IR Assignment 1

Homework Submission Guidelines

1. **Due date: 28.11.17 at 23:55**
2. Homework can be done in groups of up to 2 students
3. **WORD submissions only! The file name is: HW1_Student1ID_student2ID**
4. The work must be typed in WORD (20% grade penalty otherwise)
5. Answers can be submitted either in English or Hebrew
6. HW submission should be done via moodle in the corresponding area (by **only** one of the students)
7. Late submission penalty (20% a day) for submitting after the assignment's due date
8. Questions / clarifications and more in the dedicated discussion sub-forum.

Dry part (60%)

Indexing and Ranking Models (20%)

Malware indexing and retrieval: A malware is a malicious software program with several characteristics. Malware has a type (virus, worm, ransomware..), author and a short sentence describing it. Malware search is aimed for retrieving malware information in response to malicious activity (query) instead of documents.

1. Describe the structure of the index you would construct for malware search given a collection of malware objects. (10%)

2. Suggest a ranking model for keyword queries that addresses the malware description field and malware type field. Describe the exact way you intend to use the corpus statistics. (Hint. Treat each malware characteristic as a document and calculate tf.idf accordingly)(10%)

Vector space model (20%):

The following matrix represents the word frequencies of four documents d1, d2, d3, d4. Columns represent the documents in the above order; rows represent the vocabulary of six indexed terms a,b,c,d,e,f in that order. (Use log base 10.)

	d1	d2	d3	d4
a	0	1	1	1
b	1	2	0	1
c	2	0	0	0
d	0	0	0	0
e	1	0	1	1
f	7	5	7	2

Assume that the fraction of corpus documents in which each term appears is 10%, 10%, 20%, 5%, 50%, 90% for the terms a, b, c, d, e, and f, respectively.

1. Compute the cosine similarity between d1 and d2 where terms are represented by the tf-idf scheme. (Describe the tf-idf scheme you have used and provide details of the computation.) (5%)
2. Rank the documents in response to the query "a b f". Use the vector space model where document terms are represented by tf and query terms by tf-idf. Provide details of your computations. (5%)
3. Suggest a measure that estimates the similarity between two terms in the corpus. (10%)

Term Weighting and Ranking (10%):

1. Explain how the stemming process affects *tf.idf* (5%)
2. Describe 2 cons and 2 pros of the removal of *stopwords*. (5%)

True/False questions (10%) :

Mark each of the following sentences as true or false and give a short (but full) explanation for why your answer is correct:

1. df_t is an inverse measure of the informativeness of term t. (1%)

2. Cosine similarity and Euclidean distance are equivalent for ranking documents in response to a query under some condition. (4%)

3. Vector space-based retrieval is always more effective than Boolean retrieval. (2%)

4. In the vector space model, the higher the value of the normalization factor for a document is, the lower are the chances of retrieval for that document. (1%)

5. The stemming process increases the number of unique terms in the index (2%)

Wet part – Intro to Indri (40%)

Part A:

1. The collection for Part A is located in **docs.txt**
2. Create an Indri index using the following parameters:

```
<parameters>
  <memory>1G</memory>
  <corpus>
    <path> docs.txt  path</path>
    <class>trectext</class>
  </corpus>
  <index>Your folder and index name</index>
</parameters>
```

If the index is created correctly you will find a manifest file **inside** the index directory which looks as follows:

```
<corpus>
    <document-base>1</document-base>
    <frequent-terms>0</frequent-terms>
    <maximum-document>5</maximum-document>
    <total-documents>4</total-documents>
    <total-terms>212</total-terms>
    <unique-terms>140</unique-terms>
</corpus>
```

Run retrieval with the following parameter file:

```
<parameters>
    <memory>1G</memory>
    <index>/home/iradmin/HW1/Index</index>
    <count>5</count>
    <trecFormat>true</trecFormat>
    <baseline>tfidf,k1:1.0,b:0.3</baseline>
</parameters>
```

1. Run a query "corporation" over the collection using the above parameter file
   a. How many documents did you retrieve?
   b. How many documents did you expect to retrieve? Perform and explain the change that is needed for getting the additional documents. (Examine the text of documents.)
2. Write a query that will return document D2 first; use up to 2 words; explain your choice.
3. Write a query that will return document D1 first; use up to 2 words; explain your choice.
4. By running the query: " Michael Jackson" you will retrieve document D4.
   a. Do you think D4 is a relevant document for this query? Explain.
   b. Type a query for which D4 can be marked as relevant document; use up to 2 words; explain (refer the ranking score assigned to D4 as a result of the two queries)

Part B:

1. The files for PartB are located in /data/HW1/PartB
2. In the PartB folder you will find the following files and directories:
   a. "AP_Coll.tgz" compress file contains AP documents ("database")
   b. "queries.txt" – query file with 150 queries
   c. "qrels_AP" file – the AP relevance judgments
   d. "StopWords.xml" – the INQUERY 418 stopwords list
   e. "IndriBuildIndex.xml" – build index configuration file

3. Build 4 indexes using the given "database" directory and parameter file " IndriBuildIndex.xml".
   Report the time it takes to build each index (you can use **stopwatch** or use the "**time**" command to launch prior to IndriBuildIndex application):
   a. Index1: **Without** stopwords removal and **without** stemming.
   b. Index2: **With** stopwords removal and **without** stemming.
   c. Index3: **Without** stopwords removal and **with** stemming (Use "Krovetz" stemmer)
   d. Index4: **With** stopwords removal and **with** stemming
   (Note: Create first 4 index directories, each of which for an index version)
4. Which index version took less time to be created? Explain.

5. Run retrieval over the four indexes with the following parameter file (using tf.idf weights):

```
<parameters>
    <memory>1G</memory>
   <index>Your index Path</index>
   <count>1000</count>
   <trecFormat>true</trecFormat>
   <baseline>tfidf,k1:1.0,b:0.3</baseline>
</parameters>
```

6. In your irstudent directory, unpack the trec_eval file located in the 'parameters.tgz' file.
   Use the trec_eval application to evaluate the 4 retrieval results and complete the following table. Which retrieval result obtained the highest MAP value? Explain.

Stopword Removal	Krovetz Stemmer	MAP	P@5	P@10
Without	**Without**			
With	**Without**			
Without	**With**			
With	**With**			

Good Luck