# C:\Users\Awesome\Desktop\solr.png

# מסמך מערכת

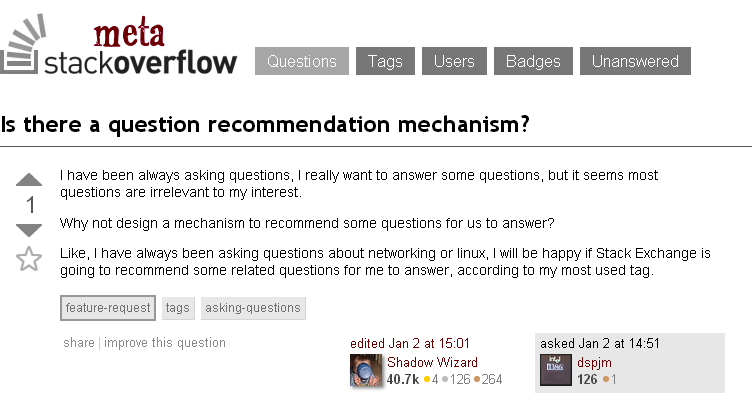
# stack**overflow** Toolkit

## כותבים :

ענבל עמרני, יזהר גלבוע, ערן הירש

# מוטיבציה

1. ניתן לראות שקיים צורך למערכת שמאפשרת למשתמש לקבל שאלות שיידע לענות עליהן.

[](http://meta.stackoverflow.com/questions/214650/is-there-a-question-recommendation-mechanism)

1. בשאלה שאין לה תשובות, מופיע בתחתית השאלה המלל הבא:



stackoverflow מעודדים משתמשים למצוא את העונה המתאים לשאלה.

1. באתר Q&A גדול אחר שנקרא Quora, יש מנגנון שמאפשר בהינתן שאלה למצוא את המשתמש המתאים לענות עליה.
2. קיים שוק חזק של אפליקציות מעל stackoverflow, האפליקציות בשוק מופיעות בקישור הבא :  
   <http://stackapps.com/questions/tagged/app>  
   המערכת שלנו תשתלב היטב בשוק הקיים.

# דרישות

הקדמה  
הפרויקט שלנו מתמקד באתר sof (stack**overflow**) כאשר נרצה לתת סט כלים על גבי האתר.  
הכלי הראשון שאנחנו רוצים ליצור הוא כלי Answerer Recommendation.

Answerer Recommendation - backend  
- בהינתן "שאלה חדשה" תחזיר את המשתמשים שהכי מתאימים לענות עליה.  
- בהינתן משתמש תחזיר את השאלות שהוא עשוי לענות עליהן.   
  
Answerer recommendation - UI

דף חיפוש  
תיבת טקסט (חיפוש) שמכניסים אליה קישור לשאלה ב- sof.  
הפלט של החיפוש יהיה משתמשים שמדורגים לפי המשתמש שהכי מתאים לענות עליה. בנוסף

למשתמש הכי מתאים ניתן להראות על איזה שאלה דומה הוא ענה.

דף רישום  
תיבת טקסט שמכניסים אליה username של משתמש ו-email.  
המשתמש יקבל למייל כל ערב רשימה של השאלות שהכי מתאימות לו לענות עליהן.

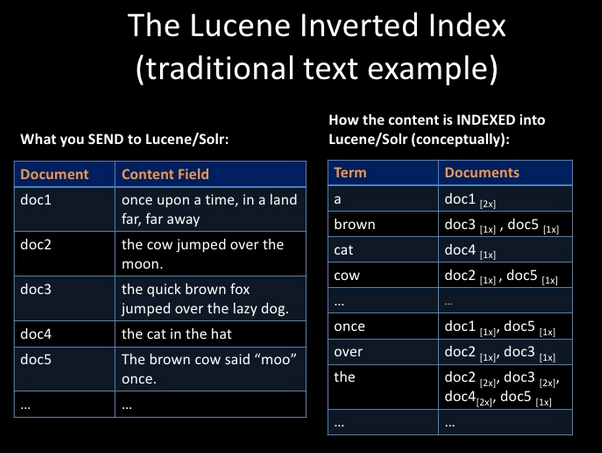
# סקירה טכנולוגית

## Solr

Solr הוא שרת חיפוש / אינדוקס המבוסס על ספריות Lucene לחיפוש / אינדוקס Full-Text.

## Lucene

Lucene היא ספריית חיפוש טקסטואלי בעלת ביצועים גבוהים, אך ניתן להתייחס אליה כספריית Token Matching ודירוג תוצאות בעלת ביצועים גבוהים, המסוגלת לבצע חיפוש טקסטואלי. אם נתרגם זאת למונחי Machine Learning, ניתן להסיק מכך ש Lucene Index הוא בעצם מטריצה רב מימדית שעבור שדה מסוים שנאנדקס נקבל מטריצה שממפה כל Token שאונדקס לכל המסמכים המכילים אותו.  
כמו כן עבור כל מסמך ניתן לקבל את ה"וקטור" של ה Tokensהמרכיבים אותו.



# אחזור מסמכים ב SOLR\LUCENE

Lucene משלבת שתי שיטות אחזור מרכזיות שמתחלקות לשני מודלים, מודל בוליאני (BIM), שלפיו מתקבלת ההחלטה איזה מסמכים הם תוצאות החיפוש, ומודל וקטורי (VSM), שלפיו מחולק ציון לכל אחד מהמסמכים שחזרו והתוצאות ממויינות לפיו.

לפי המודל הוקטורי גם מסמכים וגם שאילתות הם וקטורים ממושקלים במרחב רב מימדי שבו כל מילה (distinct index term הוא המינוח הנכון, אך בפרויקט שלנו שבו מאנדקסים טקסט אז כל term הוא בעצם מילה באנגלית) היא מימד, והמשקל הוא הערך של המילה לפי TF-IDF. אין דרישה מיוחדת של VSM שהמשקלים יהיו ב TF-IDF, אך אלגוריתם זה לרוב מניב תוצאות איכותיות.

עוד מידע על האחזור והדירוג ב Lucene:

<http://lucene.apache.org/core/3_6_2/api/core/org/apache/lucene/search/Similarity.html>

# רכיב MoreLikeThis של SOLR

רכיב MoreLikeThis נותן את היכולת לבנות חיפוש חדש על בסיס מסמכים המאונדקסים ב SOLR על מנת למצוא מסמכים נוספים ה**דומים** להם. הדמיון מוגדר על ידי המשתמש בפרמטרים המציינים איזה שדות מגדירים דמיון, וכמה השפעה לתת לכל שדה המשתתף בהגדרת הדמיון (Boosting). ישנם פרמטרים נוספים שרצוי להגדיר, אך הם פחות חשובים וניתן למצוא את כולם בדף ה Wiki של SOLR:

<https://wiki.apache.org/solr/MoreLikeThis>

הרכיב מחלץ את ה top TF/IDF terms, בעצם את המילים הכי ערכיות במסמך, על סמך השדות שהגדרנו עבורם דמיון. לאחר מכן הרכיב בונה ממילים אלו שאילתת Solr ומדרג את המסמכים שחזרו על בסיס פונק' האיחזור הבסיסית של Lucene.

# ארכיטקטורת על

## Backend

1. את כל מאגר sof אפשר להוריד באינטרנט, כ-25GB דחוסים של טקסט.
2. הקמת Solr על שרת וירטואלי מ-Google Compute Cloud.
3. אינדוקס 15 מיליון שאלות ותשובות של משתמשי sof מ-2008 ועד 2014 (קובץ Posts.xml).
   * בניית רכיב אינדוקס שמפרק את ה XML של הטקסט ובונה SolrDocuments, ושולח ל Solr ע"מ שמנוע ה Lucene יפרק אותם ל Tokens ויאנדקס.
4. בניית מנוע בסיסי שבהינתן מסמך (שאלה ב-sof) מוצא פריטים דומים לפי מספר מדדי דמיון :
   * השוואה של המילים בשאלות.
   * הכותרות של השאלות.
   * התגים שמצורפים לשאלות.
   * אורך השאלות.
   * טווחי הזמנים של השאלות (פחות או יותר באותם שנים).
   * כמה תשובות היו לשאלות.
5. תשתית לביצוע בדיקות אופטימיזציה  
   מציאת שאלות דומות והשוואה של המשתמשים שקיבלנו למשתמשים שענו בפועל לשאלות.

## UI

1. שימוש [בפרויקט Blacklight](http://projectblacklight.org/) כתשתית ל-UI.
2. בניית דף החיפוש למשתמש שיתקשר עם ה-Backend.
3. בניית דף רישום ומנגנון שליחת מיילים.

מקורות:

<http://lucene.apache.org/core/3_6_2/api/core/org/apache/lucene/search/Similarity.html>

<https://wiki.apache.org/solr/MoreLikeThis>

<https://cwiki.apache.org/confluence/display/solr/MoreLikeThis>