances such as Batch sizes, layer dimensions, input layer size, learning rate. When the batch size was reduced to lower degrees of 2 such as 16,8, 4 or 2 the result was an **overfitted** model that the error on train dataset was negligible but increased with the test dataset. Increasing the batch size to 64,128 resulted in an underfitted model as both the test and train errors were high. After analysing these results, the batch size was fixed to 32. By changing the input layer too we could also suggests us with a good model which means The lesser input layer considered, the better the confusion matrix.

3.2 Machine Learning Method

Using the traditional machine learning methods, only unigrams were considered in analysis as this itself resulted in a very large number . After iterating over the algorithms, Linear SVC proved to be the most accurate one among the considered approaches.

3.3 Comparing Neural Network with ML approach

Below table summarises the results of both approaches.

Table 3.Comparison of Confusion Matrix

RNN	Linear SVC
Full confusion matrix InfoTheory: [[15676 387] [3534 82]] Full confusion matrix CompVis: [[17218 309] [2131 21]] Full confusion matrix Math: [[13574 175] [5839 91]]	Confusion Matrix for InfoTheory : [[15777 285] [665 2951]] Confusion Matrix for CompVis : [[17426 100] [503 1649]] Confusion Matrix for Math : [[12731 1017] [1482 4448]]

Table 4. Metrics Comparison

Parameter for evaluation	Classes	RNN	Linear SVC
Accuracy (TP + TN) / Total	InfoTheory	79.24%	95.17%
	CompVis	87.60%	96.93%
	Math	69.44%	87.30%
Precision (TP/ (TP + FP))	InfoTheory	97.59%	98.22%
	CompVis	98.23%	99.42%
	Math	98.72%	92.60%
Sensitivity (Recall)(TP/TP+FN)	InfoTheory	81.60%	95.95%
	CompVis	88.98%	97.19%
	Math	69.92%	89.57%
F1 score 2TP / (2TP + FP + FN)	InfoTheory	88.88%	97.07%
	CompVis	93.38%	98.29%
	Math	81.86%	91.06%

Based on **accuracy**, Linear SVC clearly stands out to be the best method. However, RNN would play a key role when the size of the dataset is huge and adding further layers would improve the accuracy of the model.
Based on **Precision**, Except the Math model, linear SVC does best in identifying the Precision out of all the cases
Based on **Recall** , Linear SVC is high in reducing the number of False negatives
Based on **F1 score, Linear** SVC is higher for all three models
Overall, for this dataset, ML algorithms (Linear SVC) performs better than the RNN(Recurrent neural networks). However, each task would have different algorithms performing better than one another.

4.Topic Modelling

Based on the content being gathered from news sites, containing the term "Monash University", or at tagged with the label "Monash University" by an external annotator. Topic models are to be built on the content to analyse the range of topics present in the articles regarding Monash University.

4.1 Text preprocessing

Sections in code – `preprocessing of data`

The data was checked for null values and then processed with the regular text processing techniques. In this step, the pre-processing techniques like as tokenisation, lemmatisation, stop words, punctuations,bigrams and trigrams are removed. The text is converted to corpus -> bag of words using the Dictionary and then the corpus is used to build the LDA model.The final result of unique tokens and number of documents is taken.

```
Number of unique tokens: 1319
Number of documents: 366
```

Fig 7.Document and token length

4.2 LDA Model

Sections in code – `LDA MODEL`

The Latent Dirichlet Allocation model is developed by using the *gensim* library. The parameters needed for the model are setup such as Number of Topics from the corpus,chunk_size,iterations count and its passes.

Once the model is built, using the **perplexity** and **coherence** is used to determine the **num_topics** to be used. The model is iterated for topic numbers from 2 to 15 and coherence is calculated for each of the model.Two LDA models are created based on the top 2 coherence score denoting NUM_TOPIC values.

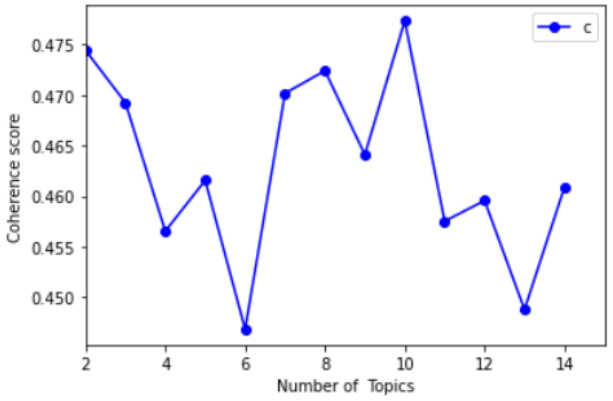

Fig 8.Coherence Plot for Number of Topics

4.3 Topic Analysis

Sections in code – `Topic Analysis`

The two LDA models analysis is built for the BODY column. Based on the coherence plot, top two Num_topics is chosen to build the model. Based on the plot we could say the top two coherence score are for number of topics **2** and **10.**The perplexity(measure to find the better model-lower perplexity is better) for two models are

-6.57 and -6.45. The coherence score(Higher value gives better results) for the two models is 0.48 and 0.46.

An LDA model is built for 2 and 10 topics to be found out from the context.

```
[(0,
  '0.018*"coronavirus" + 0.018*"february" + 0.016*"january" + 0.016*"australia" + 0.013*"people" + 0.013*"virus" + 0.012*"ch
ina" + 0.009*"case" + 0.009*"australian" + 0.008*"health"'),
 (1,
  '0.010*"university" + 0.010*"australia" + 0.008*"also" + 0.007*"area" + 0.007*"people" + 0.007*"year" + 0.007*"say" + 0.00
6*"one" + 0.006*"fire" + 0.006*"could"')]
```

Fig 9.Top_topics in model 1

Based on the above figure using the topics are interpreted with an average topic coherence of -0.4158

- Topic -1: The topic is based on corona-virus outbreak affecting people in china in January and February
- Topic -2: This article is about the bush fires affecting people and universities

The relevance in the two topics is chosen from the contents depicted in the InterTopic Distance Map.

For Number of topics as **10** , below is the figure consisting of the top-words to help us find the relevant article.

```
[(0,
  '0.038*"virus" + 0.025*"people" + 0.022*"coronavirus" + 0.018*"case" + 0.013*"symptom" + 0.013*"spread" + 0.012*"china" +
0.010*"outbreak" + 0.009*"health" + 0.009*"wuhan"'),
 (1,
  '0.017*"university" + 0.015*"say" + 0.012*"time" + 0.011*"one" + 0.010*"also" + 0.010*"could" + 0.009*"year" + 0.008*"wor
k" + 0.008*"like" + 0.008*"research"'),
 (2,
  '0.032*"fire" + 0.024*"australia" + 0.015*"year" + 0.013*"bushfires" + 0.013*"south" + 0.010*"climate" + 0.010*"people" +
0.008*"million" + 0.008*"country" + 0.008*"bushfire"'),
 (3,
  '0.037*"patient" + 0.022*"study" + 0.017*"mass" + 0.015*"per" + 0.015*"disease" + 0.014*"per_cent" + 0.014*"cent" + 0.014
*"level" + 0.014*"test" + 0.013*"also"'),
 (4,
  '0.033*"china" + 0.023*"case" + 0.021*"coronavirus" + 0.015*"death" + 0.014*"chinese" + 0.013*"wuhan" + 0.011*"number" +
0.010*"reuters" + 0.010*"sars" + 0.009*"infection"'),
 (5,
  '0.028*"area" + 0.014*"cell" + 0.013*"australian" + 0.012*"australia" + 0.012*"smoke" + 0.012*"air" + 0.011*"data" + 0.009
*"change" + 0.008*"specie" + 0.008*"climate"'),
 (6,
  '0.044*"february" + 0.032*"january" + 0.019*"australia" + 0.019*"coronavirus" + 0.012*"mask" + 0.011*"south" + 0.011*"aust
ralian" + 0.011*"people" + 0.009*"february_february" + 0.008*"january_january"'),
 (7,
  '0.017*"woman" + 0.015*"school" + 0.015*"people" + 0.012*"university" + 0.009*"student" + 0.009*"home" + 0.008*"coronaviru
s" + 0.008*"social" + 0.008*"health" + 0.008*"business"'),
 (8,
  '0.029*"australia" + 0.024*"february" + 0.017*"student" + 0.016*"january" + 0.016*"australian" + 0.016*"coronavirus" + 0.0
14*"china" + 0.012*"ban" + 0.011*"travel" + 0.009*"government"'),
 (9,
  '0.016*"people" + 0.014*"china" + 0.014*"coronavirus" + 0.014*"wuhan" + 0.013*"australia" + 0.012*"january" + 0.012*"fligh
t" + 0.011*"virus" + 0.011*"health" + 0.010*"passenger"')]
```

Fig 10.Top topics in model 2

Based on the above figure using the topics are interpreted with an average topic coherence of "-1.7484"

- Topic -1: Corona virus outbreak in china
- Topic -2: Article about time required in research and study
- Topic -3: This topic tells about bush fire affecting climatic change
- Topic -4: This is related to disease percent on patients
- Topic -5: Corona virus in comparison with SARS
- Topic -6: Amount of smoke in australia due to bush fires
- Topic -7 : Sales of face masks in australia during january and february
- Topic -8: Effect of lockdown on women and students in different sectors
- Topic -9: Article about travel ban due to corona virus
- Topic -10: Passengers health in january who travelled to australia from china

4.4 Inferences

Sections in code – `Inferences`

All the topics are not related to *Monash University.* The topics are easily comprehensible and it is easily relatable to the article based on the top topic words. These topic words which has university or monash suggests us that those topics are either directly or indirectly to the Monash University.Based on our 2 models, Monash University is related to Bushfire,climate studies and coronavirus'travel ban affecting the students.

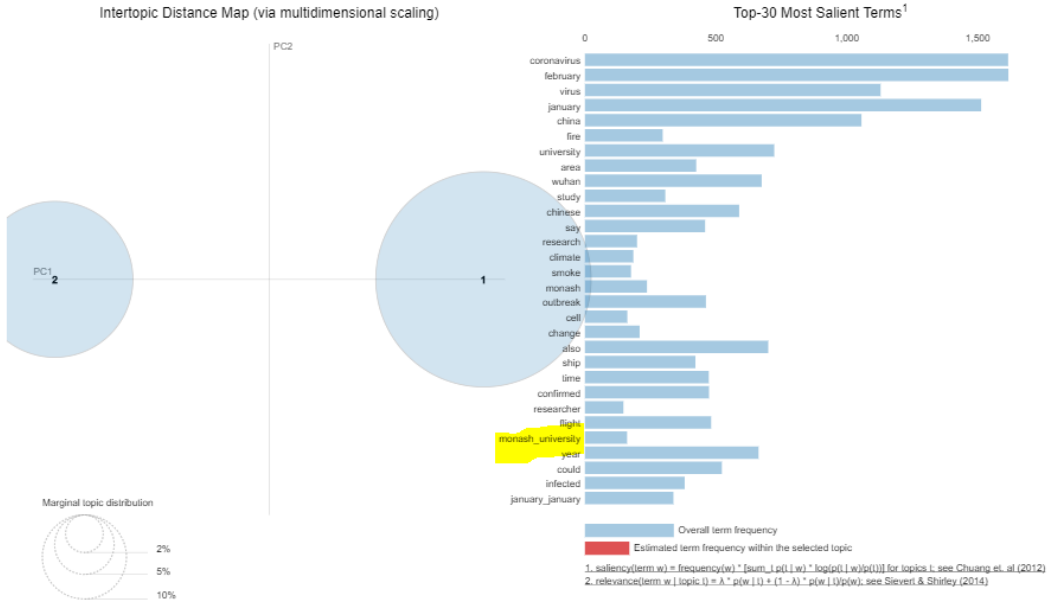

Fig 11. Visualisation for model 1

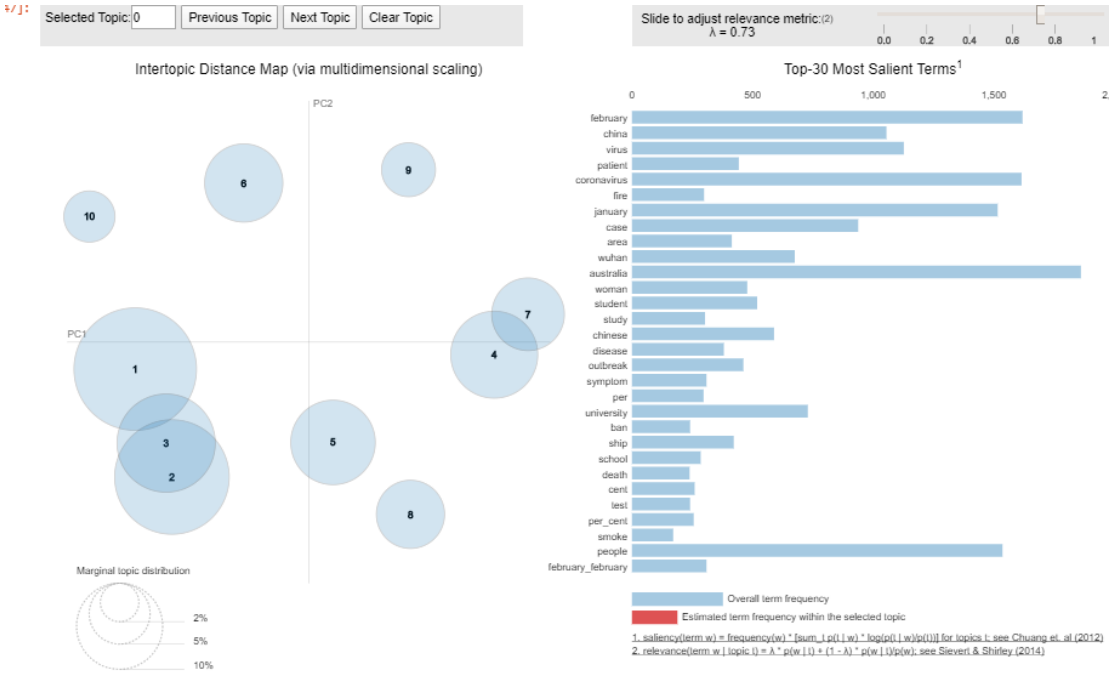

Fig 12. Visualisation for model 2

4.5 Document Mapping

Sections in code – `Document Mapping`

Based on the dominant topics we have considered two topics as examples one for **topic 1 and topic 3,** to support our inferences.

In Topic 1, that relates to the **work-rights and job**, article from the news https://www.dailymail.co.uk/news/article-8009293/Coronavirus-crisis-Australian-expatriates-China-disease-changed-lives.html dated 16/02/2020 gives a view on the employment sector of Australia being affected by Coronavirus, where the article quotes

'Australian expatriates in China say almost every aspect of their lives now revolves around the coronavirus outbreak.So far, the disease has killed over 1,600 people and infected about 70,000 people globally, including 15 people in Australia. The epicentre of the virus, Wuhan, remains almost entirely cut off from the outside world as authorities try to halt its spread.'

Related to **topic 3 on study made by Australian students on climate change** , as quoted in article http://global.chinadaily.com.cn/a/202001/08/WS5e153594a310cf3e35583232.html dated 08/01/2020 states the damage caused by the bushfires and also its adverse future climatic effects

'David Holmes, director of the Monash Climate Change Communication Research Hub at Melbourne's Monash University, said, "This isn't just about cyclical drought, which politicians like to use to normalize the current situation."In the last 15 years, Australia saw eight of its 10 warmest years on record. Climate scientists tell us that, with climate change, weather systems increasingly move poleward. This means that storm tracks that once brought moisture from the Southern Ocean right up the east coast are not reaching as far, and inland forested areas are becoming much drier," he said..'

Inference: Based on the above two sample links taken, we can identify that the in topic related to work-rights and job how it is affecting the students, employers based on corona virus and the next topic about climate change affected by bushfires being quoted by a director of Monash Researchhub at the university

4.6 Conclusion:

Sections in code – `Output`

These two models give us a wide range of information generally on corona virus and bushfires.All the topics does not involve monash university,but it is linked based on the location "Monash Hospital" or any university being referenced. Since LDA is a probabilistic model and results in coherent topics, it is easy to interpret the topics.The shortcomings of LDA is that we must know the parameters before hand developing the model and it assumes a bag of words as top topic words, also the sentence structure is not modelled which can lead to misinterpretation of the topics.Thus LDA models have both advantages and disadvantages as it is not constant everytime since it is a probabilistic model.

5. References

RNN

Tutorial materials

https://discuss.pytorch.org/t/what-is-manual-seed/5939/10
https://www.analyticsvidhya.com/blog/2020/01/first-text-classification-in-pytorch/
https://discuss.pytorch.org/t/type-object-tabulardataset-has-no-attribute/58590
https://torchtext.readthedocs.io/en/latest/data.html
https://github.com/pytorch/text/issues/430

Statistical modelling :

Tutorial materials

https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f
https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

Topic modelling:

Tutorial materials

https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21
https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0
https://datascienceplus.com/evaluation-of-topic-modeling-topic-coherence/
https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3
http://qpleple.com/perplexity-to-evaluate-topic-models/
https://stats.stackexchange.com/questions/18167/how-to-calculate-perplexity-of-a-holdout-with-latent-dirichlet-a
https://www.machinelearningplus.com/nlp/topic-modeling-visualization-how-to-present-results-lda-models/
https://cfss.uchicago.edu/notes/topic-modeling/
https://tedboy.github.io/nlps/generated/generated/gensim.models.LdaModel.log_perplexity.html

6 Appendix

Table 5. Confusion Matrices for different parameters -RNN

Input layer = total vocab length,
Embedding Layer = 100,
Hidden layer = 256
Output layer = 1
 	
batch size of 16
learning rate 1e-3.
	`Full confusion matrix for method on InfoTheory:`
	`[[12850  3213]`
	`[ 2860  756]]`
	`Full confusion matrix for method on CompVis:`
	`[[16784  743]`
	`[ 2032  120]]`
	`Full confusion matrix for method on Math:`
	`[[13446  303]`
	`[ 5784  146]]`

Input layer = total vocab length,
Embedding Layer = 100,
Hidden layer = 256
Output layer = 1
	`Full confusion matrix for method on InfoTheory:`
	`[[11674  4389]`
	`[ 2603  1013]]`
	`Full confusion matrix for method on CompVis:`
	`[[14323  3204]`

batch size of 32 learning rate 1e-3.	[1767 385]] Full confusion matrix for method on Math: [[11340 2409] [4914 1016]]
Input layer = total vocab length, Embedding Layer = 100, Hidden layer = 256 Output layer = 1 batch size of 64 learning rate 1e-3.	Full confusion matrix for method on InfoTheory: [[13886 2177] [3139 477]] Full confusion matrix for method on CompVis: [[15386 2141] [1898 254]] Full confusion matrix for method on Math: [[12039 1710] [5244 686]]
Input layer = 5000 Embedding Layer = 100, Hidden layer = 256 Output layer = 1 batch size of 16 learning rate 1e-3.	Full confusion matrix for method on InfoTheory: [[12819 3244] [2803 813]] Full confusion matrix for method on CompVis: [[16661 866] [2059 93]] Full confusion matrix for method on Math: [[13349 400] [5722 208]]
Input layer = 10000 Embedding Layer = 100, Hidden layer = 256 Output layer = 1 batch size of 16 learning rate 1e-3.	Full confusion matrix for method on InfoTheory: [[14429 1634] [3148 468]] Full confusion matrix for method on CompVis: [[17085 442] [2106 46]] Full confusion matrix for method on Math: [[13605 144] [5849 81]]
Input layer = 10000 Embedding Layer = 100, Hidden layer = 256 Output layer = 1 batch size of 64 learning rate 1e-3.	Full confusion matrix for method on InfoTheory: [[10471 5592] [2356 1260]] Full confusion matrix for method on CompVis: [[12452 5075] [1552 600]] Full confusion matrix for method on Math: [[9140 4609] [3970 1960]]
Input layer = 10000 Embedding Layer = 100, Hidden layer = 256 Output layer = 1 batch size of 32 learning rate 1e-3.	Full confusion matrix for method on InfoTheory: [[12177 3886] [2672 944]] Full confusion matrix for method on CompVis: [[14780 2747] [1781 371]] Full confusion matrix for method on Math: [[11905 1844] [5033 897]]
Input layer = 5000 Embedding Layer = 100, Hidden layer = 256 Output layer = 1 batch size of 64 learning rate 1e-3.	Full confusion matrix for method on InfoTheory: [[10702 5361] [2413 1203]] Full confusion matrix for method on CompVis: [[12595 4932] [1579 573]] Full confusion matrix for method on Math: [[9420 4329] [4081 1849]]