אוניברסיטת בן-גוריון בנגב

Ben-Gurion University of the Negev

הפקולטה למדעי ההנדסה

המחלקה להנדסת חשמל ומחשבים

Faculty of Engineering Science

Dept. of Electrical and Computer Engineering

Fourth Year Engineering Project

Final Report

AmosOz2Vec: Hebrew Literature meets Word2Vec

Project Number:	**p-2021-061**	
Student #1:	Name:	Kobi Kenzi
	ID:	318962008
Student #2:	Name:	Ronen Haim Portnikh
	ID:	205872708
Supervisors:	Dr. Dan Vilenchik	
Submitting Date:	25.07.2021	

Contents

1. Abstract 3

1.1 English Abstract 3

1.2 Hebrew Abstract 4

2. Preview 4

2.1 Motivation 4

2.2 Main Goal 4

2.3 Survey Of Existing Solution Methods 5

2.4 Solution Methods Used in This Project 5

3. Final Performance Specification 6

3.1 Changes From Preliminary Report 6

4. AmosOz2Vec’s Final Solution 7

5. Final Testing 9

5.1 Discussion 10

6. Project Difficulties 10

7. Conclusions and Recommendations 10

7.1 Pros and Cons 11

7.2 Meeting the Objectives 11

7.3 Conclusions 12

7.4 Recommendations 12

8. Bibliography 13

9. Appendix 14

1. Abstract

1.1 English Abstract

AmosOz2Vec: Hebrew Literature meets Word2Vec

Students Names: Kenzi Kobi, Portnikh Ronen Haim

kenzikob@post.bgu.ac.il

Advisor's name: Dr. Vilenchik, Dan

The recent developments in the field of machine learning and Natural Language Processing (NLP) have contributed greatly to areas not directly related to engineering, such as history, language, medicine, agriculture, etc. In recent years, there has been a significant increase in use of statistical models such as Word2Vec and BERT, which enables the representation of words as real-value vectors. With this method of representation, it is possible to use *mathematical* operations and reach conclusions concerning for example the meaning of words and sentences. Such capabilities contribute to research in theoretical fields, including language and literature.

The aim of the project is to *harness well-known NLP algorithms* such as Word2Vec into a simple GUI (Graphical User Interface) platform that solves various questions in the field of literature. Then, a prove to the product's correctness is required and investigated.

The methods used in this project were experiments on Word2Vec hyper-parameters, draws of characters graphs of Amos Oz's literature arts and output a summary of relationships between characters into excel files. To validate the results, the software's outputs were compared to graphs and excel files, created by the literature department, by computing the Jaccard index of the top k pairs of characters score. This is how our research of the Word2Vec hyper-parameters was progressed.

The results were interesting. The manual outputs were more similar to the outputs of Word2Vec models than the output of counter co-occurrence models. We conclude from the results that Word2Vec can be used on literature arts and not just to work with neutral texts.

Keywords: literature, machine learning, NLP, Word2Vec, GUI, Jaccard index

1.2 Hebrew Abstract

<u>מעבר ממחקר ידני לדיגיטלי בעזרת שיטות לימוד מכונה: עמוס עוז ואהרון אפלפלד</u>

התפתחות המחקר והפיתוח בתחום לימוד מכונה ועיבוד שפה טבעית תרם רבות לתחומים שאינם קשורים באופן ישיר להנדסה, כגון: היסטוריה, לשון, רפואה, חקלאות ועוד. בשנים האחרונות קיימת עליה משמעותית בשימוש במודלים סטטיסטיים, כגון מודל Word2Vec ומודל BERT, המאפשרים ייצוג מילים במרחב וקטורי בעל n ממדים. בעזרת שיטת ייצוג זאת, ניתן להגיע למסקנות *מתמטיות* הנוגעות למילים ומשפטים ובכך לתרום למחקר בתחומים עיוניים, ביניהם שפה וספרות.

מטרת הפרויקט היא לממשק אלגוריתמי NLP ידועים כמו Word2Vec לפלטפורמת GUI פשוטה (ממשק משתמש גרפי) הפותרת שאלות שונות בתחום הספרות. לאחר מכן, נדרשת ונחקרת הוכחה לנכונות המוצר.

השיטות בהן נעשה שימוש בפרויקט היו ניסויים בפרמטרים של Word2Vec, שרטוטי גרפים של דמויות מיצירות ספרותיות של עמוס עוז וכתיבת סיכום של יחסים בין דמויות לקבצי Excel. בכדי לאמת את התוצאות, פלטי התוכנה הושוו לגרפים ולקבצי אקסל, שנוצרו על ידי המחלקה לספרות, על ידי חישוב אינדקס ג'אקרד של ציון k זוגות הדמויות הגבוהים ביותר. כך התקדם המחקר שלנו על הפרמטרים של Word2Vec.

התוצאות היו מעניינות. התוצאות הידניות של החוקרים בספרות היו דומות יותר לפלטים של מודלי Word2Vec, שאומנו על אותן יצירות, מאשר תוצאות מודלים הסופרים הופעות משותפות של דמויות בספר. אנו מסיקים מהתוצאות כי ניתן להשתמש במודלים של Word2Vec במחקר ספרות ולא רק לעבודה עם טקסטים ניטרליים.

<u>מילות מפתח:</u> ספרות, לימוד מכונה, אינדקס ג'אקרד,NLP, Word2Vec

2. Preview

2.1 Motivation

The present state of the art in most literature studies are tools that use basic methods such as counting words etc. Our product will feature more complex and literature related methods. This will pave the way to study deeper questions in the field of literature research, and in a more effective way.

2.2 Main Goal

Our main goal in this project is to build a platform that uses machine learning methods and NLP methods to support research of literature analysis, and to make this platform accessible and friendly to non-experts in the field of ML and NLP.

2.3 Survey Of Existing Solution Methods

- Development of NLP tools for a specific literature genre or time period – each genre has its own properties such as writing style, structure and use of motifs. Novel2Vec built a Word2Vec model that fits the 19th century's properties. This model is not general and will not be precise for other writing types.
- Simple Statistics measurements- many tools use measurements to analyze literature such as counting words and co-occurrence of words. This tool is helpful, but too simple. In this project, we will use more complex tools.
- Topic analysis tools: there are numerous tools that automatically extract the main topics of a text. But this is only one aspect of literary research, and in many cases not the most interesting one form the scientific point of view.

2.4 Solution Methods Used in This Project

To deal with more complex literature research demands, it is necessary to build a mathematical-computable infrastructure that will support the following operations: upload corpuses to the system, train a Word2Vec model, define mathematical formulas to characterize relationships between characters and compute the result of the formula on the current text.

We use the Word2Vec NLP tool to estimate semantic relationships between words in general, and characters in particular. Later in this project, we find proofs that Word2Vec's train routine is able to find those relationships, something that has never been done before!

3. Final Performance Specification

Input – Any book formatted in a text file or a word document.

Output – Draws of characters heatmaps, n-closest words graphs and relationships summaries in excel files (see Fig. 2).

Resources – every personal computer that can run Python 3.8 code.

Connection to the outside world – GUI system used for communication between the user and the NLP engine.

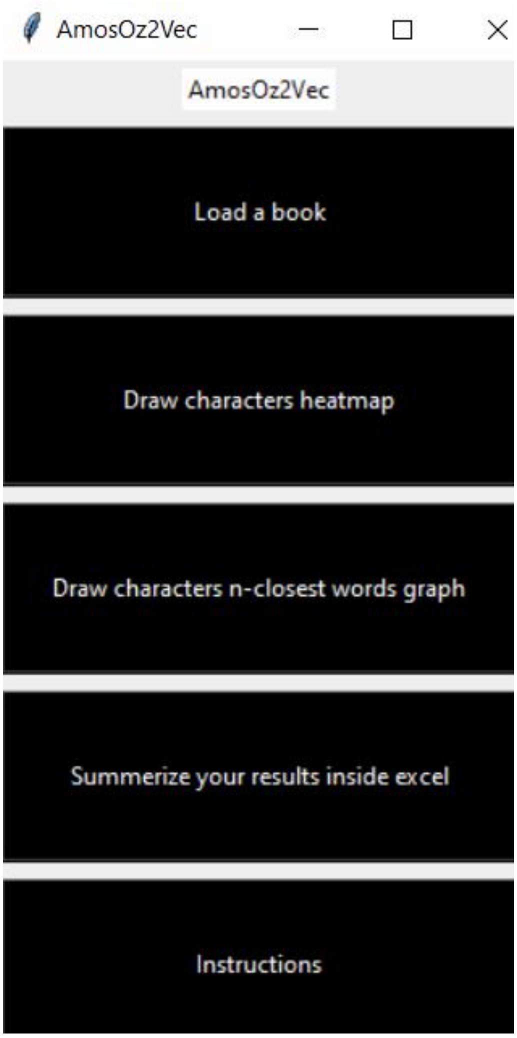
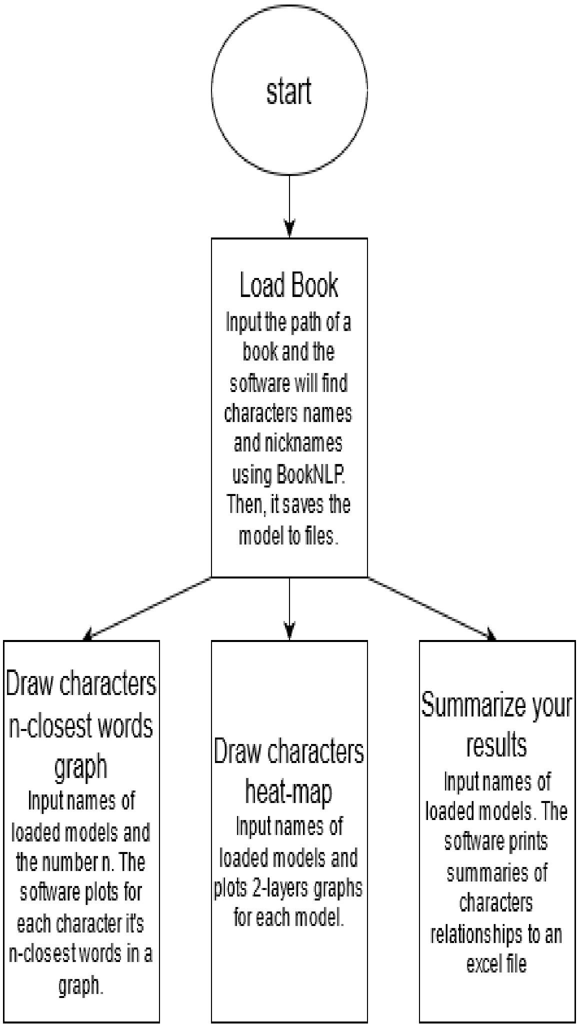

AmosOz2Vec
AmosOz2Vec
Load a book
Draw characters heatmap
Draw characters n-closest words graph
Summerize your results inside excel
Instructions

Fig. 1: The GUI system



start

Load Book
Input the path of a book and the software will find characters names and nicknames using BookNLP. Then, it saves the model to files.

Draw characters n-closest words graph
Input names of loaded models and the number n. The software plots for each character it's n-closest words in a graph.

Draw characters heat-map
Input names of loaded models and plots 2-layers graphs for each model.

Summarize your results
Input names of loaded models. The software prints summaries of characters relationships to an excel file

Fig.2: AmosOz2Vec's state machine

3.1 Changes From Preliminary Report

The specification sheet in the preliminary report was designed to deal with a specific research question about Amos Oz's and Aaron Appelfeld's novels. Throughout the work on this project, we succeed to generalize the product to deal with more book genres and writers.

4. AmosOz2Vec's Final Solution

The end-user (e.g. a researcher from some literature department) will use the GUI to upload the text corpus that is of interest. Then, the product will use BookNLP, an NLP tool that finds characters names and nicknames. AmosOz2Vec will fit a word-embedding model for the chosen corpus. Then, our neural network will start to optimize the word vectors for each word in the corpus to measure the semantic meaning for each word. Meanwhile, the product will count co-occurrences of the chosen characters by using a window size. If the distance between 2 characters in the text is at most the window size, we count this as co-occurrence between the characters. After the neural network returned the optimized word vectors, AmosOz2Vec will provide the following functionalities:

- *Draw characters heat-map* – This graph is an undirected graph, where the nodes are characters names. The edges contain 2 layers: The number's layer and the color's layer. *The number's layer* represents the cosine similarity of the word vectors, calculated by Word2Vec. The weights are numbered from 0 to 1, where 1 is the strongest relationship. This similarity measures the relationships between characters, either semantic or co-occurrence relationship. Each edge is in the graph if the cosine similarity between the characters is more than average. *The color's layer* determines if the relationship of 2 characters is semantic or co-occurrence. The color red represents a semantic meaning and blue the co-occurrence meaning. It does it by comparing the Word2Vec rank of this pair to the count co-occurrence rank. *Those 2 layers summarize all we need to know about the characters relationships!* (Fig. 3).
- *Draw characters n-closest words graph* – This undirected graph gives an intuition about the behavior of characters in the novel. We define the n-closest words to each character as the words with the highest cosine similarity with the character. We connect those words in the graph to the character node with an undirected edge. By that, we examine the character's behavior and look for words that connect between characters. (Fig. 4).
- *Summarize your results* - This option prints to excel files the Word2Vec score of characters pairs, the co-occurrence score, and the rank differential between those models. We define irregular pairs as pairs that their Word2Vec score is high, but the co-occurrence score is low and the opposite. Then we mark those pairs. In addition, we add statistics to the excel file such as average similarity between characters, std etc. We also estimate the noise of the model by computing cosine similarities between random pairs of words and print statistics about the noise.

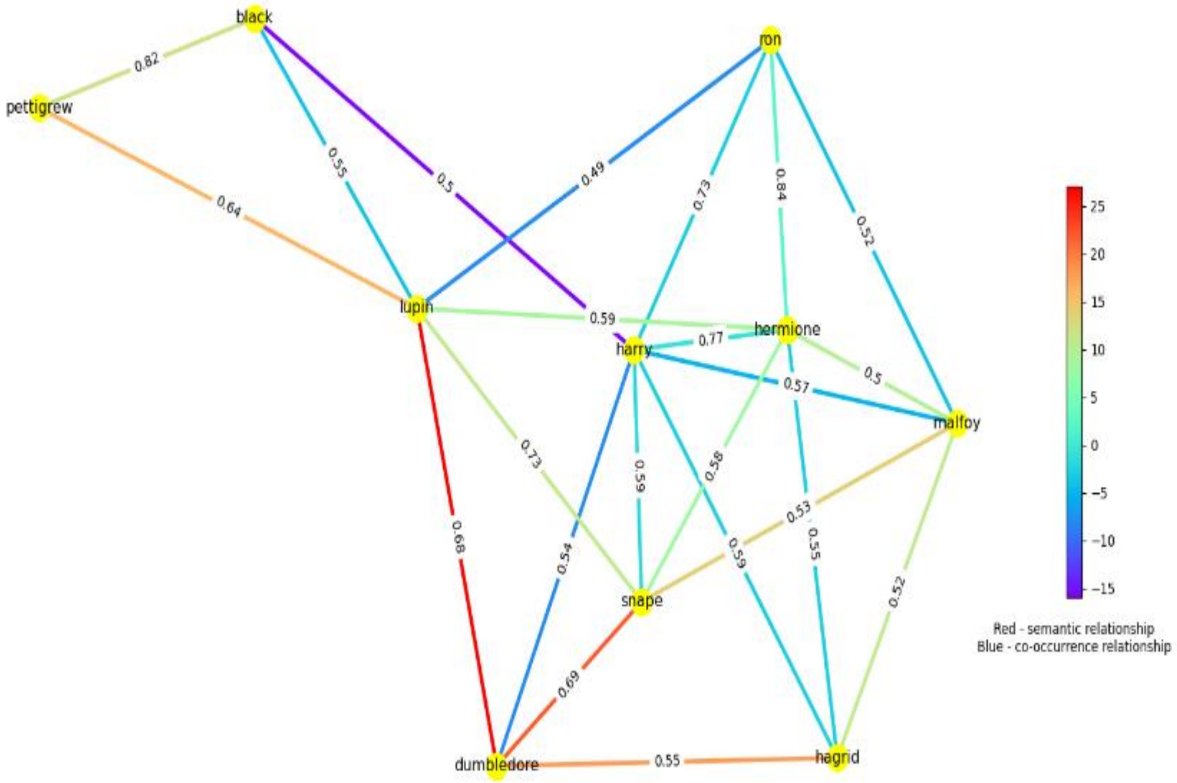


Fig. 3: characters heat-map of “Harry Potter and the prisoner from Azkaban”

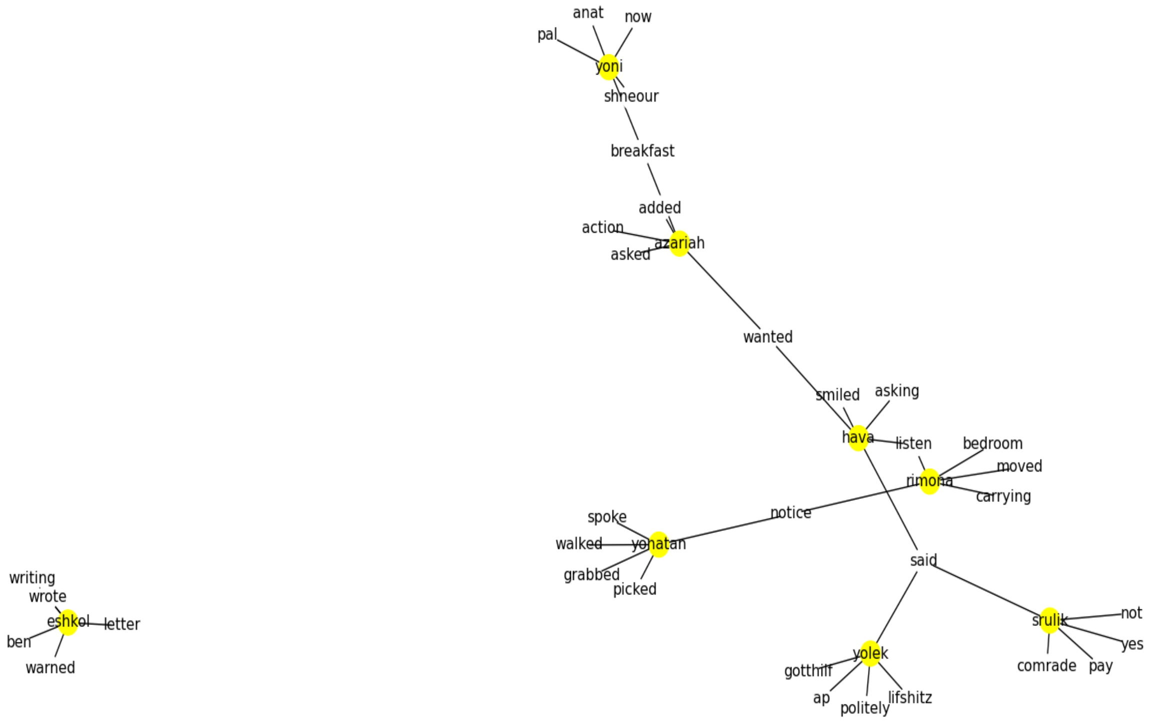


Fig. 4: 5-closest words graph of “A Perfect Peace”

5. Final Testing

To validate our results, we designed a measurable experiment called *Intersection K*. With the help and collaboration of the literature experts, we tried to measure similarity between different models to a manually-designed model of a literature novel. The novel that was tested was “A Perfect Peace” by Amos Oz. The literature experts read the novel and analyzed it. The output of this analysis method is an ordered list of characters pairs by the number of dialogs and “on stage” appearances in the novel. The automated models that analyzed the novel were 30 Word2Vec models, used to eliminate the noise of Word2Vec, and co-occurrence models. Using those models, we ranked the pairs of characters. The research question is simple: *Are Word2Vec models more similar to manual models than co-occurrence models?* We measure similarity of models using the *Jaccard index* of the K most ranked pairs between two model. Jaccard index of 1 means identical sets and 0 means foreign sets.

$$Jaccard\ Index(Set1, Set2) = \frac{size(Set1\ \cap Set2)}{size(Set1\ \cup Set2)}$$

Each list contains pairs of characters. The hyper-parameter of Word2Vec that was changed during the experiment was the window size. It is used to notice the number of words (left and right from a current word) that are relevant for the optimization of a center word vector. In AmosOz2Vec, the window size of Word2Vec and the co-occurrence models is the same.

Here are the results of Intersection K experiment:

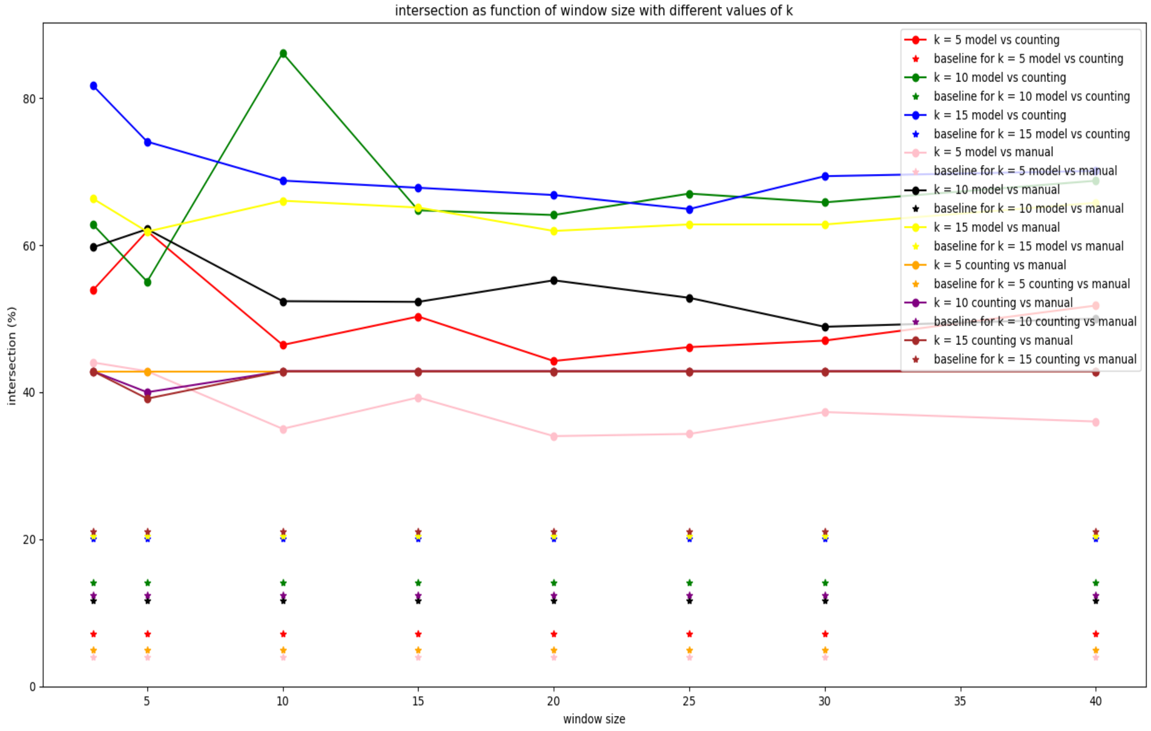


Fig. 5: Intersection K experiment on “A Perfect Peace”

The parameter K is the size of the top K ranked pairs of characters in each model. The X-axis is the window size of Word2Vec and co-occurrence. The Y-axis is the Jaccard index in percentage of two models. The dots at the bottom are noise estimations of the Jaccard index of 2 models. It is measured to validate that the results are not noisy.

5.1 Discussion

For $K \geq 10$, we see that the manual model is more similar to the Word2Vec models than to the co-occurrence models. Another interesting result is the similarity of Word2Vec models and the co-occurrence models. We conclude from those results that Word2Vec can be meaningful to measure relationships between characters, semantic and co-occurrence. This conclusion is important duo to the common use of Word2Vec in neutral texts such as Wikipedia dataset. Now we see that Word2Vec can be used to understand sub-text and literature complex relationships.

6. Project Difficulties

- *How to fit the hyper-parameters:* To use Word2Vec models, it is necessary to understand the role of hyper-parameters used to train the model, such as window size, vector size for the embedded vectors etc. *Solution:* During the year, we ran Word2Vec models on Amos Oz's novels and met the literature experts to get intuition about the hyper-parameters in literary manner. Those models helped us determine "rules of thumb" for the hyper-parameters.
- *Text length affects accuracy:* Word2Vec is commonly trained on giant datasets such as the Google news dataset. Literature novel's length in words is small relative to those datasets. The effect on the Word2Vec model is noise added to the samples. *Solution*: We estimate the noise using random sampling of words in the text. It is done by choosing 100 pairs of random words and calculate the cosine similarity between the pairs. The average cosine similarity and the std are calculated and every pair of characters with similarity under average + std is ignored.
- *BookNLP might guess names and nicknames wrong:* Nicknames of characters can be identified as characters and a separation of meaning will occur. This will be seen in the characters heat-map as two different nodes of the same character.

Note about marks numbered 3-5: We believe that those problems can be solved as the automated literature research will progress. Then, AmosOz2Vec or any other software can be "on top of things" and further applications will be developed.

7. Conclusions and Recommendations

The work on AmosOz2Vec was successful! With hard work and innovative ideas, we developed an NLP product that is simple to use and helped progress the computational research in literature and provided *proof of concept.* We plan to submit a *journal paper* to summarize our results and present our product later this year.

This project was challenging, exciting and above all, not in our comfort zone. We did not know much about machine learning in general and NLP in particular before we started the project. To gather enough experience to start this project, we participated in an online course on Stanford University, which discussed about main topics in machine learning and NLP. Then, we started to work on AmosOz2Vec. The concept was designed in steps, using weekly meetings with the literature experts. We believe this project is special for many reasons, such as the innovation in the computational research in literature domain and the co-operation between engineers and literature researchers.

7.1 Pros and Cons

The key *advantages* of this project are:

- *Savings in time, resources, and money:* AmosOz2Vec's run time, for novels with length of about 300 pages, is about 3 minutes. For the average researcher, manual reading and first analysis can take weeks. Furthermore, the resources required to run the software are common in most modern Personal Computers. This allows us to approach more potential customers. In addition, the savings in time and resources is translated to saving money directly. This can be significant with multiple use of the software for different novels and save in cumulative years of work.
- *No internet connection required:* This is an advantage for the customer and the developer. *The customer* earns full accessibility, independent of a server's state and does not waste network bandwidth. This can be achieved due to the simplicity of the software. In addition, *the developer* does not need to maintain a server to handle some functionalities. This makes the maintenance of the software cheaper.
- *Simple to use:* Most of the latest products in computational research in literature are complicated to use and require software development skills. For example: *Mallet*, a commonly used package written in Java is used to handle literature computational problems. AmosOz2Vec is designed with a simple GUI that requires no experiment in software. Furthermore, not much of computer skills are required at all.

The key *disadvantages* of this project are:

- *First-person narrator books are out of bounds:* When the story-teller is also a character in the novel, he may refer himself with "I". By that BookNLP, which finds the characters names and nicknames, will not recognize the story-teller as a character. *Solution:* We tried to find tools that can handle this problem but could not found useful tools. Therefore, we limited AmosOz2Vec to handle third-person narrator.
- *Pronoun's identification:* As of today, AmosOZ2Vec do not use an external software in order to guess the character referred with he, she, or it. *Solution:* We tried to use some tools that identify pronouns, but those were not good enough.

7.2 Meeting the Objectives

As a reminder, our *main goal* in this project was to build a platform that uses machine learning methods and NLP methods to support research of literature analysis. Another important goal was to make this platform simple to use. *We achieved those goals!* AmosOz2Vec is a simple to use platform that promotes literature research with special functionalities described earlier in this report. Most importantly, we proved in this project the potential of literature projects that use NLP tools such as Word2Vec and provided proof of concept.

The *budget* for this project was 40,500 NIS that went for salaries and computational resources, such as GPUs and TPUs. Due to the simplicity of the software and the short runtime, the use of GPUs and TPUs was not necessary! Therefore, the budget estimation was enough, and no further money was needed.

We completed most of the *Task Management* missions (the full Task management is in appendix). The only mission that was not completed was "Getting to know BERT". BERT is a large transfer learning model used to measure semantic meanings of complete sentences. We could use BERT for more accurate semantic meanings of sentences and words. The key disadvantage of BERT is the demand of large texts and corpuses for transfer learning. A single novel or a small group of novels are not enough to execute accurate transfer learning. Therefore, we did not use BERT.

7.3 Conclusions

- *NLP tools can be used to analyze "sub-text":* From the *Intersection K* experiment, we concluded that Word2Vec can be used not just for word embeddings of large datasets. It can be used to analyze sub-text as well, a very important conclusion to progress the computational research in literature or any other humanities subject.
- *AmosOz2Vec can distinguish between semantic and co-occurrence relationships:* The heat-map functionality was built to distinguish between semantic and co-occurrence relationships. There is no such automated project that does that! This creates opportunities to further projects, as we will discuss later.
- *Engineers and humanities researchers can co-operate:* We met the literature department every week and discussed the progress and the results of this project. The engineering team was responsible for the software and functionalities and the literature team was responsible for the validation of the software's results. Both teams contributed with concept thinking.

7.4 Recommendations

AmosOz2Vec can be expanded to some interesting future projects and changes:

- *Feature engineering on novels using AmosOz2Vec:* from the heat-map functionality, we believe that features for neural networks can be extracted. Those features can be used to proof statistical connections between structure of characters and ideas in literature, such as genre. Examples for features that can be extracted: number of red triangles in the heat-map graph.
- *Improved n-closest words graph:* As a reminder, the n-closest words graph attaches edges from characters nodes to the n-largest cosine similarity words in the novel. Using this graph, we can learn about the character's behavior. This can be improved by observing the n-closest adjectives or verbs for each character. With that in mind, we can examine the character's behavior more precisely.

8. Bibliography

[1] O. Avraham and Y. Goldberg, *Word Embeddings in Hebrew: Initial Results,* 2015.

[2] J. Devlin, . M.-W. Chang, K. Lee and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for,* 2018.

[3] R. Heuser, *Word Vectors in the Eighteenth Century,* 2016.

[4] I. Marienberg-Milikowsky, *Beyond digitization? Digital humanities and the case of Hebrew literature,* Ben Gurion University, 2019.

[5] Y. Neuman, H. Hames and Y. Cohen, *An information-based procedure for measuring semantic change in historical data,* ScinceDirect, 2017.

[6] A. Piper, *Data and literary study,* The university of Chicago, 2018.

[7] B. Schmidt, *Word Embeddings for the digital humanities,* 2015.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, *Distributed Representations of Words and Phrases,* 2013.

[9] Y. Neuman, Y. Cohen, N. Israeli and B. Tamir, *A proposed methodology for studying the historical trajectory of words' meaning through Tsallis entropy,* ScinceDirect, 2018.

[10] *Novel2Vec: Characterising* ,D. Greene S.´. Grayson, M. Mulvany, K. Wade, G. Meaney and
.*19th Century Fictionvia Word Embeddings*

[11] *Dynamic Word Embeddings for* ,X. Hui and Y. Zijun, S. Yifan, D. Weicong, R. Nikhil
Evolving Semantic Discovery, 2018.

[12] *The Transformation of Gender in English-Language* ,S. Lee and T. Underwood, D. Bamman
. 2018*Fiction,*

9. Appendix

Task Management:

Start Date	**End Date**	**Start Time**	**End Time**	**Portnikh Ronen Haim**	**Kenzi Kobi**	**Task**	**Status**
1/09/2020	21/09/2020	09:00:00	21:00:00	yes	yes	Stanford NLU internet course: part 1	Done
22/09/2020	30/09/2020	09:00:00	21:00:00	yes	yes	Practical project - Word2vec implementation	Done
1/10/2020	13/10/2020	09:00:00	21:00:00	yes	yes	Stanford NLU internet course: part 2	Done
14/10/2020	27/10/2020	09:00:00	21:00:00	yes	yes	Harry potter Hyper parameter tests on word2vec	Done
20/10/2020	31/10/2020	09:00:00	21:00:00	yes	yes	Work on PDR	Done
2/11/2020	01/12/2020	09:00:00	21:00:00	yes	yes	Check new visualization and analysis methods	Done
12/11/2020	21/11/2020	09:00:00	21:00:00	yes	yes	Work on preliminary report	Done
2/12/2020	08/12/2020	09:00:00	21:00:00	yes	yes	Getting to know BERT	Not Done
9/12/2020	09/12/2020	09:00:00	21:00:00	yes	yes	Primary Amos Oz corpus arrives	Done
10/12/2020	30/12/2020	09:00:00	21:00:00	yes	yes	Analyze the primary corpus	Done
31/12/2020	09/01/2021	09:00:00	21:00:00	yes	yes	Work on technology ventures	Done
11/01/2021	11/11/2020	09:00:00	21:00:00	yes	yes	Amos Oz and Aharon Appelfeld corpus arrives	Done
15/01/2021	28/02/2021	09:00:00	21:00:00	yes	yes	Exam period	Done
26/01/2021	08/02/2021	09:00:00	21:00:00	yes	yes	Analyze the full corpus	Done
20/02/2021	06/03/2021	09:00:00	21:00:00	yes	yes	Work on progress report	Done
8/03/2021	06/05/2021	09:00:00	21:00:00	yes	yes	Work on general GUI system	Done
8/05/2021	22/05/2021	09:00:00	21:00:00	yes	yes	Work on poster and presentation	Done

המלצת ציון לדו"ח מסכם

אם יש צורך, לכל סטודנט/ית בנפרד

מספר הפרויקט: P-2021-061

שם הפרויקט: AmosOz2Vec: Hebrew Literature meets Word2Vec

שם המנחה מהמחלקה: Dr. Dan Vilenchik

שם הסטודנט/ית: קובי קנזי ת.ז.: 318962008

שם הסטודנט/ית: רונן חיים פורטניך ת.ז.: 205872708

%		חלש 55-64	בינוני 65-74	טוב 75-84	ט"מ 85-94	מצוין 95-100
20	הצגת גישת הפתרון, והתכנון ההנדסי					
20	הצגת התוצאות וניתוח השגיאות					
20	הסקת מסקנות					
10	גילוי יוזמה וחריצות					
20	פתרון בעיות, מקוריות ותרומה אישית (מעבר למילוי ההנחיות)					
10	מידה בלו"ז ורמת הביצוע המעשי					

אם יש כוונה לפרסם/יפורסם מאמר, שם כתב העת ומועד משוער להגשה:

ציין אם יש כוונה לשקול המלצה כפרויקט מצטיין:

הערות נוספות: