Semantic Scholar Big Data Application

Investigating the factors for citations of academic papers

MSc Analytics: Big Data Platforms (MSCA 31013)

Team Members: Nazih Kalo, Andrew Luong, Hoang Nguyen

Date: March 10, 2020

Business Problem/Executive Summary

Background:

- Researchers (academic or industry) goal is to increase the body of knowledge in a meaningful and impactful way
- One of the hardest challenges is to [1] identify a innovative topic and [2] structure the document in a way to earn citations (a proxy for notoriety and peacefulness of a paper)

Current Condition/Problem:

- Often equipped with only their domain expertise OR big keywords search algorithms, researchers (academic or industry) often scrounge through endless papers to find an innovative topic
- This limitation does not enable researchers to see beyond their own domain OR domain specific key words

Solution and Hypothesis:

- We hypothesize that innovation and impact comes at the connection of different fields and the paper's attributes (e.g. abstract choice of words, text). As a result we aim to accomplish:
  - **Analytical:** Build two models to predict citations (impact proxy)
  - **Knowledge Base:** Develop the underlying knowledge base for relationships between different fields of study in academic research
  - **Mission**: Foster collaborate disparate fields or across disciplines

Data Sources: We ingested two data sources (Semantic Scholar and SCHIMAGO Institutions Rankings)

Data Source	AI2 Allen Institute for AI / Semantic Scholar	SCIMAGO INSTITUTIONS RANKINGS
Rationale	Individual academic paper text data as well as other features (e.g. journal, authors, in-citations, out-citations)	Journals external benchmarking for impact factors (relevant importance)
Visual	{ "id": "4cd223df721b722b1c40689caa52932a41fcc223", "title": "Knowledge-rich, computer-assisted composition of Chinese couplets", "paperAbstract": "Recent research effort in poem composition has focused on the use of automatic language generation...", "entities": ["Conformance testing", "Natural language generation", "Natural language processing", "Parallel computing", "Stochastic grammar", "Web application"], "fieldsOfStudy": ["Computer Science"], "s2Url": "https://semanticscholar.org/paper/4cd223df721b722b1c40689caa52932a41fcc223", "s2PdfUrl": "", "pdfUrls": ["https://doi.org/10.1093/llc/fqu052"	

	Title	Type	SJR	H index	Total Docs. (2018)	Total Docs. (3years)	Total Refs. (2018)	Total Cites (3years)	Citable Docs. (3years)	Cites / Doc. (2years)	Ref. / Doc. (2018)
1	CA - A Cancer Journal for Clinicians	journal	72.576 Q1	144	45	127	3078	20088	103	206.85	68.40
2	MMWR. Recommendations and reports : Morbidity and mortality weekly report. Recommendations and reports / Centers for Disease Control	journal	48.894 Q1	134	3	12	559	1043	12	86.00	186.33
3	Nature Reviews Materials	journal	34.171 Q1	61	99	195	8124	7297	104	70.16	82.06

From the Original Dataset Size: ~270GB (includes +175MM Papers), **we are utilizing about 30GB (approximately 30 MM research paper and articles as part of data pipeline.**

Data Sources: We merged the two dataset to create a complete picture of both internal factors and external factors as well (e.g. journal impact factor – SJR)

Data Source

Ai2 Allen Institute for AI / Semantic Scholar

SCIMAGO INSTITUTIONS RANKINGS

Data Dictionary

Attribute	Dtype	Definition
Id	string	S2 generated research paper ID.
Title	string	Research paper title.
Paperabstract	string	Extracted abstract of the paper.
Entities	list	Extracted list of relevant entities or topics.
S2Url	string	URL to S2 research paper details page.
S2Pdfurl	string	URL to PDF on S2 if available.
Pdfurls	list	URLs related to this PDF scraped from the web.
Authors	list	List of authors with an S2 generated author ID and name.
Incitations	list	List of S2 paper IDs which cited this paper.
Outcitations	list	List of S2 paper IDs which this paper cited
Year	int	Year this paper was published as integer.
Venue	string	Extracted publication venue for this paper.
Journalname	string	Name of the journal that published this paper.
Journalvolume	string	The volume of the journal where this paper was published.
Journalpages	string	The pages of the journal where this paper was published.
Sources	list	Identifies papers sourced from DBLP or Medline.
Doi	string	Digital Object Identifier registered at doi.org.
Doiurl	string	DOI link for registered objects.
Pmid	string	Unique identifier used by PubMed.
Fieldsofstudy	list	Zero or more fields of study this paper addresses.`

Attribute	Dtype	Definition
Rank	int	Journal Rank
Sourceid	int	
Title	string	Name of journal
Type	string	Type of journal
Issn	string	
SJR	float	Composite Journal Score
…		
Country	string	
Publisher	string	
Coverage	string	
Categories	string	

The incoming data has a mixture of both text corpus and numerical values which we can uses as predictive features

Big Data Infrastructure: We relied on the usage of GCP and Spark in order to ingest data and extract insights

Data Acquisition Strategy:

1. *wget into HDFS*

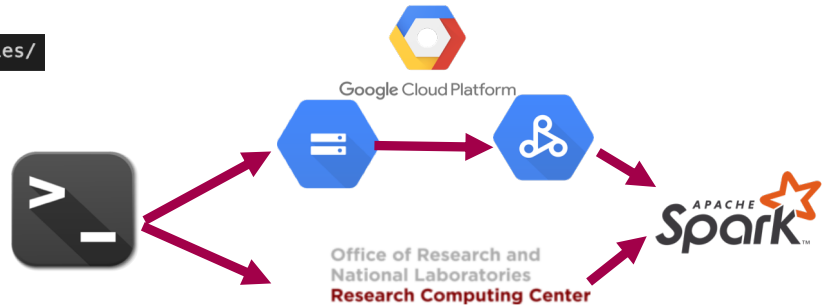
```
wget https://s3-us-west-2.amazonaws.com/ai2-s2-research-public/open-corpus/2020-02-01/manifest.txt
wget -qO- -B https://s3-us-west-2.amazonaws.com/ai2-s2-research-public/open-corpus/2020-01-13/ -i manifest.txt | hdfs dfs -put - /user/$USER/big_data_project
hdfs dfs -chmod 777 big_data_project/s2-corpus-*.gz
hdfs dfs -cat big_data_project/s2-corpus-*.gz | gunzip -d | hdfs dfs -put - big_data_project/
```

2. *Push to GCP Bucket from hdfs using gsutil*

```
hdfs dfs -cat big_data_project/*.json | gsutil -m cp -r - gs://semantic_scholar_files/unzipped_files/
```

Buckets / semantic_scholar_files / unzipped_files

Name	Size	Type	Storage class	Last modified
s2-corpus-000	1.48 GB	application/octet-stream	Standard	2/26/20, 12:12:16 AM UTC-6
s2-corpus-001	1.48 GB	application/octet-stream	Standard	2/26/20, 12:09:11 AM UTC-6
s2-corpus-002	1.49 GB	application/octet-stream	Standard	2/26/20, 12:10:33 AM UTC-6
s2-corpus-003	1.49	application/octet-	Standard	2/26/20, 12:10:55 AM UTC-



Data Processing & Modeling

1. Spin up cluster from command

```
gcloud beta dataproc clusters create ${CLUSTER_NAME} \
    --region=us-central1 \
    --zone=${ZONE} \
    --metadata 'PIP_PACKAGES=google-cloud-storage graphframes' \
    --worker-machine-type n1-standard-8 \
    --num-workers 4 \
    --image-version 1.4-debian9 \
    --initialization-actions gs://dataproc-initialization-actions/python/pip-install.sh, gs://dataproc-initialization-actions/spark-nlp/spark-nlp.sh \
    --optional-components=JUPYTER,ANACONDA \
    --max-idle 1h \
    --enable-component-gateway
```

Downloads Graphframes

4 Workers | 8vCPUS | 30GB

Autodeletes once idle

2. Submit jobs to cluster & wait…

```
gcloud dataproc jobs submit pyspark --cluster big-data-project-cluster1 --region us-central1 gs://semantic_scholar_files/01_read_write.py
gcloud dataproc jobs submit pyspark --cluster big-data-project-cluster1 --region us-central1 gs://semantic_scholar_files/02_modeling.py
gcloud dataproc jobs submit pyspark --cluster big-data-project-cluster1 --region us-central1 gs://semantic_scholar_files/03_Graph_Analysis.py
```

Data Analysis: In order to prepare the data we cleaned the data and processed the corpus of text to prep for the eventual feature engineering

Distinct Counts

	distinct_cnt
entities	1
s2PdfUrl	1
sources	5
year	85
outCitations_count	768
inCitations_count	2007
fieldsOfStudy	2816
journalName	50606
venue	63013
journalVolume	82485
journalPages	583782
pmid	2626646
outCitations	3150636
pdfUrls	3338194
inCitations	4818096
doiUrl	6957101
doi	6957166
authors	10084691
paperAbstract	11580963
title	11587325
s2Url	11625672
id	11625672

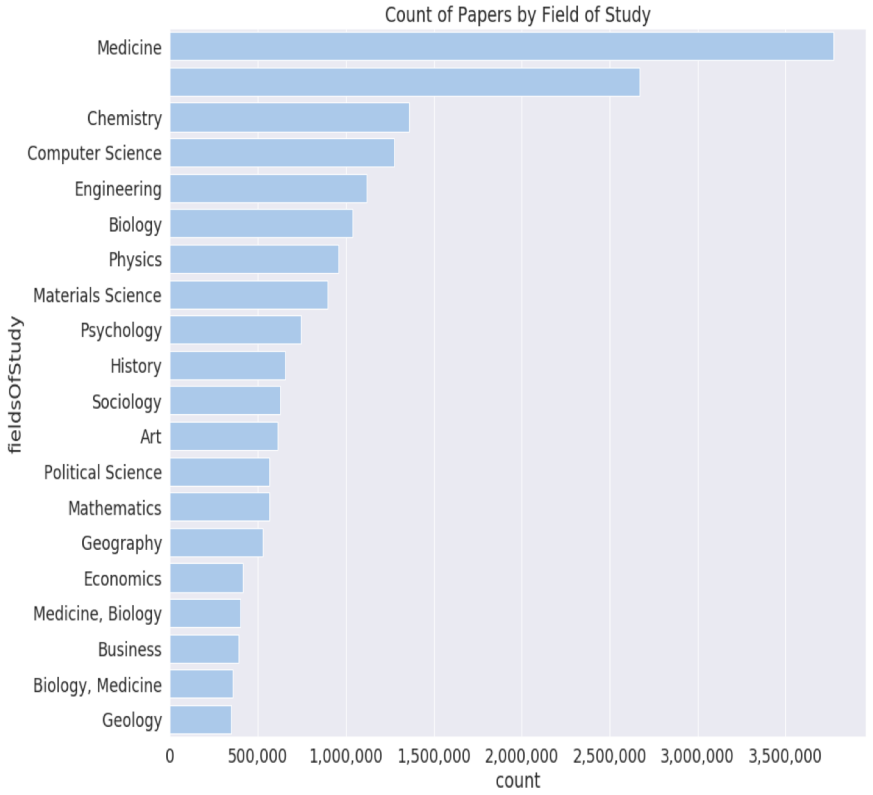
(after non-English and missing title/abstract rows)

1. Cleaning

1. Drop articles with no titles and/or abstracts
2. Impute
   1. Missing values for year of publication
   2. Missing values for SJR
3. Fixing some dtypes
   1. Unpacking arrays into strings & exploding

2. Text Preprocessing

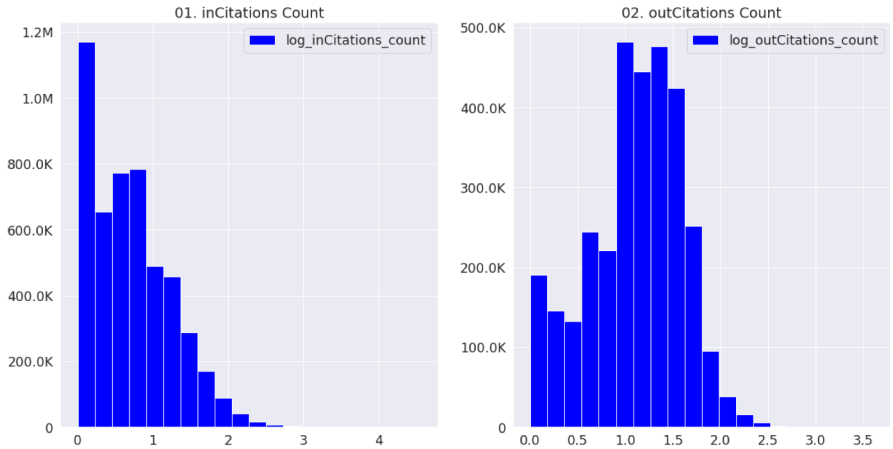
1. Clear non-alphanumerics
2. Tokenize the abstract & title
3. Remove Stop words
4. Filter short words
5. Lemmatize words



Data Analysis: We completed feature engineering and assembled them to be processed in machine learning models. In addition we utilized the word2vec features to grenade insights/similarities between papers

Feature Engineering

1. Create count columns for:
    1. inCitations (# of times a paper was cited) ~ *TARGET*
    2. outCitations (# of papers a paper cited)
    3. Authors
    4. Title wordcount
    5. Abstract wordcount
2. TFIDF using HashVectorizer
    1. Title
    2. Abstract
3. Word2Vec (i.e. Title2Vec)
4. Journal Prestige Score from SJR
    1. Nulls – Filled in values with 50% Percentile
5. Created One Hot Encodings of:
    1. Field of Study
    2. Sources



Word2Vec Insights

Defining cosine similarity

```
#Defining cosine similarity
def cos_sim(a,b):
    return float(np.dot(a, b) / (np.linalg.norm(a) * np.linalg.norm(b)))

cos_sim_udf = udf(lambda x, y: cos_sim(x,y), FloatType())
```

Creating function that takes query string as input and returns similar article titles.

String should be a list of words that are relevant to your query/search

```
def search_articles(query_string, model, n = 5):
    """Uses word embeddings trained on the corpus of research papers to find titles
    that are similar to a given search query. Provides URL for reference.

    Parameters:
    query_string (string): Description of arg1
    model (object) : pyspark Word2Vec Model instance
    n (int) : number of titles/articles to return

    Returns:
    DataFrame: Dataframe with the 'n' most similar titles to the given search query. Returns link to article.
    """

    #Put string into DF because spark W2V model only accepts a DF
    documentDF = spark.createDataFrame([(query_string.split(" "),)], ["title_tokens_filtered_lem"])
    #Get the mean vector of the query
    qry = model.transform(documentDF).collect()[0].title_wordVectors
    #Get the n most similar titles based on cosim vector distnace
    sim_titles = nlpdf.select('title','fieldsOfStudy', 'title_wordVectors', 's2Url')\
    .withColumn("coSim", cos_sim_udf(col("title_wordVectors"), array([lit(v) for v in qry])))\
    .dropna()\
    .orderBy('coSim',ascending = False).limit(n).toPandas()

    return sim_titles
```

```
pd.set_option('max_colwidth', 100)
#Let's test out the function
results_pdf = search_articles(query_string = "neural network hadoop", model = w2VM, n =5)
results_pdf
```

	title	fieldsOfStudy	title_wordVectors	s2Url	coSim
0	crossmedia hashing with neural networks	[Computer Science]	[2.1895049810409546, 0.3853525444865227, -0.5056682452559471, -0.1253671124575223, -0.849391877...	https://semanticscholar.org/paper/b5fe4731ff6a7a7f1ad8232186e84b1f944162e0	0.990144
1	selforganizing neural networks	[Computer Science]	[2.9634455839792886, 0.5228483875592549, -0.737404907743136, 0.15041999022165933, -1.28611646095...	https://semanticscholar.org/paper/a6ea2b1cbf69ee005b23f5bf1092be8ba684dda3	0.989763
2	data aggregation using neural network in wsn sing neural network in wsn sing neural network in w...	[Engineering]	[2.6220287945535445, 0.8723543286323547, -0.8760132657157049, -0.0028696780403455096, -0.6602880...	https://semanticscholar.org/paper/623a4d00878abfa680db8a069070a91775f85cce	0.989691

Examples

```
model.getVectors().count() = 16363
```

```
# Let's see how useful our wordvectors seem
w2VM.findSynonymsArray('neural', 10)
```

```
[('spiking', 0.8338825106620789),
 ('convolutional', 0.8299897313117981),
 ('rbf', 0.7949727177619934),
 ('artificial', 0.7792593240737915),
 ('adversarial', 0.7513853311538696),
 ('network', 0.7242907881736755),
 ('backpropagation', 0.7111539244651794),
 ('generative', 0.7054539918899536),
 ('unsupervised', 0.7009112238883972),
 ('clustered', 0.7007094621658325)]
```

```
# Let's see how useful our wordvectors seem
w2VM.findSynonymsArray('hadoop', 10)
```

```
[('mapreduce', 0.9746064543724 06),
 ('privacypreserving', 0.9414143562316895),
 ('querying', 0.9357867240905762),
 ('encrypted', 0.9334874153137207),
 ('scalability', 0.9307804703712463),
 ('multisource', 0.9288651943206787),
 ('virtualization', 0.9231449365615845),
 ('secured', 0.9222572445869446),
 ('cloudbased', 0.9167419075965881),
 ('runtime', 0.914556622505188)]
```


DEMONSTRATION

Classification: Now with all the data preparation, we would like to evaluate if we can at least predict if a paper will get cited – class 0 = No citations and class 1 = citations

Modeling Parameters

Training and Evaluation

3.2.1 Build Classes

In [51]:
```
from pyspark.sql import functions as f
modeling_data = modeling_data.withColumn('binary_class', f.when(f.col('inCitations_count')==0,0).otherwise(1))

from pyspark.ml.classification import LogisticRegression, LogisticRegressionModel

train_df, test_df = modeling_data.randomSplit([.7,.3],seed=100)
```

3.2.2 Set Model Paramters

In [52]:
```
# Set parameters for Logistic Regression
lgr = LogisticRegression(maxIter=5, featuresCol = 'features', labelCol='binary_class')
```

3.2.3 Train Model

In [53]:
```
# Fit the model to the data.
lgrm = lgr.fit(train_df)
```

3.2.4 Prediction

In [54]:
```
# Given a dataset, predict each point's label, and show the results.
predictions = lgrm.transform(test_df)
```

In [69]:
```
predictions_train = lgrm.transform(train_df)
```

Train and Test = 70:30
MaxIter: 5

Model Evaluations

Classification Coefficients

```
lgrm.coefficients
```


```
SparseVector(948, {0: -0.0007, 1: 0.0203, 2: -0.0005, 3: 0.0066, 4: -0.0572, 5: -0.0067, 6: -0.025, 7: 0.0593, 8: -0.1418,
9: -0.0286, 10: -0.1, 11: -0.0375, 12: -0.0094, 13: -0.0518, 14: 0.0505, 15: 0.0143, 16: 0.0955, 17: -0.0298, 18: 0.0504, 1
9: -0.0689, 20: 0.0139, 21: -0.0291, 22: -0.0309, 23: -0.0409, 24: -0.0593, 25: 0.0128, 26: -0.0355, 27: 0.1215, 28: 0.0835,
29: -0.0276, 30: -0.05, 31: -0.0143, 32: -0.0463, 33: 0.0241, 34: -0.0548, 35: 0.0511, 36: -0.114, 37: -0.1459, 38: -0.0788,
39: -0.119, 40: 0.0608, 41: -0.0809, 42: -0.0417, 43: -0.1277, 44: 0.0293, 45: 0.0047, 46: -0.1008, 47: -0.0121, 48: -0.073
6, 49: 0.1011, 50: -0.1746, 51: -0.0165, 52: 0.157, 53: -0.0175, 54: -0.042, 55: 0.0452, 56: 0.0601, 57: -0.0779, 58: 0.004,
59: 0.0816, 60: -0.035, 61: 0.1975, 62: 0.011, 63: -0.1218, 64: -0.0469, 65: -0.1313, 66: -0.0111, 67: 0.0246, 68: -0.0021,
69: -0.0652, 70: 0.0452, 71: 0.016, 72: 0.0045, 73: -0.1854, 74: -0.026, 75: -0.1158, 76: 0.068, 77: -0.1178, 78: -0.0378, 7
9: -0.0433, 80: 0.1114, 81: 0.0163, 82: -0.0587, 83: 0.021, 84: -0.0193, 85: -0.0632, 86: -0.0653, 87: 0.0167, 88: 0.0171, 8
9: 0.0473, 90: -0.0826, 91: 0.0104, 92: -0.0248, 93: -0.0215, 94: 0.0043, 95: -0.0531, 96: -0.0179, 97: 0.0502, 98: 0.0003,
99: -0.0195, 100: 0.0157, 101: 0.016, 102: -0.0357, 103: -0.0361, 104: -0.1077, 105: 0.0971, 106: 0.004, 107: -0.1555, 108:
-0.0712, 109: 0.2803, 110: -0.143, 111: -0.1309, 112: 0.1559, 113: -0.1198, 114: 0.1116, 115: 0.084, 116: -0.1317, 117: 0.09
46, 119: -0.1579, 120: 0.0044, 121: 0.0766, 122: 0.0039, 124: 0.0047, 125: 0.084, 126: 0.1655, 127: 0.2063, 128: -0.0406, 12
9: -0.1848, 131: -0.1449, 132: -0.1442, 133: 0.004, 134: -0.1613, 135: 0.1482, 136: -0.1567, 137: 0.004, 138: -0.1581, 139:
-0.147, 140: 0.1695, 141: -0.1339, 142: 0.0242, 143: 0.0426, 144: -0.1051, 145: -0.0351, 146: -0.0599, 147: -0.0388, 148: -
0.177, 150: 0.0034, 151: 0.137, 152: 0.2838, 153: 0.0518, 154: -0.038, 155: 0.2737, 156: -0.153, 157: 0.0362, 158: -0.0398,
159: 0.1625, 160: 0.3208, 161: 0.0901, 162: 0.1404, 163: -0.1416, 164: 0.3197, 165: 0.0734, 166: 0.2895, 167: -0.0385, 169:
-0.1404, 170: -0.1336, 171: 0.0754, 172: 0.0037, 173: 0.068, 174: 0.0849, 175: 0.0788, 176: 0.0661, 177: -0.1553, 178: -0.04
62, 179: -0.1453, 180: 0.1441, 181: 0.0819, 182: -0.1431, 183: 0.1517, 184: 0.0041, 185: -0.1707, 186: -0.1423, 187: -0.061
6, 188: 0.0787, 189: 0.311, 190: -0.0217, 191: -0.1323, 192: -0.0539, 193: 0.0044, 194: 0.2153, 195: -0.1614, 196: -0.1458,
197: -0.0191, 198: 0.149, 199: -0.0622, 200: 0.0412, 201: 0.2687, 202: -0.1436, 203: -0.1638, 204: -0.0183, 205: 0.0506, 20
6: 0.3543, 207: -0.4191, 208: 0.2178, 209: 0.2095, 210: 0.2081, 211: -0.3867, 212: -0.3785, 213: 0.2072, 215: 0.7808, 216:
0.2088, 218: -0.376, 219: -0.3861, 220: -0.3863, 221: -0.3763, 222: 0.7824, 223: -0.376, 224: 0.7835, 225: -0.376, 227: -0.3
805, 228: 0.781, 230: -0.3867, 894: -0.3362, 895: 0.4222, 896: -0.3761, 898: -0.1205, 899: -0.0823, 900: -0.2865, 901: -0.15
32, 902: -0.1853, 903: 0.0831, 904: -0.0999, 905: 0.2195, 906: -0.2094, 907: 0.086, 908: 0.014, 909: -0.0179, 910: 0.1394, 9
11: -0.0137, 912: -0.106, 913: 0.0587, 914: 0.0782, 915: 0.1736, 916: 0.1516, 917: -0.1119, 918: 0.0427, 919: 0.0924, 920:
0.2123, 921: 0.1809, 922: 0.2358, 923: -0.1256, 924: 0.1006, 925: -0.1248, 926: 0.0081, 927: 0.1769, 928: -0.1074, 929: -0.0
102, 930: 0.2526, 931: -0.0009, 932: -0.1766, 933: -0.0196, 934: -0.1222, 935: 0.2725, 936: 0.0068, 937: -0.2709, 938: -0.10
88, 939: 0.1495, 940: -0.0133, 941: 0.1275, 942: -0.1794, 943: -0.009, 944: 0.1145, 945: -0.0558, 946: -0.1272, 947: -0.105
6})
```


Train Accuracy = 92.56%
Train F1 = 92.57%

Test Accuracy = 78.57%
Test F1 = 69.14%

Insight: Classification uses all features and the word embed index drives prediction; however there is some room for regularization as the model seems to be overfitting

Regression: With the given data, how likely we are to predict the number of citations

Modeling Parameters

Training and Evaluation

```
train_df, test_df = modeling_data.randomSplit([.7,.3],seed=100)
```

```
from pyspark.ml.regression import LinearRegression

#Elastic Net
lr = LinearRegression(featuresCol = 'features', labelCol='inCitations_count', regParam=0.3, elasticNetParam=0.8, maxIter
lrm = lr.fit(train_df)

#coefficients
print("Coefficients: " + str(lrm.coefficients))
print("Intercept: " + str(lrm.intercept))

#model summary
print("RMSE: %f" % lrm.summary.rootMeanSquaredError)
print("r2: %f" % lrm.summary.r2)
```

Train and Test = 70:30
Regularization Parameter: 0.3
ElasticNetParam: 0.8
MaxIter: 10

Model Evaluations

Regression Coefficients

```
Coefficients: [7.605314146598465e-05,0.02126875438873086,0.0,0.0,-0.09717637658842729,0.0,0.0,0.0,0.0,-0.1007207211691633,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.10151171596710952,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.056724720532019976,0.0,0.0,0.0,0.0,0.0,-0.01
6339881214798666,0.0,0.0,0.0,0.0,0.0,-0.03573362832578906,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-0.12488981608379021,0.0,0.0,0.0,0.0,0.0,
0.0,0.12748259690449038,0.0,0.0,0.0,0.0,0.14140372026672063,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-0.08012973525615243,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.05167
210295557288,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-0.31765741568011296,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.002270977425235093,0.6166732434929431,0.0,0.0,0.0,0.0,-0.19325884452968847,0.2482196011562018,-0.135829406505911
88,0.0,0.0,0.0,-0.23269249163639208,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0156560214
78736428,0.8496340837561515,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-0.01654068019076566,0.0,-0.11423177231483461,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-0.07745621844427951,0.0,0.0,-0.0642553897882301,0.0,0.0,0.0,0.0,0.543552215869053,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.8761140193168812,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-0.6107993669969527,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-0.5436344949689229,0.
7197823699953309,0.0,0.0,0.0,0.0,-0.0864878205394704,0.0,0.0,0.0,-0.01962336994297925,0.10497994690519784,-0.039604265361673
24,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.20966183000414151,0.0,-0.11405098170070493,0.0,0.0,0.0,0.0,0.07683614043515628,0.0571758339
77832956,0.0,0.1424317444144489,0.0,-0.018807287141273995,0.0,0.0,0.0,0.0,0.15639752892169376,0.0,0.0,0.0,0.0,0.0,0.578758397237948
5,0.0,-0.11462239739087121,0.0,0.0,0.0,0.0,-0.08450130945332068,0.0,0.19693597774012392,0.0,0.0,0.0]
```

Train RMSE: 0.994727

Test RMSE: 2.323

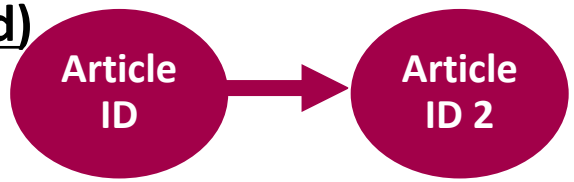
Insight: Regression does not seem to be using the word embedding as a predictor and the same issue of overfitting is still present

Graph: We also wanted to supplement to build a basic knowledge base into the paper's attributes and their relations to each other

1. Research Paper Graph (Directed)

Nodes : Articles = 11,300,912

Edges : Citations = 28,890,874

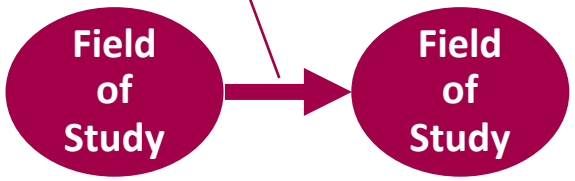


2. Fields of Study Graph (Directed)

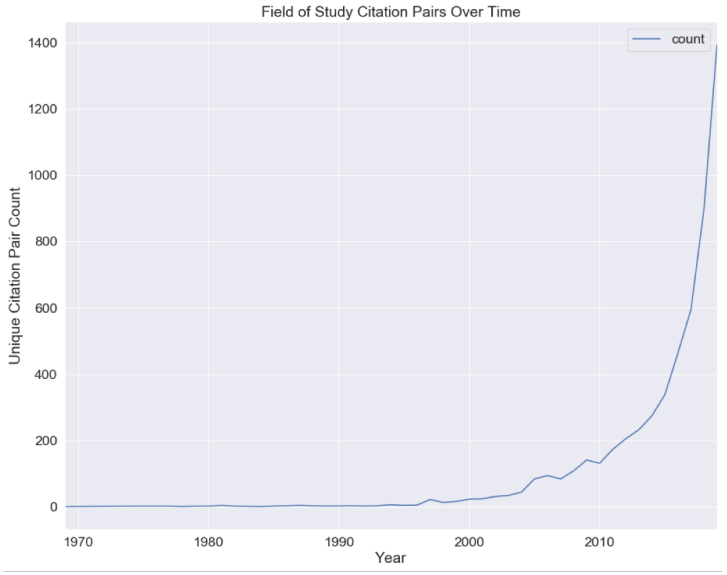
Nodes : Field of Study = 644

Edges : Citations = 211,619

Weight = citation count



Unique FOS Citation Pairs over time



InDegrees Discovery: Top 5 Papers by Citation Count

	title	fieldsOfStudy	year	paperAbstract	author_names	inCitations_count	outCitations_count
0	r a language and environment for statistical computing	Computer Science	2014	copyright r foundation for statistical computing permission is granted to make and distribute ...	[R Core Team]	36091	0
1	imagenet classification with deep convolutional neural networks	Computer Science	2012	we trained a large deep convolutional neural network to classify the million highresolution ima...	[Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton]	30738	17
2	statistical methods for assessing agreement between two methods of clinical measurement	Medicine	1986	in clinical measurement comparison of a new measurement technique with an established one is oft...	[J Martin Bland, Douglas G. Altman]	17060	0
3	evaluating structural equation models with unobservable variables and measurement error	Business	1981	the statistical tests used in the analysis of structural equation models with unobservable varia...	[Claes Fornell, David F. Larcker]	14059	0
4	gene ontology tool for the unification of biology	Biology, Medicine	2000	genomic sequencing has made it clear that a large fraction of the genes specifying the core biol...	[Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Micha...	16531	25

Motifs Discovery: Top 5 Bi-referenced papers by Citation Count

g.find("(a)-[e]->(b); (b)-[e2]->(a)")

title_a	fieldsOfStudy_a	year_a	author_names_a	inCitations_count_a	title_b	fieldsOfStudy_b	year_b	author_names_b	inCitations_count_b
principles of fluorescence spectroscopy	Chemistry	1983	[Joseph R. Lakowicz]	7745	mechanisms of tryptophan fluorescence shifts in proteins	Medicine, Chemistry	2001	[Jean-Marc Vivian, Patrik R. Callis]	328
multiplex genome engineering using crisprcas systems	Biology, Medicine	2013	[Lê Chí Công, Fei Ann Ran, David Cox, Shuailiang Lin, Robert P. J. Barretto, Naomi Habib, Patric...	4660	rnaguided human genome engineering via cas	Medicine, Biology	2013	[Prashant R. Mali, Luhan Yang, Kevin M Esvelt, John Aach, Marc Guell, James E. DiCarlo, Julie E....	2937
geodesic active contours	Computer Science	1995	[Vicent Caselles, Ron Kimmel, Guillermo Sapiro]	3991	gradient flows and geometric active contour models	Computer Science	1995	[Satyanad Kichenassamy, Arun Kumar, Peter J. Olver, Allen R. Tannenbaum, Anthony J. Yezzi]	536
rnaguided human genome engineering via cas	Medicine, Biology	2013	[Prashant R. Mali, Luhan Yang, Kevin M Esvelt, John Aach, Marc Guell, James E. DiCarlo, Julie E....	2937	multiplex genome engineering using crisprcas systems	Biology, Medicine	2013	[Lê Chí Công, Fei Ann Ran, David Cox, Shuailiang Lin, Robert P. J. Barretto, Naomi Habib, Patric...	4660
longterm recurrent convolutional networks for visual recognition and description	Computer Science, Medicine	2014	[Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, ...	1710	translating videos to natural language using deep recurrent neural networks	Computer Science	2014	[Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, Kate Saenko]	480

In building the graph knowledge base, we also were able to analyze a variety of other features

Page Ranks

fieldsOfStudy	id	pagerank
Medicine	Medicine	95.31182392139145
Medicine, Biology	Medicine, Biology	78.91553539374645
Biology, Medicine	Biology, Medicine	62.17956393990782
Psychology, Medicine	Psychology, Medicine	33.75207371641937
Computer Science	Computer Science	32.388632585174584
Mathematics, Comp...	Mathematics, Comp...	28.89974693889713
Biology	Biology	15.256319460684747
Mathematics	Mathematics	15.186701672368748
Physics	Physics	14.964267300651692
Medicine, Chemistry	Medicine, Chemistry	14.328605844937979

Triangle Counts

count	FOS_id	fieldsOfStudy	id
2353	1.0	Computer Science	Computer Science
2346	11.0	Biology, Medicine	Biology, Medicine
2296	9.0	Medicine, Biology	Medicine, Biology
2183	0.0	Medicine	Medicine
2128	25.0	Mathematics, Comp...	Mathematics, Comp...
1893	8.0	Mathematics	Mathematics
1886	22.0	Psychology, Medicine	Psychology, Medicine
1806	32.0	Computer Science,...	Computer Science,...
1794	5.0	no_FoS	no_FoS
1790	20.0	Medicine, Chemistry	Medicine, Chemistry

Most Frequent Citation Pairs (unique FOS)

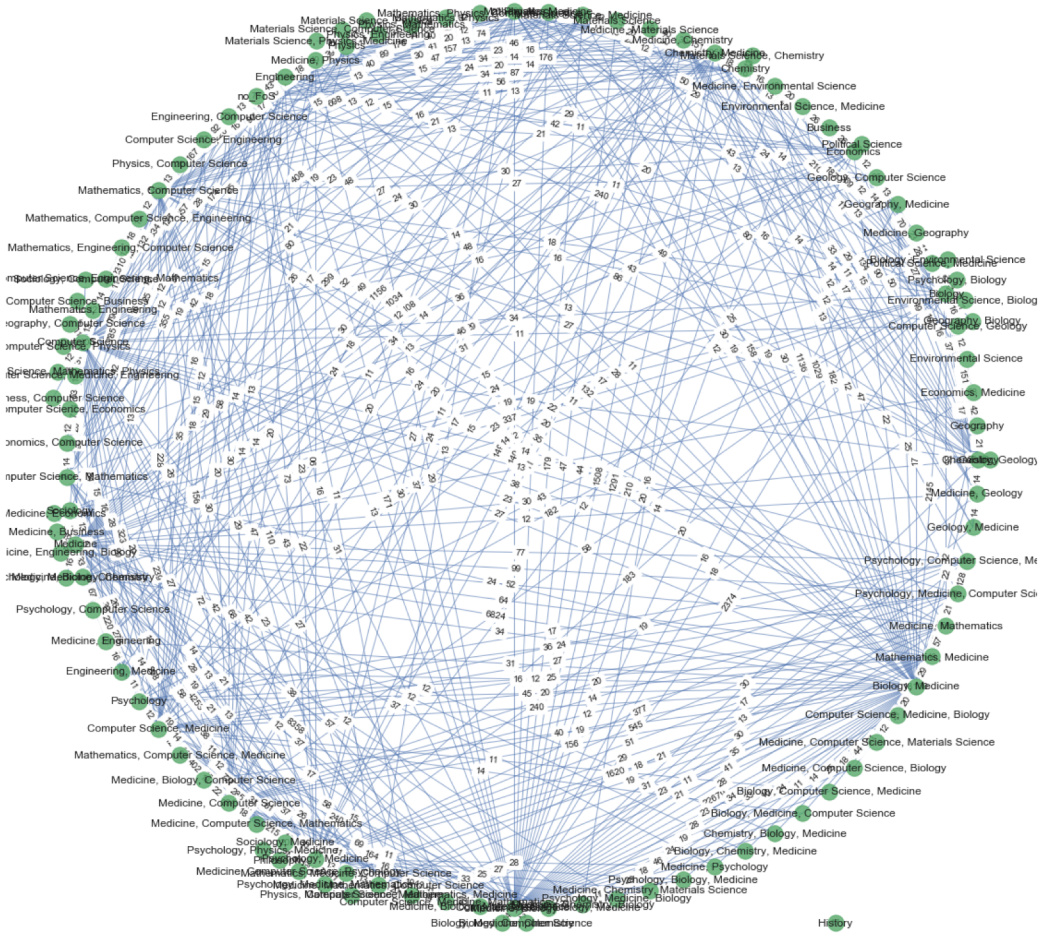
src	dst	count
Medicine	Medicine	102022
Computer Science	Computer Science	12306
Biology	Biology	9101
Physics	Physics	4534
Mathematics	Mathematics	2738

Least Frequent Citation Pairs (unique FOS)

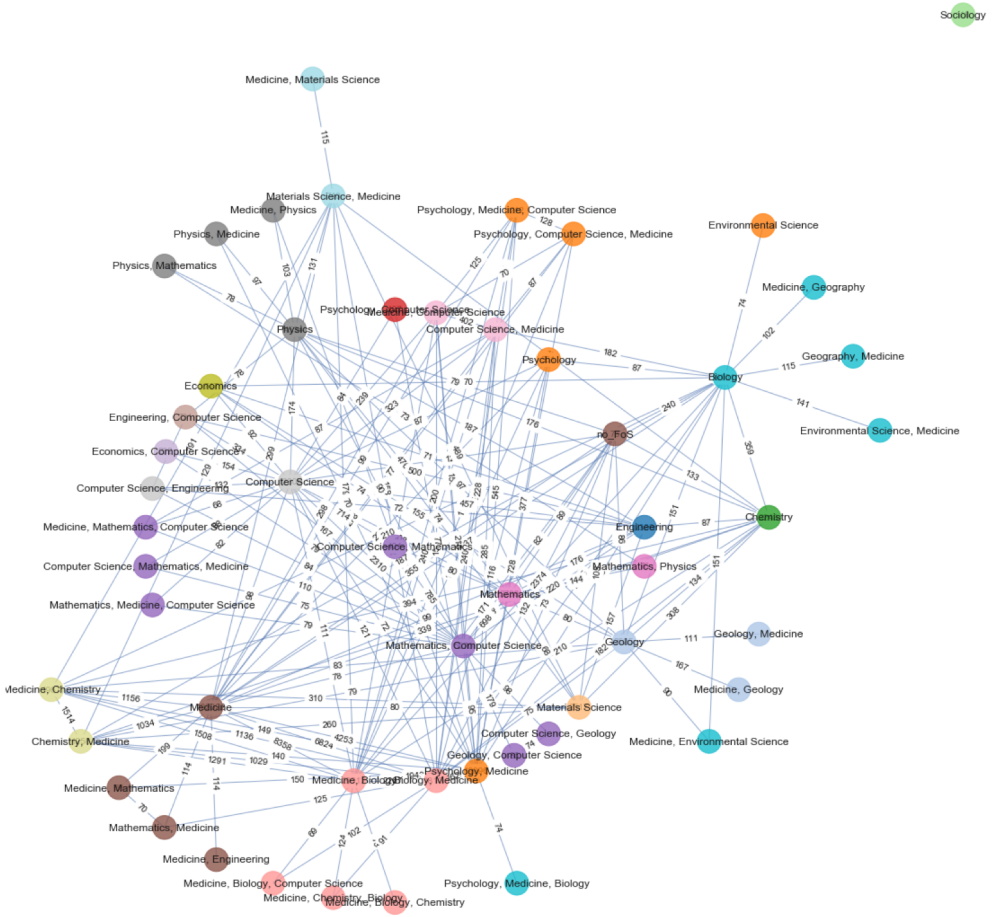
src	dst	count
Geography	Business	1
Physics	Sociology	1
Art	Psychology	1
History	Environmental Science	1
Philosophy	Geology	1

We also visualized the graphs in variety of architectures

Top 1000 Citation Pairs



Top 300 Citation Pairs Colored by LP Algorithm Communities



Label Propagation found a total of 25 communities.
Total of 290 Strong connected components

Conclusion & Next Steps

Conclusion:

With building machine learning models and graph knowledge base, this pipeline can be the foundation for which researchers can use to start to understand how to secure more citations and identify where innovation may exist in the academic body of knowledge.

Improvements & Next Steps:

Data Sources:

- Ingest more contextual information on the paper itself (beyond Abstract)
- Expand the subset of journals beyond biology or medicine related

Machine Learning

- Employ cross-validation to prevent overfitting of machine learning models

Graph Knowledge Base:

- Fields of Study disambiguation since one hot encoding for unique combinations
- Graph Studies on the articles (since the number of unique articles >> number of unique fields of study)