

**Authors:**

**Israel Avihail –** [**israel.avihail@gmail.com**](mailto:Israel.avihail@gmail.com) **,** [**avihaili@post.bgu.ac.il**](mailto:avihaili@post.bgu.ac.il)

**Ariel Epshtein**

**Maor Moav**

**Academic Supervisors:**

**Dr. Marina Litvak –** [**litvak.marina@gmail.com**](mailto:litvak.marina@gmail.com) **,** [**marinal@ac.sce.ac.il**](mailto:marinal@ac.sce.ac.il)

**Dr. Hadas Chassidim –** [**hadasch@ac.sce.ac.il**](mailto:hadasch@ac.sce.ac.il)

**Hebrew Text Simplification**

***Project Report***

Contents

[General 4](#_Toc138240165)

[Overview 4](#_Toc138240166)

[Problem Statement 4](#_Toc138240167)

[Literature Review 5](#_Toc138240168)

[1 Introduction 5](#_Toc138240169)

[2 Related work 5](#_Toc138240170)

[2.1 Lexical approach 5](#_Toc138240171)

[2.2 Syntactic Approaches 6](#_Toc138240172)

[2.3 Neural Text Simplification (NTS) 6](#_Toc138240173)

[2.3.1 Translation 6](#_Toc138240174)

[2.3.2 Compression 6](#_Toc138240175)

[2.3.3 Word Substitution 6](#_Toc138240176)

[2.3.4 Paraphrase Generation 7](#_Toc138240177)

[3 Automatic Text Simplification in Hebrew: 7](#_Toc138240178)

[3.1 Unsupervised Text Summarization for text structural simplification 7](#_Toc138240179)

[3.2 Abstractive Summarization for Text Simplification 8](#_Toc138240180)

[3.3 Generative Text Simplification using Transfer Learning 8](#_Toc138240181)

[3.4 Modular Text Simplification 8](#_Toc138240182)

[4 Evaluation: 9](#_Toc138240183)

[4.1 Measures Borrowed from other Tasks 9](#_Toc138240184)

[4.2 Simplification Specific Measures 9](#_Toc138240185)

[Project uniqueness 9](#_Toc138240186)

[System requirements 9](#_Toc138240187)

[Method 10](#_Toc138240188)

[Environment and Technologies 10](#_Toc138240189)

[Development Tools 11](#_Toc138240190)

[Architecture 12](#_Toc138240191)

[Hebrew Text Simplification System Pipeline 12](#_Toc138240192)

[Browser Plugin 12](#_Toc138240193)

[Work plan and milestones 13](#_Toc138240194)

[Risk Management 13](#_Toc138240195)

[Results 15](#_Toc138240196)

[System Result Example 15](#_Toc138240197)

[System Automatic Performance Evaluation 15](#_Toc138240198)

[System Human Evaluation 16](#_Toc138240199)

[System Browser Plugin Demonstration 18](#_Toc138240200)

[Features and the populations benefitted 19](#_Toc138240201)

[Conclusion 19](#_Toc138240202)

[Future Work 20](#_Toc138240203)

[References 21](#_Toc138240204)

Figures

[Figure 1: The lexical simplification pipeline 5](file:///C:\Users\bil\Desktop\לימודים\פרוייקט%20גמר\project_report.docx#_Toc138240245)

[Figure 2: The Syntactic Simplification Pipeline 6](file:///C:\Users\bil\Desktop\לימודים\פרוייקט%20גמר\project_report.docx#_Toc138240246)

[Figure 3: The architecture of HECTOR 8](#_Toc138240247)

[Figure 4: Implementing the System: A Step-by-Step Pipeline 12](#_Toc138240248)

[Figure 5: Browser Plugin Architecture 12](#_Toc138240249)

[Figure 6: Previous Gantt 13](file:///C:\Users\bil\Desktop\לימודים\פרוייקט%20גמר\project_report.docx#_Toc138240250)

[Figure 7: Current Gantt 13](#_Toc138240251)

[Figure 8: System Result Example 15](file:///C:\Users\bil\Desktop\לימודים\פרוייקט%20גמר\project_report.docx#_Toc138240252)

[Figure 9: Data Sample 15](#_Toc138240253)

[Figure 10: Data Word Counts 15](#_Toc138240254)

[Figure 11: Plugin Example 18](#_Toc138240255)

Tables

[Table 1: Document updates 4](#_Toc138240256)

[Table 2: Development tools 11](#_Toc138240257)

[Table 3: Risk management 14](#_Toc138240258)

[Table 4: Evaluation results 16](#_Toc138240259)

****[Link to Project](https://github.com/bilbisli/hebrew_text_simplification)

# General

1. Stakeholders:

* Research teams that study Text Simplification and summarization systems in Hebrew.
* Developers who want to create a Text Simplification system in Hebrew or use Text Simplification for a downstream NLP task.
* People with low literacy skills that struggle reading Hebrew text and are interested in a system that will make the text easier to read and understand such as immigrants, children, and the cognitively challenged.

1. Date commenced: 10.07.2022.
2. Expected completion date: 06.05.2023.
3. Document updates:

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Change Description** |
| 16.12.22 | v1.0 | Document creation |
| 19.12.22 | v1.1 | Draft submission |
| 26.01.23 | v2.0 | Translation into English |
| 16.05.23 | v3.0 | Update changes for final submission |

Table 1: Document updates

# Overview

This project aims to make a significant contribution and advance the field of Automatic Text Simplification in Hebrew by conducting research.

The project's findings will serve as a foundation for future research in this area and will be used to develop tools that can help individuals with limited reading skills in Hebrew, such as immigrants, children, and those with cognitive challenges, to better understand and engage with text. Additionally, the results of the project could be used by developers to create tools and applications that can assist these populations with reading and understanding Hebrew text.

The ultimate goal is to prepare the foundations to create a system that will make reading Hebrew text easier and more accessible for these populations and a secondary goal for this project is to create a working system which could be opened to the public.

Contrary to existing projects in this field, which were focused on other languages, this project focuses on research for the Hebrew language. Compared to other languages, the research in this field in Hebrew is at its infancy.

# Problem Statement

Hebrew text could be complex due to factors such as lengthy sentences, intricate structures, obscure vocabulary, and rich morphology. These factors can make it challenging for individuals with limited reading skills to read and fully comprehend the text.

Text simplification is a process of modifying text to make it more accessible and easier to understand, such as through the use of techniques like compression, reordering, and lexical substitution. Unfortunately, there are currently no tools available to address this issue specifically for the Hebrew language.

# Literature Review

## Introduction

Automatic Text Simplification (ATS) is a Text Simplification (TS) method that modifies text to improve readability and understandability utilizing automated processes like Natural Language Processing (NLP) or Machine Learning (ML) without losing its meaning (Shardlow, 2014). There are many factors that affect ease of reading and understanding text. Factors such as length, structure, and unfamiliar words influence text complexity greatly. They may make reading, a difficult task. Besides the academic interest, Text Simplification may serve a social value and contribute to society (Stajner, 2021). In recent years the problem of automatic text simplification was studied extensively in languages such as English and Spanish (Saggion, 2017) and many approaches were researched (Al-Thanyyan & Aqil, 2021). In Hebrew, there are many methods that are yet to be explored. The Hebrew language is known for its rich morphology, which includes complex verb conjugations, noun declensions, and intricate grammatical rules (Tsarfaty, Huitink, & Katrenko, 2006). The focus of this research is to modify text to improve readability and understandability (i.e., TS) utilizing ML and ATS methods in Hebrew language.

## Related work

Relevant work for text simplification can be divided into different approaches (Shardlow, 2014).

### Lexical approach

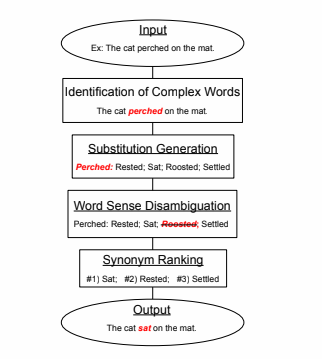
Is the task of identifying and replacing complex words with simpler substitutes. This involves no attempt to simplify the grammar of a text but instead focusses on simplifying complex aspects of vocabulary. The four steps for this approach are shown in Figure 1.

Figure 1: The lexical simplification pipeline

Firstly, the complex terms in a document must be identified. Secondly, a list of substitutions must be generated for each one. Thirdly, those substitutions should be refined to retain those which make sense in the given context. Finally, the remaining substitutions must be ranked in their order of simplicity (Shardlow, 2014).

### Syntactic Approaches

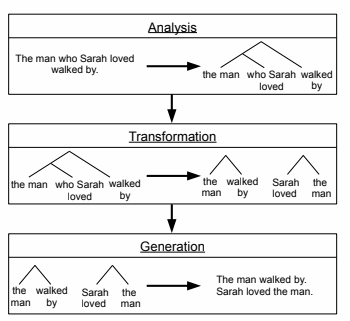
 Syntactic approaches are techniques that identify grammatical complexities in text and rewrite them into simpler structures (shown in fig. 2). Long sentences may be split into their component clauses; Sentences which use the passive form may be rewritten, and anaphora may be resolved. Poorly written text may be very difficult to engage with. Readers may struggle to follow the text, lose interest at some point in a sentence and eventually give up trying. In the case of people with cognitive impairments such as aphasia, some grammar structures may even cause a loss of meaning. Such people may not be able to distinguish between subject and object when the passive form is used (Shardlow, 2014).

Figure 2: The Syntactic Simplification Pipeline

In order to simplify the structure of text, an unsupervised automatic text summarization method may be used. An example for such method was used on Konkani Texts utilizing algorithms such as TF-IDF to score the importance of each sentence and KMeans together with the elbow method to create clusters of similar meaning sentences and choose the most relevant sentence from each cluster (D’Silva & Sharma, 2020).

### Neural Text Simplification (NTS)

### Translation

Neural Machine Translation (NMT) uses an artificial Neural Network (NN) to solve translating one language to another where the input of the network is the original language, and the output is the translation to the other language (Koehn, 2017). Contrary to previous techniques, the NTS approach, utilizes the advancements in ML to simplify text. This approach utilizes NNs where the input is the original text, and the output is a simplified version/s of the text. The learning process makes use of NMT advancements treating the original text as one language and the simplified text as the language to translate to (Qiang, 2016).

### Compression

Sentence compression is a common NLP task of shortening sentences without changing their meaning and while preserving their correct grammatical structure. The compression methodology is composed of several stages: extraction of removal candidates, representation of extracted candidates by predictive features, and candidates’ classification using a pre-trained classification model. Both supervised (Churkin, Last, Litvak, & Vanetik, 2018) and unsupervised (Vanetik, Litvak, Churkin, & Last, 2020) methods have been researched.

### Word Substitution

Automatic lexical simplification via synonym replacement was investigated using three different strategies for choosing alternative synonyms: based on word frequency, based on word length, and based on level of synonymy (Keskis¨arkk¨a, 2012) . In this approach, complex/unfamiliar words are replaced with simple/familiar words to improve readability and understandability of the text.

### Paraphrase Generation

By generating paraphrases to the original sentence that don’t lose the meaning we can create a simpler version of the sentence. Both supervised (Maddela, Alva-Manchego, & Xu, 2020) and unsupervised (Martin, Fan, Clergerie, Bordes, & Sagot, 2020) state of the art results have been reached using this method.

## Automatic Text Simplification in Hebrew

Today, most of the efforts of ATS are focused on English and similar languages, thus the performance of ATS systems is much greater in these languages. Languages which are vastly different to English, such as Hebrew, get less attention if at all. To our knowledge, this is the first attempt at ATS for Hebrew. In this article, different methods in performing ATS for Hebrew will be explored and their results will be compared, to set the stage for further improvements in future work. There are a few Hebrew NLP tools which may help with precursory tasks on the earlier stages of developing Hebrew ATS system that are worth mentioning[[1]](#footnote-2):

1. Pretrained Language Models (PLMs):
   1. Multilingual BERT (mBERT) – A pre-trained multilingual language model, originally made by taking the classic Bidirectional Encoder Representations from Transformers (BERT) model (Jacob, Ming-Wei, Kenton, & Toutanova, 2019) and fine-tuning it for multiple-languages.
   2. HeBERT - A Transformer-based model for modern Hebrew text (Chriqui & Inbal, 2021).
   3. Hebrew BERT (AlephBert) - A large pre-trained language model for Hebrew (Amit Seker, 2021).
2. Pre-processing tools:
   1. ONLP YAP - Morphological and Syntactic Analysis of Hebrew Texts (More, Seker, Basmova, & Tsarfaty, 2019). This tool can help in data preprocessing and can do the following:
   * Segmentation (division) of a sentence into morphemes (a morpheme is the smallest linguistic unit that carries meaning).
   * Labeling the different parts of speech in a sentence.
   * Dividing the words in the sentence into their lemmas (lemma - a pattern of a word that can be inflected with different grammatical inflections).
   * Defining morphological features (different forms of a word) of the words (plural/singular, male/female, past/present, first/third person).
   * Defining the dependencies between the words in the sentence.
   * A list of all the options for interpreting the sentence according to their dictionary definition (all these options constitute the field in which the search for the relevant meaning of the word in the sentence is carried out).
   1. Stanza – a collection of accurate and efficient tools for the linguistic analysis of many human languages (Qi, Zhang, Zhang, Bolton, & Manning, 2020).

### Unsupervised Text Summarization for text structural simplification

One method to simplify the structure of text is to summarize it by grouping sentences withholding the same information and choosing the most relevant sentence from each group while removing the others. A way to group similar sentences together may include encoding them into embeddings which are then used to cluster them together using a clustering algorithm. Apart from clustering, each sentence is given a score such of overall relevance/importance to the text such as TextRank (Mihalcea & Tarau, 2004) and the leading candidate from each cluster will be chosen accordingly. A challenge to this method is choosing the number of clusters to use and to deal with that are methods of choosing the right number of clusters such as the elbow and silhouette methods.

The performance of this method on Hebrew text will be dependent on how well the encoding of the sentences will be which will capture the meaning of each sentence. Upon achieving ideal embeddings, a cluster leading candidate would truly withhold the meaning of the sentences in its cluster and be the most relevant to the text out of all sentences within the cluster.

### Abstractive Summarization for Text Simplification

The goal of Abstractive Summarization (AS) is to take a long text and generate a shorter version without losing meaning. In this technique, abstractive summarization models will be utilized for text simplification in the same way that NMT was utilized by NTS. There are models such as T5 (Raffel, et al., 2020) and Bidirectional and Auto-Regressive Transformers (BART) (Lewis, et al., 2019) that were trained for Abstractive Summarization which have multilingual versions mT5 (Constant, et al., 2021) / Flan-T5 (Chung, et al., 2022) and multilingual BART (mBART) (Liu, et al., 2020) respectively. The idea is that fine-tuning such models by training on Hebrew text simplification dataset may produce positive results.

### Generative Text Simplification using Transfer Learning

Transfer learning is a method where a pre-trained transformer NN model is fine-tuned and further trained to perform a similar (yet different) task. Such method for TS was implemented when a Spanish model was trained based on the T5 model while adding additional tokens to improve quality (Sheang & Saggion, 2021). An attempt to fine tune such model further for Hebrew could be challenging due to the difference between the two languages but has a potential to perform TS in Hebrew end-to-end with one single model.

### Modular Text Simplification

In modular TS approach, different TS methods are pipelined together into one multi-stage process.

Such method was used to create a Spanish ATS system named Simplex (Horacio , et al., 2015) and more recently another system in French named HECTOR (Todirascu, et al., 2022) as shown in fig. 3.

This approach focuses upon the different tasks that improve readability and understandability both in lexical and syntax levels. A system built upon this approach may, for example, look at word simplification as one stage and sentence compression as another. Adapting each step to Hebrew may be challenging but a higher level of control will be gained enabling improvement in each step separately.

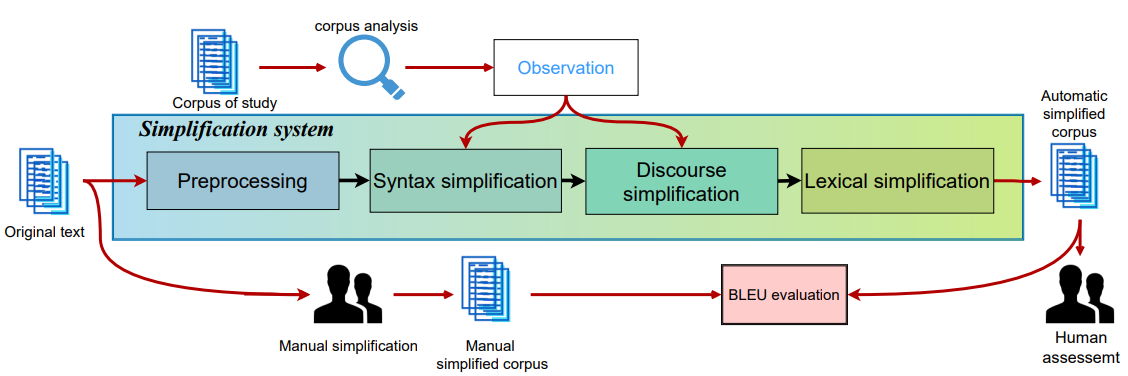


Figure 3: The architecture of HECTOR

## Evaluation

Since Automatic Text Simplification is a less researched task compared to fields such as Text Summarization and translation, evaluating such automatic systems’ performance is commonly done by using measures developed for similar tasks such as translation and summarization. This approach is not unintuitive since simplification can be looked at as translating from one domain (complex) to another (simple), also by looking at summarization, making shorter more compact text, would make it simpler to read.

In recent years there have been more research and development in the field of ATS and measures tailored to TS were developed. Nowadays these measures are used alongside measures from other domains to increase the evaluations’ robustness. It is worth mentioning that there are tools such as Easier Automatic Sentence Simplification Evaluation (EASSE) which combine such measures together and are used as a standard tool to evaluate ATS (Alva-Manchego, Martin, Scarton, & Specia, 2019). We will therefor divide the commonly used evaluation measures into two groups:

### Measures Borrowed from other Tasks

#### METEOR (Banerjee & Lavie, 2005) - Unigram matching between the machine produced translation and reference translation.​

#### BLEU (Papineni, Roukos, Ward, & Zhu, 2002) - Compares individual translated segments with a set of good quality reference translations.

#### ROUGE-1 (Lin, 2004) – Compares an automatically produced summary or translation against a reference or set of references​.

### Simplification Specific Measures

#### FKBLEU (Xu, Napoles, Pavlick, Chen, & Callison-Burch, 2016) - a combination of iBLEU (an extension of BLEU to measure diversity and adequacy of generated paraphrases) and the Flesch-Kincaid Index (FK) which measures the readability of the text.

#### SARI (Xu, Napoles, Pavlick, Chen, & Callison-Burch, 2016) - Compares the predicted simplified sentences against the reference and the source sentences.

#### SAMSA (Sulem, Abend, & Rappoport, 2018) - Compares the predicted simplified sentences against the source sentences (reference-less). Evaluates the structural aspects of the text.

# Project uniqueness

Our project focuses on the simplification and condensation of texts in the Hebrew language, in contrast to current systems which focus on simplifying texts in languages such as English or Spanish. Additionally, there is currently no system for simplifying texts for a specific target audience, and the use of existing models requires knowledge of programming languages. In contrast, our project will provide a user-friendly system for the targeted audience of the project.

# System requirements

1. Text division for paragraph level simplification.
2. Word simplification, including replacement of unfamiliar words and entities recognition.
3. Sentence level simplification, including redundant sentence removal and reordering.
4. Text recombination, paragraph, and sentence levels.

# Method

1. Data collection for future model training.
2. Splitting the text into paragraphs for paragraph level simplification and recombining them afterward.
3. Simplifying words in the text:
4. Identification of entities (NER) in the text so that they are not replaced.
5. Identification of dates, numbers and characters that are not in Hebrew so that they are not replaced.
6. Setting an index for an unknown word.
7. Identifying unfamiliar words in the text.
8. Locating candidate words includes a matching level for replacing an unfamiliar word.
9. Setting a minimum threshold for the matching level of the candidate word leading to the replacement of the unknown word.
10. Replacing unfamiliar words in the text.
11. A correction is required for words following the replacement of a word (in its vicinity).
12. Sentence level simplification:

A. Removing unnecessary parts of the text - removing phrases that appear in brackets.

B. Filtering the most relevant sentences:

1. Splitting a paragraph into sentences to filter out the less relevant sentences later.
2. Representation of a sentence in vector form to allow comparison between sentences.
3. Comparing the sentences and giving a relevance score for each sentence via TextRank algorithm.
4. Determine k sentences that will remain and create clusters accordingly using KMeans (elbow method) algorithm.
5. Choose the highest TextRank sentence and remove the others from each cluster.
6. Ordering the sentences according to importance.
7. Combining the selected sentences back to the paragraph.
8. Training a text simplification model.
9. Determining indicators for evaluating text simplification quality (measures):
   1. SARI
   2. BLEU
   3. METEOR
   4. ROUGE-1
10. Conducting experiments for various simplification methods and documenting them.
11. Summarizing the results of the research and writing an academic article.

# Environment and Technologies

Shared project environment: Google Drive[[2]](#footnote-3).

Code environment: Google Colab.

Programming language: Python.

Data: newsflash from news websites.

The reason for using Google Colab is because it is a convenient tool for performing tasks related to Machine Learning and enables run code with Google's GPU (saving run-time).

Python was chosen as the programming language, since it supports ready-made and trained models related to machine learning, and has a wide variety of convenient libraries in the field of machine learning.

Since there are no other resources in Hebrew for text simplification, newsflash was chosen as the most similar relevant data because the title of a newsflash represents a simplified/compressed version of its body thus the title was defined as the target for the model and the body as the data features. The data was comprised of 130,000 samples. The newsflash data was collected from 3 prominent Hebrew news websites: Ynet[[3]](#footnote-4), Walla[[4]](#footnote-5), Maariv[[5]](#footnote-6).

# Development Tools

|  |  |  |
| --- | --- | --- |
| **Tool** | **Description** | **Use** |
| HuggingFace transformers | Library that simplifies the use of transformers based models | 1. Using [AlephBertGimel](https://huggingface.co/imvladikon/alephbertgimmel-base-512) Hebrew language model to replace unfamiliar words. 2. Using [AlephBert](https://huggingface.co/onlplab/alephbert-base) model for named entity recognition. 3. Import tokenizer for numerical representation of the text. 4. Importing a text simplification model from another language and training it on Hebrew data. |
| SentenceTransformers | Framework for using transformers based models upon whole sentences | Using a model for numerical sentence representation (for sentence filtering and ranking) |
| wordfreq | Library that includes word occurrence frequencies in different languages | Identifying unfamiliar words according to their frequency of appearance in the language |
| networkx | Library that allows you to represent a graph and perform operations on it in a convenient way | ranking of sentences according to a specific algorithm |
| Numpy | Library that allows you to perform mathematical operations in an efficient and simple way | Building a matrix of connections between different sentences |
| Pandas | Library that includes tools for managing and presenting tabular information | Loading information and displaying it |
| Pytorch | Library that allows building and training models based on neural networks | The various models used in the project are based on it |
| BeautifulSoup | Library for pulling data out of HTML and XML files | Scraping newsflash from various sources to be used as data for training the model and evaluating performance |
| Selenium | Selenium is an open source tool which is used for automating the test cases carried out on web browsers or the web applications that are being tested using any web browser |

Table 2: Development tools

# Architecture

## Hebrew Text Simplification System Pipeline

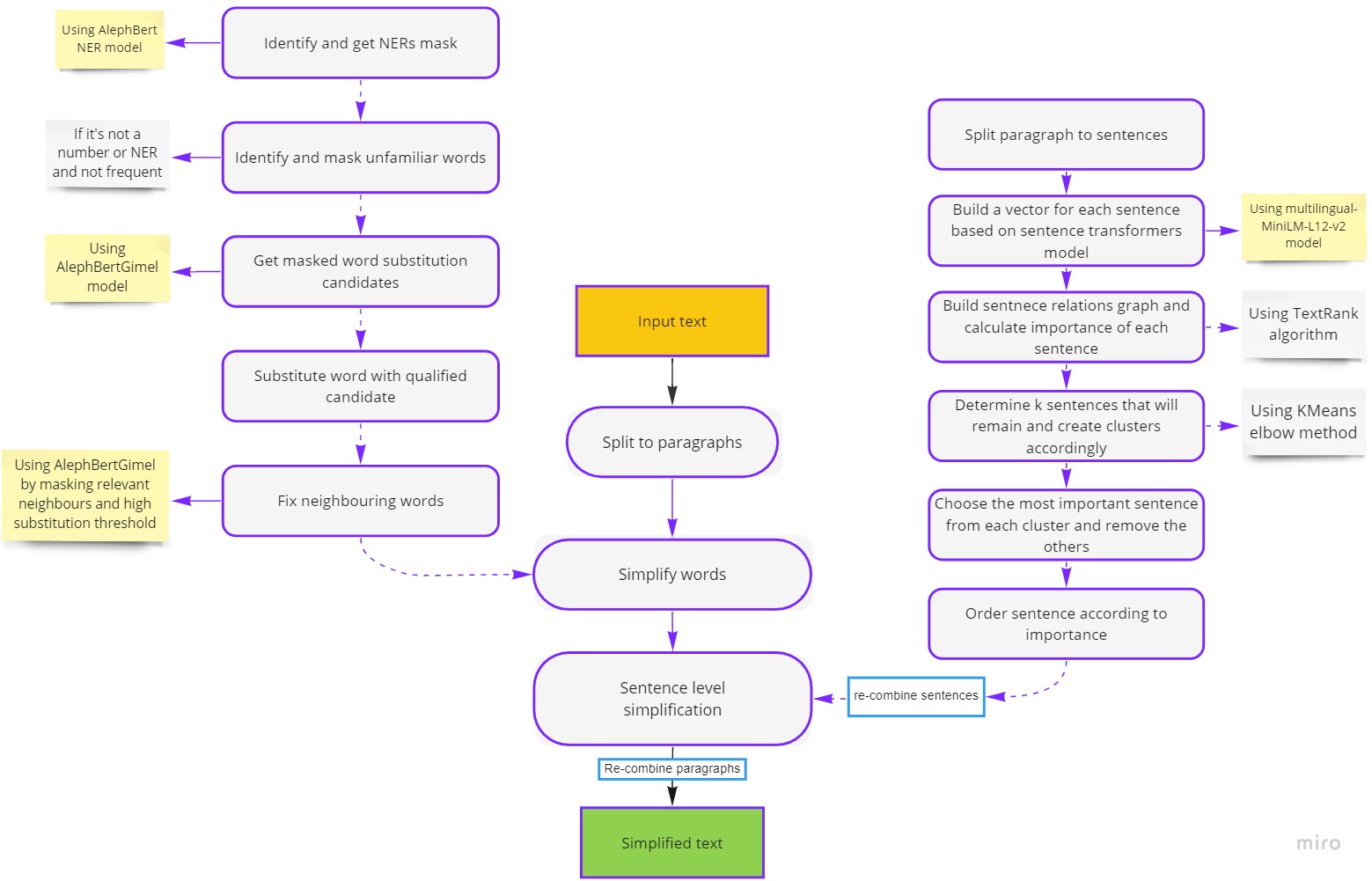


Figure 4: Implementing the System: A Step-by-Step Pipeline

## Browser Plugin[[6]](#footnote-7)

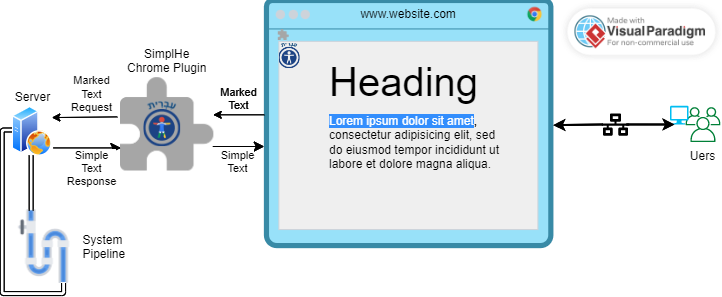


Figure 5: Browser Plugin Architecture

# Work plan and milestones

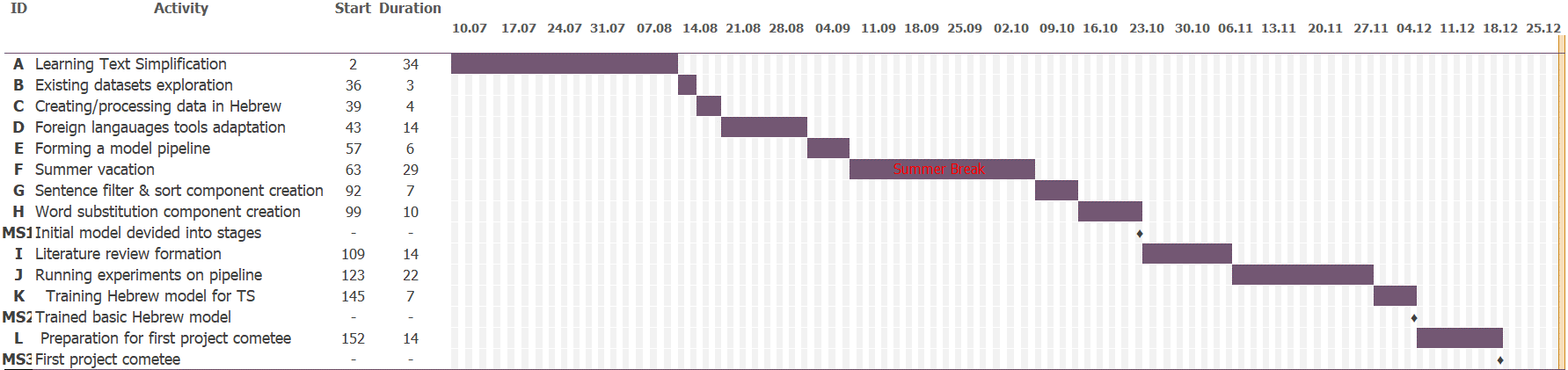
Previous Gantt:

Figure 6: Previous Gantt

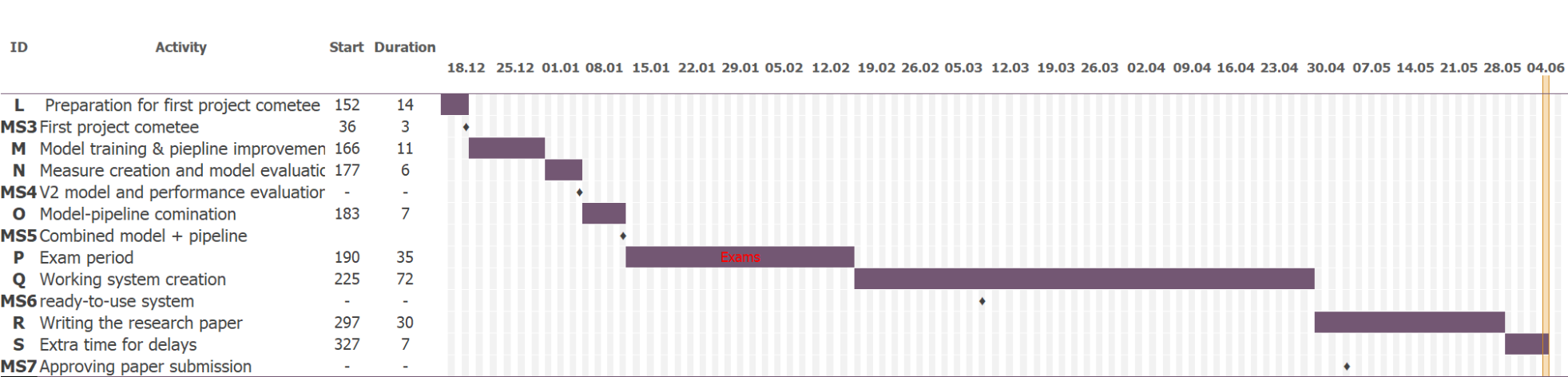
Current Gantt:

Figure 7: Current Gantt

# Risk Management

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk** | | **Impact** | **Feasibility** | **Risk level** | **Risk degree** | **Prevention/Mitigation** |
| Lack of data in the Hebrew language on the basis of which it will be possible to train a model in Hebrew | | 1. Inability to train a model in Hebrew.  2. It will not be possible to measure the performance of the model. | 5 | 4 | 20 | 1. Locating existing data.  2. Data creation.  3. Human assessment. |
| Failure to train a model from another language to the Hebrew language due to the difference between the languages | | The possibility of performing transfer learning will not materialize and it will be necessary to build a model from scratch so that the results will be less good or will require many resources to achieve good performance. | 3 | 4 | 12 | 1.Dividing the process into stages and applying the model from the other language to the stages in which the languages do work.  2. Training on a more basic but still trained model. |
| Inadequate performance of text simplification | Changing the meaning of the original sentence after the automatic simplification. | Building a system for the client will not be possible or there will be an unreliable system that does not satisfy the needs of the users (and consequently the level of use will not be low). | 4 | 4 | 16 | 1. Working at the same time on several models to increase the chance that one will succeed.  2. Dividing the process into more controllable stages for change and adjustment.  3. Performing many experiments on each model to try to improve it. |
| Unclear display of the text (gibberish). |
| Incorrect biases. |
| Technical difficulty in using existing models/tools | Downloading a ready model. | The ability to use existing models/tools will be impaired to such an extent that it will not be possible to use them. | 2 | 4 | 8 | 1. Investigating and learning how many different models work as possible.  2. Use of models that support existing tokenization models in Hebrew.  3. Consultation with the creators of the original model/tool. |
| Turning the text into a suitable input for the model (vectorization). |
| Making changes to the model we rely on. |
| Many system requirements of existing tools. |
| The performance of the model during training and turning the final model into a production ready system | Due to the large memory space it requires. | 1. Any change and attempt to train a model will take a long time and will reduce the amount of experiments that can be performed and also a lot of time will be wasted on waiting.  2. The collapse of the system/model during running/training.  3. The constraints of the model will limit how it can be used. | 5 | 4 | 20 | 1. Requirement of a suitable server from the college.  2. Use of distributed cloud services.  3. Adapting the input to the model (splitting the text for example). |
| The speed of the model operation. |
| The resources it requires. |
| The constraints of the model (e.g. accepts a certain vector length). |
| Lack of experience in the research process | Documenting the work according to accepted standards. | 1. Writing an article not of the required quality so that it is neither submitted nor accepted.  2. The comparison between the models/results will not be correct and will not give a true assessment. | 5 | 2 | 10 | 1. An inquiry into the structure and accepted standards of writing an academic paper.  2. Consultation with the team of supervisors during the writing of the article.  3. Performing a human evaluation of the results of the various models, in addition to the accepted metrics. |
| Empirical comparison between the results of the work. |
| Writing an article at an academic level. |

Table 3: Risk management

# Results

## תמונה שמכילה טקסט התיאור נוצר באופן אוטומטיSystem Result Example

Figure 8: System Result Example

## System Automatic Performance Evaluation

**Data Description**: Data includes 32,348 samples of Israeli newsflash collected from 3 prominent news agencies: Maariv, Walla and Ynet.

**Data Sample**:



Figure 9: Data Sample

**Data Details:**

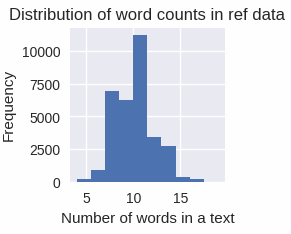
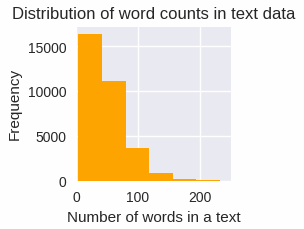


Figure 10: Data Word Counts

* Average text length (chars): 287.16
* Average word length (chars): 55.63

**System Evaluation:**

**תמונה שמכילה טקסט, צילום מסך, גופן, קבלה

התיאור נוצר באופן אוטומטי**

Table 4: Evaluation results

## System Human Evaluation

In addition to automatic evaluation, human evaluation was performed where participants were asked questions regarding the quality and satisfaction from different aspects of the system.

**Questionnaire Structure[[7]](#footnote-8):**

1. On a scale of 1 (low) to 5 (high), how effectively did the system simplify the complex words in the text?
2. Did the simplified version of the text maintain the overall meaning and essence of the original text?
3. Were there any instances where the system incorrectly simplified a word or phrase? If yes, please provide specific examples. If no, please write 'no'.
4. On a scale of 1 (low) to 5 (high), how well did the system summarize the text by including only the most important sentences?
5. Did the summary provided by the system capture the main points and key information of the original text? Please rate its effectiveness on a scale of 1 (low) to 5 (high).
6. Were there any important sentences or information that were omitted from the summary? If yes, please provide specific examples.
7. How well did the system handle the balance between simplicity and maintaining the necessary details in the text? Rate its performance on a scale of 1 (low) to 5 (high).
8. On a scale of 1 (low) to 5 (high), how satisfied are you with the overall output of the system (simplified text and summary)?
9. Would you find the simplified version of the text useful in scenarios where the original text may be difficult to understand? Please provide your response on a scale of 1 (low) to 5 (high).
10. Do you have any suggestions or feedback for improving the Hebrew text simplification system?

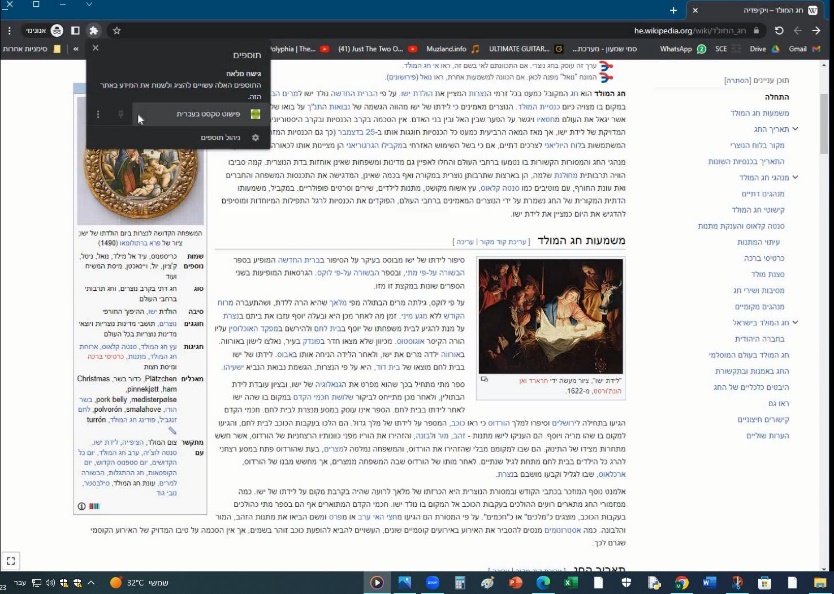
**The Participants:**

The participant were adults from the general public aged between 20-64.  
Total participants: 32.

**Forms response chart. Question title: האם הגרסה המפושטת שמרה על המשמעות הכללית ועל מהות הטקסט המקורי?
Did the simplified version of the text maintain the overall meaning and essence of the original text?. Number of responses: 32 responses.Forms response chart. Question title:  בסולם של בין 1 (נמוך) ל-5 (גבוה), כמה המערכת פישטה את המילים בטקסט?
On a scale of 1 (low) to 5 (high), how effectively did the system simplify the complex words in the text?
. Number of responses: 32 responses.Forms response chart. Question title:  בסולם של בין 1 (נמוך) ל-5 (גבוה), מה איכות תמצות הטקסט ביחס לשמירה על המשפטים החשובים ביותר בלבד?
On a scale of 1 (low) to 5 (high), how well did the system summarize the text by including only the most important sentences?
. Number of responses: 32 responses.Forms response chart. Question title: האם התמצות שהמערכת סיפקה שמרה על נקודות המידע העיקריות של הטקסט המקורי? דרג בסולם של בין 1 (נמוך) ל-5 (גבוה).
Did the summary provided by the system capture the main points and key information of the original text? Please rate its effectiveness on a scale of 1 (low) to 5 (high).. Number of responses: 32 responses.Questionnaire results:**

Forms response chart. Question title: בסולם של בין 1 (נמוך) ל-5 (גבוה), כמה אתה מרוצה עם הפלט הכללי של המערכת (תמצות ופישוט)?
On a scale of 1 (low) to 5 (high), how satisfied are you with the overall output of the system (simplified text and summary)?. Number of responses: 32 responses.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Forms response chart. Question title: האם אתה מוצא את הגרסה הפשוטה של הטקסט שימושית בתרחישים שהטקסט המקורי יכול להיות קשה להבנה? תן תשובה בסולם בין 1 (נמוך) ל-5 (גבוה).
Would you find the simplified version of the text useful in scenarios where the original text may be difficult to understand? Please provide your response on a scale of  1 (low) to 5 (high).. Number of responses: 32 responses.  
  
  
  
  
  
  
  
  
  
**Results analysis:**  
Forms response chart. Question title: באיזו רמה המערכת שמרה על האיזון בין פשטות לשמירה על הפרטים ההכרחיים בטקסט? דרג את הביצועים בין 1 (נמוך) ל-5 (גבוה) .
How well did the system handle the balance between simplicity and maintaining the necessary details in the text? Rate its performance on a scale of 1 (low) to 5 (high).. Number of responses: 32 responses.Regarding answers ranked 4 and 5, most participants found the system to be effective at simplifying words (81.3%) and summarized the text while keeping the most important sentences (84.4%). In addition, the simplified version was found to maintain the overall meaning of the original text at 81.3% of the time by the participants.  
The participants found the system to be useful 81.3% of the time but when asked if the system would be more useful if the text was more complex the percentages were higher, reaching 87.5%.

## System Browser Plugin Demonstration – [Video Demo](https://drive.google.com/file/d/1qm9folL8Myu_boD_19qqSFIcmrTtJA3B/view?usp=sharing)

A screenshot of a computer

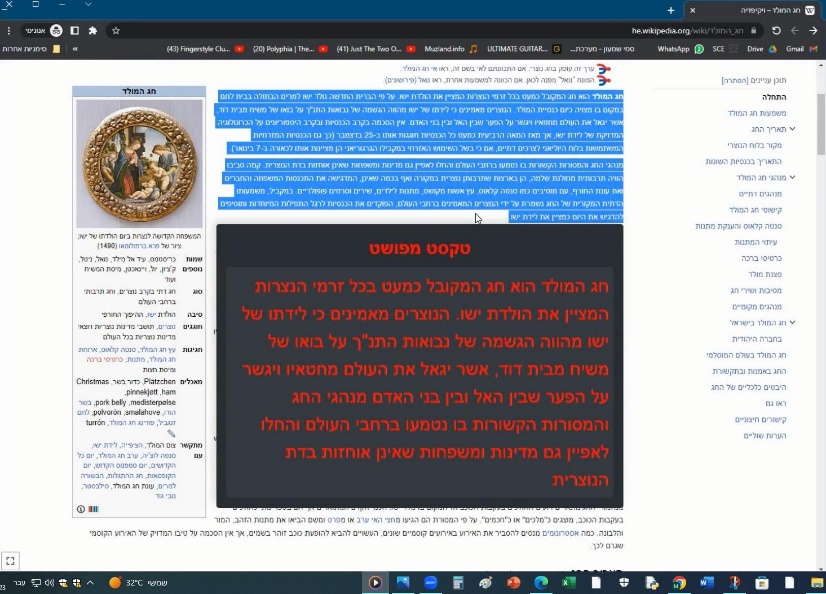
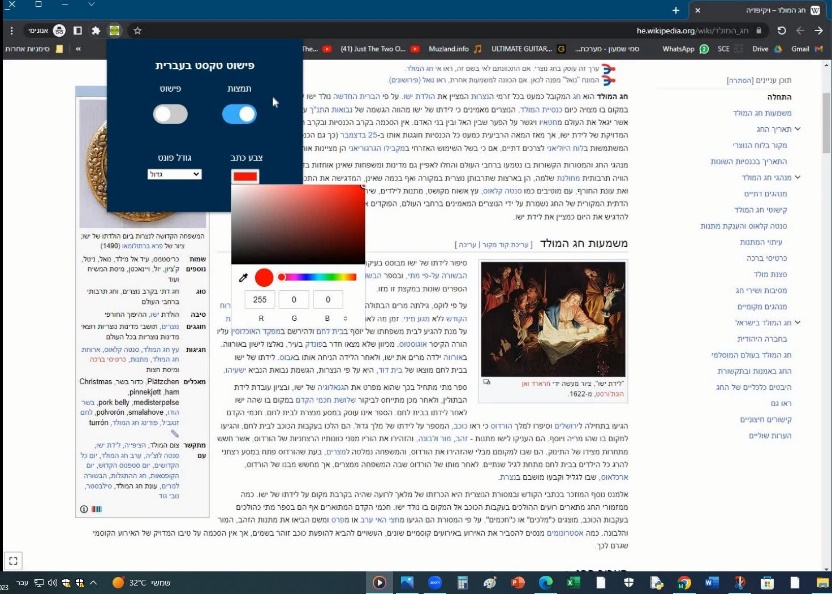
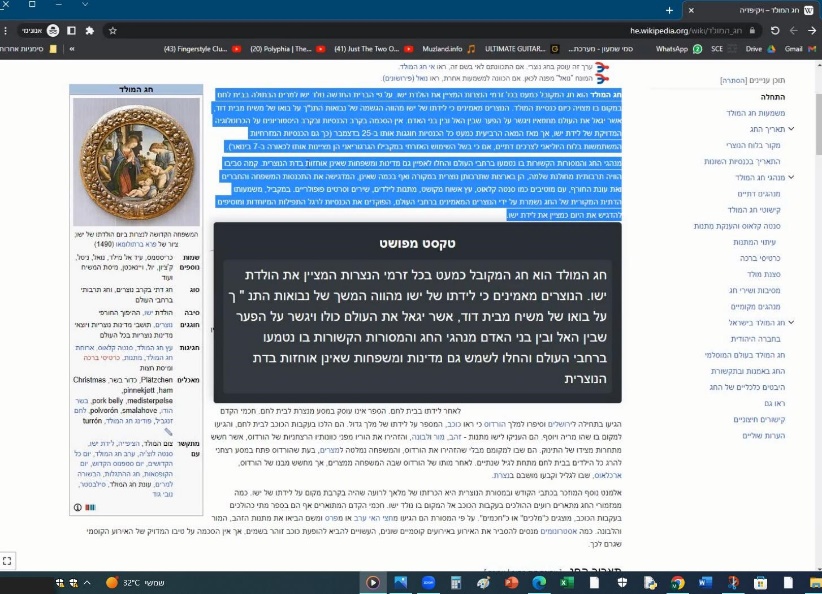
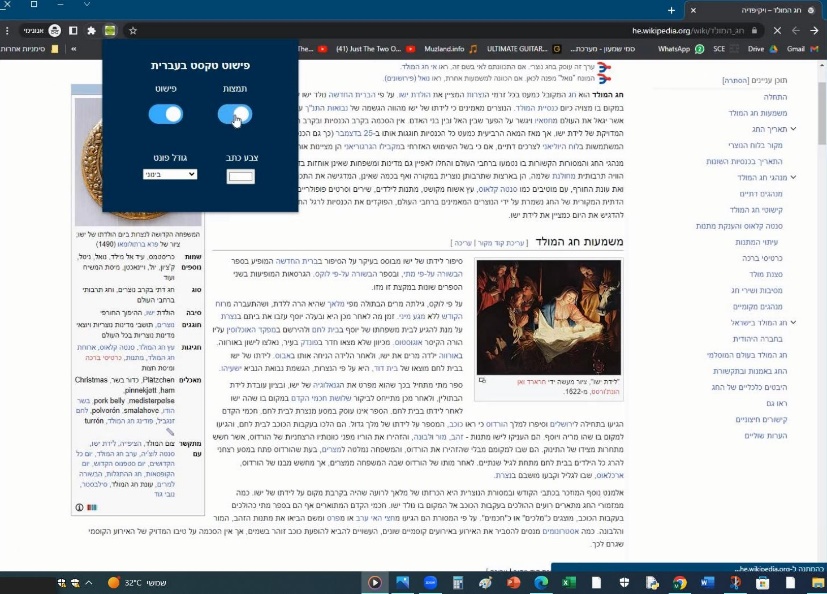
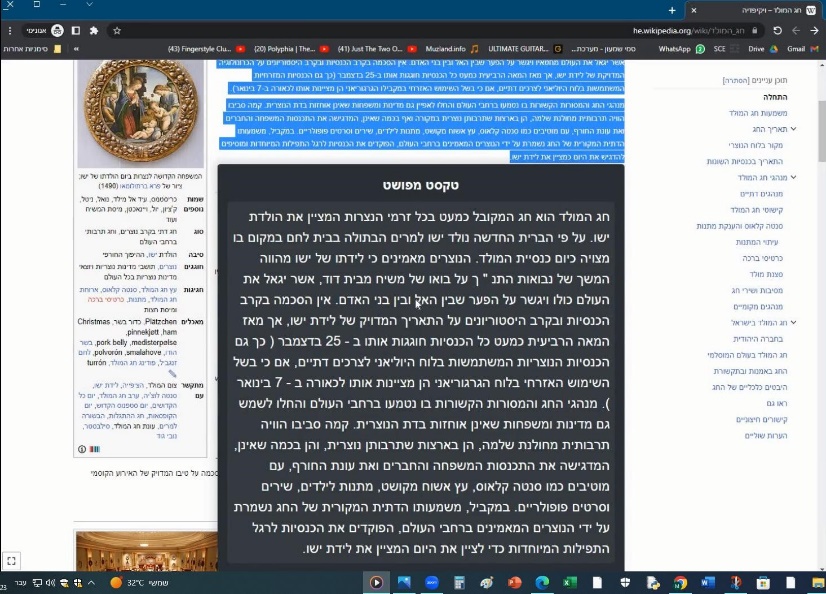
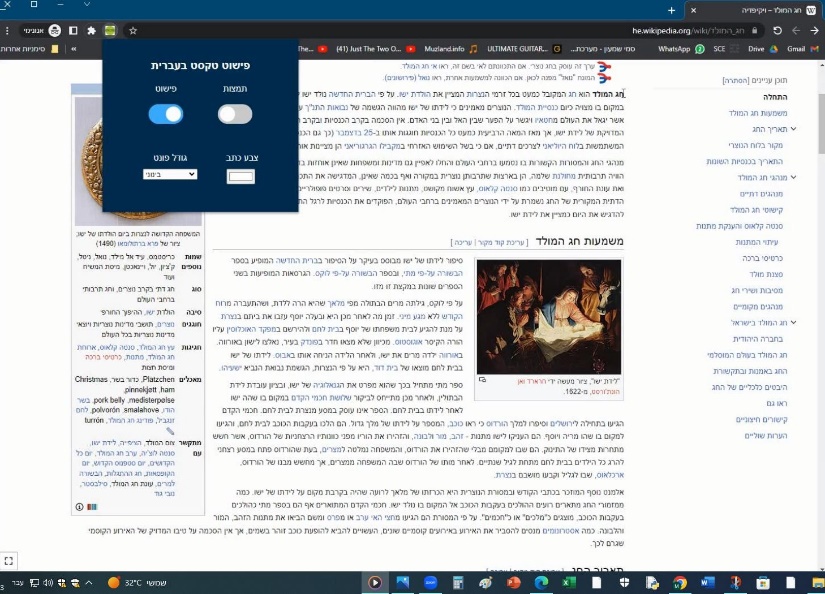
Description automatically generated

Figure 11: Plugin Example

## Features and the populations benefitted

1. Complex word simplification:
   1. Foreigners / Immigrants / Children – complex words may be unfamiliar to them.
   2. The cognitively challenged / Children – complex words may be hard to read for them.
2. Dependent word substitution:

Foreigners / Immigrants – helps learning proper grammar.

1. Sentence reordering by importance:

The cognitively challenged (ADHD) – reading important sentences first helps in case of concertation loss while reading.

1. Sentence summarization:

Foreigners / Immigrants / Children / ADHD – long sentences can be hard to follow for them.

1. Plugin accessibility:
   1. User customizable text size – people with sight impairments or reading difficulties may find the default presented text too big or too small for them to read easily.
   2. User customizable text color – The default text color might make it to see or read. This feature helps the color-blind or other people that find the color too bright/dark to adjust the system to fit their needs.

# Conclusion

In this project we present SimplHe, an Automatic Text Simplification system (ATSs) for Hebrew text, utilizing machine learning and natural language processing techniques. This system tackles the challenges of Hebrew's rich morphology and aims to enhance readability and understandability by reducing complexity factors such as structure, length, and unfamiliar words.

The project's contribution includes the development and implementation of SimplHe, as well as experiments, evaluation, and documentation of various simplification methods. The research outcomes have the potential to benefit individuals with language barriers, including children and those with linguistic challenges, by providing simplified and more accessible Hebrew text.

The methods employed in SimplHe involved data collection for future model training, paragraph-level simplification, word simplification through various identification and replacement techniques, sentence-level simplification using relevance scoring and clustering algorithms. To evaluate the performance of the system measures such as SARI, BLEU, METEOR, and ROUGE-1 were used to assess the quality of text simplification.

The Chrome browser plugin developed for making SimplHe publicly available features an interface which provided users with options to select the level of simplification they desired, focusing on unfamiliar word simplification, sentence level simplification or both. Users could activate the plugin, select text to simplify and receive a simplified version presented in a tooltip based on their selected options. In addition, features such as user customizable size and color of the simplified text presented in the tooltip were specifically designed with people of reading impairments in mind.

While SimplHe demonstrated promising results, some challenges remained, particularly in accurately replacing difficult words. Additionally, the system effectively removed unimportant sentences and summarized the text. However, further refinement is necessary to improve the accuracy of word replacement and ensure consistent performance across different text types and domains. Currently the structural simplification of the system focuses on between-sentence simplification and another improvement to the system could be to add in-sentence structural simplification.

During the work creating and improving the system, it proved to be of immense importance to keep up with the development of research in the field of Automatic Text Simplification in Hebrew as new tools and models were introduced even after we have started which improved the performance of the system substantially.

Overall, the project lays a solid foundation for future advancements in Hebrew text simplification and opens avenues for further research and improvement in order to enhance the readability and understandability of complex Hebrew text for a wide range of users.

# Future Work

1. Fine-tuning and Model Optimization: Continuously fine-tune and optimize the word simplification model to enhance its accuracy, efficiency, and ability to handle specific linguistic complexities in Hebrew. Exploring different model architectures and hyperparameter settings to improve overall performance. As better models are introduced in the future, replacing the current model used in the system has the potential to improve the systems’ performance greatly as proven during our work.
2. Adding in-sentence level structural simplification: As mentioned, the systems’ structural simplification focuses on removing irrelevant sentences and reordering the remaining by importance (between-sentence structural simplification). A step that could make text easier to read would be to simplify the structure of the sentences within each sentence itself either by training a model to do so or using other methods.
3. Substituting affected words: Improving detection of words that are affected from substituted word (to be also substituted), currently done on “neighbouring” words but can be done in a more sophisticated way such as analyzing and detecting words that depend on the substituted word (with tools such as ONLP YAP).
4. Dataset Expansion and simplification model training: Expanding the dataset used for training a machine learning model focused on Hebrew Text Simplification. Collecting more diverse and representative of Hebrew text samples from various domains. Furthermore, samples specifically focused on text simplification would improve the model's generalization, performance, relevant evaluation, and comparability.
5. Evaluation and User Feedback: Conducting extensive evaluations of the plugin's performance. Collecting user feedback and analyzing user satisfaction and usability metrics. Using this information to further refine the plugin and address any shortcomings. Adding more tools and measures such as EASSE, SAMSA and FKBLEU will also help other researchers compare the improvements of future work to the current systems’ performance.
6. Multilingual Support: Extending the plugin's capabilities to support text simplification for languages other than Hebrew. Exploring techniques to adapt the machine learning model and NLP pipeline for different languages, taking into account their specific linguistic characteristics.
7. Integration with Assistive Technologies: Investigating ways to integrate the text simplification plugin with existing assistive technologies, such as screen readers or language translation tools. This would enhance accessibility and improve the user experience for individuals with language barriers or cognitive difficulties.

# References

1. (n.d.). Retrieved from https://www.gov.il/BlobFolder/policy/molsa-volunteering-corona-27072020/he/Documents\_volunteering-corona-27072020.pdf
2. (n.d.). Retrieved from https://www.maariv.co.il/news/israel/Article-779359
3. Al-Thanyyan, S. S., & Aqil, M. A. (2021). Automated Text Simplification: A Survey. ACM Computing Surveys, 54(2), 1-36. doi:10.1145/3442695
4. Alva-Manchego, F., Martin, L., Scarton, C., & Specia, L. (2019). EASSE: Easier Automatic Sentence Simplification Evaluation. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations (pp. 49–54). Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-3009
5. Amit Seker, E. B. (2021). AlephBERT: A Hebrew Large Pre-Trained Language Model. Bar-Ilan University, Computer Science Department, Ramat-Gan, Israel.
6. Banerjee, S., & Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization (pp. 65–72). Ann Arbor, Michigan: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W05-0909
7. Chriqui, A., & Inbal, Y. (2021). HeBERT & HebEMO: a Hebrew BERT Model and a Tool for Polarity Analysis and Emotion Recognition. NFORMS Journal on Data Science, 1(1), 81-95. doi:10.1287/ijds.2022.0016
8. Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., & Fedus, W. (2022, Oct 21). Scaling Instruction-Finetuned Language Models. Google. doi:arXiv:2210.11416
9. Churkin, E., Last, M., Litvak, M., & Vanetik, N. (2018). Sentence compression as a supervised learning with a rich feature space. CICLing. Retrieved from http://www.cicling.org/2018/intranet/pre-print/papers/paper\_109.pdf
10. Constant, N., Xue, L., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., . . . Raffel, C. (2021, Mar 11). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. Google Research. doi:arXiv:2010.11934v3
11. D’Silva, J., & Sharma, D. (2020). Unsupervised Automatic Text Summarization of Konkani Texts using. International Journal of Engineering Research and Technology, 13(9), 2380-2384. Retrieved from https://dx.doi.org/10.37624/IJERT/13.9.2020.2380-2384
12. Horacio , S., Štajner, S., Bott, S., Mille, S., Rello, L., & Drndarevic, B. (2015, June). Making It Simplext: Implementation and Evaluation of a Text Simplification System for Spanish. ACM Transactions on Accessible Computing, 6(4), 1-36. doi:10.1145/2738046
13. Jacob, D., Ming-Wei, C., Kenton, L., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for. google.com. doi:10.48550/arXiv.1810.04805
14. Keskis¨arkk¨a, R. (2012). Automatic Text Simplification via Synonym Replacement. DiVA.
15. Koehn, P. (2017). Neural Machine Translation. In Statistical Machine Translation. doi:1709.07809
16. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Abdelrahman, M., Levy, O., . . . Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Facebook AI. doi:arXiv:1910.13461
17. Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. Post-Conference Workshop of ACL 2004 (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W04-1013
18. Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., . . . Zettlemoyer, L. (2020, November 1). Multilingual Denoising Pre-training for Neural Machine Translation. Facebook AI Research, 8, 726-742. doi:10.1162
19. Maddela, M., Alva-Manchego, F., & Xu, W. (2020). Controllable text simplification with explicit paraphrasing. arXiv. doi:arXiv:2010.11004
20. Martin, L., Fan, A., Clergerie, É. d., Bordes, A., & Sagot, B. (2020, May 1). MUSS: multilingual unsupervised sentence simplification by mining paraphrases. arXiv. doi:10.48550
21. Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Text. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (pp. 404–411). Barcelona, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W04-3252
22. More, A., Seker, A., Basmova, V., & Tsarfaty, R. (2019). Joint Transition-Based Models for Morpho-Syntactic Parsing: Parsing Strategies for {MRL}s and a Case Study from Modern {H}ebrew. Transactions of the Association for Computational Linguistics, 7, 33-48. doi:10.1162/tacl\_a\_00253
23. Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (pp. 311–318). Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. doi:10.3115/1073083.1073135
24. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. Association for Computational Linguistics (ACL) System Demonstrations. doi:10.48550/arXiv.2003.07082
25. Qiang, T. W. (2016). Text Simplification Using Neural Machine Translation. doi:aaai.v30i1.9933
26. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., . . . Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research, 21(140), 1-67. doi:arXiv:1910.10683
27. Saggion, H. (2017). Automatic text simplification. San Rafael, California : Morgan & Claypool.
28. Shardlow, M. (2014). A Survey of Automated Text Simplification. International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Natural Language Processing 2014.
29. Sheang, K. C., & Saggion, H. (2021). Controllable Sentence Simplification with a Unified Text-to-Text Transfer Transformer. Association for Computational Linguistics. Proceedings of the 14th International Conference on Natural Language Generation, pp. 341–352. Aberdeen, Scotland, UK: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.inlg-1.38
30. Stajner, S. (2021). Automatic Text Simplification for Social Good: Progress and Challenges. Findings of the Association for Computational Linguistics: ACL-IJCNLP, 2637-2652.
31. Sulem, E., Abend, O., & Rappoport, A. (2018). Semantic Structural Evaluation for Text Simplification. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 1, pp. 685–696. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-1063
32. Todirascu, A., Wilkens, R., Rolin, E., François, T., Bernhard, D., & Gala, N. (2022). HECTOR: A Hybrid TExt SimplifiCation TOol for Raw Texts in French. European Language Resources Association. Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 4620–4630. Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2022.lrec-1.493
33. Tsarfaty, R., Huitink, J., & Katrenko, S. (2006). The Interplay of Syntax and Morphology in Building Parsing Models for Modern Hebrew. ESSLLI (pp. 263-274). Amsterdam: University of Amsterdam.
34. Vanetik, N., Litvak, M., Churkin, E., & Last, M. (2020). An unsupervised constrained optimization approach to compressive summarization. Information Sciences, 509, 22-35. doi:10.1016
35. Xu, W., Napoles, C., Pavlick, E., Chen, Q., & Callison-Burch, C. (2016). Optimizing Statistical Machine Translation for Text Simplification. Transactions of the Association for Computational Linguistics, Volume 4. 4, pp. 401–415. Cambridge, MA: MIT Press. doi:10.1162/tacl\_a\_00107

1. Since decent performing NLP tools in Hebrew are comparatively scarce, attempts to convert tools based on other languages to work with Hebrew, will inevitably be made. [↑](#footnote-ref-2)
2. <https://drive.google.com/drive/folders/1w6Yq1v1UVCXDtdyIBlBDgopkYXFkA659?usp=sharing> [↑](#footnote-ref-3)
3. <https://www.ynet.co.il/news/category/184> [↑](#footnote-ref-4)
4. <https://news.walla.co.il/breaking> [↑](#footnote-ref-5)
5. <https://www.maariv.co.il/> [↑](#footnote-ref-6)
6. <https://github.com/bilbisli/hebrew_text_simplification> [↑](#footnote-ref-7)
7. <https://forms.gle/p4ozDftv87V9zK96A> [↑](#footnote-ref-8)