On Bi-gram Graph Attributes

Thomas Konstantinovsky[1], Matan Mizrachi[1]

[1] Department of Computer Science, Holon Institute of Technology, Golomb 52, Israel

Correspondence: Thomas Konstantinovsky, Department of Computer Science, Holon Institute of Technology, Golomb 52, Israel.

Abstract

We propose a new approach to text semantic analysis and general corpus analysis using, as termed in this article, a "bi-gram graph" representation of a corpus. The different attributes derived from graph theory are measured and analyzed as unique insights or against other corpus graphs, attributes such as the graph chromatic number and the graph coloring, graph density and graph K-core. We observe a vast domain of tools and algorithms that can be developed on top of the graph representation; creating such a graph proves to be computationally cheap, and much of the heavy lifting is achieved via basic graph calculations. Furthermore, we showcase the different use-cases for the bi-gram graphs and how scalable it proves to be when dealing with large datasets.

Keywords: graph theory; natural language processing; machine learning; computer science

1. Introduction

Corpus representation is central to natural language processing. However, the default approach of representing a corpus usually revolves around the bag of words representation or other vectorization methods. This paper highlights the benefits and use cases of a representation based on inner word relationships derived from the bi-grams of a given corpus.

Previous works that used similar methods revolve around solving a specific problem using a graph representation (Masséet al, 2008), suggested a new way for grounding the meanings of certain words in sensorimotor categories (Reimer and Hahn, 1988), proposed a model of knowledge-based text condensation that resembles today's well-known knowledge-graphs.

(Giannakopoulos and Karkaletsis, 2009), showed promising effectiveness and suggests that n-gram graphs and operators can constitute an effective and updatable text representation method that strengthens some of the algorithms proposed in this article.

However, many graph attributes are left untouched in natural language processing due to the different representations available.

Previous works also highlighted the benefits of N-gram flexibility with the well-structured representation of directed graphs and their applications towards classification problems, (Violos et al, 2018).

Representing a corpus as a weighted directed graph of its bi-grams as defined in section 3 not only is computationally cheaper and more scalable in comparison to standard representation methods but also allows a vast domain of analysis as each bi-gram graph represents not only the inner connection between the words and the classic conditional probabilistic model but also a broader semantic meaning that is unique to each corpus. Moreover, as we present in the article, there are novel tools and algorithms that can be applied to those graphs both individually and against different pairs deriving new insights yet to be explored.

The additional benefits gained from the algorithms based on such a representation are similar to the benefits derived via transfer learning when comparing to deep learning methodologies (Thrun and Pratt,2012), i.e., after creating a graph that represents the semantics of a particular corpus, the same graph can be used to apply the graph-based algorithms proposed in this article on other corpora in a way using one semantic field to derive information regarding another.

2. The Bi-gram Graph Representation

Throughout the entire paper, $BG_x = (V, E, W)$ is a weighted directed, finite graph, based on corpus

$$x = \{t_1, t_2, \ldots, t_n : t_i = w_1 w_2 \ldots w_m \text{ is the } i\text{th text of the corpus}\} \tag{1}$$

with *V* as the set of words that appears in at least one of *x*'s texts and *E*, which is defined by:

$$E = \{(w_p, w_q) : (w_p, w_q) \subseteq \overset{..}{\underset{i=1}{\cup}} t_i\} \tag{2}$$

The weights set *W* is defined by:

$$W = \{\omega(e) = \text{the number of times that } e \text{ appears as a bi-gram in } x, \forall e \in E\} \tag{3}$$

Note that punctuation marks have been removed from *x* prior to generating its Bi-gram Graph. For example,

let $x = \{$"I love eating pizza", "I usually enjoy having a pizza when it rains outside", "The art of making a pizza"$\}$, so its bi-gram graph is the following:
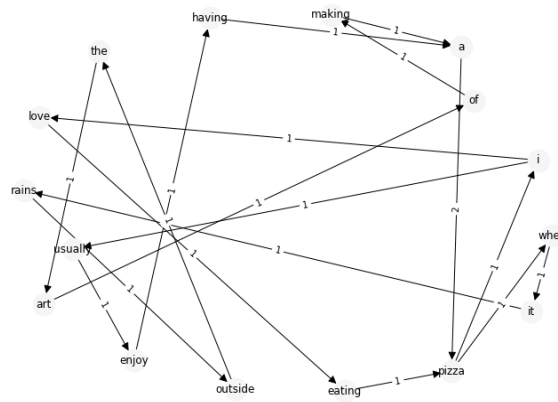

Figure 1. The Bi-gram graph of the corpus *x*

3. Bi-gram Graph Attributes

3.1 Chromatic Number

The main highlight of this article is the different insights gained by deriving the chromatic number and coloring of the graph; Formally, Let $G = (V, E)$ be any graph. Then,

$$C(G) = \{c_v : \forall v, u \in V, (v, u) \in E \Rightarrow c_v \neq c_u\} \tag{4}$$

is the chromatic embedding of G, and

$$\chi(G) = max\{C(G)\} \tag{5}$$

is the chromatic number of G.

Mainly, most of the new approaches we present are based on the coloring of the graph and the different relations the graph coloring has within itself and in comparison to other bi-gram graphs. A graph coloring is achieved by labeling each node with a color such that no two vertices sharing the same edge have the same color. After coloring a bi-gram graph, each node or each unique word from a given corpus is assigned with a color tag; this tag is heavily dependent on the corpus context as the different edges are formed based on the variability of the sentences in the corpora. Having a unique context-based color tag for each word by itself provides an interesting meta-feature for different machine learning models, potentially elevating the performance of weak models by introducing context-based features; Moreover, in this article, we propose the "Chromatic Vectorizer," which takes advantage of those tags in order to embed sequences of words based on a bi-gram graph coloring. As stated in the introduction, the suggested approach of using bi-gram graphs as corpora representation allows for somewhat of a "transfer learning" methodology. After creating a bi-gram graph for certain corpora and calculating the graph coloring, the same coloring can be used to embed or analyze texts from different corpora by tagging the intersecting words with

the chromatic labels of the "pre-trained" bi-gram graph in a way it can be thought of as "context projection" we project the context derived from one corpus onto another.

3.2 K-Core

A K-core of graph BG_x is defined as the maximal connected subgraph of BG_x in which all vertices have a degree of at least k. K-cores are mainly mentioned with respect to k-degenerate graphs, (Lick and White, 1970), and have been shown in various cases, such as in (Altaf-Ul-Amine et al,2003), that they can be useful when dealing with prediction-related objectives.

We propose a different perspective on the K-core of a graph by addressing problems such as text summarization and dimensionality reduction; the hypothesis we present is that the K-core of BG_x where the maximal k is selected, represents a summarized version of the text where unlike classical text summarization that focuses on maintaining human readability, the K-core of a bi-gram graph with the maximal k represent

the main context of a corpus eliminating any redundant edges that may represent noise or irrelevant data. From a machine learning perspective, the K-core of a bi-gram graph can be used as an imputer for redundant words by removing all words that have been left out of the K-core; the size of the corpus goes down by a magnitude that is dependent on its original size.

3.3 Chromatic Similarity Coefficient

We propose a new similarity coefficient between two bi-gram graphs based on intersecting nodes and their chromatic numbers.

Definition 1.

Let $BG_1 = (V_1, E_1, W_1), BG_2 = (V_2, E_2, W_2)$ be the bi-gram graphs of corpora 1 and 2 respectively, and

$$I = ||V_1 \cap V_2|| \tag{6}$$

$$IC = ||(V_1 \cap V_2) \wedge (C(V_1) \cap C(V_2)|| \tag{7}$$

where I is defined as the number of nodes in the intersection between the two graphs, i.e., the number of similar unique words in both corpora and IC is defined as the number of intersecting words that also share the same chromatic tag, i.e., the "color" of the node. Then,

$$\Psi(BG_1, BG_2) = \begin{cases} \dfrac{I}{IC}, & I > 0 \\ 0, & I = 0 \end{cases} \tag{8}$$

is the chromatic similarity coefficient between BG_1 and BG_2.

3.4 Random Chromatic Walker

The random chromatic walker is yet another approach we propose based on the coloring of the bi-gram graph, and its purpose is to provide a fast and versatile approach to text generation. The algorithm we propose generates an array that randomly selects a number J such that $1 \ J \ \chi(BGx)$ and using a beta distribution to compensate for the non-uniform distribution of color labels. The resulting sentence is the concatenation of the paths between two randomly selected words according to the color labels generated in the first array of random color labels.

Algorithm 1: Random Chromatic Walker

Result: A randomly generated sentence based on a bi-gram graph SNT_LEN = the length of the randomly generated color label array;

CHROM_VEC = a vector of length equal to SNT_LEN filled with randomly selected chromatic number as described above, that are smaller or equal to the chromatic number of a given bi-gram graph;

LAST_WORD = randomly select a word that is tagged with the color at CHROM_VEC[0]; RESULTING_SENTENCE = "";

for *number in CHROM_VEC[1:]* **do**

    CUR_WORD = generate a random word with the color equal to number;PATH = find_path(LAST_WORD,CUR_WORD); RESULTING_SENTENCE += PATH[:-1];

    LAST_WORD = CUR_WORD;

end

---

    return RESULTING_SENTENCE

This approach allows for many different and unique results depending on the protocol by which the path between two words was calculated. In our work, we tested with fair results the following protocols:

1. Maximum weight path.
2. Minimum weight path.
3. Maximum density path.
4. Minimum density path.

Where density is defined as the sum of all out and in degrees of a given path, or more formally:

$$\text{Density}(\text{Path}) = \sum_{\text{firt word in path}}^{\text{last word in path}} \delta^-_{\text{word}} + \delta^+_{\text{word}} \tag{9}$$

3.5 Chromatic Sentence Embedding

The last graph coloring-based algorithm we propose in this article is a new approach to text embedding. When working with machine learning models, the standard approach is to transform a given text into some vector space representation in order to be able to feed the text into a model; common text vectorization algorithms range from observational methods such as the bag of words ,(Harris,1954), and TF-IDF vectorization ,(Aizawa,2003), to more advanced and state of the art deep learning model such as GloVe,(Pennington,2014), and word2vec,(Mikolov ,2013), all these approaches differ in both in their benefits and significance with respect to a given task as well as in their computational complexity. The "chromatic vectorization" proposed by us performs an embedding computational similar to the bag of words approach, but instead of representing each text as a representation of words frequency, the chromatic vectorizer embeds the context in a given corpus. The algorithm for chromatic vectorization is dependent on the coloring of a given bi-graph graph, meaning that, similarly to earlier statements, a transfer learning approach can be applied, and a graph of a corpus with the desired context can be used to chromatic vectorizer a given text. The algorithm is defined as:

Algorithm 2: Chromatic Embedding

Result: A text embedding based on a graph coloring initialization;

chromatic_vector = array of zeros with length equal to the number of words in text;

chromatic_dictionary = a dictionary extracted from a bi-gram graph coloring that maps each node to a color label;

for *index,word in **enumerate(text_words)*** **do if**

    word not in chromatic_dictionary **then**

        chromatic_vector[index] =   1;_

else

chromatic_vector[index] = chromatic_dictionary[word];

end end

return chromatic_vector

An important note to be made as for the chromatic vectorizer concerns the hypothesis of its efficiency; the chromatic embedding space does not meet the requirements of an injective function, meaning two different sentences can have the same chromatic embedding, a question arises when analyzing such property as for the contextual meaning of two sentences that are mapped to the same embedding yet are different, and we yet found such a case after preforming the embedding algorithm on over 20 datasets ranging from 4,000 samples up to 150,000 samples from different contextual themes. We hypothesize that sentences mapped to the same vector represent the same underlying context with respect to a certain theme observed in a bi-gram graph.

4. Results

Throughout the research, the different methods and metrics proposed in this article were applied to over 20 corpora of different and similar contexts. The results shown below showcase the different outcomes derived using the methods in this paper.

4.1 Graph Coloring POS and NER Interrelations
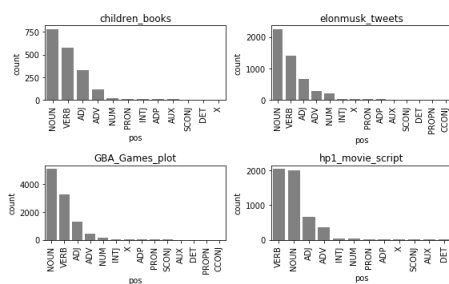

Figure 2. Distribution of POS Per Color Label   Figure 3.Distribution of NER Per Color Label
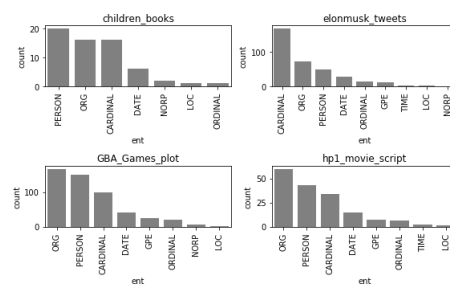
There is a keynote to be made as to the distribution of different attributes associated with color tags; the first is regarding the non-uniform distribution of part-of-speech tags associated with each tag see Figure 2; we observed that in all the corpora we tested, the same distribution has appeared where the first color tag is mostly made out of nouns the second from verbs and so on as can be seen in Figure 2, the hypothesis we propose

goes to a new approach of part of speech tagging using the insight derived, i.e., replacing the current computationally expensive part-of-speech algorithms and assigning a part of speech tag according to a given color tag with some deviation as the part of speech distribution in each color tag contains other tags as well.

As for named entities, different corpora have shown that although the position of the entities and their quintets change, see Figure 3, the main contents of the leading   entities is retained thought various corpora, the slight changes observed may indicated true contextual difference even though the same amount of entities appear inside the corpora.

4.2 K-core Dimensionality Reduction

The K-core dimensionality reduction and noise reduction approach we proposed in section 3.2 demonstrated below shows outstanding results when used in classification-based machine learning pipelines. In the example shown below, a corpus of spam and ham SMSes1 was taken as a classic example for text classification in machine learning natural language processing; the corpus was cleared of stopwords and converted into a bag of

words representation. Using a naive Random Forest model with default parameters, we received an accuracy score of 97%, the confusion matrix of that same process is demonstrated in Figure 4. The bi-gram graph of the SMS corpus contains 10730 nodes and 41389 as we hypothesized earlier; many of these nodes are redundant and potentially can be imputed. The K-core of maximal K in this graph results in a new graph of 165 nodes and 3736 edges, using only the words left in the K-core graph, the same naive Random Forest model with default parameters achieved an accuracy of 91%, and its confusion matrix can be seen in Figure 5. After reducing the original quantity of unique words to only 10%, we observe only a minor reduction in accuracy.
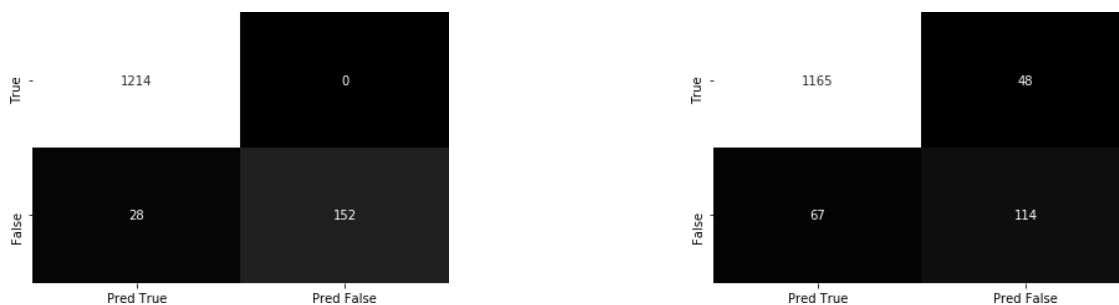

Figure 4. Before Imputing Words Out- side of The K-Core Figure 5. After Imputing Words Out- side of The K-Core

In comparison to traditional dimensionality reduction algorithms such as principal component analysis or singular value decomposition, the result of the K-core extraction is directly interpretable and are a result of imputation, unlike the formerly mentioned algorithms, which project the original data onto new vector spaces that are much less trivial to interpret with respect to the original data. Also, note that there is one degree of freedom when using K-core as a method to reduce the dimensionality, although selecting the maximum K of a graph allows extraction of the most significant context and connections between the words in the original corpus, lower K values can be selected retaining more information and more dimensions.

4.3 Chromatic Similarity Coefficient

When analyzing the results of the Chromatic Similarity Coefficient computation between our corpora,
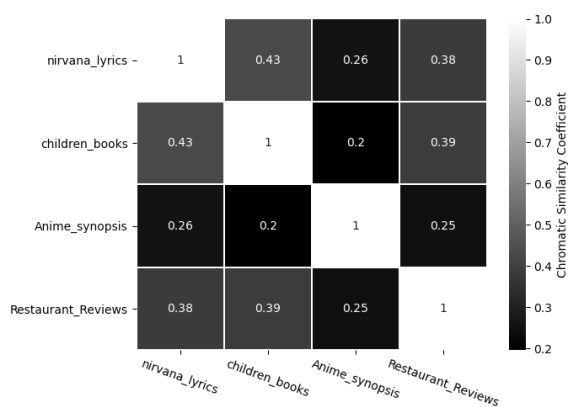

Figure 6. Ψ Similarity Matrix Between Different Corpora

we wanted to estimate the quality of the Chromatic Similarity Coefficient as a similarity coefficient see Figure 6, in comparison with other similarity measures, such as Cosine Similarity and Jaccard Index (Jaccard,1912). The

Cosine Similarity measure is defined by:

$$\cos(A, B) = \frac{A \cdot B^T}{||A|| \cdot ||B||} \quad :A, B \in \mathrm{R}^n \tag{10}$$

where, in our case, *A, B* are the TF-IDF embeddings of corpora *x, y* respectively, and the *Jaccard Index* measure, is defined by:

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{11}$$

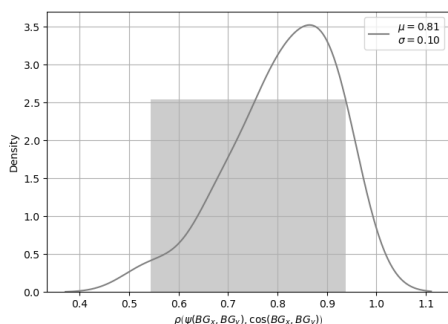where *A, B* are *x, y* corpora's sets of unique words, respectively.

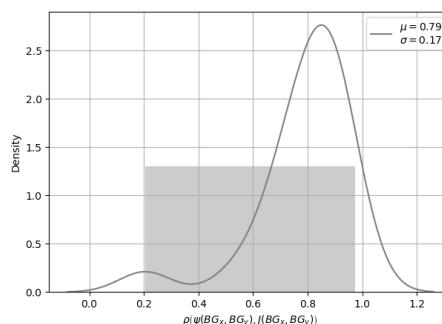Figure 7. Cosine Similarity in Relation to Ψ

Figure 8. Jaccard Index in Relation to Ψ

Our comparison is done by examining the distribution of the Pearson's correlation Coefficient, that is denoted by ρ, between each pair of our corpora Chromatic Similarity Coefficient to their Cosine Similarity and Jaccard Index. The results shown in Figures 7 and 8 demonstrate how Ψ is similar on average to both Jaccard Index and Cosine Similarity based on TF-IDF embedding, even though the formula of Ψ is closer to the Jaccard Index formula, there is a stronger correlation on average between the TF-IDF embedded Cosine Similarity. Overall, even though Ψ is different to the tested similarity measures, it brings an additional contextual consideration that is represent via the coloring of the bi-gram graph i.e. the IC component in Ψ's formula.

4.4 Chromatic Random Walker

When testing the chromatic random walker, different path protocols lead to various results depending on the context of the corpus on which the bi-gram graph was built; the below examples show three different sentences generated using three different protocols on three different corpora.

Corpus: Nirvana Lyrics; **Protocol**: Heaviest Path

'calm inside seems think happy think one wanna fun think happy knowright sure

killed toe jam booger stomach acid'

Corpus: Movie Synopses; **Protocol**: Heaviest Density

'rescue young doctor because finding mysterious deaths young police man gods

mean nothing'

> **Corpus**: Terminator 3 Script; **Protocol**: Lightest Path
>
> 'like i see you dying never we cry to reasons die birds fly happily when we take off'

Note that stop words were removed from all corpora during preprocessing, yet some protocols achieve remarkably good readability and logic results. One of the most significant advantages of the chromatic random walker is its versatile nature in that it can adapt different scoring protocols for paths and different path traversals.

5. Conclusions

Recent work in the field of natural language processing has taken significant steps towards uncovering a vast domain of deep learning application when processing and modeling text data. Nevertheless, there are "simpler" mathematical structures that allow us to extract insights and resources that resemble those achieved via deep learning, especially when dealing with large corpora, mathematical structures such as this paper.

Throughout this paper, we have constructed and proposed different approaches to text generation, corpus analysis, dimensionality reduction, and similarity comparison.

The methods presented rely on a scalable graph data structure that does not grow linearly or exponentially with the data; instead, the inner connections between words capture the context of the corpus more accurately by updating the edge weights between words.

Such scalability allows for a stable working methodology when approaching large text data sets or large corpora.

Although the results generated via the algorithms in this paper cannot compare to the state of the art deep learning architectures in the quality of their performance,

the methods can provide a fast and reliable basis for large datasets where deep learning might not be an option in terms of memory and computation time.

A key remark to be made is the real hidden potential of the "bi-gram graph" representation, as mentioned in section 4.2; after constructing such a graph for a given corpus or corpora, the graph holds vast amount of information towards those corpora, especially from a contextual perspective, those "pre-trained" graphs can be manipulated and applied on entirely different corpora achieving real scalability and mobility similarly to what we today see in the world of deep learning, where each user can use open-source pre-trained architectures to "finetune" and fit the model towards his needs.

6. Future Work

The main purpose of this paper was to introduce different use-cases and metrics derived from unexplored bi-gram graph attributes, mainly the chromatic number and the coloring of such graphs. Ongoing and future work includes establishing robust rules and methods concerning the coloring of a bi-gram graph and the different patterns emerging when scaling it to many datasets. We have seen that the coloring of bi- gram graphs allows us to derive a new similarity coefficient and various text generation and dimensionality reduction techniques. However, graphs associated with datasets of a similar topic have a similar structure. We intend to describe how such datasets interrelate, yielding a solution that can unfold a computationally cheap solution for large-scale corpus analysis.

References

A. Massé, et al. (2008). How Is Meaning Grounded in Dictionary Definitions? *ArXiv* abs/0806.3710 (2008). https://doi.org/10.3115/1627328.1627331

Akiko, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing and Management, 39*(1), 45-65. https://doi.org/10.1016/S0306-4573(02)00021-3

Don, R. L., & Arthur, T. W. (1970). k-Degenerate Graphs. *Canadian Journal of Mathe- matics, 22*(5), 1082-1096. https://doi.org/10.4153/CJM-1970-125-1

George, G., & Vangelis, K. (2009). *N-gram graphs: Representing documents and document sets in summary system evaluation.*

Jeffrey, P., Richard, S., & Christopher, M. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 14,* 1532-1543.

John, V. et al. (2018). Text Classification Using the N-Gram Graph Representation Model Over High Frequency Data Streams. *Frontiers in Applied Mathematics and Statistics, 4,* 41.

https://doi.org/10.3389/fams.2018.00041

Kaggle @darkrubiks.   *Nirvana Lyrics*. 2021. Retrieved from
    https://www.kaggle.com/darkrubiks/nirvana-lyrics

Kaggle @Thomas Konstantin. *Highly Rated Children Books And Stories*. 2021. Retrieved from
    https://www.kaggle.com/thomaskonstantin/highly-rated-children-books-and-stories

Kaggle @Thomas Konstantin. *Top 10000 Anime Movies ,OVA's and Tv-Shows*. 2021. Retrieved from
    https://www.kaggle.com/thomaskonstantin/top- 10000- anime- movies- ovas- and- tvshows

Kaggle @Vicky. *Restaurant Customer Reviews*. 2019. Retrieved from
    https://www.kaggle.com/vigneshwarsofficial/reviews

Md. Altaf-Ul-Amine et al. (2003). Prediction of Protein Functions Based on K-Cores of Protein- Protein
    Interaction Networks and Amino Acid Sequences. *Genome Informatics, 14,* 498-499.

Paul, J. (1912). THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1. *Phytologist, 11*(2), 37-50.
    https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

Reimer, U., & Hahn, U. (1988). Text condensation as knowledge base abstraction. *Proceedings. The Fourth
    Conference on Artificial Intelligence Applications*, 338-344. https://doi.org/10.1109/CAIA.1988.196128

Thrun, S., & Pratt, L. (2012). *Learning to Learn*. Springer US, 2012. ɪsbn: 9781461555292. Retrieved from
    https://books.google.co.il/books?id=X%5C_jpBwAAQBAJ

Tomas, M. et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781
    [cs.CL].

Zellig, S. H. (1954). Distributional Structure. *WORD, 10*(2-3), 146-162.
    https://doi.org/10.1080/00437956.1954.11659520