

# **Knowledge Discovery and Management**

## **Project Phase - I Report**

**By**

**Team 1:**

**Sai Venkatesh Gatiganti (Class ID: 08)**

**Karthik Reddy Vundela (Class ID: 43)**

**Sai Chaitanya Manne (Class ID: 20)**

**Sri Chaitanya Patluri (Class ID: 32)**

## Objective:

To design a Semantic Search Engine that provides search results on movies based on reviews obtained from Amazon.

## Expected Outcome:

To obtain search results based on context besides keywords for better accuracy. For example, if a user searches for a specific director of movies, all of his/her well known works are shown in the results.

## Project Domain: Movies (Semantic Search Engine)

## Dataset:

Amazon Review Data collected by Julian McAuley, UCSD -

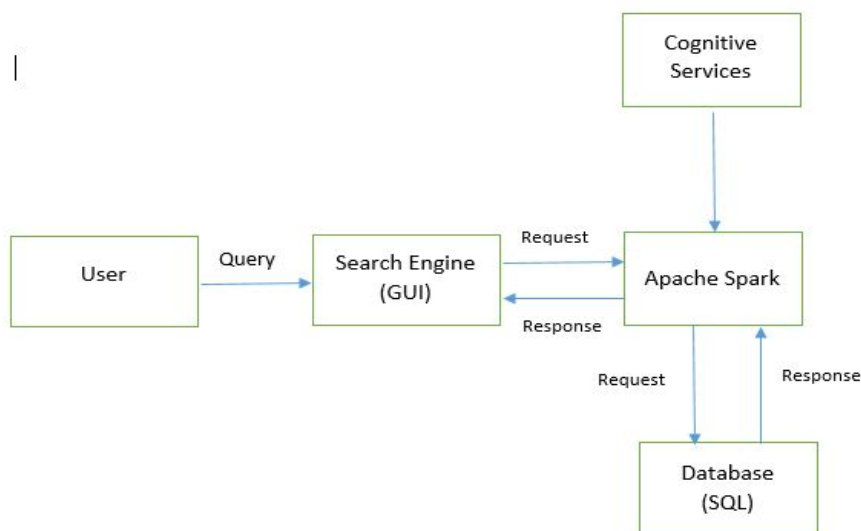
<http://jmcauley.ucsd.edu/data/amazon/links.html>

*Image-based recommendations on styles and substitutes* J. McAuley, C. Targett, J. Shi, A. van den Hengel SIGIR, 2015

*Inferring networks of substitutable and complementary products* J. McAuley, R. Pandey, J. Leskovec, Knowledge Discovery and Data Mining, 2015

Wikipedia - <https://dumps.wikimedia.org/>

## System Architecture:



## Example Data 1 :

The third movie produced by Howard Hughes, this gem was thought to be lost. It was recently restored and shown on TCM (12/15/04). The plot is a familiar one - two WW I soldiers escape from a German prison camp (guarded by an extremely lethargic German shepherd, who practically guides them out of the camp), stow away on a ship, and end up in "Arabia", where they rescue the lovely Mary Astor. The restoration is very good overall, although there are two or three very rough sequences. The production is very good, and there are some very funny scenes. And did I mention that Mary Astor is in it? The film won an Academy Award for the now-defunct category of "Best Direction of a Comedy".

## NLP Workflow :



## Output:

### Lemmatization:

[the, third, movie, produce, by, Howard, Hughes, ,, this, gem, be, think, to, be, lose, ., it, be, recently, restore, and, show, on, TCM, -lrb-, 12/15/04, -rrb-, ., the, plot, be, a, familiar, one, -, two, ww, I, soldier, escape, from, a, german, prison, camp, -lrb-, guard, by, a, extremely, lethargic, german, shepherd, ,, who, practically, guide, they, out, of, the, camp, -rrb-, ,, stow, away, on, a, ship, ,, and, end, up, in, `` , Arabia, " , ,, where, they, rescue, the, lovely, Mary, Astor, ., the, restoration, be, very, good, overall, ,, although, there, be, two, or, three, very, rough, sequence, ., the, production, be, very, good, ,, and, there, be, some, very, funny, scene, ., and, do, I, mention, that, Mary, Astor, be, in, it, ?, the, film, win, a, academy, award, for, the, now-defunct, category, of, `` , best, direction, of, a, comedy, " , .]

### Parts of Speech Tagging:

[(The: DT), (third: JJ), (movie: NN), (produced: VBN), (by: IN), (Howard: NNP), (Hughes: NNP), (: :), (this: DT), (gem: NN), (was: VBD), (thought: VBN), (to: TO), (be: VB), (lost: VBN), (: :), (It: PRP), (was: VBD), (recently: RB), (restored: VBN), (and: CC), (shown: VBN), (on: IN), (TCM: NNP), (-LRB-: -LRB-), (12/15/04: CD), (-RRB-: -RRB-), (: :), (The: DT), (plot: NN), (is: VBZ), (a: DT), (familiar: JJ), (one: CD), (-: :), (two: CD), (WW: NN), (I: PRP),

(soldiers: NNS), (escape: VB), (from: IN), (a: DT), (German: JJ), (prison: NN), (camp: NN), (-LRB-: -LRB-), (guarded: VBN), (by: IN), (an: DT), (extremely: RB), (lethargic: JJ), (German: JJ), (shepherd: NN), (: ,), (who: WP), (practically: RB), (guides: NNS), (them: PRP), (out: IN), (of: IN), (the: DT), (camp: NN), (-RRB-: -RRB-), (: ,), (stow: VB), (away: RB), (on: IN), (a: DT), (ship: NN), (: ,), (and: CC), (end: VB), (up: RP), (in: IN), (`` ``), (Arabia: NNP), (" "), (: ,), (where: WRB), (they: PRP), (rescue: VBP), (the: DT), (lovely: JJ), (Mary: NNP), (Astor: NNP), (: .), (The: DT), (restoration: NN), (is: VBZ), (very: RB), (good: JJ), (overall: RB), (: ,), (although: IN), (there: EX), (are: VBP), (two: CD), (or: CC), (three: CD), (very: RB), (rough: JJ), (sequences: NNS), (: .), (The: DT), (production: NN), (is: VBZ), (very: RB), (good: JJ), (: ,), (and: CC), (there: EX), (are: VBP), (some: DT), (very: RB), (funny: JJ), (scenes: NNS), (: .), (And: CC), (did: VBD), (I: PRP), (mention: VB), (that: DT), (Mary: NNP), (Astor: NNP), (is: VBZ), (in: IN), (it: PRP), (? :), (The: DT), (film: NN), (won: VBD), (an: DT), (Academy: NN), (Award: NN), (for: IN), (the: DT), (now-defunct: JJ), (category: NN), (of: IN), (`` ``), (Best: JJS), (Direction: NN), (of: IN), (a: DT), (Comedy: NN), (" "), (: .)]

## Parsing:

(ROOT (S (S (NP (DT The) (JJ third) (NN movie)) (VP (VBN produced) (PP (IN by) (NP (NNP Howard) (NNP Hughes))))) (: ,) (NP (DT this) (NN gem)) (VP (VBD was) (VP (VBN thought) (S (VP (TO to) (VP (VB be) (VP (VBN lost))))))) (: .)))

(ROOT (S (NP (PRP It)) (VP (VBD was) (ADVP (RB recently)) (VP (VBN restored) (CC and) (VBN shown) (PP (IN on) (NP (NP (NNP TCM)) (PRN (-LRB- -LRB-) (NP (CD 12/15/04)) (-RRB- -RRB-))))) (: .)))

(ROOT (S (NP (DT The) (NN plot)) (VP (VBZ is) (NP (NP (DT a) (ADJP (JJ familiar) (NP-TMP (CD one)))) (: -) (NP (NP (CD two) (NN WW)) (SBAR (S (NP (PRP I) (NNS soldiers)) (VP (VP (VB escape) (PP (IN from) (NP (NP (DT a) (JJ German) (NN prison) (NN camp)) (PRN (-LRB- -LRB-) (VP (VBN guarded) (PP (IN by) (NP (NP (DT an) (RB extremely) (JJ lethargic) (JJ German) (NN shepherd)) (: ,) (SBAR (WHNP (WP who)) (S (NP (RB practically) (NNS guides)) (NP (PRP them))))) (PP (IN out) (PP (IN of) (NP (DT the) (NN camp)))) (-RRB- -RRB-))))) (: ,) (VP (VB stow) (ADVP (RB away)) (PP (IN on) (NP (DT a) (NN ship)))) (: ,) (CC and) (VP (VB end) (PRT (RP up)) (PP (IN in) (`` ``) (NP (NNP Arabia)) (" ")))) (: ,) (SBAR (WHADVP (WRB where)) (S (NP (PRP they)) (VP (VBP rescue) (NP (DT the) (JJ lovely) (NNP Mary) (NNP Astor))))) (: .)))

(ROOT (S (NP (DT The) (NN restoration)) (VP (VBZ is) (ADJP (RB very) (JJ good)) (ADVP (RB overall)) (: ,) (SBAR (IN although) (S (NP (EX there)) (VP (VBP are) (NP (NP (CD two)) (CC or) (NP (CD three) (ADJP (RB very) (JJ rough)) (NNS sequences)))))) (: .)))

(ROOT (S (S (NP (DT The) (NN production)) (VP (VBZ is) (ADJP (RB very) (JJ good)))) (: ,) (CC and) (S (NP (EX there)) (VP (VBP are) (NP (DT some) (ADJP (RB very) (JJ funny)) (NNS scenes)))) (: .)))

(ROOT (SQ (CC And) (VBD did) (NP (PRP I)) (VP (VB mention) (SBAR (S (NP (DT that) (NNP Mary) (NNP Astor)) (VP (VBZ is) (PP (IN in) (NP (PRP it)))))) ( . ?)))  
 (ROOT (S (NP (DT The) (NN film)) (VP (VBD won) (NP (DT an) (NN Academy) (NN Award)) (PP (IN for) (NP (NP (DT the) (JJ now-defunct) (NN category)) (PP (IN of) (`` ``) (NP (NP (JJS Best) (NN Direction)) (PP (IN of) (NP (DT a) (NN Comedy)))) (" ")))) ( . )))

## Name Entity Recognition:

[(The: O), (third: ORDINAL), (movie: O), (produced: O), (by: O), (Howard: PERSON), (Hughes: PERSON), (: O), (this: O), (gem: O), (was: O), (thought: O), (to: O), (be: O), (lost: O), (: O), (It: O), (was: O), (recently: DATE), (restored: O), (and: O), (shown: O), (on: O), (TCM: ORGANIZATION), (-LRB-: O), (12/15/04: DATE), (-RRB-: O), (: O), (The: O), (plot: O), (is: O), (a: O), (familiar: O), (one: NUMBER), (-: O), (two: NUMBER), (WW: O), (I: O), (soldiers: O), (escape: O), (from: O), (a: O), (German: MISC), (prison: O), (camp: O), (-LRB-: O), (guarded: O), (by: O), (an: O), (extremely: O), (lethargic: O), (German: MISC), (shepherd: O), (: O), (who: O), (practically: O), (guides: O), (them: O), (out: O), (of: O), (the: O), (camp: O), (-RRB-: O), (: O), (stow: O), (away: O), (on: O), (a: O), (ship: O), (: O), (and: O), (end: O), (up: O), (in: O), (``: O), (Arabia: O), (": O), (: O), (where: O), (they: O), (rescue: O), (the: O), (lovely: O), (Mary: PERSON), (Astor: PERSON), (: O), (The: O), (restoration: O), (is: O), (very: O), (good: O), (overall: O), (: O), (although: O), (there: O), (are: O), (two: NUMBER), (or: O), (three: NUMBER), (very: O), (rough: O), (sequences: O), (: O), (The: O), (production: O), (is: O), (very: O), (good: O), (: O), (and: O), (there: O), (are: O), (some: O), (very: O), (funny: O), (scenes: O), (: O), (And: O), (did: O), (I: O), (mention: O), (that: O), (Mary: PERSON), (Astor: PERSON), (is: O), (in: O), (it: O), (? : O), (The: O), (film: O), (won: O), (an: O), (Academy: MISC), (Award: MISC), (for: O), (the: O), (now-defunct: O), (category: O), (of: O), (``: O), (Best: O), (Direction: O), (of: O), (a: O), (Comedy: O), (": O), (: O)]

## Co-Reference Resolution Graph:

[CHAIN1-["third" in sentence 1], CHAIN2-["Howard Hughes" in sentence 1], CHAIN3-["The third movie" in sentence 1, "It" in sentence 2], CHAIN4-["this gem" in sentence 1], CHAIN6-["TCM -LRB- 12/15/04 -RRB-" in sentence 2], CHAIN7-["12/15/04" in sentence 2], CHAIN8-["one" in sentence 3], CHAIN9-["two" in sentence 3, "two" in sentence 4], CHAIN10-["The plot" in sentence 3, "a familiar one - two WW I soldiers escape from a German prison camp -LRB- guarded by an extremely lethargic German shepherd , who practically guides them out of the camp -RRB- , stow away on a ship , and end up in `` Arabia "" in sentence 3], CHAIN12-["two WW I soldiers escape from a German prison camp -LRB- guarded by an extremely lethargic German shepherd , who practically guides them out of the camp -RRB- , stow away on a ship , and end up in `` Arabia "" in sentence 3], CHAIN13-["I soldiers" in

sentence 3, "they" in sentence 3], CHAIN14-["I" in sentence 3, "I" in sentence 6], CHAIN15-["a German prison camp -LRB- guarded by an extremely lethargic German shepherd , who practically guides them out of the camp -RRB-" in sentence 3], CHAIN16-["an extremely lethargic German shepherd , who practically guides them" in sentence 3], CHAIN17-["an extremely lethargic German shepherd" in sentence 3], CHAIN18-["practically guides" in sentence 3, "them" in sentence 3], CHAIN20-["the camp" in sentence 3], CHAIN21-["a ship" in sentence 3], CHAIN22-["Arabia" in sentence 3], CHAIN24-["the lovely Mary Astor" in sentence 3, "that Mary Astor" in sentence 6], CHAIN26-["three" in sentence 4], CHAIN27-["The restoration" in sentence 4], CHAIN28-["two or three very rough sequences" in sentence 4], CHAIN29-["three very rough sequences" in sentence 4], CHAIN30-["The production" in sentence 5, "it" in sentence 6], CHAIN31-["some very funny scenes" in sentence 5], CHAIN35-["The film" in sentence 7], CHAIN36-["an Academy Award" in sentence 7], CHAIN37-["the now-defunct category of `` Best Direction of a Comedy "" in sentence 7], CHAIN38-["Best Direction of a Comedy" in sentence 7], CHAIN39-["a Comedy" in sentence 7]]

## Implementation of Core NLP:

```

Properties props = new Properties();
props.setProperty("annotators", "tokenize, split, pos, lemma, ner, parse, dcoref");
StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

// read some text in the text variable
String text = "The third movie produced by Howard Hughes, this gem was thought to be lost. It was recently restored and shown on TV."

// create an empty Annotation just with the given text
Annotation document = new Annotation(text);

// run all Annotators on this text
pipeline.annotate(document);

// these are all the sentences in this document
// a CoreMap is essentially a Map that uses class objects as keys and has values with custom types
List<CoreMap> sentences = document.get(CoreAnnotations.SentencesAnnotation.class);

for (CoreMap sentence : sentences) {
    System.out.println("\n+ sentence):
    // traversing the words in the current sentence
    // a CoreLabel is a CoreMap with additional token-specific methods
    for (CoreLabel token : sentence.get(CoreAnnotations.TokensAnnotation.class)) {

```

Run Main

```

[CHAIN1-["third" in sentence 1], CHAIN2-["Howard Hughes" in sentence 1], CHAIN3-["The third movie" in sentence 1, "It" in sentence 2], CHAIN4-["this gem" in sentence 1], CHAIN5-["(The: O) (third: ORDINAL) (movie: O) (produced: O) (by: O) (Howard: PERSON) (Hughes: PERSON) (,: O) (this: O) (gem: O) (was: O) (thought: O) (to: O) (The: O) (third: ORDINAL) (movie: O) (produced: O) (by: O) (Howard: PERSON) (Hughes: PERSON) (,: O) (this: O) (gem: O) (was: O) (thought: O) (to: O) (be: O) (lost: O)"]
Process finished with exit code 0

```

Compilation completed successfully in 1s 919ms (2 minutes ago)

## Information Extraction/Retrieval Techniques

- TF-IDF (Term Frequency and Inverse Document Frequency)
- Spark NLP and TFIDF (ML PIPELINE)

## TF-IDF (without NLP):

The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \times \log N / \text{df}_t$$

## Example data 2:

The third movie produced by Howard Hughes, this gem was thought to be lost.

It was recently restored and shown on TCM (12/15/04).

The plot is a familiar one - two WW I soldiers escape from a German prison camp (guarded by an extremely lethargic German shepherd, who practically guides them out of the camp), stow away on a ship, and end up in "Arabia", where they rescue the lovely Mary Astor.

The restoration is very good overall, although there are two or three very rough sequences.

The production is very good, and there are some very funny scenes.

And did I mention that Mary Astor is in it? The film won an Academy Award for the now-defunct category of "Best Direction of a Comedy".

## TF-IDF output:

(1048576,[3370,3555,32957,53406,84049,96852,104860,115276,191910,226916,239006,470326,700601,1011071],[0.3364722366212129,1.252762968495368,1.252762968495368,1.252762968495368,0.15415067982725836,0.8472978603872037,1.252762968495368,0.8472978603872037,1.252762968495368,0.8472978603872037,1.252762968495368,1.6945957207744073,1.252762968495368,1.252762968495368])

(1048576,[3370,84049,96727,96852,226916,276362,390388,470326,848057,996263,1021711],[0.3364722366212129,0.15415067982725836,0.5596157879354227,0.8472978603872037,0.8472978603872037,1.252762968495368,1.252762968495368,1.6945957207744073,1.252762968495368,1.252762968495368,1.252762968495368])

(1048576,[3139,3159,3707,45091,84049,102223,117481,230759,257968,278320,400778,413342,774020,903705],[1.252762968495368,0.8472978603872037,1.252762968495368,1.252762968495368,0.15415067982725836,1.252762968495368,0.8472978603872037,1.252762968495368,1.252762968495368,1.252762968495368,1.252762968495368,1.252762968495368,1.252762968495368,1.252762968495368])

(1048576,[2379,3551,82878,96727,117481,361745,928040,957294,1015702],[1.252762968495368,0.8472978603872037,1.252762968495368,0.5596157879354227,0.8472978603872037,1.252762968495368,1.252762968495368,1.252762968495368,1.252762968495368])

(1048576,[73,97,3117,3365,3370,3543,65975,78021,84049,99455,101577,104564,114801,117910,141151,179454,293627,335338,413095,655301,702156,756285,928344,960806,1045892],[0.

8472978603872037,0.8472978603872037,0.8472978603872037,0.8472978603872037,0.336472  
2366212129,1.6945957207744073,1.252762968495368,1.252762968495368,0.15415067982725  
836,1.252762968495368,1.252762968495368,1.252762968495368,0.8472978603872037,1.2527  
62968495368,1.252762968495368,1.252762968495368,0.8472978603872037,1.2527629684953  
68,1.252762968495368,1.252762968495368,1.252762968495368,1.252762968495368,1.252762  
968495368,1.252762968495368,1.252762968495368])  
(1048576,[45,73,97,2784,3117,3159,3365,3370,3543,3551,3739,6058,84049,96727,100571,110  
182,110414,114801,115276,117694,268508,293627,298209,337143,355421,355696,361058,378  
038,391433,395273,413212,413224,439691,569137,648036,687929,819329,830563,840102,886  
163,900327,910062,948865,966783,975493],[1.252762968495368,0.8472978603872037,2.5418  
935811616112,1.252762968495368,0.8472978603872037,0.8472978603872037,0.84729786038  
72037,0.3364722366212129,0.8472978603872037,0.8472978603872037,1.252762968495368,1.  
252762968495368,0.15415067982725836,0.5596157879354227,1.252762968495368,1.2527629  
68495368,1.252762968495368,1.6945957207744073,0.8472978603872037,1.252762968495368  
,1.252762968495368,0.8472978603872037,1.252762968495368,1.252762968495368,1.2527629  
68495368,1.252762968495368,1.252762968495368,1.252762968495368,1.252762968495368,1.  
252762968495368,1.252762968495368,1.252762968495368,1.252762968495368,1.2527629684  
95368,1.252762968495368,1.252762968495368,1.252762968495368,1.252762968495368,2.505  
525936990736,1.252762968495368,1.252762968495368,1.252762968495368,1.2527629684953  
68,1.252762968495368,1.252762968495368])

### **Top TF-IDF values:**

(German,2.505525936990736)  
(very,1.6945957207744073)  
(of,1.6945957207744073)  
(be,1.252762968495368)  
(Direction,1.252762968495368)  
(Comedy",1.252762968495368)  
(Astor.,1.252762968495368)  
(escape,1.252762968495368)  
(some,1.252762968495368)  
(film,1.252762968495368)  
(recently,1.252762968495368)  
(camp,1.252762968495368)  
(category,1.252762968495368)  
(Howard,1.252762968495368)  
(three,1.252762968495368)  
(Hughes,,1.252762968495368)



(lethargic,1.252762968495368)  
(sequences.,1.252762968495368)  
(restoration,1.252762968495368)  
(that,1.252762968495368)

### **TF-IDF using spark NLP:**

By using NLP we can remove the unwanted words like an, in, of, and, as etc. using stop word remover.

### **Output:**

```
[WrappedArray(third, movie, produced, howard, hughes,, gem, thought, lost.), (20, [0, 5, 6, 9, 11, 13, 16], [0.5596157879354227, 1.1192315758708453, 0.8472978603872037, 0.3364722366212129, 0.3364722366212129, 0.3364722366212129, 0.5596157879354227]), 0]
```

```
[WrappedArray(recently, restored, shown, tcm, (12/15/04).), (20, [2, 3, 10, 11, 14], [0.8472978603872037, 0.5596157879354227, 0.5596157879354227, 0.3364722366212129, 0.8472978603872037]), 1]
```

```
[WrappedArray(plot, familiar, one, -, two, ww, soldiers, escape, german, prison, camp, (guarded, extremely, lethargic, german, shepherd,, practically, guides, camp)), stow, away, ship,, end, "arabia",, rescue, lovely, mary, astor.), (20, [0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18, 19], [1.1192315758708453, 1.6945957207744073, 0.5596157879354227, 1.6945957207744073, 0.5596157879354227, 1.6945957207744073, 0.8472978603872037, 2.2384631517416906, 0.5596157879354227, 1.0094167098636386, 1.252762968495368, 0.3364722366212129, 0.8472978603872037, 1.252762968495368, 1.252762968495368, 0.6729444732424258, 1.1192315758708453]), 2]
```

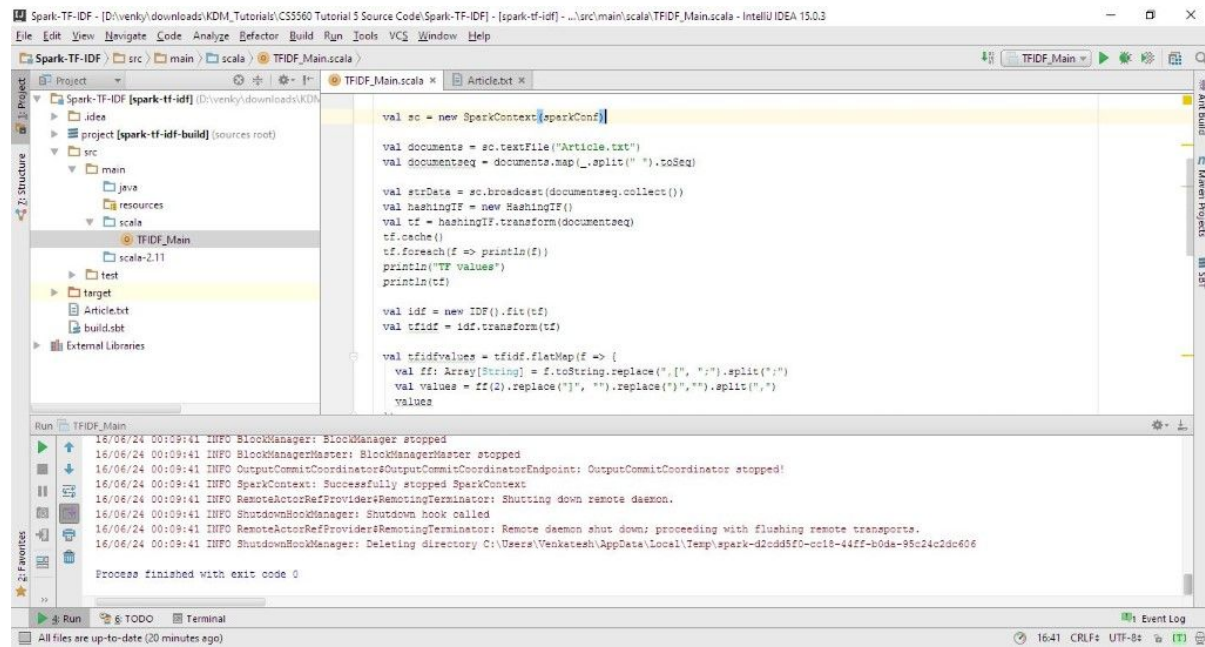
```
[WrappedArray(restoration, good, overall,, although, two, three, rough, sequences.), (20, [8, 9, 13, 16, 18, 19], [1.1192315758708453, 0.3364722366212129, 0.3364722366212129, 1.1192315758708453, 0.3364722366212129, 0.5596157879354227]), 3]
```

```
[WrappedArray(restoration, good, overall,, although, two, three, rough, sequences.), (20, [8, 9, 13, 16, 18, 19], [1.1192315758708453, 0.3364722366212129, 0.3364722366212129, 1.1192315758708453, 0.3364722366212129, 0.5596157879354227]), 4]
```

```
[WrappedArray(mention, mary, astor, it?, film, academy, award, now-defunct, category, "best, direction, comedy".), (20, [0, 3, 4, 5, 7, 9, 10, 11, 18], [1.1192315758708453, 0.5596157879354227, 1.6945957207
```

744073,0.5596157879354227,1.6945957207744073,0.3364722366212129,0.5596157879354227  
,0.3364722366212129,0.3364722366212129)),5]

## Implementation of TF-IDF:



```
val sc = new SparkContext(sparkConf)

val documents = sc.textFile("Article.txt")
val documentsseq = documents.map(_.split(" ").toSeq)

val strData = sc.broadcast(documentsseq.collect())
val hashingTF = new HashingTF()
val tf = hashingTF.transform(documentsseq)
tf.cache()
tf.foreach(f => println(f))
println("TF values")
println(tf)

val idf = new IDF().fit(tf)
val tfidf = idf.transform(tf)

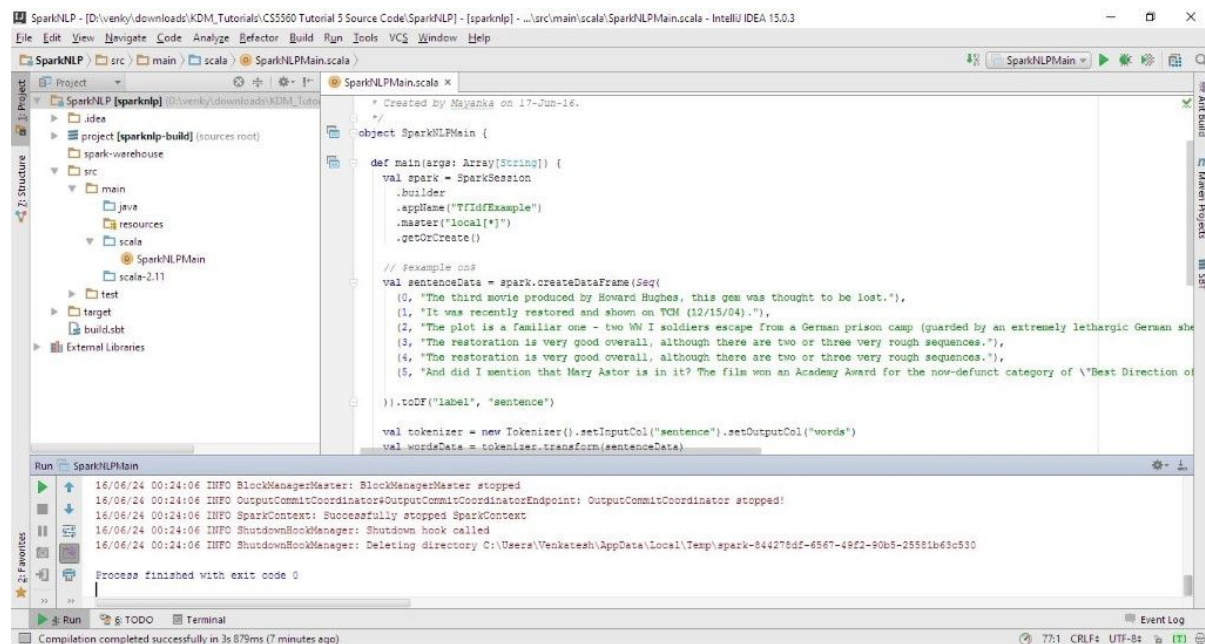
val tfidfvalues = tfidf.flatMap(f => {
  val ff: Array[String] = f.toString.replace(",", ";").split(";")
  val values = ff(2).replace("]", "").replace("'", "").split(",")
  values
})
```

Run TFIDF\_Main

```
16/06/24 00:09:41 INFO BlockManager: BlockManager stopped
16/06/24 00:09:41 INFO BlockManagerMaster: BlockManagerMaster stopped
16/06/24 00:09:41 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/06/24 00:09:41 INFO SparkContext: Successfully stopped SparkContext
16/06/24 00:09:41 INFO RemoteActorRefProviders$RemovingTerminator: Shutting down remote daemon.
16/06/24 00:09:41 INFO ShutdownHookManager: Shutdown hook called
16/06/24 00:09:41 INFO RemoteActorRefProviders$RemovingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
16/06/24 00:09:41 INFO ShutdownHookManager: Deleting directory C:\Users\Venkatesh\AppData\Local\Temp\spark-820dd5f0-cc18-44ff-b0da-95c24c2dc606

Process finished with exit code 0
```

## Implementation of TF-IDF using NLP:



```
object SparkNLPMain {
  def main(args: Array[String]) {
    val spark = SparkSession
      .builder
      .appName("TFIDFExample")
      .master("local[*]")
      .getOrCreate()

    // Example on8
    val sentenceData = spark.createDataFrame(Seq(
      (0, "The third movie produced by Howard Hughes, this gem was thought to be lost."),
      (1, "It was recently restored and shown on TCM (12/15/04)."),
      (2, "The plot is a familiar one - two WW I soldiers escape from a German prison camp (guarded by an extremely lethargic German she"),
      (3, "The restoration is very good overall, although there are two or three very rough sequences."),
      (4, "The restoration is very good overall, although there are two or three very rough sequences."),
      (5, "And did I mention that Mary Astor is in it? The film won an Academy Award for the now-defunct category of \"Best Direction of")
    )).toDF("label", "sentence")

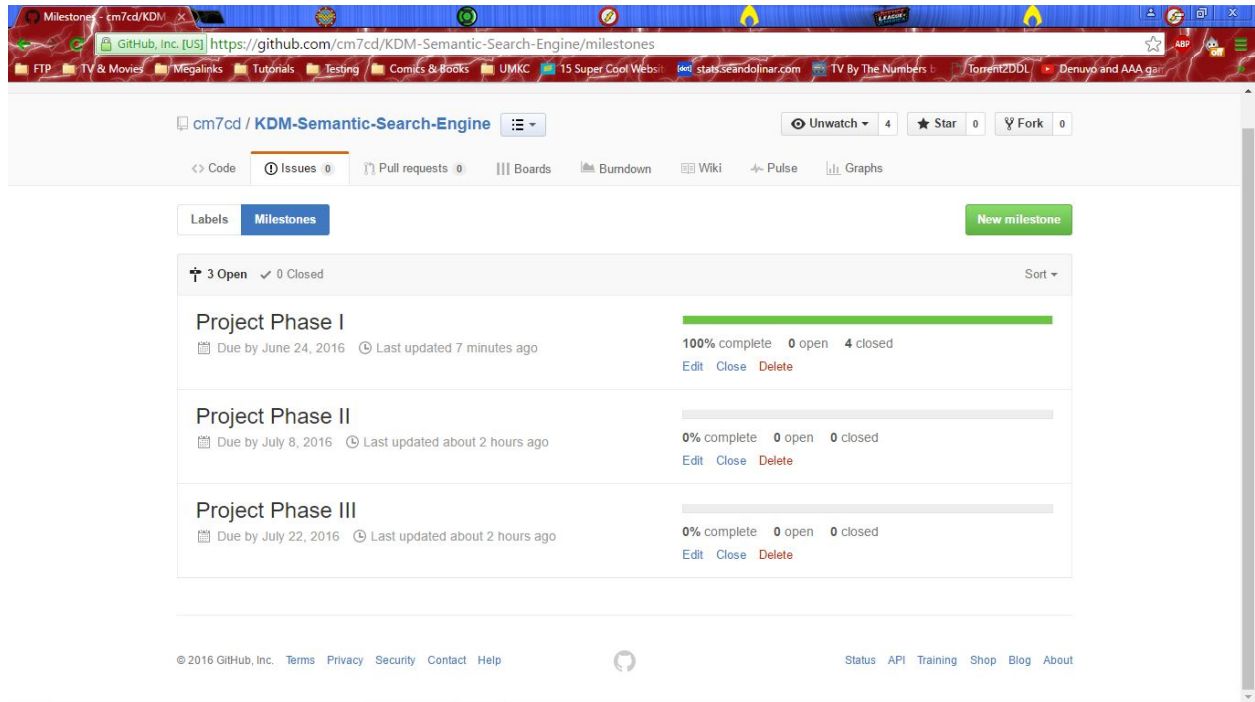
    val tokenizer = new Tokenizer().setInputCol("sentence").setOutputCol("words")
    val wordsData = tokenizer.transform(sentenceData)
```

Run SparkNLPMain

```
16/06/24 00:24:06 INFO BlockManagerMaster: BlockManagerMaster stopped
16/06/24 00:24:06 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/06/24 00:24:06 INFO SparkContext: Successfully stopped SparkContext
16/06/24 00:24:06 INFO ShutdownHookManager: Shutdown hook called
16/06/24 00:24:06 INFO ShutdownHookManager: Deleting directory C:\Users\Venkatesh\AppData\Local\Temp\spark-844278df-6847-49f2-90b5-25581b63c530

Process finished with exit code 0
```

## Milestones:

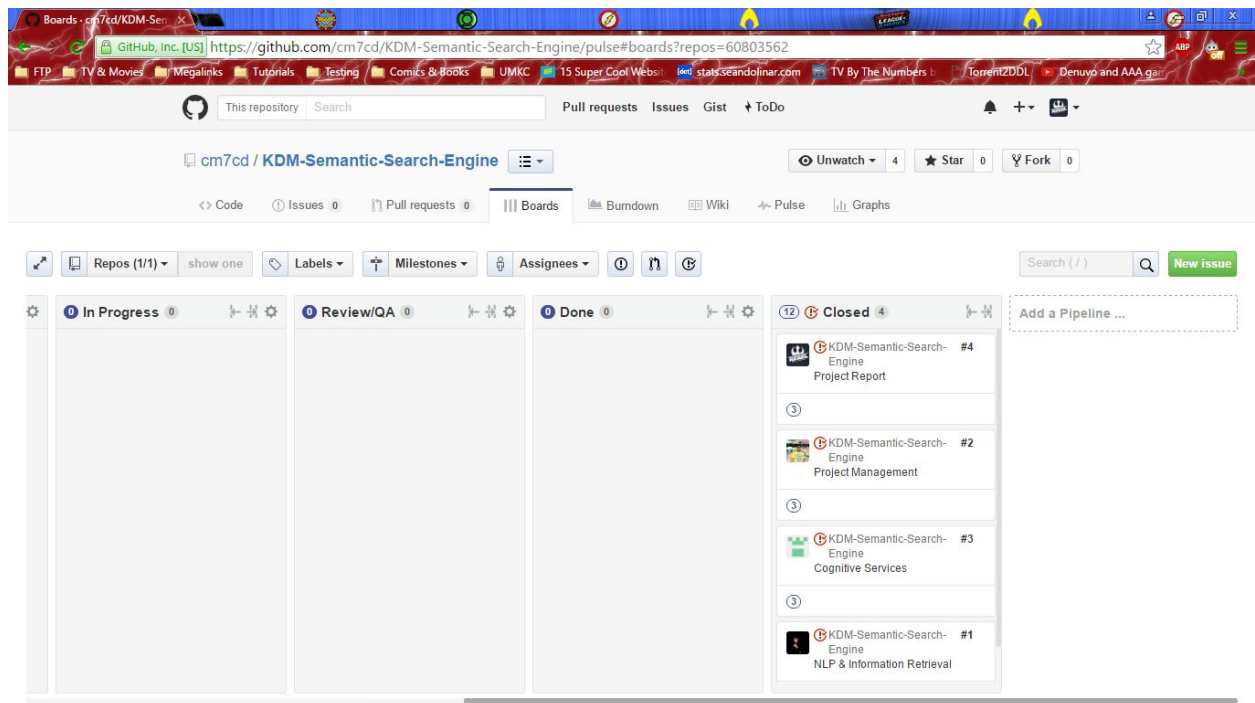


The screenshot shows the GitHub Milestones page for the repository **cm7cd / KDM-Semantic-Search-Engine**. The page displays three milestones:

- Project Phase I**: Due by June 24, 2016. 100% complete (0 open, 4 closed).
- Project Phase II**: Due by July 8, 2016. 0% complete (0 open, 0 closed).
- Project Phase III**: Due by July 22, 2016. 0% complete (0 open, 0 closed).

Each milestone includes a progress bar, a list of issues, and options to edit, close, or delete the milestone. The page also features a 'New milestone' button and a 'Sort' dropdown.

## Issues:

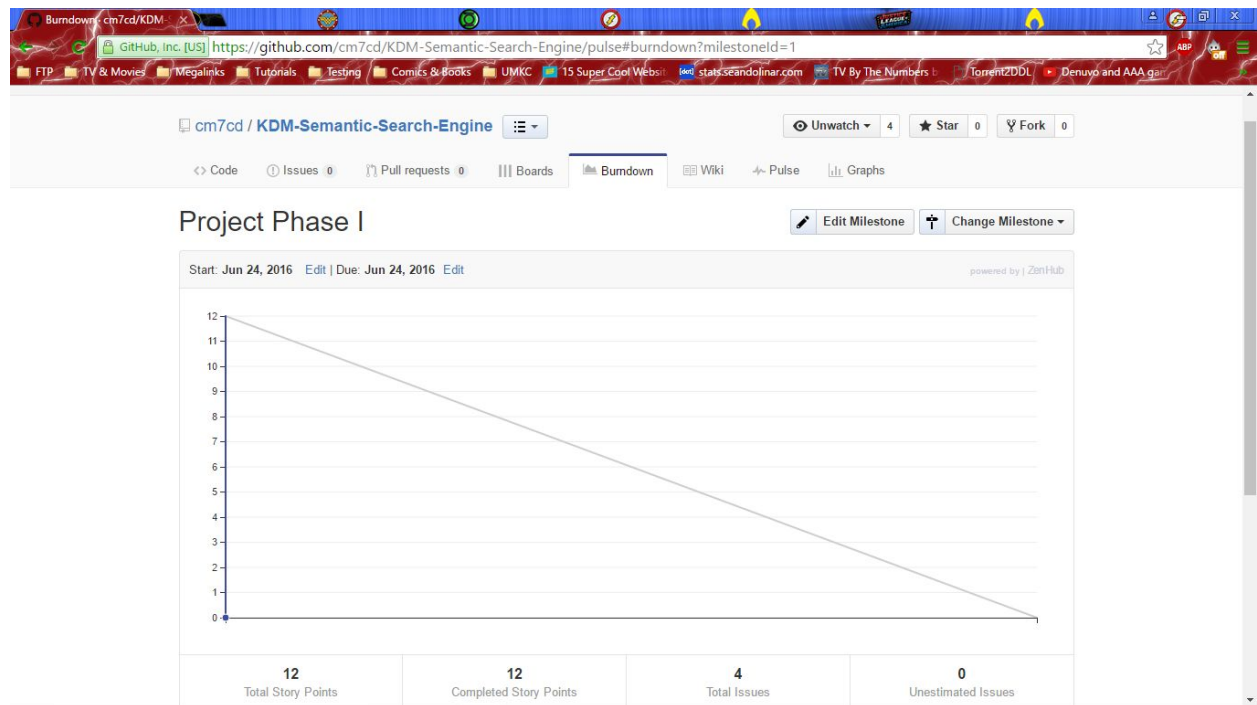


The screenshot shows the GitHub Issues page for the repository **cm7cd / KDM-Semantic-Search-Engine**. The page displays a Kanban board with four columns: **In Progress** (0 issues), **Review/QA** (0 issues), **Done** (0 issues), and **Closed** (4 issues). The **Closed** column contains four issues:

- #4**: KDM-Semantic-Search-Engine Project Report
- #2**: KDM-Semantic-Search-Engine Project Management
- #3**: KDM-Semantic-Search-Engine Cognitive Services
- #1**: KDM-Semantic-Search-Engine NLP & Information Retrieval

The page also features a search bar, a 'New Issue' button, and a 'Add a Pipeline ...' button.

## Burndown Chart:



**GitHub URL:** <https://github.com/cm7cd/KDM-Semantic-Search-Engine/>

## Bibliography:

- <http://jmcauley.ucsd.edu/data/amazon/links.html>
- <http://nlp.stanford.edu/nlp/>
- <https://en.wikipedia.org/>