זמן ההגשה והבדיקה הפרונטאלית יקבע עם ועד הכתה..

בניית מנוע לאחזור מידע

בפרוייקט זה עליכם לבנות מנוע לאחזור מסמכים ממאגר מסמכים. שפת התכנות היא java או C# . **העבודה בזוגות בלבד .**

הוראות ההגשה מפורטות למטה. בהמשך יפורסם מועד לבדיקה פרונטלית. בבדיקה תדרשו להציג תוכנה עובדת, להסביר את אופן עבודתה, ותיבחן יכולת המנוע על שאילתות חדשות, שתוצגנה למנוע בעת הבדיקה. **חובה להגיש מנוע עובד כדי לעבור את הקורס (יש להגיע בהרכב מלא של הזוג לבחינה, יתכן ציון שונה לסטודנטים בזוג על פי התרשמות הבוחן)**.

## באתר הקורס תמצאו:

1. מאגר מסמכים – מתוך מסמכים אלו המנוע יתבקש להחזיר מסמכים רלוונטים. המסמכים לקוחים מ TREC.
2. שאילתות – 10 שאילתות (topics) עליהן המנוע צריך לענות. שימו לב כי לכל שאילתא יש מספר מזהה בן שלוש ספרות. יש להשתמש רק ב title הנמצא ב topic
3. רשימת stop - words
4. הסבר לשימוש בתוכנת ההערכה - הסבר על העבודה עם ה TREC-EVAL
5. תוכנת הערכה TREC-EVAL
6. קובץ QRELS – לשימוש ב TREC-EVAL. הקובץ מכיל הערכות רלוונטיות (relevance judgments) (כלומר, לכל זוג של שאילתא/ מסמך האם המסמך רלוונטי לשאילתא) .

## סעיף א: בנייה

יש לבנות מנוע חיפוש על פי אחד המודלים שנלמדו בכיתה על פי בחירתכם. המנוע צריך לאפשר שאילתות ללא אופרטורים בין המילים (כלומר המשתמש לא מציין and או (or וכן לתמוך בשאילתות של ביטויים שיוגדרו על ידי גרשיים כגון “information retrieval” . המנוע יכול לפרש את השאילתות הרגילות (ללא גרשיים) על פי בחירתכם ) and או or או שילוב כלשהו של שניהם).

יש לבנות מנוע הכולל את המחלקות הבאות:

***(הערה: חובה*** להשתמש בשמות המחלקות שיוצגו להלן. ניתן להוסיף מחלקות נוספות כראות עינכם.)

:

1. **ReadFile** - מחלקה שתקרא את מאגר המסמכים. בכל קובץ יש הרבה מסמכים, צריך טיפול לזיהוי תחילה של כל מסמך.
2. **Parser** - מחלקה שתפרק את המאגר ל terms. ה parser צריך להתאים למסמכים במאגר המסמכים. אפשר להתייחס רק לטקסט שנמצא בין התגים <TEXT>. יש להוריד STOP-words על פי רשימת ה stopwords שפורסמה..). אין צורך להתיחס באופן מיוחד ל סימני הפיסוק.

ה Parser יופעל הן על מאגר המסמכים והן על השאילתות שתישלחנה למנוע. ניתן (אך לא חובה) להשתמש ב parser מוכן.

1. - **Stemmer** יש לבנות מודול של stemmer על פי porter (אפשר להשתמש בקוד פתוח), כדי לאפשר הרצות השוואה עם ובלי ה stemmer כפי שמפורט בהמשך.
2. **Indexer** - ליצירת קבצים הופכיים. ה- Indexer מקבל את המילים מה parser ובונה inverted index . ה Index צריך לכלול מילון שיועלה לזכרון הראשי בחיפוש . המילון ימומש במבנה פשוט של מערך ממויין (או טבלת Hash) או עץ. יש לבנות גם קובץ Posting שיאוחסן בדיסק שיכלול מידע נוסף על המילים והמסמכים לפי בחירתכם, ולפחות את מיקום של כל מילה במסמך.
3. **Utility index** - - יש לאפשר שירות נוסף של חיפוש על פי שדה מסויים, לפי בחירתכם, שמופיע ב meta-data של המסמכים (שנה, מקור וכו'), לשם כך יש ליצור אינדקס מיוחד על פי השדה שבחרתם ולאפשר בממשק חיפוש מסמכים על פי ערכי השדה הנ"ל.

**יש לאפשר הרצה של סעיפים 6-8 בנפרד מסעיפים 1-5. כלומר, בניית האינדקסים (הכנת המאגר) היא תהליך ארוך שצריך להתבצע לפני הבדיקה הפרונטאלית. עם זאת יש לאפשר לבודק הרצה של בניית אינדקס במידת הצורך של הכנת המאגר.**

1. **Ranker** - מדרג את התשובות לשאילתות על פי נוסחת דירוג שפיתחתם. הנוסחה צריכה להתחשב לפחות במספר המופעים המנורמל של מילה במסמך, ובקרבה בין מילות השאילתא כאשר הן מופיעות במסמך .
2. **Searcher** - לביצוע שאילתות. ה Searcher- יקבל שאילתא , ינתח אותה בהתאם לניתוח הטקסט שנעשה על המסמכים, ויחזיר עד 20 מסמכים רלוונטים ביותר לשאילתא מתוך מאגר המסמכים שפורסם באתר. אם המנוע מוצא פחות מ-20 מסמכים רלוונטים, הוא יחזיר פחות. הדירוג יתבצע על ידי ה Ranker שכתבתם. ה Searcher יופעל על אוסף הtopics מ Trec שפורסמו באתר. השאילתא צריכה להיות מורכבת משדה title שלtopic בלבד. לשם כך ניתן ליצור בצורה ידנית קובץ שאילתות, הכולל אך ורק ה titles מתוך קובץ השאילתות (אין צורך בעיבוד אוטומטי של ה topic.) הפלט של ה- Searcher צריך להתאים לפורמט של Trec-Eval וצריך לכלול את הזיהוי של המסמכים הרלוונטים.
3. GUI - הממשק לשאילתות ולתשובות צריך להיות בסיסי. הוא צריך לאפשר הכנסה של שאילתא בודדת וגם לאפשר קריאה של קובץ טקסט שיש בו שאילתות כמו כן רצוי לאפשר שמירה של כל התוצאות בקובץ אחד. הממשק צריך לתמוך ב utility שהגדרתם (בסעיף 5).

## סעיף ב: הערכה

יש להריץ את המנוע על 10 השאילתות שהוגדרו. את התוצאות יש לשמור בקובץ תוצאות אחד מרוכז. יש לבצע שתי סדרות של הרצות: אחת עם stem ואחת ללא .stem. כל הרצה בקובץ נפרד. יש להשתמש בקובץ QRELS ובתוכנת TREC\_Eval כדי לחשב –

1. Precision ו Recall כוללים לכל שאילתא
2. Precision ו Recall כוללים למנוע.
3. Precision ב – 5, 10 ו-20 מסמכים לכל שאילתא ובממוצע למנוע.

## הוראות הגשה:

תאריך ההגשה יפורסם באתר. בכדי לקבל את מלוא הניקוד על הפרוייקט יש להיות נוכחים בבדיקה הפרונטלית, וכן להגיש לפי הפירוט הבא:

לספריית ה ftp (הכתובת תפורסם בהמשך):

תיקייה אחת. שם התיקייה תורכב ממספרי ת"ז של המגישים. לדוגמא:012345678\_012345677

בתיקייה יש לשים את הקבצים הבאים:

1. למתכנתים ב C# : קובץ exe של התוכנית. למתכנתים בג'אווה: executable jar file. כלומר, קובץ jar אותו ניתן להריץ דרך ה command line בצורה הבאה:

C:\java –jar *filename*.jar

1. קובץ readm;;;;e.txt עם הוראות הרצה.
2. קוד מקור מתועד.

לתא של איליה פרידמן:

דו"ח פרוייקט.

על הדו"ח לכלול את הפרקים הבאים:

1. עיצוב - הסבר מפורט על אופן פעולת המנוע שבניתם. אם הוספתם מחלקות, יש להסביר מה מטרתן ואיך הן פועלות.
2. תיעוד מפורט של אלגוריתם הדירוג (ranking) בו השתמשתם בסעיף 4. יש לתאר את הנתונים בקובץ ה posting שתומכים באלגוריתם. ההסבר צריך להיות מפורט ומובן ברמה כזו שמתכנת יוכל לקרוא אותו ולתכנת את האלגוריתם על פי ההסבר.
3. הערכה של המנוע - פלט של 10 שאילתות \_(כולל השאילתות) עם ובלי stem ופלט של תוכנת TREC-EVAL לחישוב המדדים כמפורט בסעיף ב.
4. סיכום – באילו בעיות נתקלתם וכיצד פתרתם אותם. מה היה הקושי הכי גדול בפרוייקט? המלצות לשיפור.

פירוט הערכת המנוע:

הרכב הציון:

40% קוד עובד – (להזכירכם, אי אפשר לעבור את הקורס ללא מנוע עובד).

10% - התמצאות בקוד ובנוסחאות

20% - דוח

20% - איכות התוצאות – האיכות נמדדת על פי תוצאות המדדים באופן יחסי לשאר הסטודנטים, כך שהתוצאות הטובות ביותר יזכו לציון מקסימלי בסעיף זה, תוצאות סבירות יקבלו לפחות חצי מהנקודות בסעיף זה.

3% - מקוריות הפיתרון.

7% - יעילות הפתרון (מקום וזמן) י

יורדו 10 נקודות על כל יום איחור בהגשה הן של הקוד והן של הדו"ח, מכל סיבה שאינה מוכרת רשמית על ידי האוניברסיטה (מילואים, אשפוז וכו')..שני השותפים בכל זוג חייבים להיות בבדיקה הפרונטלית ולהיות מסוגלים לענות על שאלות הבודק. ייתכן ציון שונה לסטודנטים באותו צוות על פי התרשמות הבוחן.

# בהצלחה!

ברכה ואיליה