כריית טקסט – תרגיל בית מספר 1

תרגיל זה מטרתו לבחון את הדרכים השונות להכין טקסט לניתוח, לתרגל שימוש בפונקציות ניקוי הקיימות ולהבין את האתגרים והנושאים אליהם יש לשים לב כאשר מבצעים ניקיון והכנה של טקסט.

סט הנתונים בו נשתמש מכיל טוויטים שסווגו לסנטימנט חיובי ושלילי. מטרתנו בעתיד תהיה לבנות מסווג שיסווג טוויטים לטוויטים חיוביים וטוויטים שליליים באמצעות ניתוח של תוכן הטקסט.
בשלב זה עלינו להכין את הקובץ לניתוח. קובץ הנתונים מופיע באתר הקורס בשם Sentiment180_2000.csv, כל שורה במסד זה טוויט.

להלן תיאור משתני המסד:

שם משתנה	תיאור
Polarity	0 = שלילי 4 = חיובי
ID	מספר מזהה של הטוויט
Date	תאריך הטוויט
Query	שאילתא
User	שם משתמש שיצר את הטוויט
Text	התוכן של הטוויט

ניתן לקרוא פרטים נוספים על סט הנתונים בקישורים הבאים:

http://help.sentiment140.com/for-students/

https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf

1. טענו את סט הנתונים ל R לתוך Data frame.
2. (1 נק') צרו קורפוס המכיל את הטקסטים הנ"ל.
3. (3 נק') נקו את הטקסט בהתאם למה שנלמד בכיתה ובהתאם למטרת הניתוח. דווחו לפלט באילו פונקציות השתמשתם לניקוי ומאיזו סיבה (ציינו לפחות חמישה ניקיונות שביצעתם). ציינו אילו עוד פעולות הייתם מבצעים בעתיד כדי לנקות את הקובץ בצורה מיטבית.
4. (3 נק') שנו את סדר הפונקציות בהן השתמשתם בסעיף הקודם כך שיווצר סדר שגוי. פרטו שני שינויים שביצעתם המראים כיצד שינוי סדר הפעולות פגע בביצוע הניקוי. הציגו כל שינוי בנפרד. בקוד יש להציג את הניקוי המקורי (בשאלה 3), ואת שתי שיטות הניקוי הנוספות. גם אם שיניתם רק שורה אחת יש להציג את כל פונקציות הניקוי כך שניתן יהיה להריץ ולראות את השינוי. דווחו לפלט מה השינוי שביצעתם, למה הוא גרם והציגו דוגמא מהטקסט עבור כל שינוי.
5. (1 נק') צרו מטריצת מונחים-מסמכים, Term Document Matrix (TDM) , המכילה את כל המונחים לאחר ניקוי. כמה מונחים יש במטריצה? מה ה sparsity שלה? דווחו לפלט.

6. (1 נק') סננו מהמטריצה את כל המונחים המופיעים בשני מסמכים או פחות. מה ה sparsity כעת? דווחו לפלט
7. (1 נק') צרו טבלה המכילה את 20 המונחים המופיעים בתדירות הגבוהה ביותר בקורפוס אחרי ניקוי. הטבלה תכיל מונח ומספר הפעמים שהוא מופיע בטקסט. הדפיסו 6 שורות ראשונות של הטבלה ודווחו לפלט.

אופן ההגשה:

- הגשה בזוגות או ביחידים.
- הגשה דרך אתר למידה.
- יש להגיש קובץ R מתועד, וקובץ פלט הגשה בפורמט word/ pdf המכיל את התשובות לשורות המסומנות בצהוב.
- אין לכווץ את קבצי ההגשה (zip).

קובץ פלט לתרגיל מספר1

חלק 2

סעיף	מטלה	פתרון
3	פונקציות ניקוי בהן השתמשתם וסיבה (לפי הסדר בקוד)	**הסבר מקדים:** תהליך הניקוי בוצע בשני שלבים, והוא מהווה שלב חשוב בתהליכי עיבוד שפה טבעית אשר עוזרת למנוע "רעש" בניתוח הנתונים ובכך משפרת את איכות הנתונים. השלב הראשון הוא הניקוי על ידי שימוש בביטויים רגולריים ,(Regular Expressions) אשר מאפשרים לזהות ולהסיר תבניות טקסט מוגדרות מראש, כגון שמות משתמשים המתחילים ב-@ או כתובות אינטרנט המתחילות ב .http-שלב זה בוצע על טקסטים מסוגcharacter בגלל שביטויים רגולריים מותאמים לעבודה עם מחרוזות טקסט. בשלב השני, הנתונים עובדו בעזרת ספריית (Text Mining) TM לאחר המרה לאובייקט מסוג ,Corpus המהווה מבנה נתונים ייעודי לעיבוד טקסטים. בעזרת TM בוצעו פעולות ניקוי נוספות, בהן המרה לאותיות קטנות, הסרת סימני פיסוק, הסרת מספרים, הסרת מילים נפוצות (stop words) וטיפול ברווחים מיותרים. שני שלבים אלו משלימים זה את זה, שכן TM לא יכולים לפעול על מבנה של character, שגן דורשת נתונים בפורמט קורפוס . **חלק א'** - לפני יצירת אובייקט הקורפוס : (שימוש בביטויים רגולריים (regular expression)) 1. הסרת כל המשתמשים שמתחילים ב@ ועד לרווח הראשון.(ע"י פונקציית gsub) הסבר : נסיר את כל שמות המשתמשים שכן הדבר אינו חשוב לנו לניתוח הסנטימנט 2. הסרת כל הכתובת שמתחילם ב http ועד לרווח הראשון (ע"י פונקציית gsub) הסבר : נסיר את כל הכתובות שכן הדבר אינו חשוב לנו לניתוח הסנטימנט **חלק ב'** - אחרי יצירת אובייקט הקורפוס : (שימוש בספרית TM) 1. המרה מאותיות גדולות לאותיות קטנות ע"י פונקציית (content_transformer(tolower

		הסבר: בשביל שהמחשב יבין שמילה עם אותיות גדלות זה אותו מילה עם אותיות קטנות צריך להמיר את כל המילים למכנה משתוף של אותיות קטנות כי לניתוח להסנטימנט הדבר אינו חשוב וכי רק מוסיף יותר רעש למודל. 2. הסרה של כל סימני הניקוד/פיסוק ע"י פונקציית removePunctuation הסבר: סימני פיסוק יכולים להוביל לגרסאות שונות של אותה מילה (למשל, "מילה" ו-"מילה,"). הסרתם יוצרת אחידות ומונעת ריבוי בלתי נחוץ של מאפיינים במודל. בנוסף, סימני פיסוק אינם נושאים משמעות סמנטית. 3. הסרת של כל המספרים ע"י פונקציית removeNumbers הסבר: מספרים אינם מכילים משמעות סמנטית ישירה, ולכן בטקסטים שמטרתם להבין הקשר או ניתוח רגשי, המספרים אינם רלוונטיים ועלולים להסיט את תשומת הלב מהנושא המרכזי. 4. הסרת של מילים נפוצות מתוך רשימה "english" ע"י פונקציית removeWords הסבר: מילים נפוצות מופיעות כמעט בכל הטקסטים ואינן מוסיפות ערך סמנטי ייחודי. הסרתן מאפשרת למודל להתרכז במילים משמעותיות המספקות תובנות ייחודיות. 5. הסרת של כל הרווחים המיותרים ע"י פונקציית stripWhitespace הסבר: רווחים מיותרים עלולים לשבש תהליכים כמו טוקניזציה (פירוק טקסט למילים), יצירת וקטורים וניתוח סטטיסטי. הסרתם מייעלת את הדיוק של התהליכים הללו. לדוגמה, "שלום ילד" (עם שני רווחים) ו-"שלום ילד" (רווח אחד) עלולים להתפרש כשתי מחרוזות שונות על ידי האלגוריתם.
3	אילו ניקיונות הייתם מבצעים בעתיד	1. הסרת אימוג'ים (יכול להיות שלא ארצה למחוק את זה כי זה עוזר לניתוח הסנטימנט) 2. הסרה של מילים עם שכיחות נמוכה 3. הפעלת פעולת Stemming בשביל למצוא את שורש המילה ובכך יאפשר צמצום של מילים וקישור של מילים דומות.

		4. הוספת לרשימת המילים הנפוצות מילים שחוזרות ואיני רוצה אותם לניתוח.
4	שינוי ראשון שביצעתם (ביחס לסדר המקורי), נא לתאר את השינוי שבוצע ולתת דוגמא.	הסרת הרווחים לפני הסרת מילים נפוצות – ייתכן וישנם רווחים שנוצרו לאחר מכן בעקבות הסרה לדוגמה: **remove white spaces** [1] " awww thats a bummer you shoulda got david carr of third day to do it d" [1] "is upset that he cant update his facebook by texting it and might cry as a result school today also blah" [1] " i dived many times for the ball managed to save the rest go out of bounds" [1] "my whole body feels itchy and like its on fire " [1] " no its not behaving at all im mad why am i here because i cant see you all over there " [1] " not the whole crew " **remove stopwords** [1] " awww thats bummer shoulda got david carr third day d" [1] " upset cant update facebook texting might cry result school today also blah" [1] " dived many times ball managed save rest go bounds" [1] " whole body feels itchy like fire " [1] " behaving im mad cant see " [1] " whole crew "
4	שינוי שני שביצעתם (ביחס לסדר המקורי), נא לתאר את השינוי שבוצע ולתת דוגמא.	הסרה של סימני הניקוד לפני הסרת שם המשתמש המתחיל ב @ - הורדת ה-@ תגרום לחוסר חוקיות שכתוצאה מכך לא יהיה ניתן להוריד את שמות המשתמשים ובכך יגרום להוספת רעש למודל. **lowercase** [1] "@switchfoot - awww, that's a bummer. you shoulda got david carr of third day to do it. ; d"

[1] "is upset that he can't update his facebook by texting it... and might cry as a result school today also. blah!" [1] "@kenichan i dived many times for the ball. managed to save 50% the rest go out of bounds" **remove punctuation** [1] "switchfoot awww thats a bummer you shoulda got david carr of third day to do it d" [1] "is upset that he cant update his facebook by texting it and might cry as a result school today also blah" [1] "kenichan i dived many times for the ball managed to save 50 the rest go out of bounds" **remove mentions (@)** [1] "switchfoot awww thats a bummer you shoulda got david carr of third day to do it d" [1] "is upset that he cant update his facebook by texting it and might cry as a result school today also blah" [1] "kenichan i dived many times for the ball managed to save 50 the rest go out of bounds"		
4625 - כולל כל המילים מאורך 3 ומעלה 4775 - כולל כל המילים מאורך 2 ומעלה נשאיר את כל המילים מאורך 2 ומעלה כי לדוגמה מילה "no " יכולה להשפיע לנו על ניתוח הסנטימנט	מספר מונחים	5
100%	sparsity	5
935 (כולל המילים מאורך 2 ומעלה) • נשאיר את כל המונחים באורך 2 ומעלה המופיעים בשלושה מסמכים ויותר – כלומר מונח שמופיע ב-3 מסמכים או יותר **ייכלל** במטריצה.	מספר מונחים	6
99%	sparsity	6

7	מונחים נפוצים	Term Frequency im 235 just 182 good 116 now 103 get 102 dont 99 ניתן להוסיף למילים הנפוצות את המילה im אשר אינה נחוצה לנו לניתוח הסנטימנט

```
# file: EX1

# Name: Niv Kotek, Genadi Birfir

# ID: 208236315, 312741622


# Using libraries

library(stringi)

library(textstem)

library(tm)

# set working directory

setwd("C:/Users/niv/Desktop/Text mining/EX1")

#~~~
```

```
#~~~
```

```
# Cleaning test by function
check <- function(text.cor,name) {
  cat(name,"\n\n")
  for (i in 1:6) {
    print(text.cor[[i]][1])
  }
  cat("\n\n")
}
#~~~
```

```
# remove numbers from twits
text.cor <- tm_map(text.cor, removeNumbers)
check(text.cor,"remove numbers")

# remove stopwords
text.cor<- tm_map(text.cor, removeWords,stopwords("english"))
check(text.cor,"remove stopwords")

# remove white spaces
text.cor<- tm_map(text.cor,stripWhitespace)
check(text.cor,"remove white spaces")
```

```
#~~~
```

```
#Create corpus
text.vec<- VectorSource(text.df)
text.cor<- VCorpus(text.vec)
# transferring data to lowercase
text.cor <- tm_map(text.cor,content_transformer(tolower))
check(text.cor,"lowercase")
# remove punctuation
text.cor <- tm_map(text.cor, removePunctuation)
check(text.cor,"remove punctuation")
# remove numbers from twits
text.cor <- tm_map(text.cor, removeNumbers)
check(text.cor,"remove numbers")
# remove white spaces
text.cor<- tm_map(text.cor,stripWhitespace)
check(text.cor,"remove white spaces")
# remove stopwords
text.cor<- tm_map(text.cor, removeWords,stopwords("english"))
check(text.cor,"remove stopwords")
```

```
########### Change order 2 ###########

#text coloum into new data frame

text.df<-Sentiments.df[,'Text']

head(text.df)

# remove all the characters after the http

text.df <- gsub("http\\S*", "", text.df)

head(text.df)

#Create corpus

text.vec<- VectorSource(text.df)

text.cor<- VCorpus(text.vec)

# transferring data to lowercase

text.cor <- tm_map(text.cor,content_transformer(tolower))

check(text.cor,"lowercase")

# remove punctuation

text.cor <- tm_map(text.cor, removePunctuation)

check(text.cor,"remove punctuation")
```

```
# Remove mentions (e.g., @username)
text.cor <- tm_map(text.cor, content_transformer(function(x) gsub("@\\S+", "", x)))
check(text.cor, "remove mentions (@)")
# remove numbers from twits
text.cor <- tm_map(text.cor, removeNumbers)
check(text.cor,"remove numbers")
# remove stopwords
text.cor<- tm_map(text.cor, removeWords,stopwords("english"))
check(text.cor,"remove stopwords")
# remove white spaces
text.cor<- tm_map(text.cor,stripWhitespace)
check(text.cor,"remove white spaces")
#~~~
```

```
inspect(text.tdm)

#~~~
```

```
term_frequency <- rowSums(text.mat)

# Sort terms by frequency in descending order
sorted_terms <- sort(term_frequency, decreasing = TRUE)
# Create a data frame of the top 20 terms
top_terms <- data.frame(
  Term = names(sorted_terms)[1:20],
  Frequency = sorted_terms[1:20]
)

# Display the table - 6
head(top_terms)
```