Data Management, Warehousing and Analytics
Assignment 4

In this assignment, the typical processes of sentiment and semantic analysis are explored. The advent of big data processing technologies has not only provided solutions to store and process humongous amounts of data but also opened doors to new fields of study. They also complemented the existing fields of machine learning, natural language processing, computer vision etc, by providing a solution to deal with extremely large datasets which may include millions of documents and training data. These data cannot be handled by conventional processing environments and thus the models being built may stay suboptimal unless they use big data processing solutions underneath.

Spark is one of the popular, trending big data processing frameworks which also significantly boosts the speed of building machine learning models. Spark is 100x faster than Hadoop's MapReduce when it comes to machine learning [1]. This assignment uses Spark to explore the basic processes of sentiment and semantic analysis.

Sentiment Analysis

Sentiment Analysis is a study of processing data to classify the opinion/emotion within text data. It is a sub-field of Natural language Processing (NLP) and involves text analysis and computational linguistics to classify/predict the sentiment within the data [6]. It has a lot of applications in business, government surveys, and administration. For example, knowing the customers opinion on the line of business or a certain product without explicit feedback could give veracious insights.

In this assignment, 5763 tweets have been analyzed to extract the opinions in them. It uses the simple *Bag of Words* model for analysis. All the tweets are cleaned to remove punctuation, non ascii symbols and special symbols. Bag of words is extracted from each tweet, which contains the set of words and corresponding frequencies in that particular tweet. The script, *generate_bow.py* generates the bag of words and writes them to the collection, *bagOfWords* in MongoDB. These bags of words have been extracted from MongoDB and presented in the CSV file, *bag_of_words.csv* as shown in Figure-1.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
tweet_id	text	Bag_of_Words																				
26	RT	{'httpstcopqespjbsjj': 1, 'covid19': 1, 'warchamp7': 1, 'rt': 1, 'season': 1, 'canada': 1, 'cancels': 1, 'hockey': 1}																				
29	RT @Emily	{'advises': 1, 'henry': 1, 'breaking': 1, 'all': 1, 'us': 1, 'rt': 1, 'canada': 1, 'out': 1, 'columbians': 1, 'travel': 1, 'bonnie': 1, 'nonessential': 1, 'dr': 1, 'emilyvance': 1, 'british': 1, 'against': 1, 'including': 1, 'of': 1, 'the': 1}																				
474	RT @spect	{'pm': 1, 'for': 1, 'breaking': 1, 'trudeau': 1, 'tests': 1, 'spectatorindex': 1, 'rt': 1, 'coronavirus': 1, 'sophie': 1, 'positive': 1, 'wife': 1, 'of': 1, 'canadas': 1}																				
964	@Dontee	{'this': 1, 'are': 1, 'guys': 1, 'system': 1, 'but': 1, 'donteewrites': 1, 'facilities': 1, 'play': 1, 'just': 1, 'tooupgrade': 1, 'applies': 1, 'build': 1, 'reasonable': 1, 'sector': 1, 'gallery': 1, 'to': 3, 'you': 1, 'theyd': 1, 'these': 1, 'on': 1, 'education': 1, 'groun																				
1677	@brucefar	{'weird': 1, 'dalhousieu': 1, 'normal': 1, 'brucefanjoy': 1, 'pronounciation': 1, 'moving': 1, 'coming': 1, 'quite': 1, 'amp': 1, 'i': 1, 'nb': 1, 'to': 1, 'understood': 1, 'here': 1, 'strange': 1, 'maritimes': 1, 'university': 1, 'from': 1, 'ottawa': 2, 'street':																				
1697	RT @Rico	{'have': 1, 'am': 1, 'so': 1, 'opportunity': 1, 'womens': 1, 'rt': 1, 'i': 1, 'work': 1, 'to': 2, 'coach': 1, 'happy': 1, 'international': 1, 'ricoscott6': 1, 'with': 1, 'fortunate': 1, 'many': 1, 'very': 1, 'privilege': 1, 'day': 1, 'and': 2, 'the': 1}																				
1806	@Letterke	{'for': 2, 'in': 2, 'your': 1, 'guys': 1, 'show': 1, 'wayne': 1, 'but': 1, 'good': 1, 'up': 1, 'keep': 1, 'immensely': 1, 'unfortunate': 1, 'it': 1, 'awesome': 1, 'minister': 1, 'i': 2, 'work': 1, 'letterkennyprob': 1, 'to': 1, 'was': 1, 'thats': 1, 'february': 1, 'phe																				
1950	@HalifaxB	{'down': 1, 'shop': 1, 'working': 1, 'in': 1, 'is': 1, 'neither': 1, 'etc': 1, 'so': 1, 'all': 1, 'just': 1, 'it': 1, 'bos': 1, 'thing': 1, 'credit': 1, 'or': 1, 'seems': 1, 'i': 1, 'lloyds': 1, 'machines': 1, 'garage': 1, 'payment': 1, 'can': 1, 'my': 1, 'halifaxbank': 1, 'either																				
2040	@domesti	{'this': 1, 'are': 1, 'lol': 1, 'try': 1, 'honestly': 1, 'saying': 1, 'up': 1, 'keeps': 1, 'i': 1, 'now': 1, 'you': 1, 'my': 1, 'halifaxbank': 1, 'issues': 1, 'domesticqueen34': 1, 'even': 1, 'make': 1, 'app': 1, 'sorry': 1, 'halifax': 1, 'we': 1, 'having': 1, 'later': 1, 'co																				
2214	RT @Dave	{'is': 1, 'screenwriter': 1, 'calderdale': 1, 'choir': 1, 'hall': 1, 'a': 1, 'rt': 1, 'sally': 1, 'freedom': 1, 'daveexaminer': 1, 'at': 1, 'acclaimed': 1, 'receiving': 1, 'her': 1, 'town': 1, 'halifax': 1, 'of': 1, 'borough': 1, 'wainwright': 1, 'the': 1}																				
2250	RT @tinay	{'dorian': 1, 'yhz': 1, 'remember': 1, 'crises': 1, 'had': 1, 'disasters': 1, 'it': 1, 'a': 1, 'rt': 1, 'impacts': 1, 'such': 1, 'amp': 3, 'on': 1, 'poor': 1, 'tinayoh': 1, 'disabled': 1, 'more': 1, 'halifax': 1, 'elderly': 1, 'hurricane': 1, 'the': 2}																				
2453	I miss Livin	{'in': 1, 'i': 1, 'miss': 1, 'halifax': 1, 'living': 1}																				
2509	RT @Jacol	{'a': 1, 'httpstcor4ubb0z2oj': 1, 'rt': 1, 'lot': 1, 'see': 1, 'we': 1, 'jacobwhite26': 1, 'of': 1, 'hams': 1}																				
2529	RT @MyH	{'this': 2, 'are': 1, 'httpstcoloworkaojs': 1, 'weeks': 1, 'if': 1, 'it': 1, 'awesome': 1, 'watch': 1, 'rt': 1, 'checking': 1, 'myhalifax': 1, 'here': 1, 'week': 1, 'video': 1, 'love': 1, 'youd': 1, 'we': 1, 'right': 1, 'wed': 1, 'the': 1}																				
2927	Canada	{'intlstudents': 1, 'in': 2, 'have': 1, 'httpstcob4uxf8l06o': 1, 'seen': 1, 'three': 1, 'experts': 1, 'explain': 1, 'methodologies': 1, 'join': 1, 'systems': 1, 'evaluating': 1, 'httpstco4j2qk7g2fa': 1, 'bangladesh': 1, 'highered': 1, 'as': 1, 'pakistan': 1, 'the																				
3091	RT @deep	{'incompetent': 1, 'for': 1, 'politicians': 1, 'pensio': 1, 'is': 2, 'career': 1, 'a': 1, 'rt': 1, 'canada': 1, 'cpc': 1, 'grifting': 1, 'deepgreendesign': 1, 'throng': 1, 'ignorant': 1, 'realizing': 1, 'cpchq': 1, 'of': 1, 'golden': 1, 'and': 1}																				
3506	@ErinOTo	{'this': 1, 'but': 1, 'racism': 1, 'so': 1, 'youre': 1, 'canada': 1, 'law': 1, 'race': 1, 'not': 1, 'petermackay': 1, 'erinotoolemp': 1, 'into': 2, 'sharia': 1, 'concerned': 1, 'about': 2, 'creeping': 2}																				
3764	Canadian I	{'his': 1, 'fears': 1, 'trudeau': 1, 'office': 1, 'minister': 1, 'coronavirus': 1, 'confirms': 1, 'over': 1, 'canadian': 1, 'justin': 1, 'selfisolating': 1, 'httpstcopidbugj43l': 1, 'prime': 1}																				
4590	got the	{'down': 1, 'semester': 1, 'for': 1, 'despite': 1, 'this': 1, 'worship': 1, 'have': 1, 'are': 1, 'is': 2, 'got': 1, 'remainder': 1, 'but': 1, 'students': 1, 'saying': 1, 'our': 1, 'httpstcoxrkzelrvwe': 1, 'glory': 1, 'a': 1, 'dont': 1, 'tears': 1, 'amp': 3, 'that': 1, 'hea																				
4823	If you're fr	{'in': 1, 'have': 1, 'harbour': 1, 'immune': 1, 'if': 1, 'youre': 2, 'swam': 1, 'coronavirus': 1, 'to': 1, 'now': 1, 'from': 1, 'halifax': 1, 'and': 1, 'the': 2}																				
4894	The Wome	{'for': 2, 'due': 1, 'in': 1, 'women': 1, 'try': 1, 'originally': 1, 'covid19': 1, 'event': 1, 'world': 2, 'as': 1, 'has': 1, 'womens': 1, 'championship': 1, 'they': 1, 'to': 2, 'cancelled': 1, '1': 1, '2021': 1, 'been': 1, 'cancellation': 1, 'planned': 1, 'april': 1, 're																				
5385	Quebec Pr	{'health': 1, 'for': 2, 'premier': 1, 'people': 2, 'this': 1, 'in': 1, 'is': 3, 'legault': 1, 'than': 1, 'mandatory': 1, 'selfisolate': 1, 'francois': 1, 'abroad': 1, 'systems': 1, '250': 1, 'all': 1, 'employees': 1, 'public': 1, 'sector': 1, 'banning': 1, 'quebec': 1, 'to'																				
5409	Eight years	{'stillwaiting': 1, 'for': 3, 'support': 1, 'children': 1, 'across': 1, 'years': 2, 'dream': 2, 'are': 1, 'calling': 1, 'parliament': 1, 'voted': 1, 'a': 1, 'canada': 1, 'to': 2, 'become': 1, 'shannens': 2, 'equal': 1, 'education': 1, 'httpstcokdw4nlpp7s': 1, 'ago':																				
5556	Canada	{'for': 2, 'trump': 1, 'instead': 1, 'covid19': 1, 'himself': 1, 'traveling': 1, 'ignoring': 1, 'amp': 3, 'canada': 1, 'testing': 2, 'there': 1, 'block': 1, 'results': 1, 'need': 1, 'on': 1, 'education': 2, 'disease': 2, 'then': 1, 'management': 2, 'while': 1, 'from':																				
65	RT @mari	{'people': 1, 'this': 1, 'have': 1, 'is': 1, 'seen': 1, 'facilities': 1, 'longterm': 1, 'as': 1, 'rt': 1, 'care': 1, 'most': 1, 'at': 1, 'outbreak': 1, 'marialavis': 1, 'from': 1, 'vulnerable': 1, 'lynnvalley': 1, 'we': 1, 'group': 1, 'of': 1, 'the': 2}																				
191	RT @spect	{'pm': 1, 'for': 1, 'breaking': 1, 'trudeau': 1, 'tests': 1, 'spectatorindex': 1, 'rt': 1, 'coronavirus': 1, 'sophie': 1, 'positive': 1, 'wife': 1, 'of': 1, 'canadas': 1}																				
418	@pigeonk	{'years': 1, 'this': 1, 'in': 1, 'have': 1, 'are': 1, 'is': 1, 'oh': 1, 'fucking': 1, '4': 2, 'included': 1, 'so': 2, 'coast': 1, 'well': 1, 'americans': 1, 'west': 1, 'last': 1, 'canada': 1, 'im': 1, 'i': 1, 'that': 1, 'know': 1, 'relationships': 2, 'difficult': 1, 'brits': 1, 'on':																				
541	130 confir	{'in': 1, 'already': 1, 'so': 1, 'confirmed': 1, 'country': 1, 'canada': 1, 'i': 1, 'to': 1, 'go': 1, 'my': 1, 'back': 1, '130': 1, 'cases': 1, 'cant': 1}																				

Figure 1: Bag of Words

In the second step, the bag of words of each tweet is compared against the lists of positive and negative words to find the polarity (opinion class) of the matching words in each of the tweets. The script, *find_polarity.py* does this task. The findings are written *findings.json* file. This file is then transformed to *findings_tableau.xlsx* by the script, *findings_for_tableau.py,* for making it ready to be visualized in tableau. (Figure 2)

tweet_id	tweet	match	polarity	frequency
26	RT @Warc	N/A	neutral	0
29	RT @Emily	breaking	negative	1
474	RT @spect	positive	positive	1
474	RT @spect	breaking	negative	1
964	@Donteev	reasonable	positive	1
1677	@brucefar	strange	negative	1
1677	@brucefar	weird	negative	1
1697	RT @Rico$	fortunate	positive	1
1697	RT @Rico$	happy	positive	1
1697	RT @Rico$	privilege	positive	1
1697	RT @Rico$	work	positive	1
1806	@Letterke	awesome	positive	1
1806	@Letterke	enjoyed	positive	1
1806	@Letterke	enough	positive	1
1806	@Letterke	fortunate	positive	1
1806	@Letterke	good	positive	1
1806	@Letterke	phenomen	positive	1
1806	@Letterke	work	positive	1
1806	@Letterke	unfortunat	negative	1
1950	@HalifaxB	N/A	neutral	0
2040	@domesti	issues	negative	1
2040	@domesti	sorry	negative	1
2214	RT @Dave	acclaimed	positive	1
2214	RT @Dave	freedom	positive	1
2250	RT @tinay	disabled	negative	1
2250	RT @tinay	poor	negative	1
2453	I miss Livin	miss	negative	1

Figure 2: findings_tableau.xlsx

These findings are then visualized in Tableau for the words and corresponding frequencies in each of the sentiment classes, using a word cloud. (Figure 3)

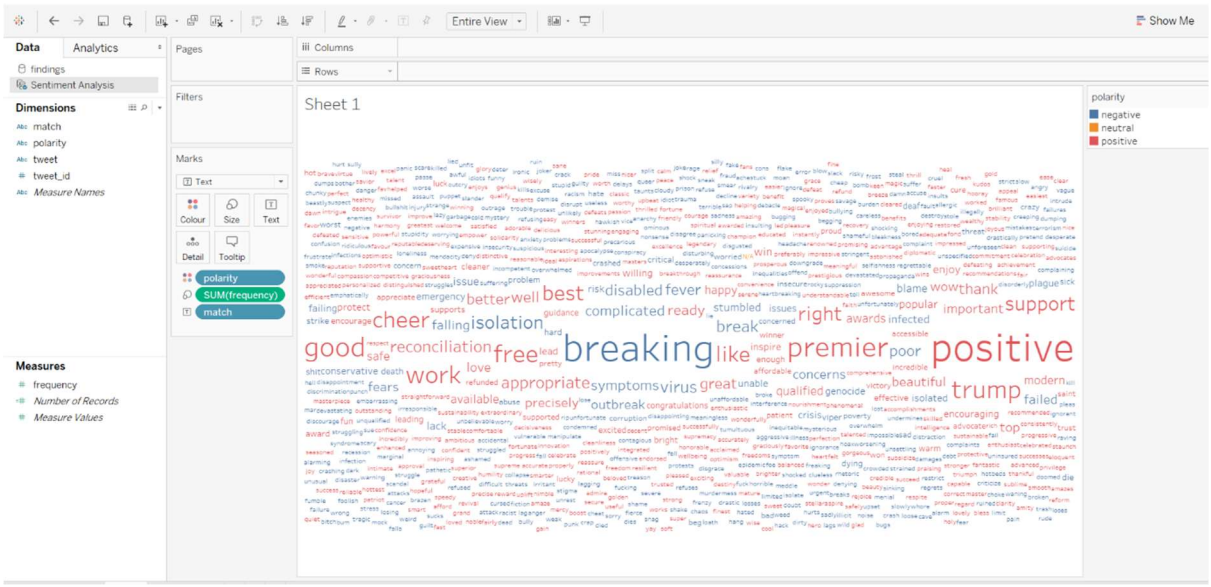

Figure 3: Word Cloud Visualization in Tableau

Semantic Analysis

Semantic Analysis is a study which uses phrases, clauses, sentences and syntactic structures within text data to classify or cluster the documents into similar semantic classes[6]. In other words, semantic analysis is used for analyzing and extracting the meaning and classify or cluster the documents with respect to similarity in the meaning and context. Semantic Analysis is an application of Natural Language Processing. It is mainly used for searching appropriate, similar documents based on query.

This assignment uses methods like TF-IDF (Term Frequency-Inverse Document Frequency) weighting and relative frequency measures to analyze the relevance and context of the documents with respect to the query keywords. TF-IDF is a statistic which measures how important a word is to document in a corpus. TF_IDF value increases as the relevance of the word increases with respect to the document. If more documents contain the given word, it means that the word is of less relevance to the given document and hence the TF-IDF value decreases.

TF-IDF is used as a text vectorization (feature extraction and feature vectors creation from text) method. A document can be expressed as a feature vector whose feature names are the words in the text and component magnitudes are the corresponding TF-IDF weights. These documents (as feature vectors) can then be subjected to similarity analysis methods like cosine similarity and can be analyzed for spatial correlation and similarity[6]. This assignment involves calculating the Inverse Document Frequencies, and Real Frequencies only. *Cosine Similarity* analysis is beyond the scope of this assignment.

For this task, news data have been used. These data are cleaned and loaded into MongoDB. The script, *idf.py* generates the inverse document frequencies for each of the query keywords as shown in Figure 4.

	total_docs	df	total_by_c	idf
Canada	80	25	3.2	1.678
education	80	2	40	5.322
university	80	8	10	3.322
Dalhousie	80	5	16	4
Halifax	80	5	16	4
business	80	2	40	5.322

Figure 4: IDF Table

The script, *relative_frequency.py* calculates the relative frequency and generates the relative frequency tables for each of the keywords. Figure 5 shows the relative frequency table for the keyword 'Canada' and Figure 6 shows the relative frequency table for the keyword 'university'

Document	Total_wor	Term_freq	Relative_frequency
1	92	2	0.022
2	70	1	0.014
3	117	2	0.017
4	87	0	0
5	83	0	0
6	108	0	0
7	93	2	0.022
8	91	2	0.022
9	97	7	0.072
10	87	0	0
11	87	2	0.023
12	104	5	0.048
13	113	0	0
14	92	1	0.011
15	93	2	0.022
16	98	0	0
17	76	0	0
18	94	0	0
19	112	0	0
20	64	0	0
21	107	0	0
22	73	0	0
23	84	0	0
24	95	0	0
25	90	0	0
26	93	0	0
27	92	0	0
28	94	0	0

Canada_rel_freq

Figure 5: RF of 'Canada'

Document	Total_wor	Term_freq	Relative_frequency
1	92	0	0
2	70	0	0
3	117	0	0
4	87	0	0
5	83	0	0
6	108	0	0
7	93	0	0
8	91	0	0
9	97	0	0
10	87	0	0
11	87	0	0
12	104	0	0
13	113	0	0
14	92	0	0
15	93	0	0
16	98	0	0
17	76	1	0.013
18	94	1	0.011
19	112	1	0.009
20	64	1	0.016
21	107	0	0
22	73	0	0
23	84	0	0
24	95	0	0
25	90	0	0
26	93	0	0
27	92	0	0
28	94	0	0

university_rel_freq

Figure 6: RF of 'university'

All the scripts, tables, figures, output files have been attached in the submission zip.

References:

[1] "Apache Spark™ - Unified Analytics Engine for Big Data", Spark.apache.org, 2020. [Online]. Available: https://spark.apache.org/. [Accessed: 13- Apr- 2020].

[2] " Opinion Observer: Analyzing and Comparing Opinions on the Web.", Proceedings of the 14th International World Wide Web conference (WWW-2005) [Online]. Available: http://ptrckprry.com/course/ssd/data/positive-words.txt [Accessed: 13- Apr- 2020].

[3] " Opinion Observer: Analyzing and Comparing Opinions on the Web.", Proceedings of the 14th International World Wide Web conference (WWW-2005) [Online]. Available: http://ptrckprry.com/course/ssd/data/negative-words.txt [Accessed: 13- Apr- 2020].

[4] "pyspark.sql module — PySpark 2.4.5 documentation", *Spark.apache.org*, 2020. [Online]. Available: https://spark.apache.org/docs/latest/api/python/pyspark.sql.html. [Accessed: 13- Apr- 2020].

[5] "pyspark.sql.dataframe — PySpark master documentation", *People.eecs.berkeley.edu*, 2020. [Online]. Available: http://people.eecs.berkeley.edu/~jegonzal/pyspark/_modules/pyspark/sql/dataframe.html. [Accessed: 13- Apr- 2020].

[6] "Introduction to Information Retrieval". [Online]. Available: https://nlp.stanford.edu/IR-book/pdf/06vect.pdf. [Accessed: 13- Apr- 2020].