

TEXT ANALYTICS USING RAPID MINER

By

Aaron Ward

Supervisor(s):

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
B.SC IN COMPUTING AND INFORMATION TECHNOLOGY
AT
INSTITUTE OF TECHNOLOGY BLANCHARDSTOWN
DUBLIN, IRELAND
2017

Declaration

I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, except where otherwise stated. I/We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references.

I/We understand that plagiarism, collusion, and copying are grave and serious offences and accept the penalties that would be imposed should I/we engage in plagiarism, collusion or copying. I acknowledge that copying someone elses assignment, or part of it, is wrong, and that submitting identical work to others constitutes a form of plagiarism. I/We have read and understood the colleges plagiarism policy 3AS08.

This material, or any part of it, has not been previously submitted for assessment for an academic purpose at this or any other academic institution. I have not allowed anyone to copy my work with the intention of passing it off as their own work.

Dated: 2017

Author:

Aaron Ward

Text Analytics Using Rapid Miner

Business Understanding

This following sections shall describe the business objectives, the mining objectives, a analysis and plan for the proposed project.

Business Objective

The objective of this project is to mine unstructured data using Rapid Miner.

Data Mining objective

- The main objective is to mine articles about Machine Learning, Deep Learning and Robotics.
- Web crawling will be implemented to find related text on the topics.
- Preprocessing steps shall be put in place in order to produce the most predictive words for modeling.
- Multiple clustering and classification models will be used to identify the texts.

Project analysis

In terms of a cost benefit analysis, this project deems relatively efficient due to the benefits out weighing the costs. The proposed assignment poses minimal risk as only a few events may delay the production of the project, such as unavailable texts and misclassification,

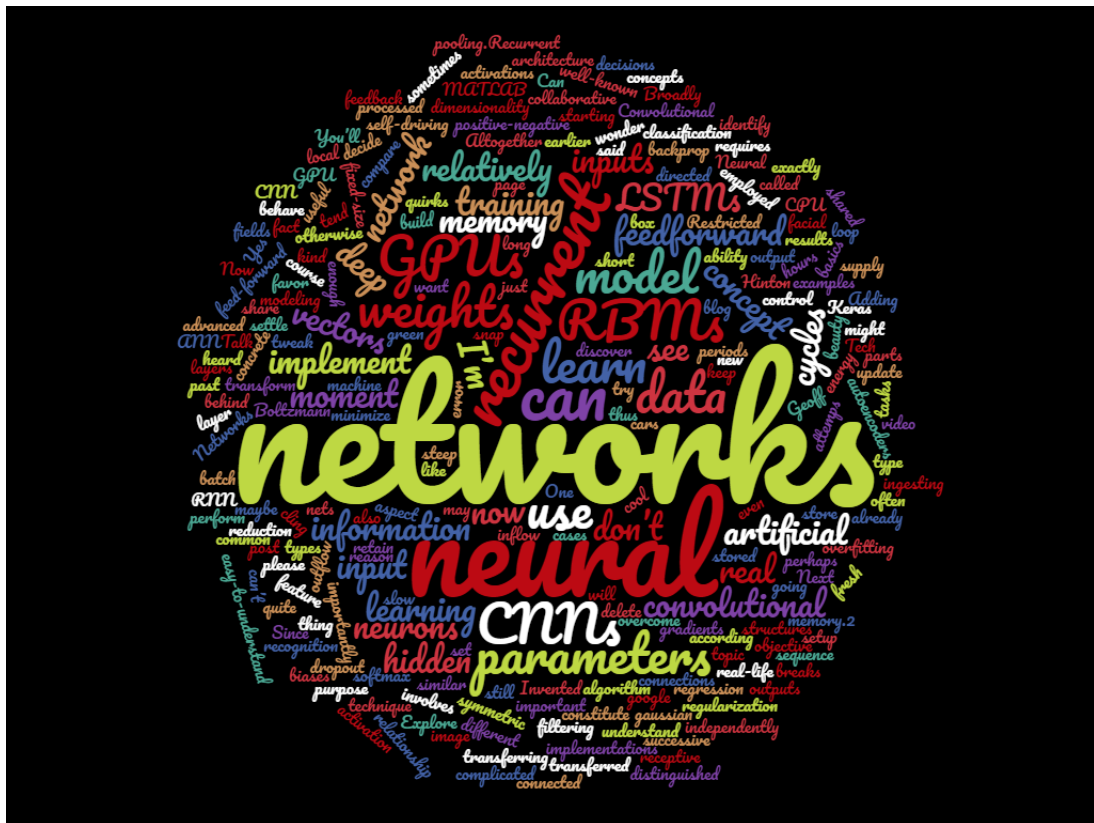
which are very unlikely. The resources required for this project include: The RapidMiner software and all its operators related to the data mining objective, articles from online resources, 13 documents based on 3 categories and an online word cloud service for visualisation of predictive words or phrases.

Project plan

There are a number of steps taken in relation to this project.

- Initially, Three categories are decided on. Two topics should be relatively similar, and the last one will be unrelated. In this case, they are based on Deep Learning, Machine Learning and robotics.
- Thirteen texts are sourced online for each category. Ten shall be used for training datasets and three shall be used for testing. Web crawling will be implemented using RapidMiner for three of those texts. The steps taken will be documented
- These source articles will be downloaded into their respective folders based on their class label.
- A word cloud will be created to visualise the frequently occurring terms.
- Preprocessing steps shall be experimented with to compare stemmers, determine which words are predominantly more predictive, apply pruning and comparing accuracy based on different vectors and documenting the results.
- Apply two algorithms for clustering and classification and discuss the accuracy of them.
- Lastly, an evaluation of the project will be performed, which will assess the overall result in relation with the business objectives.

Figure 1: Word cloud for the machine learning category



Lastly, we see the terms generated for the robotics category. Evidently, these words are not related to machine learning or deep learning as they share very few words to the documents in the other two categories. The prominent words found here are **robots**, **humans** or **humanoid** and **robotics**

Data Preparation

The initial step in the preprocessing phase consists of making a stop word list based on the word clouds. This is followed by the creation of a synonyms list to group relating terms under the one word. In terms of Rapid Miner processes a number of steps are taken in order to prepare the data for mining.

Generating a Bag of Words

Process Documents from Files: this operator generates word vectors based on text from multiple files, in this case, the texts that have been collected initially. Within the *Process Documents from Files* operator, a number of other techniques are used to cleanse the data.

Tokenizing: to split the text into tokens.

Transform cases: which changes everything to the letter case you specify.

Filter Stopwords (English): This operator filters out all the commonly occurring words in the English language which are not useful for text analysis.

Filter Tokens by Length: Upon inspection of the word vector that was generated, it was noted that some characters were not filtered by the *Filter Stopwords (by length)* operator as there were still words like "a" and "on" still being seen, therefore this operator was utilized in order to remove these terms. A range between 3 and 25 characters was set for this.

After these initial steps, the documents were processed and a number of 690 attributes were generated before pruning. Absolute pruning was implemented. The parameters set for this were between 3 and 15. Many of the attributes were lost as a result of this, with a number of just 45 attributes. In response to this, the parameters were adjusted to be between 2 and 15. This resulted in 155 attributes.

The **Filter Stopwords (Dictionary)** was applied to remove out the words that were unrelated or unimportant in the word clouds. After the stop word filtering, 120 attributes remained. Example of a few of these stop words can be seen in Figure .

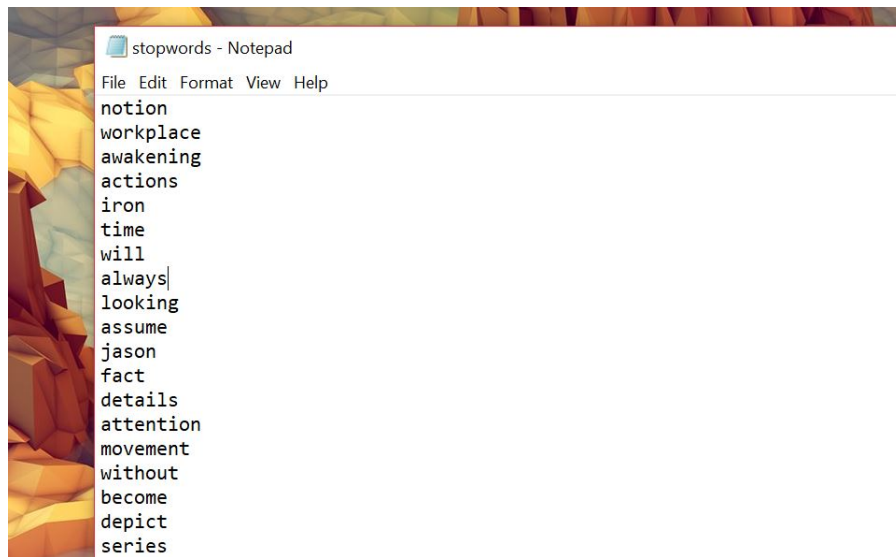


Figure 4: Some of the Dictionary Stop Words

Following inspection of the terms generated, new stop words were added to the list, bringing it down to 107 attributes.

Stemming

In order to derive words down to their root stem, two algorithms were utilized, Lovins Stemmer and Port Stemmer. 122 attributes were a result of using Lovins stemmer, reduces the quality as it transforms words like "action" to "act" or completely transforms certain terms into words that do not make any sense, for example: the word "activation" became "activ".

Secondly, Porters stemmer produced 118 attributes, which is a minor improvement as it produced less malformations of words. Also to be noticed, it picked up on synonyms of certain words. For example, words like "robotics" and "robot's" all became robot. However, some words were have been completely transformed. Therefore, it is a better approach to apply a stem dictionary.

A dictionary was created using not only with the words noticed in the word clouds, but the words that appeared in the file processing steps. The application of this list of synonyms reduced the dimensionality to 86 attributes. It can be seen that applying words like "LSTM",

"CNN", "network" under a more generalized term such as **neural-network**, 41 occurrences appear in the document for Deep Learning. This trend can be also be seen with terms like "nearest-neighbor" "k-nn" to simply fall under the term **knn**, which appears 21 times in the documents for machine learning and 1 time in the document for deep learning. Furthermore, the word "humanoid", a descriptive term that is used in the phrase "humanoid robot", appears entirely within the category of robotics at a cardinality of 13.

Word	Attribute Name	Tota... ↓	Docum...	machin...	deep le...	robotics
neural-network	neural-network	41	7	0	41	0
robot	robot	29	10	0	0	29
knn	knn	22	4	21	1	0
training	training	17	7	12	5	0
naive-bayes	naive-bayes	16	4	16	0	0
model	model	14	7	9	4	1
humanoid	humanoid	13	6	0	0	13
algorithm	algorithm	11	6	10	1	0
data	data	10	4	7	3	0
classification	classification	9	5	8	1	0
decision-tree	decision-tree	9	4	9	0	0

Figure 5: Results of Word Generated Using Synonym List

Generate N-Grams

Experimentation was conducted on using N-Gram generation instead of tokenizing because of certain combinations of words like "artificial neural network" or "restricted boltzmann machine". Initially, the **Generate N-Grams (Characters)** operator was used and the length was set to 25 as there are 25 characters in the phrase *artificial neural network*. Upon evaluation it was noted that there were 365 attributes in the dataset, along with certain phrases that did not seem predictive or relevant to the categories of the documents. Therefore N-gram generation will not be used through the remaining phases of this project.

Creating a Document Vector

The previous section focuses mostly on the terms that appear in the documents. This section will dwell on the cardinality of the terms within the documents. A document vector will be generated, which will be the data for the clustering and classification algorithms to be trained on. Firstly, a cross validation building block with a decision tree operator was added in order test the accuracy of different vector types, in this case: **Term Occurance** and **TD-IDF**.

TD-IDF

With TD-IDF, using absolute pruning between 2 and 15, an accuracy of 86.67% was achieved. See Figure 6 for the RapidMiner confusion matrix.

accuracy: 86.67% +/- 16.33% (mikro: 86.67%)

	true machine learning	true deep learning	true robotics	class precision
pred. machine learning	9	3	0	75.00%
pred. deep learning	1	7	0	87.50%
pred. robotics	0	0	10	100.00%
class recall	90.00%	70.00%	100.00%	

Figure 6: Accuracy using

Secondly, pruning by rank was applied with a below rank of 0.5 and 0.95. However, this showed poor results as the accuracy only 20.00% and number of 495 attribute generated. Percentual Pruning was applied to the model with a prune below percentage of 3.0 and a prune above percentage of 30.0. This resulted in an accuracy of 46.67%.

Term Occurrence

With Term Occurrence, a better score was achieved, with a 90.00% accuracy which can be seen in Figure 7. This was achieved using absolute pruning with a range between 2 and 15.

accuracy: 90.00% +/- 15.28% (mikro: 90.00%)

	true machine learning	true deep learning	true robotics	class precision
pred. machine learning	10	3	0	76.92%
pred. deep learning	0	7	0	100.00%
pred. robotics	0	0	10	100.00%
class recall	100.00%	70.00%	100.00%	

Figure 7: Accuracy using

Similar to TD-IDF, the ranking prune method was used with the same metrics, which returned a slightly higher accuracy score of 23.33%.

Overall, it is evident that term occurrence using the absolute pruning method proves to be the most accurate for vector creation. Furthermore, in order to use this data for training when using clustering or classification algorithms, it needs to be saved. Therefore, the **store** operator is used within the process.

Modeling

Modeling techniques

To test the data that has been prepared two approaches will be taken. Firstly, two clustering algorithms will be used to see if they can identify groups in which the documents reside. It will be evaluated whether or not does the cosine numerical measure improve accuracy of

the model and seen whether minimum or maximum linkage is better suited for the dataset that has been generated. Additionally, two classification algorithms applied and evaluated on which returned the most better results on the testing documents. In terms of algorithms, for clustering K-Means and Agglomerative Clustering algorithms shall be used. For classification, K-NN and naive bays shall be used.

Clustering

K-Means

The k-means algorithm was first applied for evaluation. The initial configurations are as follows. K (the amount of clusters) was set to 3, 10 max runs were used, the measurement type was set to numerical and the **EuclideanDistance** was used as a symmetric distance measurement. The model performs rather poorly, misgrouping some of the documents into the wrong classes, with majority of the documents being grouped into *cluster_2*. See Figure 8 for a scatter plot of K-means.

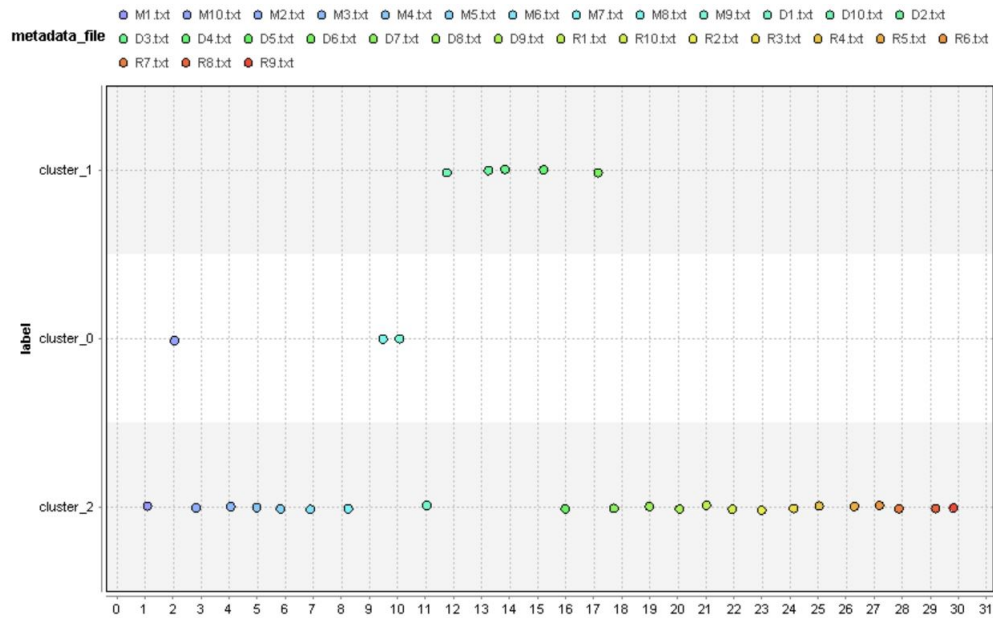


Figure 8: K-means Clustering Scatter Plot

After this the **CosineSimilarity** asymmetric similarity measure was applied. This improved the performance of the model with majority of the documents being grouped in their respective classes. See Figure 9. for visualisation of improved performance.

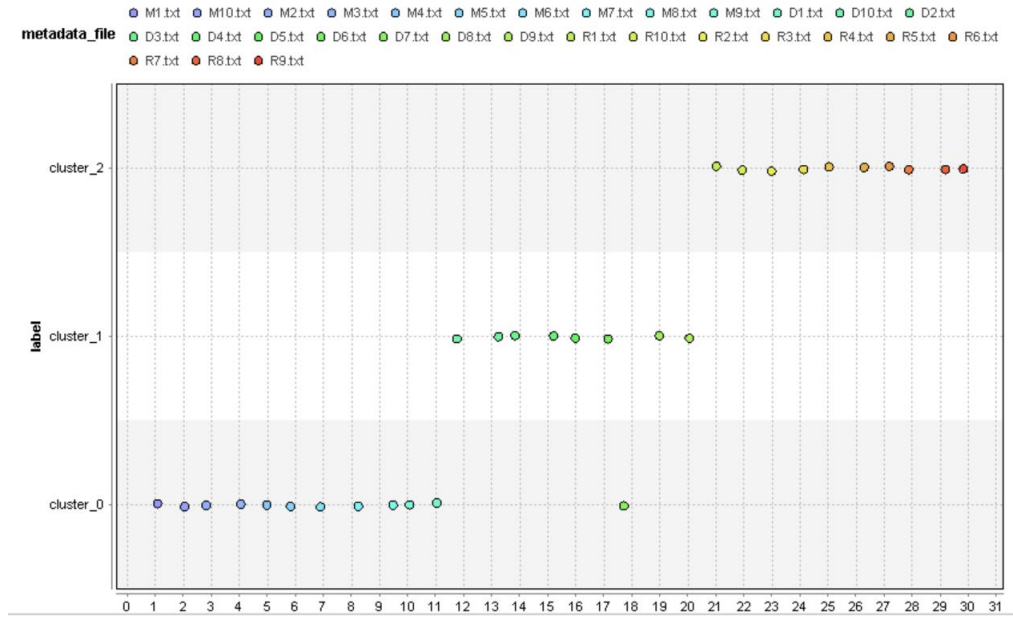


Figure 9: K-means Clustering Scatter Plot Using Cosine Similarity

Agglomerative Clustering

For agglomerative clustering the parameter configurations are as follows: the measurement type was set to numeric, the measurement was set to **CosineSimilarity**, the number of clusters is equal to three and the minimum (single) linkage was applied initially. The model performed very well, identifying all the documents into the appropriate clusters.

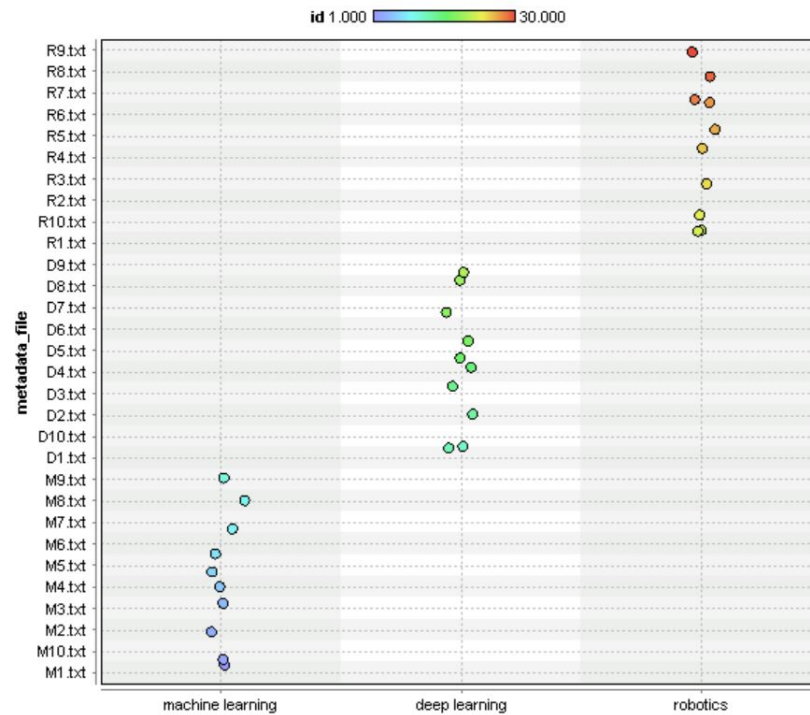


Figure 10: Agglomerative Clustering Using Minimum Linkage

Complete linkage was then applied to the model, this did not make any increase or decrease in performance. However, due to a bug with RapidMiner, the dendrogram was unviewable as it cut off half of the diagram, therefore it is uncertain to comment on which method is more desirable.

Classification

This section deals with classification of the document vector. Two models shall be used to assess the accuracy. Ideally, a support vector machine would be used, as it is known to be the most accurate of predictive models, but as it only supports a binary class input and we have three, K-NN and Naive Bayes shall be used.

K-NN

Niave Bayes

Test Design

Build and Assess the Model

Evaluation

Clustering

Subjective Evaluation

Objective Evaluation

Classification

Project review

Project deployment

Appendices

Appendix A

Rapid Miner Processes in XML