Knowledge Discovery and Management

SS2017

Project Submission-Final Phase

Instructor: Dr. Yugyung Lee

Team Members:

Rashmi Tripathi

Table of Contents

1. Introduction

There are a lot of open source deep learning projects available on the internet. These existing projects can be used to develop new innovative projects. The challenge is to understand the functionalities implemented in these projects. Also, these projects use many deep learning models. By knowing the functionalities and the deep learning technologies used, the developers can decide if they want to use the part of the logic in their project. In this project, we are building a model that can analyze a source code and provide results containing the functional semantics of the code as well as their dependencies. The model also segregates the code into clusters based on their functionalities

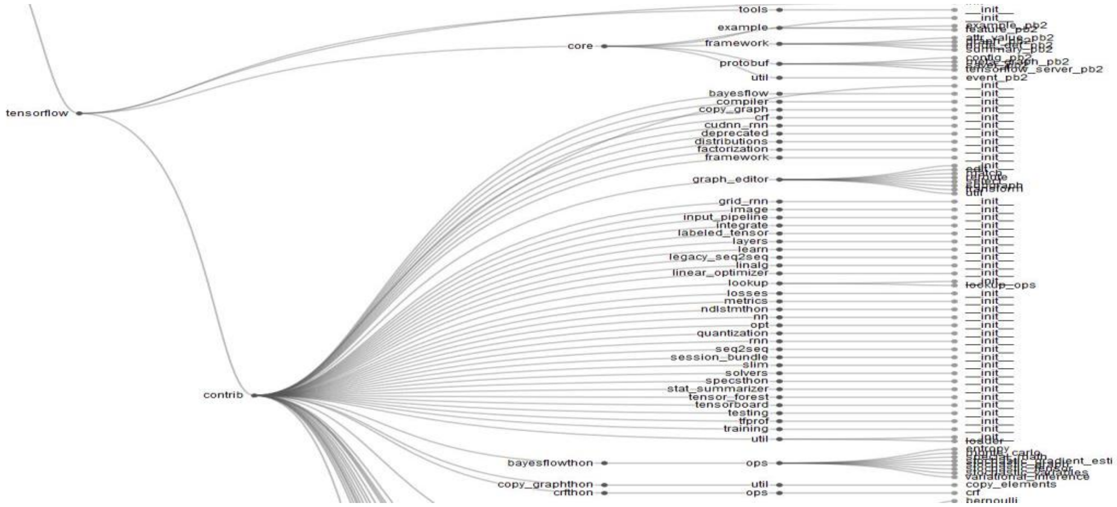
As part of this project we will focus on the relationships between the various modules of **Tensorflow**. TensorFlow is open source project created by Google Brains Twehaimch can be used to build robust and useful deep learning for machine learning purposes providing a Python API and C++ Api.

Prior work in Tensorflow Study

A crawler is created to crawl through each python module to associate the methods with respective packages and classes. There are total number of 1356 API available and their respective packages are known. Counts are shown below as per experiment:

Structure	Total
Packages	319
Modules (Number of python	268
Classes	Still in progress
Important API’s	1356

Below diagram shows the various package flow in Tensorflow:



2. Datasets

The first step for any analysis is generation of data. Here we are working on Tensorflow open source. As already discussed in prior understanding we have collected the packages we want to look into.

Any python file represented by .py extension will have the following members:

Import modules/Packages	Defines dependency on other modules to function
Class Decorators	
Classes	Classes name can describe about the kind of model it using
Functions	Method names can signify the kind of operation being done in this particular file
Functions Decorators	

This is a challenging task to do. To collect the dependency of any module one need to have knowledge of the following function in python.

Dir([object]) in python return the list of names in the current local scope in case of no argument. In case of argument, it will lists the name of attributes for the object.

"The default `dir()` mechanism behaves differently with different types of objects, as it attempts to produce the most relevant, rather than complete, information:

- If the object is a module object, the list contains the names of the module's attributes.

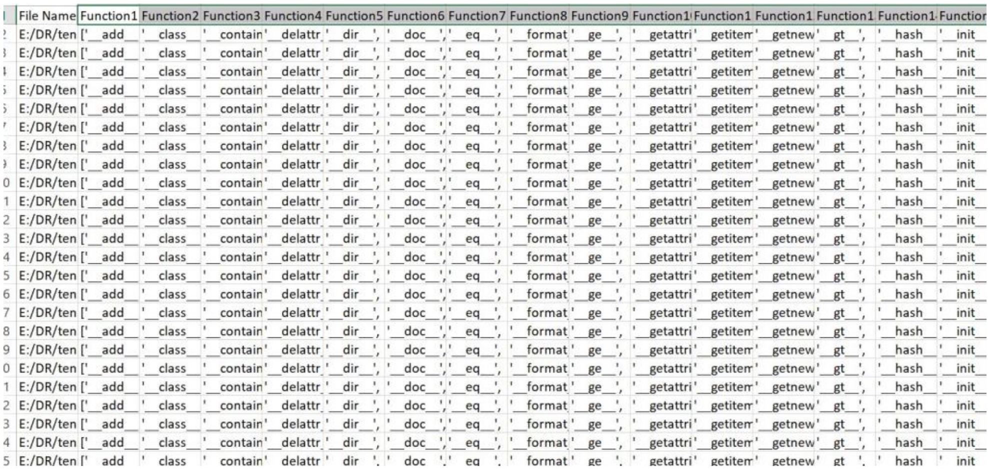
- If the object is a type or class object, the list contains the names of its attributes, and recursively of the attributes of its bases.
- Otherwise, the list contains the object's attributes' names, the names of its class's attributes, and recursively of the attributes of its class's base classes."

Program to get the module details

```
import os
import a

def get_filepaths(directory):
    file_paths = []  # List which will store all of the full filepaths.

    # Walk the tree.
    for root, directories, files in os.walk(directory):
        for filename in files:
            # Join the two strings in order to form the full filepath.
            #print("root",root)
            filepath=root.replace("\\","/")
            #print("root", root)
            filepath=filepath+"/"+filename
            #filepath = os.path.join(root, filename)
            file_paths.append(filepath)  # Add it to the list.
            print("filepath : ",filepath," members are :",dir(filepath))

    return file_paths  # Self-explanatory.

# Run the above function and store its results in a variable.
full_file_paths = get_filepaths("E:/DR/tensorflow/python/ops")


print(dir("E:/DR/tensorflow/python/ops/math_ops.py"))

print(dir(a))
```

Output

===

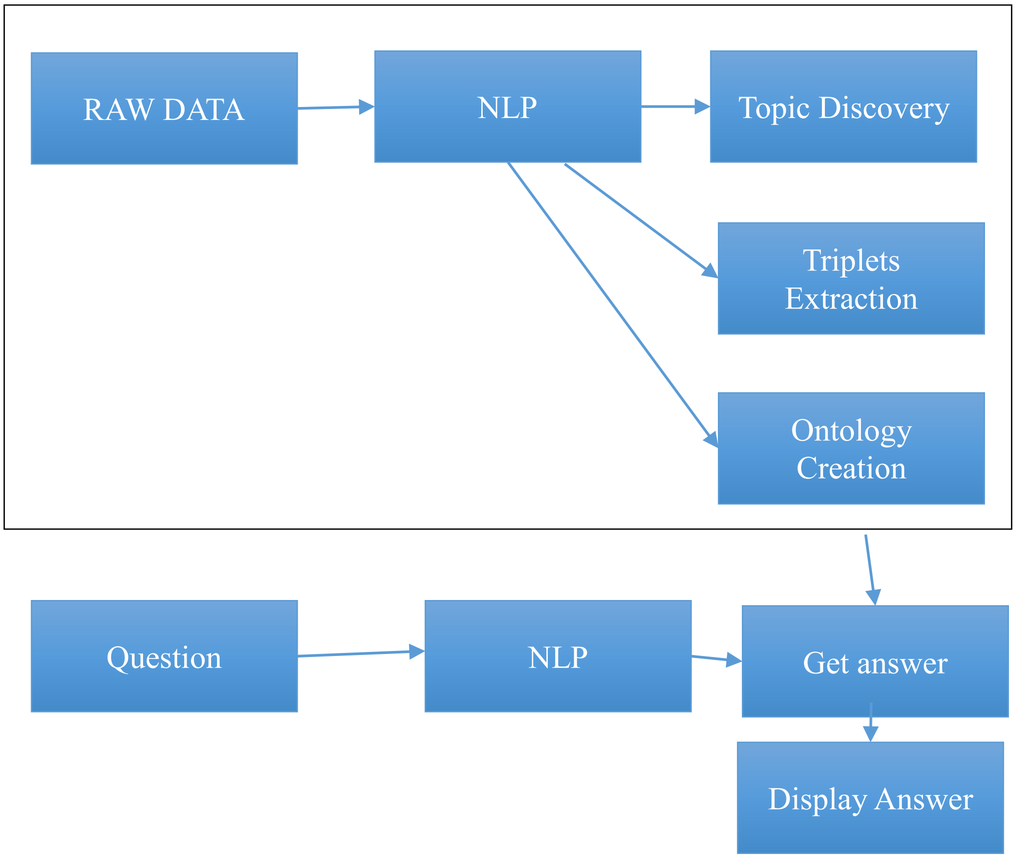


1	File Name	Function1	Function2	Function3	Function4	Function5	Function6	Function7	Function8	Function9	Function1	Function1	Function1	Function1	Function1	Functior
2	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
3	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
4	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
5	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
6	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
7	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
8	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
9	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
0	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
1	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
2	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
3	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
4	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
5	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
6	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
7	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
8	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
9	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
0	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
1	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
2	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
3	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
4	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__
5	E:/DR/ten	['__add__	'__class__	'__contain	'__delattr	'__dir__',	'__doc__',	'__eq__',	'__format	'__ge__',	'__getattri	'__getiten	'__getnew	'__gt__',	'__hash__	'__init__

3. Architecture

Below is architecture of our project overflow. We collected raw data in CSV format as shown in step 2. It consists of several steps before answering questions to answer. The question and answer both need to be processed using NLP to get the matching answer.

- Preprocessing was done on raw data to remove unwanted/stop words. Stop words which we removed are given below:
  - Init
  - Self
  - Scope
  - None
  - (
  -)
  - :
  - -



- **NLP**

Then we extracted the classes List, functions list and import list from the preprocessed data. We used lemmatization in this step.

"***Lemmatization*** *usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma .*
We used Stanford dictionary and libraries to get the lemmatized data."[2]

- **Feature Generation using Information Retrieval (TFIDF, Word2Vec)**

Then TFIDF was implemented on the same to get the top frequency words. There were some usual imports like absolute, math, division which are usual imports required for any deep learning project or mathematical computations using machine learning.

"*Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus."* [3]

- **Triplet Generation <S, P, O>: Information Extraction (OpenIE, WordNet)**

As relations are very important for any knowledge graph generation we need to find the relations between files, imports, classes and functions.

We used our own rules to create triplets in this case as the OpenIE was not sufficient for a peculiar dataset like this.

Sample Triplets:

```
lossesimpl,hasImport,confusionmatrix,obj
lossesimpl,hasFunction,computeweightedloss,obj
rnncellimpl,hasClass,RNNCellobject
```

- **Answer & Question Categorization: LDA & Machine Learning (Clustering, Classification)**

In this step we are using LDA to categorize our data. We tried to get top 20 topics.
LDA

"*In natural language processing, latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. LDA is an example of a topic model"[4]*

- **Generation of your Knowledge Graph (KG) using your answer data**

Using Protégé we generated our knowledge graph. We identified the classes, object properties, data properties, individuals and triplets to get OWL file which can be visualized using WEBOwl.

- **Question Answering using your KG and SPARQL/SWRL**

Then we get the question from the user and corresponding answer was generated from the KG.

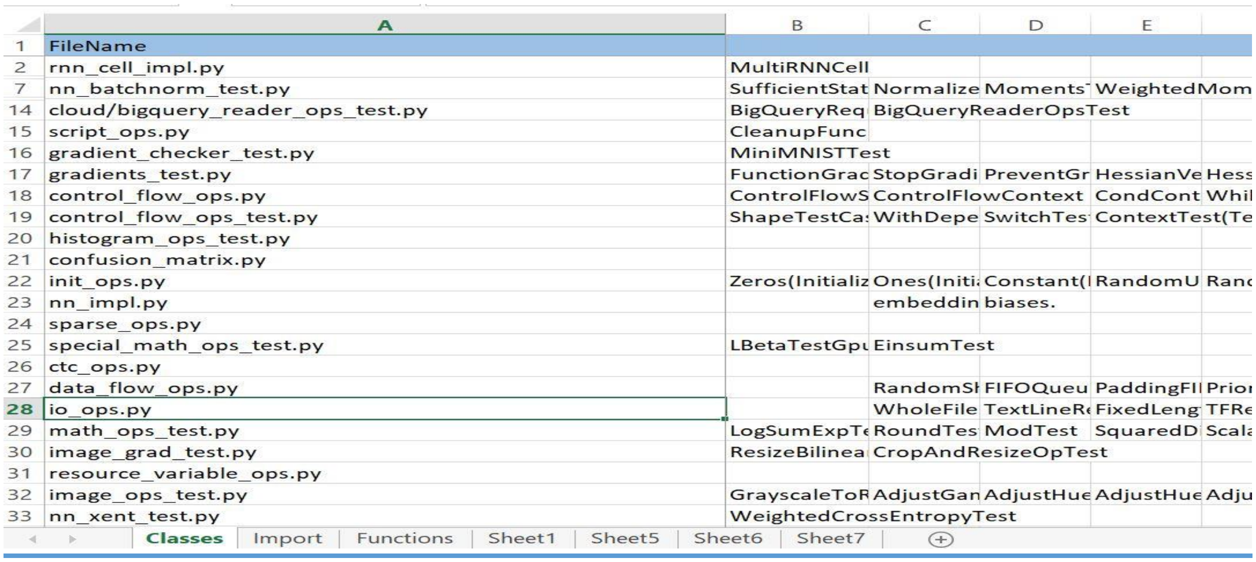
4. Implementation

Tools and Technologies used:

IDE: Intellij 2.4
Language: Apache Spark, Scala, Java
Library: Stanford NLTK
Tools: Protégé, Webowl

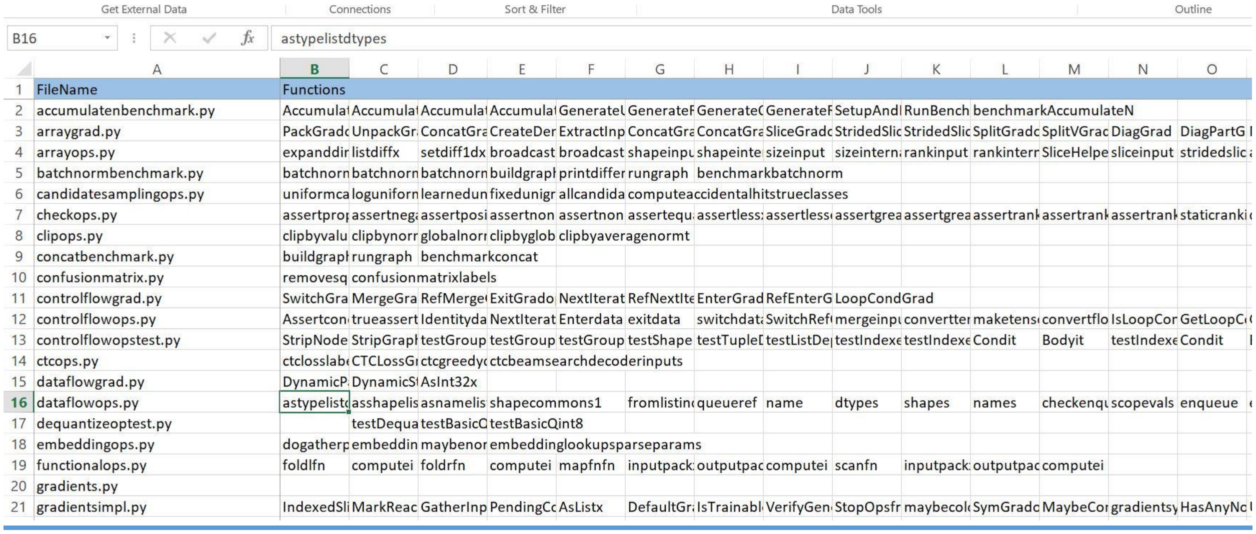
Below are the screenshots for the implementation phase:

Datasets view



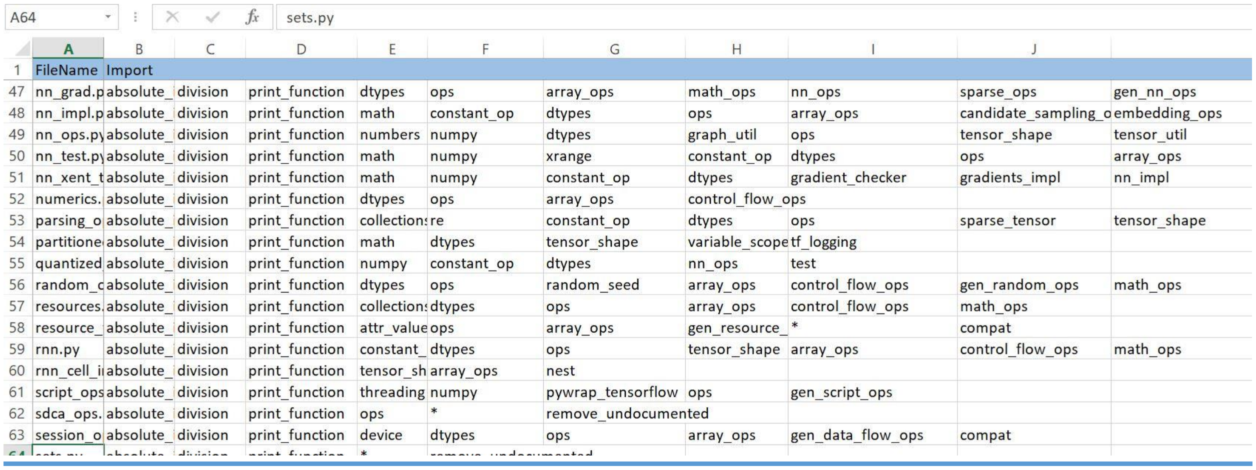
	A	B	C	D	E	
1	FileName					
2	rnn_cell_impl.py	MultiRNNCell				
7	nn_batchnorm_test.py	SufficientStat	Normalize	Moments	WeightedMom	
14	cloud/bigquery_reader_ops_test.py	BigQueryReq	BigQueryReaderOpsTest			
15	script_ops.py	CleanupFunc				
16	gradient_checker_test.py	MiniMNISTTest				
17	gradients_test.py	FunctionGrac	StopGradi	PreventGr	HessianVe	Hess
18	control_flow_ops.py	ControlFlowS	ControlFlowContext		CondCont	Whil
19	control_flow_ops_test.py	ShapeTestCa:	WithDepe	SwitchTes	ContextTest(Te	
20	histogram_ops_test.py					
21	confusion_matrix.py					
22	init_ops.py	Zeros(Initializ	Ones(Initi:	Constant(I	RandomU	Ranc
23	nn_impl.py		embeddin	biases.		
24	sparse_ops.py					
25	special_math_ops_test.py	LBetaTestGpu	EinsumTest			
26	ctc_ops.py					
27	data_flow_ops.py		RandomSh	FIFOQueu	PaddingFI	Prior
28	io_ops.py		WholeFile	TextLineRe	FixedLeng	TFRe
29	math_ops_test.py	LogSumExpTe	RoundTes	ModTest	SquaredD	Scala
30	image_grad_test.py	ResizeBilinea	CropAndResizeOpTest			
31	resource_variable_ops.py					
32	image_ops_test.py	GrayscaleToR	AdjustGan	AdjustHue	AdjustHue	Adju
33	nn_xent_test.py	WeightedCrossEntropyTest				

Classes | Import | Functions | Sheet1 | Sheet5 | Sheet6 | Sheet7



B16 astypelistdtypes

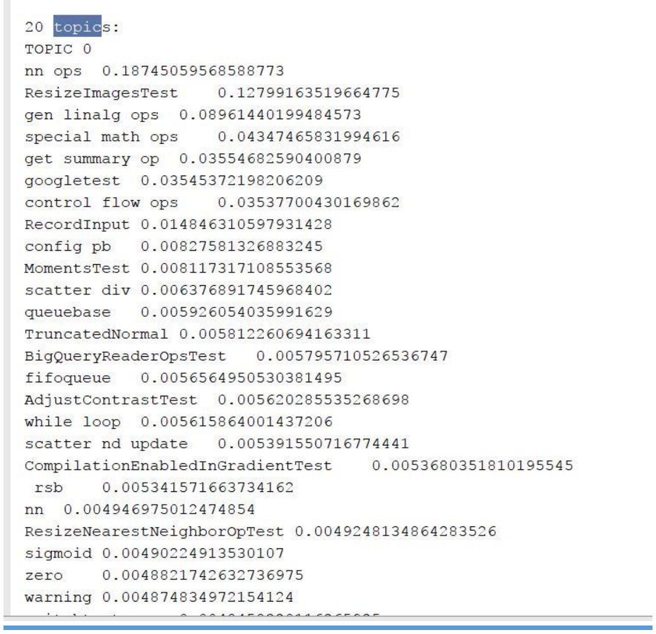
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	FileName	Functions														
2	accumulatenbenchmark.py	Accumulat	Accumulat	Accumulat	Accumulat	GenerateU	GenerateF	GenerateC	GenerateF	SetupAndI	RunBench	benchmarkAccumulateN				
3	arraygrad.py	PackGradc	UnpackGr:	ConcatGra	CreateDer	ExtractInp	ConcatGra	ConcatGra	SliceGradc	StridedSlic	StridedSlic	SplitGradc	SplitVGrac	DiagGrad	DiagPartG	I
4	arrayops.py	expanddir	listdiffx	setdiff1dx	broadcast	broadcast	shapeinpu	shapeinte	sizeinput	sizeinterna	rankinput	rankinterr	SliceHelpe	sliceinput	stridedslic	a
5	batchnormbenchmark.py	batchnorn	batchnorn	batchnorn	buildgraph	printdiffer	rungraph	benchmarkbatchnorm								
6	candidatesamplingops.py	uniformca	loguniforn	learnedun	fixedunigr	allcandida	computeaccidentalhitstrueclasses									
7	checkops.py	assertprop	assertnega	assertposi	assertnon	assertnon	assertequ:	assertless:	assertless	assertgrea	assertgrea	assertrank	assertrank	assertrank	staticranki	(
8	clipops.py	clipbyvalu	clipbynorr	globalnorr	clipbyglob	clipbyaveragenormt										
9	concatbenchmark.py	buildgraph	rungraph	benchmarkconcat												
10	confusionmatrix.py	removesq	confusionmatrixlabels													
11	controlflowgrad.py	SwitchGra	MergeGra	RefMerge	ExitGrado	NextIterat	RefNextIte	EnterGrad	RefEnterG	LoopCondGrad						
12	controlflowops.py	Assertcon	trueassert	Identityda	NextIterat	Enterdata	exitdata	switchdat:	SwitchRef	mergeinpu	convertter	maketens	convertflo	IsLoopCor	GetLoopC	(
13	controlflowopstest.py	StripNode	StripGraph	testGroup	testGroup	testGroup	testShape	testTupleI	testListDe	testIndexe	testIndexe	Condit	Bodyit	testIndexe	Condit	I
14	ctcops.py	ctclosslab	CTCLossGr	ctcgreedy	ctcbeamsearchdecoderinputs											
15	dataflowgrad.py	DynamicP.	DynamicS	AsInt32x												
16	dataflowops.py	astypelist	asshapelis	asnamelis	shapecommons1		fromlistin	queueref	name	dtypes	shapes	names	checkenqu	scopevals	enqueue	e
17	dequantizeoptest.py		testDequa	testBasicQ	testBasicQint8											
18	embeddingops.py	dogatherp	embeddin	maybenor	embeddinglookupsparseparams											
19	functionalops.py	foldlfn	computei	foldrfn	computei	mapfnfn	inputpack:	outputpac	computei	scanfn	inputpack:	outputpac	computei			
20	gradients.py															
21	gradientsimpl.py	IndexedSli	MarkReac	GatherInp	PendingCc	AsListx	DefaultGr:	IsTrainabl	VerifyGen	StopOpsfr	maybecol	SymGradc	MaybeCor	gradientsy	HasAnyNc	l



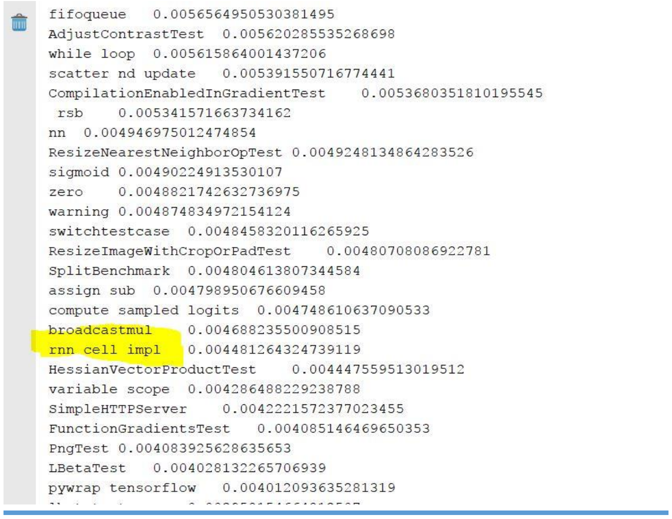
A64 sets.py

	A	B	C	D	E	F	G	H	I	J	
1	FileName	Import									
47	nn_grad.p	absolute_	division	print_function	dtypes	ops	array_ops	math_ops	nn_ops	sparse_ops	gen_nn_ops
48	nn_impl.p	absolute_	division	print_function	math	constant_op	dtypes	ops	array_ops	candidate_sampling_o	embedding_ops
49	nn_ops.py	absolute_	division	print_function	numbers	numpy	dtypes	graph_util	ops	tensor_shape	tensor_util
50	nn_test.py	absolute_	division	print_function	math	numpy	xrange	constant_op	dtypes	ops	array_ops
51	nn_xent_t	absolute_	division	print_function	math	numpy	constant_op	dtypes	gradient_checker	gradients_impl	nn_impl
52	numerics.	absolute_	division	print_function	dtypes	ops	array_ops	control_flow_ops			
53	parsing_o	absolute_	division	print_function	collections	re	constant_op	dtypes	ops	sparse_tensor	tensor_shape
54	partitione	absolute_	division	print_function	math	dtypes	tensor_shape	variable_scope	tf_logging		
55	quantized	absolute_	division	print_function	numpy	constant_op	dtypes	nn_ops	test		
56	random_c	absolute_	division	print_function	dtypes	ops	random_seed	array_ops	control_flow_ops	gen_random_ops	math_ops
57	resources.	absolute_	division	print_function	collections	dtypes	ops	array_ops	control_flow_ops	math_ops	
58	resource_	absolute_	division	print_function	attr_value	ops	array_ops	gen_resource_	*	compat	
59	rnn.py	absolute_	division	print_function	constant_	dtypes	ops	tensor_shape	array_ops	control_flow_ops	math_ops
60	rnn_cell_i	absolute_	division	print_function	tensor_sh	array_ops	nest				
61	script_ops	absolute_	division	print_function	threading	numpy	pywrap_tensorflow	ops	gen_script_ops		
62	sdca_ops.	absolute_	division	print_function	ops	*	remove_undocumented				
63	session_o	absolute_	division	print_function	device	dtypes	ops	array_ops	gen_data_flow_ops	compat	
64	sets.py	absolute_	division	print_function	*	remove_undocumented					

LDA(top 20 topics)

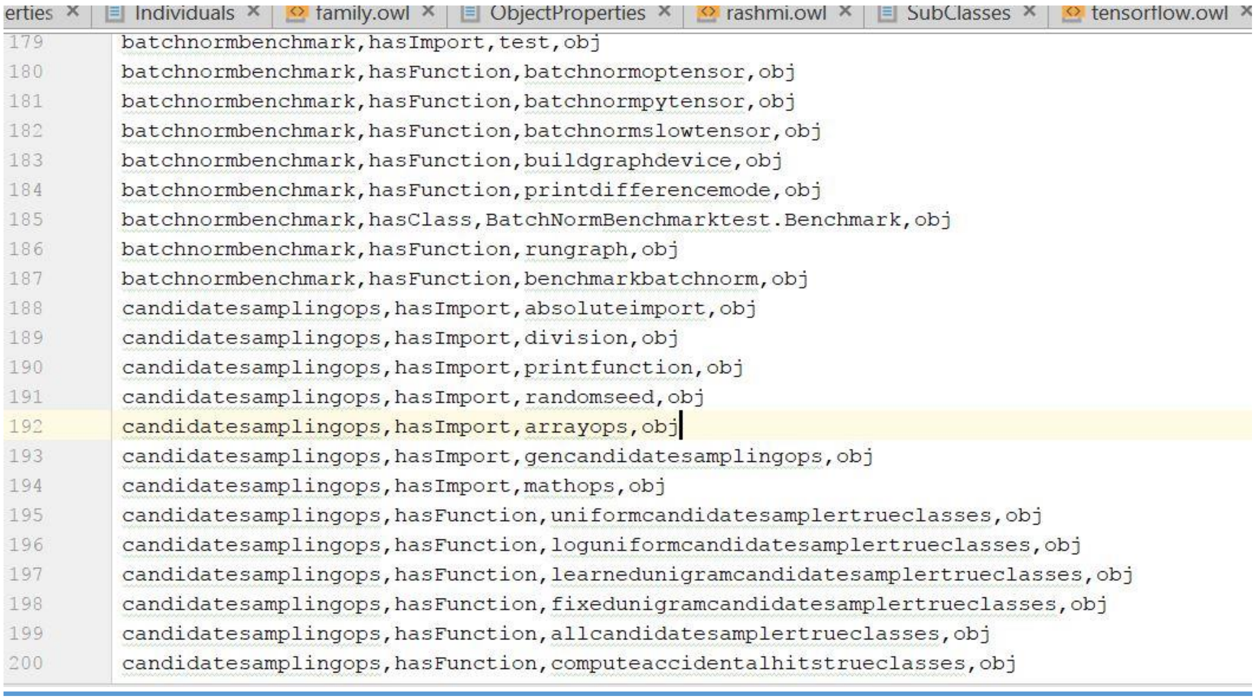


```
20 topics:
TOPIC 0
nn ops  0.18745059568588773
ResizeImagesTest    0.12799163519664775
gen linalg ops  0.08961440199484573
special math ops    0.04347465831994616
get summary op  0.03554682590400879
googletest  0.03545372198206209
control flow ops    0.03537700430169862
RecordInput 0.014846310597931428
config pb   0.00827581326883245
MomentsTest 0.008117317108553568
scatter div 0.006376891745968402
queuebase   0.005926054035991629
TruncatedNormal 0.005812260694163311
BigQueryReaderOpsTest   0.005795710526536747
fifoqueue   0.0056564950530381495
AdjustContrastTest  0.005620285535268698
while loop  0.005615864001437206
scatter nd update   0.005391550716774441
CompilationEnabledInGradientTest    0.0053680351810195545
 rsb    0.005341571663734162
nn  0.004946975012474854
ResizeNearestNeighborOpTest 0.0049248134864283526
sigmoid 0.00490224913530107
zero    0.0048821742632736975
warning 0.004874834972154124
```



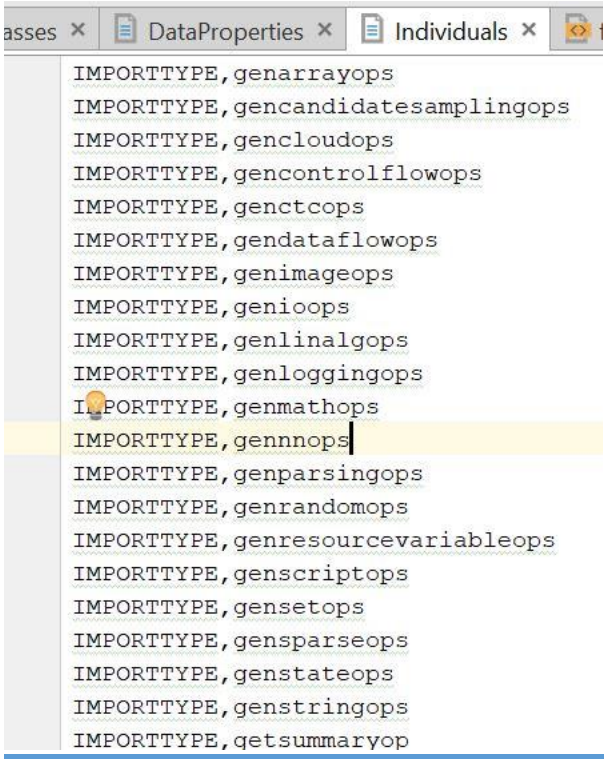
```
fifoqueue   0.0056564950530381495
AdjustContrastTest  0.005620285535268698
while loop  0.005615864001437206
scatter nd update   0.005391550716774441
CompilationEnabledInGradientTest    0.0053680351810195545
 rsb    0.005341571663734162
nn  0.004946975012474854
ResizeNearestNeighborOpTest 0.0049248134864283526
sigmoid 0.00490224913530107
zero    0.0048821742632736975
warning 0.004874834972154124
switchtestcase  0.0048458320116265925
ResizeImageWithCropOrPadTest    0.00480708086922781
SplitBenchmark  0.004804613807344584
assign sub  0.004798950676609458
compute sampled logits  0.004748610637090533
broadcastmul    0.004688235500908515
rnn cell impl   0.004481264324739119
HessianVectorProductTest    0.004447559513019512
variable scope  0.004286488229238788
SimpleHTTPServer    0.0042221572377023455
FunctionGradientsTest   0.004085146469650353
PngTest 0.004083925628635653
LBetaTest   0.004028132265706939
pywrap tensorflow   0.004012093635281319
```

Triplets Extraction

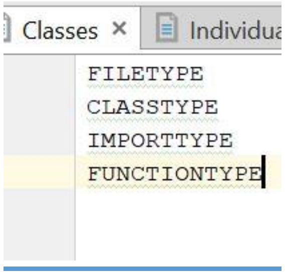


```
batchnormbenchmark,hasImport,test,obj
batchnormbenchmark,hasFunction,batchnormoptensor,obj
batchnormbenchmark,hasFunction,batchnormpytensor,obj
batchnormbenchmark,hasFunction,batchnormslowtensor,obj
batchnormbenchmark,hasFunction,buildgraphdevice,obj
batchnormbenchmark,hasFunction,printdifferencemode,obj
batchnormbenchmark,hasClass,BatchNormBenchmarktest.Benchmark,obj
batchnormbenchmark,hasFunction,rungraph,obj
batchnormbenchmark,hasFunction,benchmarkbatchnorm,obj
candidatesamplingops,hasImport,absoluteimport,obj
candidatesamplingops,hasImport,division,obj
candidatesamplingops,hasImport,printfunction,obj
candidatesamplingops,hasImport,randomseed,obj
candidatesamplingops,hasImport,arrayops,obj
candidatesamplingops,hasImport,gencandidatesamplingops,obj
candidatesamplingops,hasImport,mathops,obj
candidatesamplingops,hasFunction,uniformcandidatesamplertrueclasses,obj
candidatesamplingops,hasFunction,loguniformcandidatesamplertrueclasses,obj
candidatesamplingops,hasFunction,learnedunigramcandidatesamplertrueclasses,obj
candidatesamplingops,hasFunction,fixedunigramcandidatesamplertrueclasses,obj
candidatesamplingops,hasFunction,allcandidatesamplertrueclasses,obj
candidatesamplingops,hasFunction,computeaccidentalhitstrueclasses,obj
```

Knowledge Graph



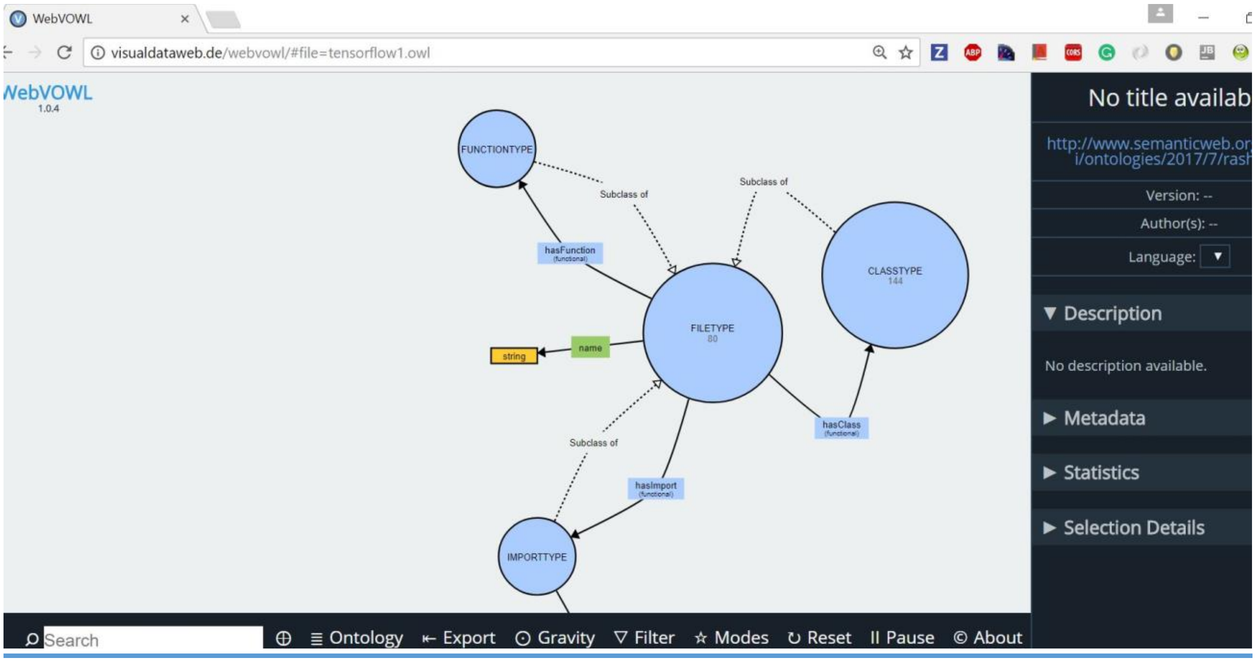
```
IMPORTTYPE,genarrayops
IMPORTTYPE,gencandidatesamplingops
IMPORTTYPE,gencloudops
IMPORTTYPE,gencontrolflowops
IMPORTTYPE,genctcops
IMPORTTYPE,gendataflowops
IMPORTTYPE,genimageops
IMPORTTYPE,genioops
IMPORTTYPE,genlinalgops
IMPORTTYPE,genloggingops
IMPORTTYPE,genmathops
IMPORTTYPE,gennnops
IMPORTTYPE,genparsingops
IMPORTTYPE,genrandomops
IMPORTTYPE,genresourcevariableops
IMPORTTYPE,genscriptops
IMPORTTYPE,gensetops
IMPORTTYPE,gensparseops
IMPORTTYPE,genstateops
IMPORTTYPE,genstringops
IMPORTTYPE,getsummaryop
```



```
FILETYPE
CLASSTYPE
IMPORTTYPE
FUNCTIONTYPE
```

Final Knowledge Graph

As you can see classes and individuals are mapped into the graph. We tried using a bit more complex graph for whole dataset and it has limitation for 10 MB data.



5. Results and Evaluation

The classification has been done using decision tree and random forest approach. We got following accuracy:

Random Forest	35%
Decision Tree	40%

Question answering system was tried using Open IE Triplets approach and SPARQL. Both failed due to the following reasons:

As dataset is not any regular text data, triplets were not generated properly and manually doing it will be like string matching approach in any language. It is not properly methodology and hence need to find a proper way to do it.

SPARQL and protégé have lot of limitations. For such a huge dataset there is no automatic way. Even axioms generation through excel sheet in Protégé application fails with no relevant information to debug further. The ontology created through code was not validated in Webowl and also not getting loaded into protégé. Even partial generation fails sometimes and sometimes get success. This is very strange and need to be debug further.

6. Project management

Github URL

https://github.com/rashmitripathi/CS5560RashmiLab_knowledge_discovery_mgmt/tree/master/KDM%20Project

No project management as only single member in the team.

7. Future Work

- As variables are random in nature, need to figure out a way to extract variables from each python file.
- What about other members of python file ?How to extract information from them.
- For such kind of dataset how to generate triplets using any standard library and what can be the efficient way.

8. References

1. https://docs.python.org/2/library/functions.html
2. https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
3. http://www.tfidf.com/
4. https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation