TF-IDF_SparkRDDs

May 9, 2025

[1]:
```
sc
```

```
VBox()
```

```
Starting Spark application
```

```
<IPython.core.display.HTML object>
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
SparkSession available as 'spark'.
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
<SparkContext master=yarn appName=livy-session-1>
```

[2]:
```
import re

stopwords = set(["a", "as", "able", "about", "above", "according",␣
 ↪"accordingly",
            "across", "actually", "after", "afterwards", "again", "against",␣
 ↪"aint", "all", "allow",
            "allows", "almost", "alone", "along", "already", "also",␣
 ↪"although", "always", "am", "among",
            "amongst", "an", "and", "another", "any", "anybody", "anyhow",␣
 ↪"anyone", "anything", "anyway",
            "anyways", "anywhere", "apart", "appear","appreciate",␣
 ↪"appropriate", "are", "arent", "around",
            "as", "aside", "ask", "asking", "associated", "at", "available",␣
 ↪"away", "awfully", "be", "became",
            "because", "become", "becomes", "becoming", "been", "before",␣
 ↪"beforehand", "behind",
            "being", "believe", "below", "beside", "besides", "best",␣
 ↪"better", "between", "beyond",
            "both", "brief", "but", "by", "cmon", "cs", "came", "can", "cant",␣
 ↪"cannot", "cant",
            "cause", "causes", "certain", "certainly", "changes", "clearly",␣
 ↪"co", "com", "come",
```

```
            "comes", "concerning", "consequently", "consider", "considering",␣
↪"contain", "containing",
            "contains", "corresponding", "could", "couldnt", "course",␣
↪"currently", "definitely",
            "described", "despite", "did", "didnt", "different", "do", "does",␣
↪"doesnt", "doing",
            "dont", "done", "down", "downwards", "during", "each", "edu",␣
↪"eg", "eight", "either",
            "else", "elsewhere", "enough", "entirely", "especially", "et",␣
↪"etc", "even", "ever",
            "every", "everybody", "everyone", "everything", "everywhere",␣
↪"ex", "exactly", "example",
            "except", "far", "few", "ff", "fifth", "first", "five",␣
↪"followed", "following", "follows",
            "for", "former", "formerly", "forth", "four", "from", "further",␣
↪"furthermore", "get",
            "gets", "getting", "given", "gives", "go", "goes", "going",␣
↪"gone", "got", "gotten",
            "greetings", "had", "hadnt", "happens", "hardly", "has", "hasnt",␣
↪"have", "havent",
            "having", "he", "hes", "hello", "help", "hence", "her", "here",␣
↪"heres", "hereafter",
            "hereby", "herein", "hereupon", "hers", "herself", "hi", "him",␣
↪"himself",
            "his", "hither", "hopefully", "how", "howbeit", "however", "i",␣
↪"id", "ill", "im", "ive",
            "ie", "if", "ignored", "immediate", "in", "inasmuch", "inc",␣
↪"indeed", "indicate",
            "indicated", "indicates", "inner", "insofar", "instead", "into",␣
↪"inward", "is",
            "isnt", "it", "itd", "itll", "its", "its", "itself", "just",␣
↪"keep", "keeps", "kept",
            "know", "knows", "known", "last", "lately", "later", "latter",␣
↪"latterly", "least",
            "less", "lest", "let", "lets", "like", "liked", "likely",␣
↪"little", "look", "looking",
            "looks", "ltd", "mainly", "many", "may", "maybe", "me", "mean",␣
↪"meanwhile", "merely",
            "might", "more", "moreover", "most", "mostly", "much", "must",␣
↪"my", "myself",
            "name", "namely", "nd", "near", "nearly", "necessary", "need",␣
↪"needs", "neither",
            "never", "nevertheless", "new", "next", "nine", "no", "nobody",␣
↪"non", "none", "noone",
```

```
            "nor", "normally", "not", "nothing", "novel", "now", "nowhere",␣
↪"obviously", "of",
            "off", "often", "oh", "ok", "okay", "old", "on", "once", "one",␣
↪"ones", "only",
            "onto", "or", "other", "others", "otherwise", "ought", "our",␣
↪"ours", "ourselves",
            "out", "outside", "over", "overall", "own", "particular",␣
↪"particularly",
            "per", "perhaps", "placed", "please", "plus", "possible",␣
↪"presumably", "probably",
            "provides", "que", "quite", "qv", "rather", "rd", "re", "really",␣
↪"reasonably",
            "regarding", "regardless", "regards", "relatively",␣
↪"respectively", "right", "said",
            "same", "saw", "say", "saying", "says", "second", "secondly",␣
↪"see", "seeing",
            "seem", "seemed", "seeming", "seems", "seen", "self", "selves",␣
↪"sensible", "sent",
            "serious", "seriously", "seven", "several", "shall", "she",␣
↪"should", "shouldnt",
            "since", "six", "so", "some", "somebody", "somehow", "someone",␣
↪"something",
            "sometime", "sometimes", "somewhat", "somewhere", "soon", "sorry",␣
↪"specified", "specify",
            "specifying", "still", "sub", "such", "sup", "sure", "ts", "take",␣
↪"taken", "tell", "tends",
            "th", "than", "thank", "thanks", "thanx", "that", "thats",␣
↪"thats", "the", "their", "theirs",
            "them", "themselves", "then", "thence", "there", "theres",␣
↪"thereafter", "thereby",
            "therefore", "therein", "theres", "thereupon", "these", "they",␣
↪"theyd",
            "theyll", "theyre", "theyve", "think", "third", "this", "thorough",
            "thoroughly", "those", "though", "three", "through", "throughout",␣
↪"thru",
            "thus", "to", "together", "too", "took", "toward", "towards",␣
↪"tried", "tries",
            "truly", "try", "trying", "twice", "two", "un", "under",␣
↪"unfortunately",
            "unless", "unlikely", "until", "unto", "up", "upon", "us", "use",␣
↪"used",
            "useful", "uses", "using", "usually", "value", "various", "very",␣
↪"via", "viz",
            "vs", "want", "wants", "was", "wasnt", "way", "we", "wed", "well",␣
↪"were", "weve",
```

```
            "welcome", "well", "went", "were", "werent", "what", "whats",␣
 ↪"whatever", "when",
            "whence", "whenever", "where", "wheres", "whereafter", "whereas",␣
 ↪"whereby",
            "wherein", "whereupon", "wherever", "whether", "which", "while",␣
 ↪"whither", "who",
            "whos", "whoever", "whole", "whom", "whose", "why", "will",␣
 ↪"willing", "wish",
            "with", "within", "without", "wont", "wonder", "would", "would",␣
 ↪"wouldnt", "yes",
            "yet", "you", "youd", "youll", "youre", "youve", "your", "yours",␣
 ↪"yourself",
            "yourselves", "zero"])

def termify(line):
    terms = []
    words = re.findall(r'[^\W_]+', line)
    for word in words:
        lowered = word.lower()
        if (len(lowered) > 1) and (lowered not in stopwords) and (not re.
 ↪search(r'^\d*$', lowered)):
            terms.append(lowered)
    return terms
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…

[3]:
```
# point to S3 location for this folder
tfidf=sc.wholeTextFiles('s3://aws-emr-studio-247682200909-us-east-1/
 ↪1716062555134/e-8AZYNSASTJA7YMJE429MQZSN8/textcorpora/')
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…

[4]:
```
# function to get a docid from a file path
def get_docid(filepath):
    return filepath.split('/')[-1][: -4]
```

VBox()

FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…

below codes read the document corpus (directory) and produces TF-IDF values for each (term, doc-id) pair using Spark RDDs in every phase

[8]:
```
tfidf.flatMap(lambda x: [(term, get_docid(x[0])) for term in termify(x[1])]).
↪take(2)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[('emma', 'austen-emma'), ('jane', 'austen-emma')]
```

[9]:
```
term_frequencies = tfidf.flatMap(lambda x: [(term, get_docid(x[0])) for term in␣
↪termify(x[1])]) \
                            .map(lambda x: ((x[0], x[1]), 1))


term_frequencies.take(5)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[(('emma', 'austen-emma'), 1), (('jane', 'austen-emma'), 1), (('austen',
'austen-emma'), 1), (('volume', 'austen-emma'), 1), (('chapter', 'austen-emma'),
1)]
```

[10]:
```
term_frequencies = tfidf.flatMap(lambda x: [(term, get_docid(x[0])) for term in␣
↪termify(x[1])]) \
                            .map(lambda x: ((x[0], x[1]), 1)) \
                            .reduceByKey(lambda a, b: a + b)

term_frequencies.take(5)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[(('emma', 'austen-emma'), 865), (('austen', 'austen-emma'), 1), (('volume',
'austen-emma'), 3), (('woodhouse', 'austen-emma'), 314), (('handsome', 'austen-
emma'), 38)]
```

[11]:
```
# Calculating document frequencies (DF)
document_frequencies = term_frequencies.map(lambda x: (x[0][0], 1)) \
                                       .reduceByKey(lambda a, b: a + b)

document_frequencies.take(5)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[('persuasion', 8), ('jane', 4), ('chapter', 9), ('walter', 5),
('somersetshire', 3)]
```

[ ]:

[12]:
```
doc_term_counts = tfidf.map(lambda x: (get_docid(x[0]), len(termify(x[1]))))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[13]:
```
doc_term_counts.take(5)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[('austen-emma', 53278), ('austen-persuasion', 28637), ('austen-sense', 40397),
('austin-persuasion', 28637), ('bible-kjv', 294290)]
```

[14]:
```
term_frequencies_join = term_frequencies.keyBy(lambda t: t[0][0])
document_frequencies_join = document_frequencies.keyBy(lambda t: t[0])
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[15]:
```
term_frequencies_join.take(3)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[('emma', (('emma', 'austen-emma'), 865)), ('austen', (('austen', 'austen-
emma'), 1)), ('volume', (('volume', 'austen-emma'), 3))]
```

[16]:
```
document_frequencies_join.take(3)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[('persuasion', ('persuasion', 8)), ('jane', ('jane', 4)), ('chapter',
('chapter', 9))]
```

[17]:
```
x=doc_term_counts.take(1)
x
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
[('austen-emma', 53278)]
```

[18]:
```
doc_term_counts.lookup(x[0][0])[0]
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
53278
```

[19]:
```
z=('persuasion', ((('persuasion', 'austen-persuasion'), 7), ('persuasion', 8)))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

[20]:
```
z[1][0][0][1]
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
'austen-persuasion'
```

[21]:
```
doc_term_counts.lookup(z[1][0][0][1])[0]
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
28637
```

[22]:
```
term_frequencies_join.join(document_frequencies_join).take(5)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
[('persuasion', ((('persuasion', 'austen-persuasion'), 7), ('persuasion', 8))),
('persuasion', ((('persuasion', 'austen-emma'), 11), ('persuasion', 8))),
('persuasion', ((('persuasion', 'austen-sense'), 13), ('persuasion', 8))),
('persuasion', ((('persuasion', 'austin-persuasion'), 7), ('persuasion', 8))),
('persuasion', ((('persuasion', 'bible-kjv'), 1), ('persuasion', 8)))]
```

[23]:
```
doc_term_counts_dict = dict(doc_term_counts.collect())

# Broadcasting the dictionary
```

```
doc_term_counts_broadcast = sc.broadcast(doc_term_counts_dict)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[24]:
```
doc_term_counts_broadcast.value['austen-persuasion']
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
28637
```

[ ]:
```

```

[25]:
```
doc_term_counts_dict = dict(doc_term_counts.collect())

# Broadcast the dictionary
doc_term_counts_broadcast = sc.broadcast(doc_term_counts_dict)


# Computing TF-IDF
tfidf = term_frequencies_join.join(document_frequencies_join) \
                        .map(lambda x: ((x[1][0][0][0], x[1][0][0][1]), 1000000␣
↪* (x[1][0][1] / doc_term_counts_broadcast.value[x[1][0][0][1]] /␣
↪x[1][1][1])))
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

[26]:
```
tfidf.take(2)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[(('persuasion', 'austen-persuasion'), 30.5548765582987), (('persuasion',
'austen-emma'), 25.8080258267953)]
```

[ ]:
```

```

[27]:
```
tfidf.sortBy(lambda t: t[0][0]).sortBy(lambda t: t[0][1]).take(5)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[(('23rd', 'austen-emma'), 18.7694733285784), (('24th', 'austen-emma'),
18.7694733285784), (('26th', 'austen-emma'), 18.7694733285784), (('28th',
'austen-emma'), 37.5389466571568), (('7th', 'austen-emma'), 18.7694733285784)]
```

[28]:
```
sample = [
    ('arm', 'milton-paradise'),
    ('ashtoreth', 'bible-kjv'),
    ('decided', 'edgeworth-parents'),
    ('enchanting', 'whitman-leaves'),
    ('indebted', 'austen-emma'),
    ('inspection', 'austen-emma'),
    ('knives', 'chesterton-thursday'),
    ('material', 'melville-moby_dick'),
    ('reconciliation', 'austen-persuasion'),
    ('splash', 'bryant-stories')
]
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

[30]:
```
sample = sc.parallelize(sample)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

[31]:
```
tfset = tfidf.keyBy(lambda t: t[0])
sampleset = sample.keyBy(lambda t: t)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

[32]:
```
joined_rdd = tfset.join(sampleset)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

[33]:
```
joined_rdd.take(5)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
 ↪layout=Layout(height='25px', width='50%'),…
```

```
[('decided', 'edgeworth-parents'), ((('decided', 'edgeworth-parents'),
16.702577207663143), ('decided', 'edgeworth-parents'))), (('material',
```

```
'melville-moby_dick'), ((('material', 'melville-moby_dick'), 8.011023167879001),
('material', 'melville-moby_dick'))), (('arm', 'milton-paradise'), ((('arm',
'milton-paradise'), 22.629656617590488), ('arm', 'milton-paradise'))),
(('reconciliation', 'austen-persuasion'), ((('reconciliation', 'austen-
persuasion'), 21.824911827356217), ('reconciliation', 'austen-persuasion'))),
(('indebted', 'austen-emma'), ((('indebted', 'austen-emma'),
10.725413330616227), ('indebted', 'austen-emma')))]
```

[34]:
```
joined_rdd.count()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
10
```

[35]:
```
# Reformatting the join result to get tuples of the form ((term, docid),␣
↪tfidf_value)
reformatted_rdd = joined_rdd.map(lambda x: (x[0], x[1][0][1]))



reformatted_rdd.take(3)
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[(('decided', 'edgeworth-parents'), 16.702577207663143), (('material',
'melville-moby_dick'), 8.011023167879001), (('arm', 'milton-paradise'),
22.629656617590488)]
```

[36]:
```
# Sorting the reformatted RDD by term
sorted_rdd = reformatted_rdd.sortBy(lambda x: x[0][0])

sorted_rdd.collect()
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```

```
[(('arm', 'milton-paradise'), 22.629656617590488), (('ashtoreth', 'bible-kjv'),
10.194026300587854), (('decided', 'edgeworth-parents'), 16.702577207663143),
(('enchanting', 'whitman-leaves'), 4.450932915539097), (('indebted', 'austen-
emma'), 10.725413330616227), (('inspection', 'austen-emma'),
3.1282455547630663), (('knives', 'chesterton-thursday'), 15.05638616619239),
(('material', 'melville-moby_dick'), 8.011023167879001), (('reconciliation',
'austen-persuasion'), 21.824911827356217), (('splash', 'bryant-stories'),
34.77535123104743)]
```

[37]:
```
tfidf = None
```

```
VBox()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:',␣
↪layout=Layout(height='25px', width='50%'),…
```