*תיק פרויקט גמר*

**<Sentence compression using supervised machine learning>**

Diagram

Description automatically generated with medium confidence

**המשתתפים: מר יבגני גאיסינסקי**

**מנחה אקדמי: ד"ר מרינה ליטבק, ד"ר הדס חסידים**

**תוכן עניינים**

[סקר ספרות: 3](#_Toc87096968)

[יעדים: 6](#_Toc87096969)

[יישום: 7](#_Toc87096970)

[מימוש: 8](#_Toc87096971)

[אפיון דרישות: 9](#_Toc87096972)

[ניהול סיכונים: 10](#_Toc87096973)

[תוכנית עבודה: 11](#_Toc87096974)

[תוצאות ומסקנות: 12](#_Toc87096975)

## 

## **סקר ספרות:**

**הקדמה:**

בעשר השנים האחרונות היו הרבה מחקרים מוקדשים לצמצום משפטים. פישוט משפטים נועד להפוך משפטים לקלים יותר לקריאה ולהבנה. צמצום משפטים היא משימה שמטרתה ליצור משפטים קצרים מהנתון תוך שמירה על התוכן החיוני.

בנוסף מטרתה של צמצום משפטים היא לייצר סיכום של משפט בודד ששומר על המידע החשוב ביותר תוך שמירה על משמעותו. המשימה משכה תשומת לב רבה בשל הפוטנציאל שלה ליישומים כגון סיכום טקסט, כתוביות, והצגת טקסט על מסכים קטנים.

המטרה העיקרית של פישוט משפטים היא להפחית את המורכבות הלשונית של הטקסט, תוך שמירה על המידע והמשמעות המקוריים שלו. משימת הפישוט הייתה נושא למספר מאמצי מידול בשנים האחרונות בשל הרלוונטיות שלה עבור יישומי NLP וניתוח טקסטים. לדוגמה, רכיב פישוט יכול לשמש כשלב עיבוד מקדים לשיפור הביצועים של מנתחים, סיכומים ומתווי תפקידים סמנטיים.

**הבעיות:**

* צמצום משפטים היא משימת יצירת השפה הטבעית של קיצור אוטומטי של משפטים. מכיוון שדחיסות טובות צריכות להיות דקדוקיות ולשמור על משמעות חשובה, הן חייבות להיות מוערכות לאורך שני ממדים אלה. בניגוד למשימות אחרות, כמו תרגום מכונה, אין הערכה משותפת עבור דחיסה.
* קושי אחר בהערכת מערכות דחיסה בצורה הוגנת הוא שקשה להגדיר מדד אוטומטי חסר פניות***[9]***. הערכה אוטומטית מסתמכת על השוואה לתקן זהב יחיד בטווח קבוע מראש, מה שמגביל מאוד את סוגי הדחיסות שניתן לשפוט בצורה הוגנת***[2]***.
* המחיקות הן עצמאיות. מבנה התלות של משפט עשוי להיות ללא שינוי כאשר מילים תלויות אינן נמחקות כיחידה***[7]***. דוגמאות למילים שיש להתייחס אליהן כיחידה אחת כוללות שלילות ופריטי קוטביות שלילית או ביטויים מסוימים של מספר מילים***[5]***.
* ממחקר שנערך, נמצא כי לא קיימות מערכות לצמצום משפטים בשפה העברית, לכן נראה בהמשך פתרון בעיה זאת בפרויקט.

**פתרונות קיימים:**

רוב המחקרים חולקים שני מאפיינים: הם מסתמכים על תחביר, והם מפוקחים. מידת התלות בתחביר משתנה בין השיטות. חלקם משתמשים במנתח כדי לזהות ולאחר מכן לשמור על קשרים חשובים מסוימים***[6]***, אך אינם דורשים השלמה לנתח , או להשתמש בייצוג תחבירי כדי לחלץ תכונות.

**בדרך כלל מערכות הדחיסה המודרניות המבוססות על מחיקה מאמצות גישת גיזום עצים או גישת תיוג רצף. הראשון משתמש במידע תחבירי כדי לנווט על עץ תחבירי של משפט ולהחליט אילו חלקים ממנו להסיר*[3]*. עם הופעת המודלים של רצף לרצף אפשר היה לדלג על שלב הניתוח התחבירי ולפתור את המשימה ישירות, על ידי עיבוד משפט אחד בכל פעם וקבלת החלטות בינאריות אם לשמור אסימון או למחוק אותו. היתרונות של גישות כאלה כוללים סיכוי נמוך יותר להחדיר התפשטות שגיאות מהחלטות ניתוח שגויות, כמו גם מהירות אימון והסקת מסקנות גבוהות יותר*[4]*.**

גישות רשתות עצביות כבשו את תחום עיבוד השפה הטבעית (NLP) בסערה. במיוחד, גרסאות של רשתות זיכרון לטווח קצר (LSTM) הניבו תוצאות מרשימות בחלק ממשימות ה-NLP הקלאסיות כגון תיוג חלקי דיבור, ניתוח תחבירי ותיוג תפקידים סמנטיים***[8]***. אפשר לשער שטבעם החוזר של המודלים הללו הוא שמאפשר את התוצאות הללו.

**פתרונות של פרויקט:**

צמצום המשפטים זאת הבעיה שניתן לבצע כסיווג בינארי של מילים במשפט. הפרויקט תתבסס על הנתונים שנאספו בשפה העברית ויישום מודלי סיווג כמו SVM, LR וכד'.

הנתונים שנאספו הם מאמרים או הסברי טפסים מאתרים שונים כמו ביטוח לאומי, עירייה, שירותי בריאות. הטקסטים מחולקים למשפטים נפרדים. המשפטים עברו סיווג בינארי לצורך הגדרת משפט לניתן לפישוט או לא. בפרויקט נעשה שימוש במערכת לבניית עץ תלות ומציאת חלקי דיבור לכל מילה במשפט***[10]***. לפי תכונות האלה אנו נדע איזה מילים במשפט פחות חשובים ויהיה ניתן להסירם. תיוג חלק מהדיבור (POS) הוא משימת NLP קלאסית, שבה עיצוב מבנה הפלט חשוב להשגת ביצועים עדכניים.

הפרויקט זה מציג מודלי למידה שונים כמו linear regression, random forest, SVM ו-RNN. רשת נוירונים מבוסס מעבר מנורמל גלובלי, המשיג תוצאות מתקדם של תיוג חלקי דיבור, ניתוח תלות ודחיסת משפטים. בפרויקט זה נשתמש ב-supervised machine learning כדי להשתמש בתכונות של משפטים וסיווגם. נצטרך לדעת איזה משפטים כן ניתנים לצמצום ואיזה לא***[1]***. וגם נראה על איזה מילים צריך להתמקד כדי לדעת מה צריך להוריד ומה צריך להשאיר במשפט. ב-unsupervised model היינו צריכים לחלק נתונים לקבוצות ספציפיות וללמוד מזה מידע רלוונטי. במקום זה נשתמש בתכונות של משפטים שנאספו ונעשה בהם עיבוד.

**מקורות מאמרים:**

1. Altun, K. F. (2013). Supervised models. In K. F. Altun, *Overcoming the Lack of Parallel Data in Sentence Compression* (pp. 1-9). Zurich: Google Inc.
2. Courtney Napoles, B. V.-B. (2011). Pitfalls of Automatic Evaluation. In B. V.-B. Courtney Napoles, *Evaluating sentence compression: Pitfalls and suggested remedies* (pp. 1-5). Human Language Technology Center of Excellence: Johns Hopkins University.
3. Daniel Andor, C. A. (2016). Dependency Parsing. In K. G. Alessandro Presta, *Globally Normalized Transition-Based Neural Networks* (pp. 6-8). New York, NY: Google Inc.
4. Emily Pitler. (2010). Finding the Best Sentence. In E. Pitler, *Methods for Sentence Compression* (pp. 15-19). University of Pennsylvania: Department of Computer and Information Science.
5. Filippova, K. (2010). Multi-sentence Compression. In K. Filippova, *Multi-Sentence Compression: Finding Shortest Paths inWord Graphs* (pp. 2-6). Zurich: Google Inc.
6. Hidetaka Kamigaito, Manabu Okumura. (2020). Sentence Compression. In M. O. Hidetaka Kamigaito, *Syntactically Look-Ahead Attention Network for Sentence Compression* (pp. 1-9). Tokyo Institute of Technology: Institute of Innovative Research.
7. Jonathan Mallinson, R. S. (2018). Neural Machine Translation Training. In R. S. Jonathan Mallinson, *Sentence Compression for Arbitrary Languages via Multilingual Pivoting* (pp. 3-6). Edinburgh: University of Edinburgh.
8. Lapata, X. Z. (2017). Reinforcement Learning for Sentence. In X. Z. Lapata, *Sentence Simplification with Deep Reinforcement Learning* (pp. 3-7). Edinburgh: University of Edinburgh.
9. Puzikov, Y. (2021). BERT-based Sentence Compression. In Y. Puzikov, *Evaluation Discrepancy Discovery: A Sentence Compression Case-study* (pp. 2-6). Darmstadt: Technical University of Darmstadt.
10. Yotam, Shichel. (2020). Feature Engineering. In S. Yotam, *Complexity-Function-Based Sentence* (pp. 15-18). Be'er Sheva: BEN-GURION UNIVERSITY OF THE NEGEV.

## **יעדים:**

## **יישום:**

## **מימוש:**

## **אפיון דרישות:**

## **ניהול סיכונים:**

## **תוכנית עבודה:**

## **תוצאות ומסקנות:**