

# Knowledge Base Management Simplified with EZ Vector-KB

Michael Green, Dr. Adrienne Raglin, Dr. Douglas Summers-Stay and Dr. Andre Harrison

Army Research Laboratory, Adelphi MD 20783, USA  
{michael.a.green85.ctr, adrienne.j.raglin.civ,  
douglas.a.summers-stay.civ, andre.v.harrison2.civ}@mail.mil

**Abstract.** Vector-KB is a framework that leverages information from knowledge bases like ConceptNet to perform reasoning tasks. It includes many functions to take advantage of a knowledge base's collection of relationships between thousands of words, but the framework's complexity can make usage difficult. We introduce the EZ Vector-KB interface, which combines the Vector-KB functions into an intuitive single-window search application, making the system easier to utilize and modify.

**Keywords:** Knowledge base, Vector-KB, ConceptNet, Interface.

## 1 Introduction

With the ever-increasing sources of data from Internet of Things devices comes the challenges of utilizing that data appropriately. Moreover, is the need to make sense of that data. Knowledge bases are used to store data and form an important step in analyzing and understanding that data. Consider a first responder situation, where workers preemptively assess an environment to safely navigate it. They are provided with images and videos of the area alongside software that identifies objects. Introducing a knowledge base that can identify relationships between the detected objects and assist in making decisions on which aspects of the environment are dangerous. Vector-KB is a framework that utilizes semantic knowledge bases, such as ConceptNet, to perform deductive reasoning [2]. When provided with a combination of words and a relationship, Vector-KB returns a list of paths from one word to all relevant answers by exploiting the knowledge base's relationships between words. The framework defines many functions to run searches on the knowledge base, add new information, and adjust the parameters that influence search results, but it can be rather intricate to operate. The current iteration of Vector-KB is command-line based, requiring prior knowledge on the framework's inner workings before its MATLAB function calls can be made. To determine if an easier approach for managing Vector-KB could be created that saves time without sacrificing functionality, the EZ Vector-KB interface was developed. EZ Vector-KB is a MATLAB graphical user interface designed to support all the functions present in the Vector-KB system as a menu-based search application. All functions are

operated through text fields, buttons, and drop-down menus. This design approach allows for more intuitive and automated knowledge base management.

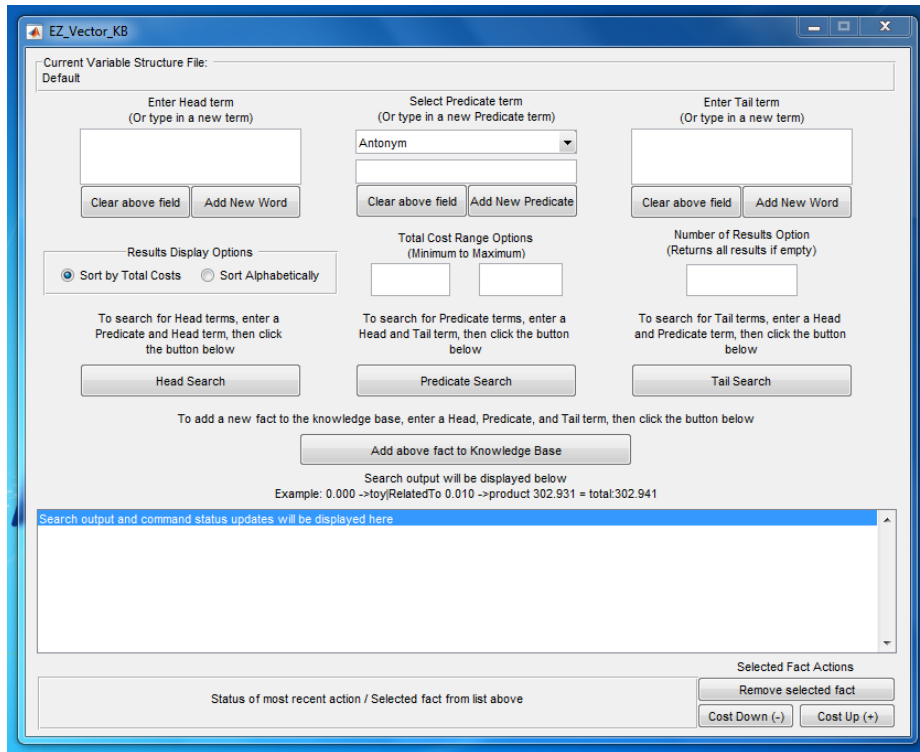


Fig. 1. A screenshot of the EZ Vector-KB interface.

## 2 Background: ConceptNet and Vector-KB

### 2.1 ConceptNet

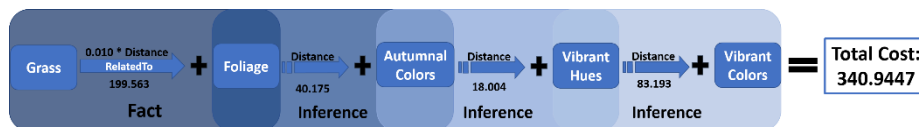
ConceptNet is an open-source knowledge base comprised of words, phrases, and terms people use and the common-sense relationships between them. Relationships made between terms are called edges, where each edge is a triple statement comprised of a head term, a predicate, and then a tail term (i.e. verb => IsA => word) and stored as a data structure with named fields [1][2].

The information in ConceptNet's knowledge base has been collected from multiple verified sources and designed to be scalable and applied to a multitude of scenarios. One such example is an artificial intelligence system that utilized ConceptNet's knowledge base to automatically generate CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) games with commonsense answers as a way of distinguishing humans apart from malicious robots [5]. By defining the

relations between words with edges, ConceptNet allows computer systems to better understand the meaning of words that people use [1].

## 2.2 Vector-KB

The Vector-KB framework sources information from a semantic knowledge base, like ConceptNet, and utilizes that information for deductive reasoning. The primary benefit of Vector-KB is that while the extent of knowledge for a network like ConceptNet is limited to the edges it contains, Vector-KB can make inferences, or well-informed conclusions, based on existing edges, creating new associations between words that do not use predicates to connect them. Terms and relations, also referred to as words and predicates, are stored as vectors, string arrays and numeric arrays that tie string values to their corresponding vectors. These vectors are known as semantic vectors or word vectors, and are learned from large bodies of text, in such a way that words similar in meaning are nearby each other, and words with similar relationships are displaced from each other by similar vectors. In Vector-KB, Edges, as defined in ConceptNet, are now called facts, and retain the triple statement structure (head term, predicate, and tail term). When performing searches, Vector-KB calculates the total cost of each answer returned, relying on the Euclidean distance between vectors within the knowledge base. The cost of each fact is the product of the predicate's weight and the distance between two terms. Since the cost of inferring a relationship between terms is just the distance, inferences have a higher cost. The total cost for an answer is the sum of the weighted distances of each fact and the raw distances of each inference [3][4]. An illustration of how facts and inferences differ and how total cost is determined can be seen below in Figure 2.



**Fig. 2.** An illustration of the fact and inferences that link the terms “grass” and “vibrant colors” together alongside the total cost calculation.

A benefit of the framework’s vector-based organization is that elements such as the distance between terms can be modified. This capability subsequently introduces control over aspects like the cost of facts and total cost of answers. Many functions were developed to leverage this advantage and are compiled together within the EZ Vector-KB interface.

## 3 The EZ Vector-KB Interface

### 3.1 Overview

As stated previously, the Vector-KB interface combines all the Vector-KB functions into a menu-based application, sharing the information that functions return between each other. The interface currently uses ConceptNet as its knowledge base, both as the source for existing information and the save location for new information. Searches of the knowledge base are executed by entering two of the three elements that make up a fact and the interface will return a list of answer terms with their total costs within the knowledge base. Each answer in the list is selectable, with the chosen answer highlighted and displayed at the bottom of the interface window, allowing for modification of individual answers. Optional parameters to alter the results before and after executing queries are also provided. All the features present in EZ Vector-KB are based on existing Vector-KB functions and convert user input into function parameters. The memory required to run the interface is no higher than running the functions on the command-line, so system performance is not increased. Below are some distinctions that make EZ Vector-KB unique from other knowledge base interfaces, followed by descriptions of every feature present in the interface.

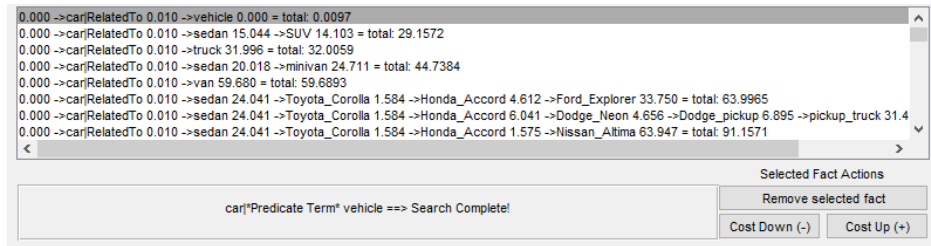
### 3.2 Related Work

As EZ Vector-KB is the only user interface developed for Vector-KB, we compare the interface to other graphical wrappers for existing knowledge base APIs (Application Programming Interface). Many of their functions can be found and have been expanded upon with our interface. Such examples include the Google Knowledge Graph Search API allows for filtered queries like EZ Vector-KB, where lists of entities within the Knowledge Graph can be returned by searching by filters like the name or ID of an entity. Search results are sorted by score, with higher values corresponding to better matches that satisfy the search parameters [6]. Another example is WordNet Search for the Wordnet system, an online application which returns multiple definitions of words and phrases alongside their location IDs within the system [7]. One aspect shared between these applications is that both are read-only and limited to knowledge base queries, whereas EZ Vector-KB includes information management with functions for adding, removing, and modifying knowledge base items.

### 3.3 Flexible Query Searching

To take full advantage of ConceptNet's collection of facts and Vector-KB's reasoning capabilities when returning answers, EZ Vector-KB's search feature can find answers for head, predicate, and tail terms. A user fills in any two of the three search fields, such as a head and predicate term, or a head and tail term, and the corresponding answers will be relevant responses for the third field according to the knowledge base and Vector-KB. When searching for either predicate or tail terms, the general arrangement of each result is (head term => predicate term => tail term = total cost). When searching

for head terms, the ordering of terms is reversed from the perspective of Vector-KB (tail term => predicate term => head term), as the search function is based on tail searching by replacing the head term input for a tail. The interface aesthetically rearranges the reversed paths to resemble the look of the standard format, making answers easier to comprehend.



**Fig. 3.** A sample of the results returned from searching with head term “car” and tail term “vehicle”.

### 3.4 Managing Words, Predicates and Facts

Although ConceptNet boasts a multitude of facts, its extent of knowledge is limited by the words and predicates it contains. Referring back to the first responder situation, a knowledge base may require additional information about electric cables and how they can be located near water. For this reason, adding new words, predicates and facts to the knowledge base is necessary. If a user inputs a word that is not present in ConceptNet, a warning will be displayed, and the user will have the option to add the word with the click of a button. This feature is available in the head and tail fields. Adding new predicates work in a similar manner, and each new entry will be immediately displayed in the drop-down menu of predicates. All three fields are case insensitive and automatically handle whitespace characters to match the format of the head, tail and predicate terms already present in ConceptNet. Some examples include “Apple” converted to “apple”, and “has color” translated to “HasColor”. Creating new facts requires input in all three search fields to match ConceptNet’s triple statement format. Terms and facts already present in the knowledge base cannot be added as duplicates, and a warning will be displayed if any are detected. This safeguard prevents inconsistencies when searching and modifying the cost of terms. Not all the facts in ConceptNet may be relevant to a user’s context and removing said facts is one way to make the knowledge base more concentrated. For example, while adding “cable => RelatedTo => powerline” may be factual, a consequence of Vector-KB’s design is that additional unintended relationships can be created occasionally, such as adding “cable => RelatedTo => ornament”. At the interface’s present state of development, only answers that consist of one fact, meaning the answer has no additional facts or inferences, can be selected and deleted by clicking the "Remove Fact Button". Providing these management functions grants the user control over the knowledge base’s content.

### **3.5 Increasing and Decreasing the Total Cost of Facts**

In some cases, a query's search results may contain the user's desired answer, but that answer may be towards the end of the list of results, meaning the total cost of the answer is noticeably high. This can be especially bothersome for searches that return upwards of 100 answers. For certain answers to appear higher within search results, the total cost of individual facts can be increased or decreased by moving head and tail terms closer to or further away from each other within the knowledge base. Like removing facts, currently only standalone facts can have their total cost adjusted. Facts can be altered by clicking on the desired result and then either the "Cost Down" or "Cost Up" buttons. The distance adjustment between terms is predetermined and was set based on prior testing of values that would return a noticeable degree of change. It is worth noting that changing the distance between two words will also affect the distance between any other facts within ConceptNet that include either term. Due to the impact of adjusting cost, the factors that precipitate the adjusting should be noted and consistent for this version of the knowledge base. If the adjustment is not universal a separate specific or context focused knowledge base version should strongly be considered.

### **3.6 Results Sorting and Filtering**

Before running a search, there are three optional fields that can be edited to alter how the results are sorted and how many will be returned. By default, search results are sorted by total cost from lowest to highest but they can also be sorted alphabetically according to the answer terms. Should a desired answer term have a high cost or be difficult to find due to a large amount of answers, sorting by alphabetical order provides another option to find intended answers quicker. Because Vector-KB returns every feasible path from head to tail terms within ConceptNet, some of these results are very high in cost and could contain very weak inferences. To filter out these answers, the user can determine how many answers will be returned and specify a minimum to maximum total cost range for answers. If both the maximum or minimum are empty, this option is disregarded, but will operate with a singular boundary if either field has a value. Combining these filter options with the ability to influence the importance and existence of facts promote a more focused knowledge base.

### **3.7 Importing and Exporting Variable Structures**

To streamline the process of retaining specialized data, all the required variables for operating the interface are saved into a MATLAB structure. These variables include a copy of ConceptNet, arrays for the collection of words and predicates as vectors, strings and indices, and the parameters for search queries. Before the interface loads, it will ask the user if they wish to import an existing "variable structure" file containing all the necessary values that need to be loaded into memory.

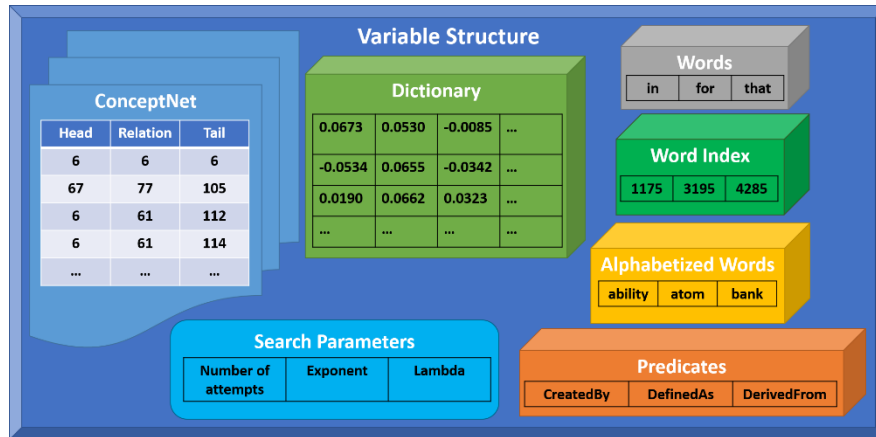


Fig. 4. Variable Structure File Diagram

The filename and its path location are displayed at the top of the window. Upon closing the interface, the user is given the option to save or discard any changes that were made. New structures can be created from Vector-KB's default variables and can be modified or saved as new files for future use. An option to use the existing default variables is also present, but changes will not be saved when closing. Allowing the use of multiple variable structures means the knowledge base and dictionaries can be specialized for a user's specified purpose and reduces the number of irrelevant information returned when searching for desired results.

## 4 Conclusion

Development of EZ Vector-KB is ongoing, with some of its features still in progress. The most prominent improvement to be made involves the fact removal and cost adjustment functions, so that answers with extended chains of reasoning can be modified. This would require a more detailed text parsing algorithm to distinguish each fact and implication, and have the option to modify each fact individually.

The intended use of this interface is to leverage the reasoning capabilities of Vector-KB and apply them to support future research in object recognition and in internet of things experimentation. For object recognition the query inputs will be sourced from text describing the objects identified from a database of images and return all possible relationships the objects have to each other, with the result being a description of all the objects present in a given image. To improve Vector-KB's depiction of images, new words, predicates, and facts can be added alongside adjusting the total cost of existing facts, or irrelevant facts can be removed to promote stronger relationships. For internet of things experimentation, the process is similar but may involve information from different types of sensors. With a more automated approach to managing the Vector-KB

framework, the EZ Vector-KB interface ties these efforts together and promotes advancement of both endeavors.

## References

1. Speer, R., Havasi, C.: Representing General Relational Knowledge in ConceptNet 5. LREC (2012).
2. Speer, R., Havasi, C. (2017). ConceptNet. Retrieved from <http://conceptnet.io/>
3. Summers-Stay, D.: Deductive and Analogical Reasoning on a Semantically Embedded Knowledge Graph. In International Conference on Artificial General Intelligence, pp. 112-122. Springer, Cham (2017).
4. DJSekora - vector-kb (2018). <https://github.com/DJSekora/vector-kb>
5. Yu, H., Riedl, M.: Automatic generation of game-base captchas. Procedural Content Generation (2015).
6. Google APIs Explorer (2015). Retrieved from <https://developers.google.com/apis-explorer/#p/kgsearch/v1/kgsearch.entities.search>
7. WordNet Search - 3.1 (2019). Retrieved from <http://wordnetweb.princeton.edu/perl/webwn>