דו"ח פרוייקט גמר - עקרונות שפות תכנות

אתר החדשות הטובות

Omri Attiya & Shira Ezra

Principles of Programming Languages Spring/2019, Ben Gurion University, Israel

משימת הפרוייקט

אנחנו חיים בעולם מתועש, מורכב ומסוכן. רוב הדברים שמעניינים את הציבור הם הדברים שקריטיים לו: הישרדות, פוליטה ודברים קיצוניים שקרו ממש מעבר לפינה. לצערו רוב הדברים האלה הם חדשות "רעות". מה זה רעות? אפשר להתווכח על זה אבל לרוב האנשים זה אומר "דברים שאני לא רוצה שיקרו לי", או "זה משהו שאני ממש אתבאס ממנו". החדשות האלה מדווחות בתקשורת: טלויזיה רדיו ואתרי חדשות למיניהם. רוב החדשות האלו הן בנימה מפחידה, עצובה או מלחיצה, כי אין מה לעשות, זה מה שמעניין את רוב האנשים וצריך להספיק לדווח על זה כדי להספיק לדווח על הסיפור הרע הבא. מצב זה יוצר דחיקה של החדשות הטובות והנעימות מחיינו. רצינו ליצור פלטפורמה שתסנן את החדשות השליליות ובכך תאפשר למשתמש חוויה נחמדה בקריאת החדשות.

לכן החלטנו שהמשימה תיהיה ליצור אתר שאוסף חדשות חיוביות ובאמצעות Machine Learning מסננים ומציגים רק את החדשות הטובות.

דרכי פעולה

היו לנו שתי בעיות עיקריות בפרויקט:

1. למידת מכונה ו-sentiment analysis .
2. להביא כתבות מאתרים

1. למידת מכונה ו-sentiment analysis: אחרי קריאה במאמרים בנושא ראינו שדרך נפוצה לזהות אם משפט הוא חיובי או שלילי הוא ניתוח באמצעות n-gram, ולכן החלטנו להשתמש ב- 1 gram . בנוסף, אנחנו משתמשים בהתפלגות מילים כדי להוציא פיצ'רים: כלומר סופרים כמה פעמים מילה מופיע בכל הדטא שלנו, ומוציאים את 5000 המילים הנפוצות ביותר.

   החלטנו להשתמש ב-NLTK על מנת לנתח את המשפטים בקלות, בנוסף במעבדה השתמשנו בכלי זה כדי לבצע משהו דומה ולכן היה לנו יותר פשוט ואינטואיטיבי לעשות זאת. בנוסף ניסינו להשתמש ב 2-gram , וstop_words, אבל הוספת אלא הורידו לנו את התוצאות ולכן החלטנו לוותר עליהם על מנת לפשט את המודל.

המודל עם הביצועים הטובים ביותר היו Naïve Bayes.

2. כתבות מאתרים: כרגע אנחנו מתממשקים עם 3 אתרי חדשות מוכרים ומוערכים: BBC, CNN ו –The New York Times . בהתחלה התחברנו לכל אחד מהאתרים עם ה-developer API שלהם. לאחר כמה חיפושים באינטרנט נתקלנו ב-API חדש ונוח שנותן גישה להמון אתרי חדשות ומוציא מהם מידע על הכתבות שלהם: כותרת, תמונה וקישור לכתבה.
לאחר מכן אנחנו מבצעים כריית תוכן על עמודי החדשות, כאשר לכל עמוד יש מבנה HTML שונה וממנו אנחנו מחלצים את התוכן של הכתבה, ואותו מעבירים במודל ביצרנו. אם המודל מסווג אותו כחיובי אנחנו מציגים את הכתבה.

הוראות התקנה של הכלי

בפרוייקט השתמשנו בכלים הבאים (מערכת הפעלה windows 10):

1. Pip install flask-socketio
2. Pip install newsapi-pyhton
3. Pip install nltk
4. Pip install beautifulsoup4
5. Pip install flask

הסטוריה של שימוש בכלי

לנו אישית יצא להשתמש כמעט בכל הכלים במהלך העבודות בקורס ובמטלות הבית. ישנם המון שימושים לכלים שבחרנו, אך לא התעמקנו בפרויקטים ספיציפים. כן נעזרנו באינטרנט כדי להבין איך עושים דברים מסויימים אבל לא מצאנו עבודות של שימוש בכלי. אנחנו יודעים שמשתמשים ב beautifulsoup כדי לבצע כריית תוכן, בflask כדי לעבד בקשות get & post וב-nltk כדי לעבד שפה טבעית.

Datasets

מצאנו מאגר מידע ב- Kaggle. המאגר מכיל כ –10K משפטים חיובים ושלילים בהתפלגות של כ-50%. בחרנו במאגר זה מאחר והמאגר הכיל משפטים בשפה גבוהה, שמתאים לכתבות בעיתונים ואתרי חדשות, וכי הוא מכיל משפטים חיוביים ושליליים, שזה בדיוק מה שאנחנו רוצים לסווג.

מקור - https://www.kaggle.com/chaitanyarahalkar/positive-and-negative-sentences

שימוש בכלי

- flask – השתמשנו בכלי זה כדי לעבד בקשות get ו-post ולהציג עמודי html של האתר שלנו.
  פונקציות עיקריות: *render_template, route, url_for*
- flask socket io – השתמשנו בכלי זה כדי לשלוח בצורה אסינכרונית הודעות לעמודי ה-html ולשרת כדי להוסיף כתבות בזמן אמת בלי לחכות שכל הכתבות יטענו.
  פונקציות עיקריות: *emit, socketio.on(event)*
- Beautifulsoup – השתמשנו בכלי זה כדי לבצע כריית תוכן של עמודי html של אתרי החדשות השונים אליהם התממשקנו.
  פונקציות עיקריות: *find_all*
- Nltk – השתמשנו בכלי זה כדי לעבד מידע טקסטואלי וכדי לבנות מודל של שלמידת מכונה עליו.
  פונקציות עיקריות: *word_tokenize, ngram, sent_tokenize, FreqDist, NaiveBayesClassifier*
- Newsapi – השתמשנו בכלי זה כדי לקבל מידע עבור חדשות. המידע מועבר דרך צד שלישי (newsapi) שמאחזר מבני json שמכילים חדשות מאתרים שונים לפי פרמטרים שונים.
  פונקציות עיקריות: *get_everything*

תוצאות

הגענו לתצאות של accuracy 75% ואתר מדהים שמראה בעיקר כתבות חיוביות.

מסקנות

המסקנות אליהן הגענו בפרויקט הן:

1. יש לנו עוד הרבה ללמוד על למידת מכונה. זה תחום שאנחנו אישית לא התנסנו בו אף פעם והפעם הראשונה שלנו בו הייתה בקורס הזה, וזה לא פשוט בכלל. אנחנו חושבים שבהינתן עוד זמן יכולנו לשפר את התוצאות שלנו.
2. לאחר הרצת הכלי הגענו למסקנה שרוב הכתבות הטובות הן בנושא הספורט.
3. לעצב אתר שיראה ויתפקד כמו שצריך זה עבודה לא פשוטה, מאתגרת וכיפית בסה"כ.
4. ניתוח טקסטואלי הוא הרבה יותר קשה מניתוח של דטא אמפירי – זה לא משהו שקל למדוד או להעריך. צריך להתחשב בהמון דברים כמו: הקשר של מילה במשפט, נטייה, שורש, נושא וכו'.

הערה

כדי להריץ את האתר:

1. להתקין את כל הכלים שפורטו למעלה.
2. להיכנס לתקיית הפרוייקט מה-cmd ולהריץ את הפקודה: python -m flask run
3. להיכנס מדפדפן chrome לכתובת: http://127.0.0.1:5000/main