Graph Theory: Neo4j Microsoft News Logs Dataset (MINDS)

PES1UG20CS620
Adarsh Subhas Nayak
Section: K

PES1UG20CS629
B G Sourav
Section: K

Introduction to Dataset:

Microsoft News Logs Dataset (MINDS)

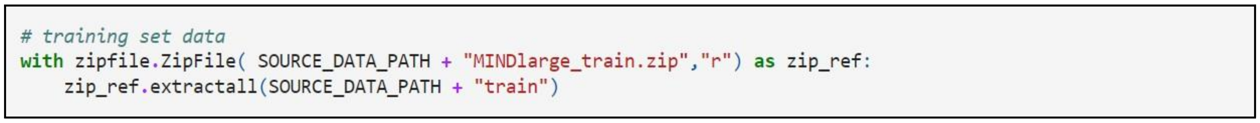
Microsoft News Dataset (MIND) is a Large Dataset that is a sample of 1 million anonymised users and their click behaviors collected from the Microsoft News website.

MIND contains about 160k English news articles and more than 15 million impression logs generated by 1 million users. Every news article contains rich textual content including title, abstract, body, category and entities. Each impression log contains the click events, non-clicked events and historical news click behaviors of this user before this impression. To protect user privacy, each user was de-linked from the production system when securely hashed into an anonymised ID.

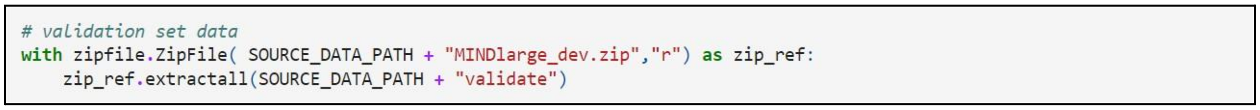
Dataset Reference: Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu and Ming Zhou. MIND: A Large-scale Dataset for News Recommendation. ACL 2020.

Graph Database:

Unzip Source data



```
# training set data
with zipfile.ZipFile( SOURCE_DATA_PATH + "MINDlarge_train.zip","r") as zip_ref:
    zip_ref.extractall(SOURCE_DATA_PATH + "train")
```



```
# validation set data
with zipfile.ZipFile( SOURCE_DATA_PATH + "MINDlarge_dev.zip","r") as zip_ref:
    zip_ref.extractall(SOURCE_DATA_PATH + "validate")
```

News Related Graph Elements

```
col_names = ['newsId', 'category', 'subcategory', 'title', 'abstract', 'url', 'titleEntities','abstractEntities']
news_train_df = pd.read_csv('./data/train/news.tsv', sep = '\t', header = None, names = col_names)
news_validate_df = pd.read_csv('./data/validate/news.tsv', sep = '\t', header = None, names = col_names)
```

```
news_df = news_and_entities_df.drop(columns = ['titleEntities', 'abstractEntities'])
```

News Nodes

```
news_df.to_csv(f'{NEO4J_IMPORT_DIR}/news.csv', index = False)
news_df
```

	newsId	category	subcategory	title	abstract	url
0	N88753	lifestyle	lifestyleroyals	The Brands Queen Elizabeth, Prince Charles, an...	Shop the notebooks, jackets, and more that the...	https://assets.msn.com/labs/mind/AAGH0ET.html
1	N45436	news	newsscienceandtechnology	Walmart Slashes Prices on Last-Generation iPads	Apple's new iPad releases bring big deals on l...	https://assets.msn.com/labs/mind/AABmf2I.html
2	N23144	health	weightloss	50 Worst Habits For Belly Fat	These seemingly harmless habits are holding yo...	https://assets.msn.com/labs/mind/AAB19MK.html
3	N86255	health	medical	Dispose of unwanted prescription drugs during ...	NaN	https://assets.msn.com/labs/mind/AAISxPN.html
4	N93187	news	newsworld	The Cost of Trump's Aid Freeze in the Trenches...	Lt. Ivan Molchanets peeked over a parapet of s...	https://assets.msn.com/labs/mind/AAJgNsz.html
...	...	...	...	...	...	...
172942	N17002	tv	tvnews	More millennials looking to slow aging process...	Who wouldn't love to turn back Father Time and...	https://assets.msn.com/labs/mind/BBWr3iL.html
172944	N91753	lifestyle	lifestylehoroscope	Which Zodiac Is The Nicest? Zodiac Ranking Fro...	Is your zodiac sign the nicest or the meanest?...	https://assets.msn.com/labs/mind/BBWIskd.html
173129	N1106	travel	internationaltravel	Overindulge in Delicious Snacks, Meals, and Tr...	Your appetite's about to be on easy street.	https://assets.msn.com/labs/mind/BBWD0dt.html
173166	N37954	tv	tvnews	'The Walking Dead' star Josh McDermitt says Th...	Josh McDermitt, who plays Eugene, tells Inside...	https://assets.msn.com/labs/mind/BBWyOPQ.html
173468	N115690	sports	baseball_mlb	Baseball history unpacked, November 8	Walton's RoY, Perez re-signs, and other stories	https://assets.msn.com/labs/mind/BBWsWhf.html

Category/Subcategory Nodes and Relationships

```
sub_category_rel_df = news_df[['newsId','category', 'subcategory']]
sub_category_rel_df.to_csv(f'{NEO4J_IMPORT_DIR}/news-belongs-to-sub-category.csv', index = False)
sub_category_rel_df
```

```
category_rel_df = news_df[['category','subcategory']].drop_duplicates()
category_rel_df.to_csv(f'{NEO4J_IMPORT_DIR}/sub-category-of-category.csv', index = False)
category_rel_df
```

```
category_df = category_rel_df[['category']].drop_duplicates()
category_df.to_csv(f'{NEO4J_IMPORT_DIR}/categories.csv', index = False)
category_df
```

```
sub_category_df = category_rel_df[['subcategory']]
sub_category_df
```

Prepare Title WikiData Entities

```
raw_title_entity_df = news_and_entities_df.drop(
    columns = ['category', 'subcategory', 'title', 'abstract', 'url', 'abstractEntities'])
```

```
def get_entities(entity_string):
    entity_list = []
    if not pd.isna(entity_string):
        entity_list = ast.literal_eval(entity_string)
    return entity_list

raw_title_entity_df['titleEntitiesFormatted'] = raw_title_entity_df.titleEntities.apply(get_entities)
raw_title_entity_df
```

Import User

```
user_df = pd.concat(user_dfs, ignore_index=True).drop_duplicates()
user_df.to_csv(f'{NEO4J_IMPORT_DIR}/users.csv', index=False)
user_df
```

Approximate News Times

```
news_approx_time_df = pd.concat(news_approx_time_dfs, ignore_index=True)\
    .groupby('newsId').agg({'approxTime':'min'}).reset_index()
news_approx_time_df.to_csv(f'{NEO4J_IMPORT_DIR}/news-approx-time.csv', index=False)
news_approx_time_df
```

Load Data Into Neo4j

```python
## Using an ini file for credentials, otherwise providing defaults
HOST = 'neo4j://localhost'
DATABASE = 'neo4j'
PASSWORD = 'password'

if NEO4J_PROPERTIES_FILE is not None and os.path.exists(NEO4J_PROPERTIES_FILE):
    config = configparser.RawConfigParser()
    config.read(NEO4J_PROPERTIES_FILE)
    HOST = config['NEO4J']['HOST']
    DATABASE = config['NEO4J']['DATABASE']
    PASSWORD = config['NEO4J']['PASSWORD']
    print('Using custom database properties')
else:
    print('Could not find database properties file, using defaults')
```

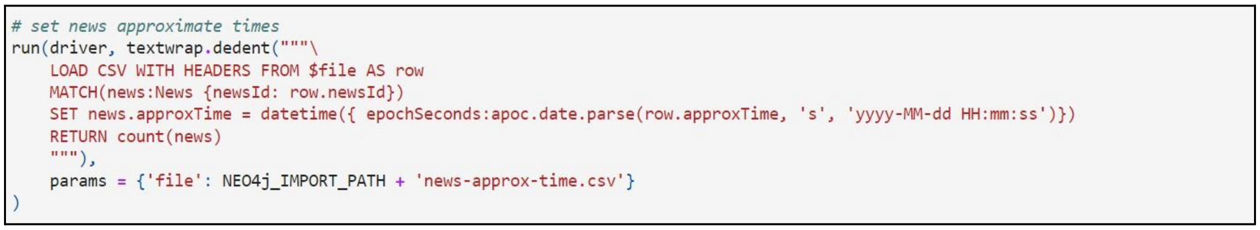
```python
# helper function
def run(driver, query, params=None):
    with driver.session() as session:
        if params is not None:
            return [r for r in session.run(query, params)]
        else:
            return [r for r in session.run(query)]
```

```python
driver = GraphDatabase.driver(HOST, auth=(DATABASE, PASSWORD))
```

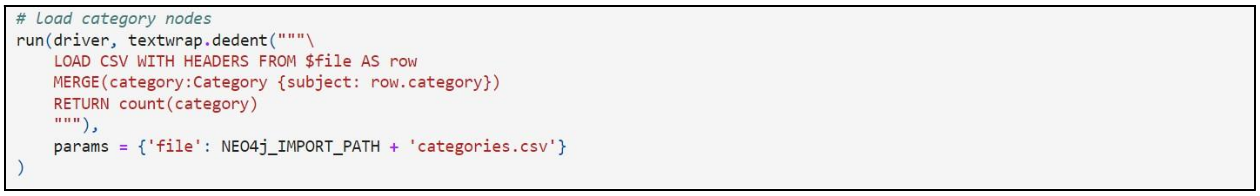
Load Nodes

```python
# user nodes
run(driver, textwrap.dedent("""\
    USING PERIODIC COMMIT 10000
    LOAD CSV WITH HEADERS FROM $file AS row
    MERGE(user:User {userId:row.userId})
    RETURN count(user)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'users.csv'}
)
```

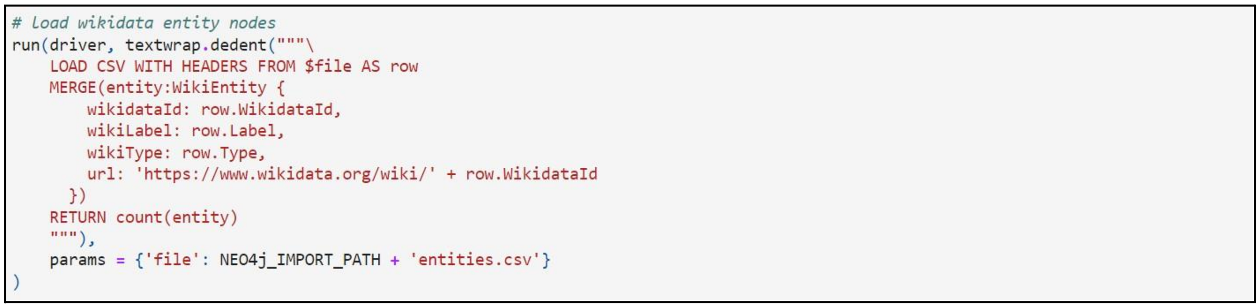
```python
# news nodes
run(driver, textwrap.dedent("""\
    LOAD CSV WITH HEADERS FROM $file AS row
    MERGE(news:News {
      newsId: row.newsId,
      category: row.category,
      subcategory: row.subcategory,
      title: row.title,
      url: row.url})
    WITH row, news
    WHERE row.abstract IS NOT null
    SET news.abstract = row.abstract
    RETURN count(news)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'news.csv'}
)
```



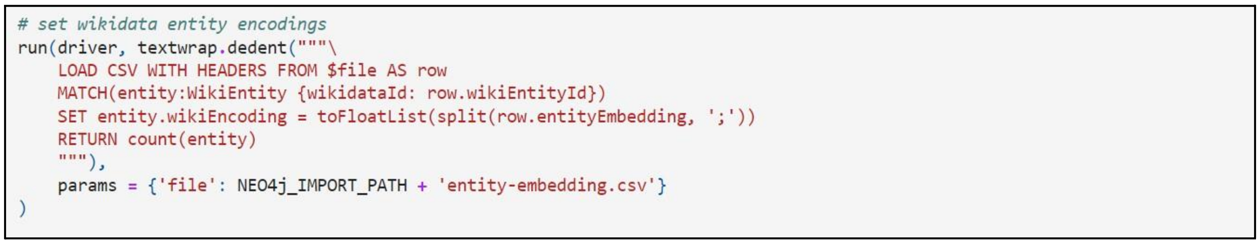
```
# set news approximate times
run(driver, textwrap.dedent("""\
    LOAD CSV WITH HEADERS FROM $file AS row
    MATCH(news:News {newsId: row.newsId})
    SET news.approxTime = datetime({ epochSeconds:apoc.date.parse(row.approxTime, 's', 'yyyy-MM-dd HH:mm:ss')})
    RETURN count(news)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'news-approx-time.csv'}
)
```



```
# load category nodes
run(driver, textwrap.dedent("""\
    LOAD CSV WITH HEADERS FROM $file AS row
    MERGE(category:Category {subject: row.category})
    RETURN count(category)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'categories.csv'}
)
```

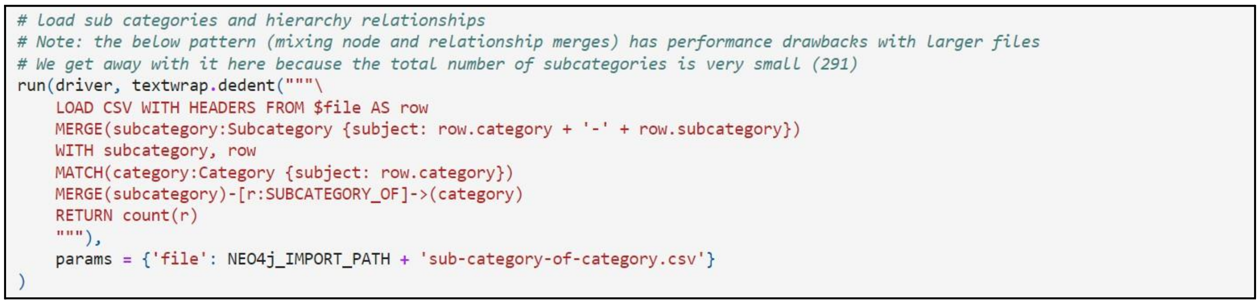


```
# load wikidata entity nodes
run(driver, textwrap.dedent("""\
    LOAD CSV WITH HEADERS FROM $file AS row
    MERGE(entity:WikiEntity {
        wikidataId: row.WikidataId,
        wikiLabel: row.Label,
        wikiType: row.Type,
        url: 'https://www.wikidata.org/wiki/' + row.WikidataId
      })
    RETURN count(entity)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'entities.csv'}
)
```



```
# set wikidata entity encodings
run(driver, textwrap.dedent("""\
    LOAD CSV WITH HEADERS FROM $file AS row
    MATCH(entity:WikiEntity {wikidataId: row.wikiEntityId})
    SET entity.wikiEncoding = toFloatList(split(row.entityEmbedding, ';'))
    RETURN count(entity)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'entity-embedding.csv'}
)
```

Load Relationships



```
# load sub categories and hierarchy relationships
# Note: the below pattern (mixing node and relationship merges) has performance drawbacks with larger files
# We get away with it here because the total number of subcategories is very small (291)
run(driver, textwrap.dedent("""\
    LOAD CSV WITH HEADERS FROM $file AS row
    MERGE(subcategory:Subcategory {subject: row.category + '-' + row.subcategory})
    WITH subcategory, row
    MATCH(category:Category {subject: row.category})
    MERGE(subcategory)-[r:SUBCATEGORY_OF]->(category)
    RETURN count(r)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'sub-category-of-category.csv'}
)
```

```
# load news-subcategory relationships
run(driver, textwrap.dedent("""\
    LOAD CSV WITH HEADERS FROM $file AS row
    MATCH(subcategory:Subcategory {subject: row.category + '-' + row.subcategory})
    MATCH(news:News {newsId: row.newsId})
    MERGE(news)-[r:BELONGS_TO_SUBCATEGORY]->(subcategory)
    RETURN count(r)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'news-belongs-to-sub-category.csv'}
)
```

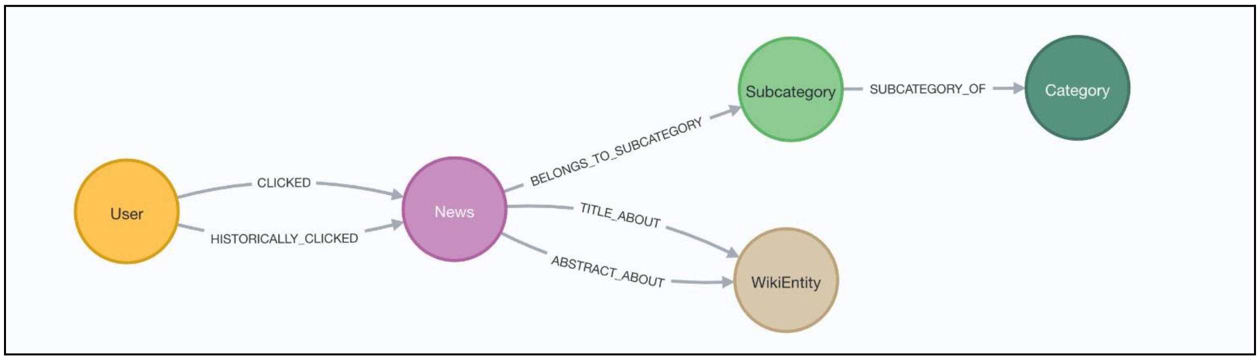
```
# load news-title-about-wikiData-entity relationships
run(driver, textwrap.dedent("""\
    LOAD CSV WITH HEADERS FROM $file AS row
    MATCH(news:News {newsId: row.newsId})
    MATCH(entity:WikiEntity {wikidataId: row.WikidataId})
    MERGE(news)-[r:TITLE_ABOUT{ confidence: toFloat(row.Confidence)}]->(entity)
    RETURN count(r)
    """),
    params = {'file': NEO4j_IMPORT_PATH + 'news-title-about.csv'}
)
```

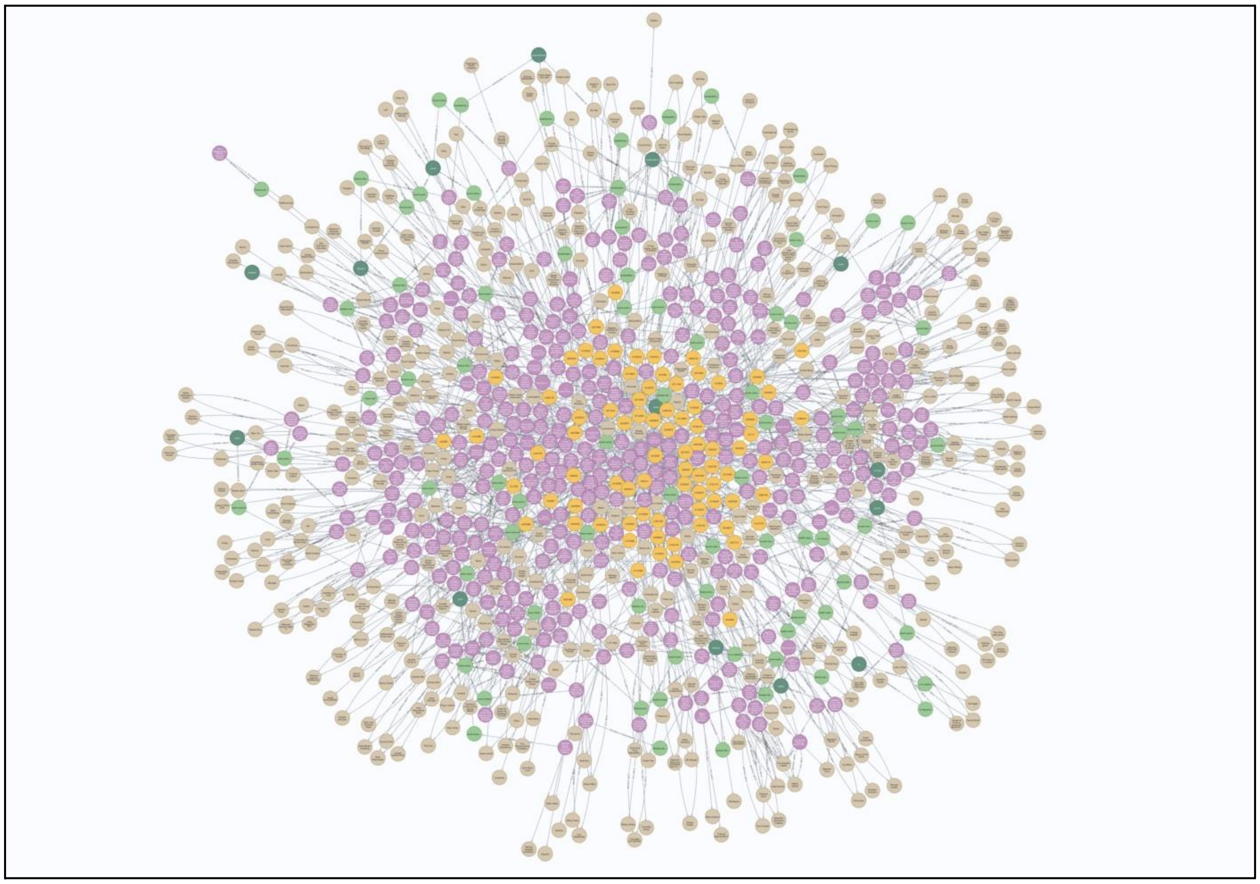
```
# load user click relationships
for i in range(click_iters):
    print(f'Loading clicks {i + 1} of {click_iters}')
    run(driver, textwrap.dedent("""\
        LOAD CSV WITH HEADERS FROM $file AS row
        MATCH(user:User {userId: row.userId})
        MATCH(news:News {newsId: row.newsId})
        MERGE(user)-[r:CLICKED {
          splitSet: row.splitSet,
          impressionId: row.impressionId,
          impressionTime: datetime({ epochSeconds:apoc.date.parse(row.time, 's', 'yyyy-MM-dd HH:mm:ss')})
        }]->(news)
        RETURN count(r)
        """),
        params = {'file': f'{NEO4j_IMPORT_PATH}clicks-{i}.csv'}
    )
```

```
# load user historic click relationships
for i in range(historic_click_iters):
    print(f'Loading historic clicks {i + 1} of {historic_click_iters}')
    run(driver, textwrap.dedent("""\
        LOAD CSV WITH HEADERS FROM $file AS row
        MATCH(user:User {userId: row.userId})
        MATCH(news:News {newsId: row.newsId})
        MERGE(user)-[r:HISTORICALLY_CLICKED {splitSet: row.splitSet}]->(news)
        RETURN count(r)
        """),
        params = {'file': f'{NEO4j_IMPORT_PATH}historic-clicks-{i}.csv'}
    )
```

Visualise the graph

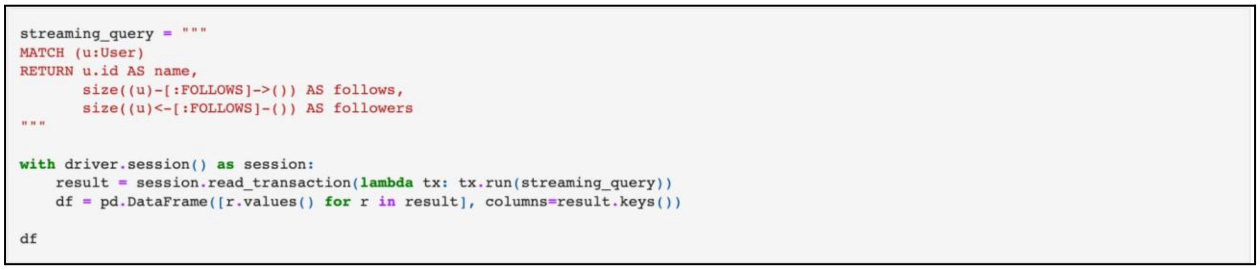
```
neo4j$ CALL db.schema.visualization();
```


User
CLICKED
HISTORICALLY_CLICKED
News
BELONGS_TO_SUBCATEGORY
Subcategory
SUBCATEGORY_OF
Category
TITLE_ABOUT
ABSTRACT_ABOUT
WikiEntity

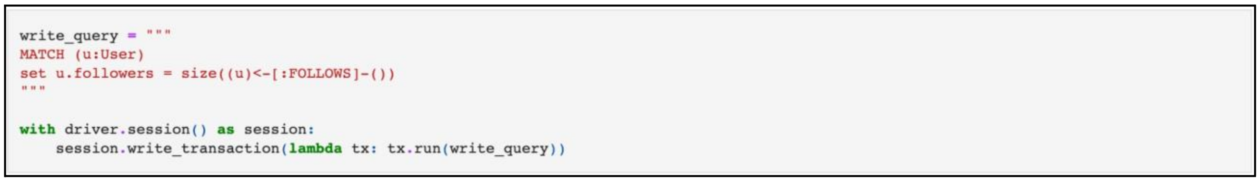


Centrality Measure

Degree Centrality



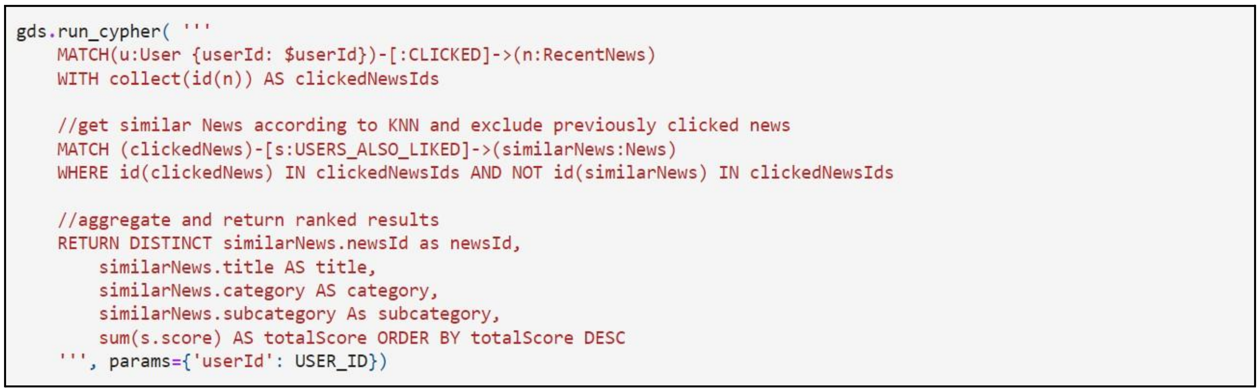
```
streaming_query = """
MATCH (u:User)
RETURN u.id AS name,
       size((u)-[:FOLLOWS]->()) AS follows,
       size((u)<-[:FOLLOWS]-()) AS followers
"""

with driver.session() as session:
    result = session.read_transaction(lambda tx: tx.run(streaming_query))
    df = pd.DataFrame([r.values() for r in result], columns=result.keys())

df
```



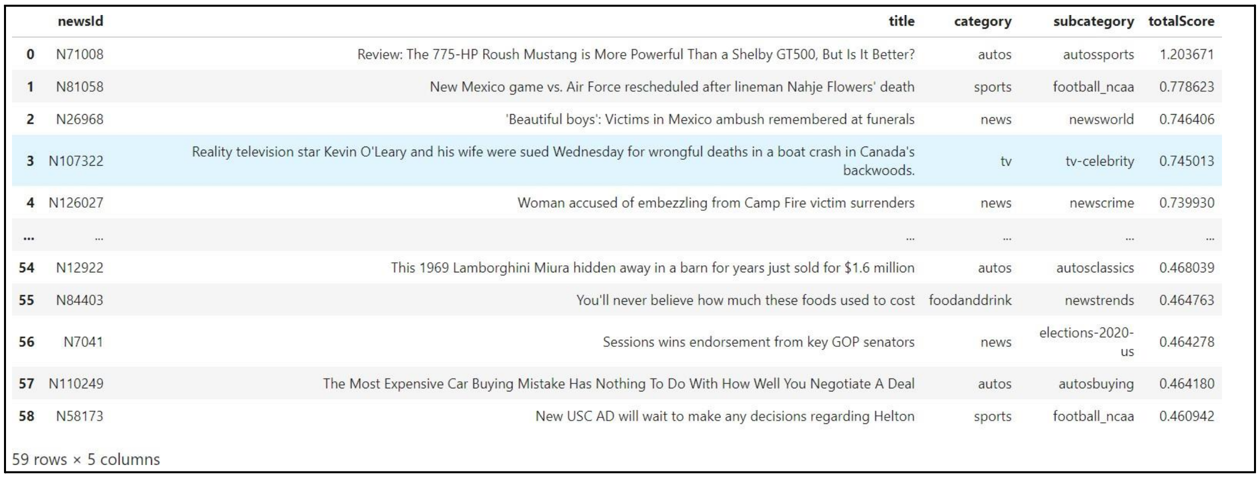
```
write_query = """
MATCH (u:User)
set u.followers = size((u)<-[:FOLLOWS]-())
"""

with driver.session() as session:
    session.write_transaction(lambda tx: tx.run(write_query))
```

Collaborative Filtering with USER_ALSO_LIKED Relationships

Collaborative filtering query for user U218584 using KNN and aggregate similarity:



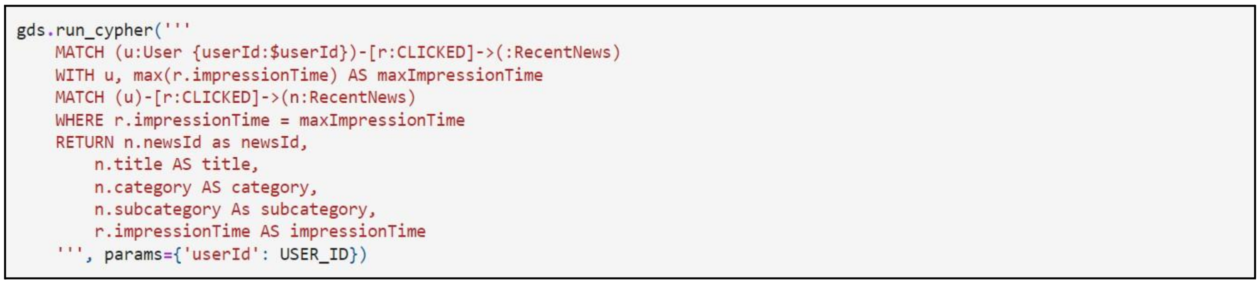
```
gds.run_cypher( '''
    MATCH(u:User {userId: $userId})-[:CLICKED]->(n:RecentNews)
    WITH collect(id(n)) AS clickedNewsIds

    //get similar News according to KNN and exclude previously clicked news
    MATCH (clickedNews)-[s:USERS_ALSO_LIKED]->(similarNews:News)
    WHERE id(clickedNews) IN clickedNewsIds AND NOT id(similarNews) IN clickedNewsIds

    //aggregate and return ranked results
    RETURN DISTINCT similarNews.newsId as newsId,
        similarNews.title AS title,
        similarNews.category AS category,
        similarNews.subcategory As subcategory,
        sum(s.score) AS totalScore ORDER BY totalScore DESC
    ''', params={'userId': USER_ID})
```



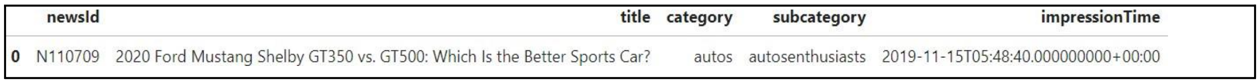
	newsId	title	category	subcategory	totalScore
0	N71008	Review: The 775-HP Roush Mustang is More Powerful Than a Shelby GT500, But Is It Better?	autos	autossports	1.203671
1	N81058	New Mexico game vs. Air Force rescheduled after lineman Nahje Flowers' death	sports	football_ncaa	0.778623
2	N26968	'Beautiful boys': Victims in Mexico ambush remembered at funerals	news	newsworld	0.746406
3	N107322	Reality television star Kevin O'Leary and his wife were sued Wednesday for wrongful deaths in a boat crash in Canada's backwoods.	tv	tv-celebrity	0.745013
4	N126027	Woman accused of embezzling from Camp Fire victim surrenders	news	newscrime	0.739930
...	...	...	...	...	...
54	N12922	This 1969 Lamborghini Miura hidden away in a barn for years just sold for $1.6 million	autos	autosclassics	0.468039
55	N84403	You'll never believe how much these foods used to cost	foodanddrink	newstrends	0.464763
56	N7041	Sessions wins endorsement from key GOP senators	news	elections-2020-us	0.464278
57	N110249	The Most Expensive Car Buying Mistake Has Nothing To Do With How Well You Negotiate A Deal	autos	autosbuying	0.464180
58	N58173	New USC AD will wait to make any decisions regarding Helton	sports	football_ncaa	0.460942

59 rows × 5 columns

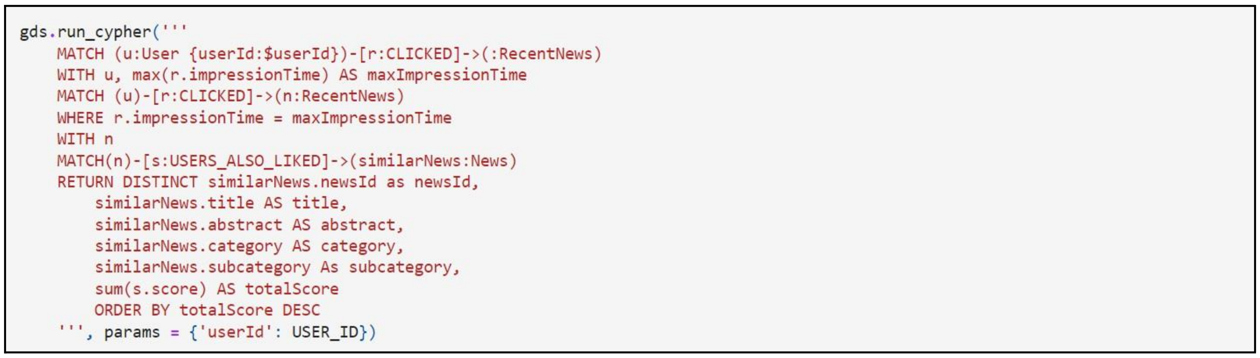
Content Based filtering recommendations Based on Latest Viewed Content



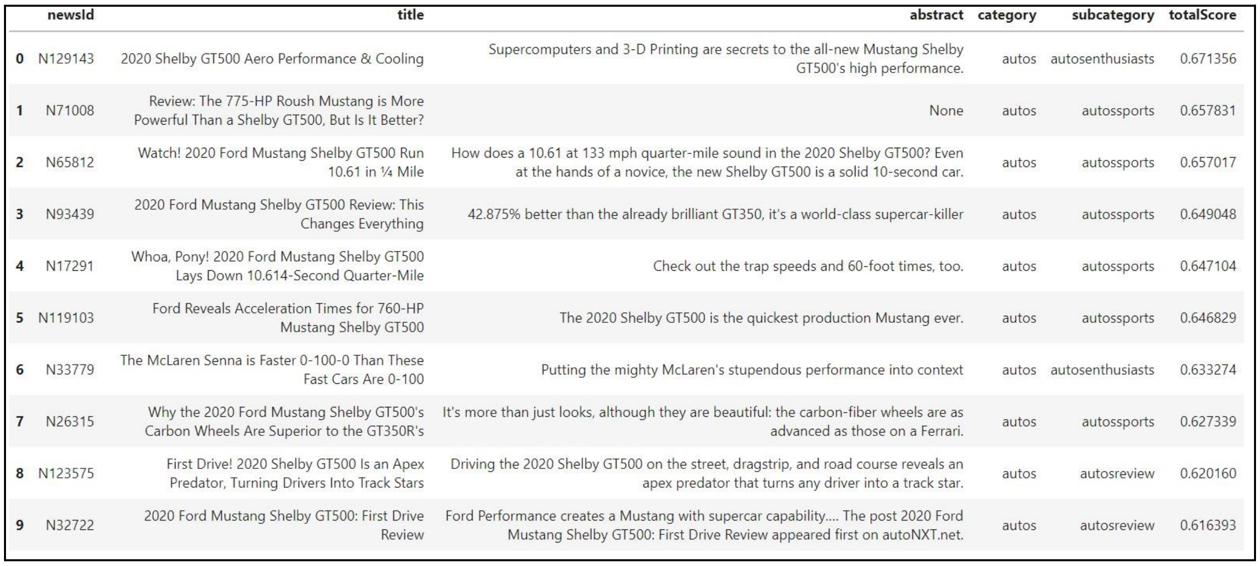
```
gds.run_cypher('''
    MATCH (u:User {userId:$userId})-[r:CLICKED]->(:RecentNews)
    WITH u, max(r.impressionTime) AS maxImpressionTime
    MATCH (u)-[r:CLICKED]->(n:RecentNews)
    WHERE r.impressionTime = maxImpressionTime
    RETURN n.newsId as newsId,
        n.title AS title,
        n.category AS category,
        n.subcategory As subcategory,
        r.impressionTime AS impressionTime
    ''', params={'userId': USER_ID})
```



	newsId	title	category	subcategory	impressionTime
0	N110709	2020 Ford Mustang Shelby GT350 vs. GT500: Which Is the Better Sports Car?	autos	autosenthusiasts	2019-11-15T05:48:40.000000000+00:00



```
gds.run_cypher('''
    MATCH (u:User {userId:$userId})-[r:CLICKED]->(:RecentNews)
    WITH u, max(r.impressionTime) AS maxImpressionTime
    MATCH (u)-[r:CLICKED]->(n:RecentNews)
    WHERE r.impressionTime = maxImpressionTime
    WITH n
    MATCH(n)-[s:USERS_ALSO_LIKED]->(similarNews:News)
    RETURN DISTINCT similarNews.newsId as newsId,
        similarNews.title AS title,
        similarNews.abstract AS abstract,
        similarNews.category AS category,
        similarNews.subcategory As subcategory,
        sum(s.score) AS totalScore
        ORDER BY totalScore DESC
    ''', params = {'userId': USER_ID})
```



	newsId	title	abstract	category	subcategory	totalScore
0	N129143	2020 Shelby GT500 Aero Performance & Cooling	Supercomputers and 3-D Printing are secrets to the all-new Mustang Shelby GT500's high performance.	autos	autosenthusiasts	0.671356
1	N71008	Review: The 775-HP Roush Mustang is More Powerful Than a Shelby GT500, But Is It Better?	None	autos	autossports	0.657831
2	N65812	Watch! 2020 Ford Mustang Shelby GT500 Run 10.61 in ¼ Mile	How does a 10.61 at 133 mph quarter-mile sound in the 2020 Shelby GT500? Even at the hands of a novice, the new Shelby GT500 is a solid 10-second car.	autos	autossports	0.657017
3	N93439	2020 Ford Mustang Shelby GT500 Review: This Changes Everything	42.875% better than the already brilliant GT350, it's a world-class supercar-killer	autos	autossports	0.649048
4	N17291	Whoa, Pony! 2020 Ford Mustang Shelby GT500 Lays Down 10.614-Second Quarter-Mile	Check out the trap speeds and 60-foot times, too.	autos	autossports	0.647104
5	N119103	Ford Reveals Acceleration Times for 760-HP Mustang Shelby GT500	The 2020 Shelby GT500 is the quickest production Mustang ever.	autos	autossports	0.646829
6	N33779	The McLaren Senna is Faster 0-100-0 Than These Fast Cars Are 0-100	Putting the mighty McLaren's stupendous performance into context	autos	autosenthusiasts	0.633274
7	N26315	Why the 2020 Ford Mustang Shelby GT500's Carbon Wheels Are Superior to the GT350R's	It's more than just looks, although they are beautiful: the carbon-fiber wheels are as advanced as those on a Ferrari.	autos	autossports	0.627339
8	N123575	First Drive! 2020 Shelby GT500 Is an Apex Predator, Turning Drivers Into Track Stars	Driving the 2020 Shelby GT500 on the street, dragstrip, and road course reveals an apex predator that turns any driver into a track star.	autos	autosreview	0.620160
9	N32722	2020 Ford Mustang Shelby GT500: First Drive Review	Ford Performance creates a Mustang with supercar capability.... The post 2020 Ford Mustang Shelby GT500: First Drive Review appeared first on autoNXT.net.	autos	autosreview	0.616393

Analysis and Inference

- News tends to be most relevant when it is recent and can lose relevance quickly with the passage of time. As a general rule, good news recommenders should avoid stale and/or outdated news. In this case we can use Microsoft impressions as a proxy and only consider news articles that were included in an impression within our sample time window. To accomplish this, we can use a node attribute called approxTime that reflects the minimum impression time for the news article.
- A limitation of CF is that it can only recommend content with user feedback. As such, news articles with no user clicks cannot be used for CF here. In application, this is sometimes referred to as a cold-start problem. It can be resolved with content based recommendation and other hybrid approaches.
- We do see a significant reduction in the number of news articles, from 104K to about 22K. But being more recent and well connected, this news is also likely to be more relevant, so we should be able to improve our recommender by narrowing the focus to these news articles. We can also see these improvements reflected in the click distributions.We can see now that every news article was clicked at least once, and at least 75% of the news articles have been recently clicked.

Conclusion

Recommendation systems is one of the pinnacle of graph theory applications especially in a world rife with technology. Every company and field will want to leverage the power of graph theory combined with artificial intelligence. Recommendation systems are primarily used in predicting/recommending TV shows and movies and also to solve the market-basket analysis problem. In this project deal with the news article recommendation, we are just barely scratching the surface for graph-based Recommender Systems in this post. Just within CF there are many different ways we could expand our analysis, as well as different ways we could optimize it to improve predictive performance. This can be explored with the current graph using the category, subcategory, and wikiData nodes in addition to other news article attributes. We focus on the recommendation of news using collaborative filtering and content based filtering using KNN (K Nearest Neighbours) based on the latest viewed content using **Neo4j**.