

# Reader Sentences

**tl;dr** This suite of software is akin to a back-of-the-book index but on steroids. It is a tool for reading.

## Introduction

This directory contains a suite of software used to index and then search databases of sentences, and its purpose is to facilitate a question/answer interface to Distant Reader study carrels. For example, you could ask the system "Who killed Hector?" and get back an answer something like this:

*In the context of the story, it is clear that Hector was killed by Achilles. The passage mentions several instances where Hector is injured or threatened by Achilles, including when Achilles "struck him with a spear and gave him a wound in the groin" (lines 170-171) and when he "dragged him away to cut off his head and take the body to fling before the dogs of Troy" (line 234). Additionally, the passage states that Hector was killed by Achilles, with the fatal blow delivered under the Trojan wall (line 258).*

Alternatively, you could ask the system "What is knowledge?", and get back something like this:

*In this context, knowledge is seen as an existing entity that cannot be created or formed, but rather derived from experience. The mind interacts with its environment to gain knowledge, and all knowledge is based on personal experience. Knowledge is not just theoretical or contemplative, but also practical and applicable in the world. However, it is recognized that knowledge itself depends on a level of "strategic ignorance" to reach manageable conclusions. While information can be a tool for generating knowledge, knowledge belongs only to an individual person. The concept of knowledge is closely tied to the idea of self-knowledge, which can only be realized through face-to-face encounters with others.*

**Very important!** This system is not intended nor expected to return *the* answers to given questions. Instead it is intended to return *plausible* answers, and you are expected to use the results as discussion points or to use traditional reading techniques for the purpose of verification.

Think of this system as a tool to supplement your reading. Create a collection of texts, index (model) the collection, search the index, interact with the model, and in the end, garner a better understanding of the collection. Think of the whole process as a sort of interactive discussion with a book. As such, this system implements a form of reading.

## (Not So) Quick Start

First, you will need to install the Distant Reader Toolbox, and if you get past this step, then the rest ought to be relatively easy (famous last words):

```
pip install reader-toolbox
```

Then you will need to create and/or download at least one Distant Reader study carrel. For example, download the Iliad and the Odyssey by Homer:

```
rdr download author-homer-gutenberg
```

Next, you will need to install the Reader Sentences scripts:

```
git pull https://github.com/ericleasemorgan/reader-sentences.git
```

(Because zero websites, let alone URLs, are permanent, I have cached the Reader Sentences software at <https://distantreader.org/blog/reader-sentences/reader-sentences.zip>, just in case.)

Change directories to the just downloaded repository:

```
cd reader-sentences
```

Now, read and cache all the sentences in a given study carrel:

```
./bin/carrel2sentences.py author-homer-gutenberg
```

At this stage, it is quite likely there will be missing Python modules. Do your best to install them.

The next step is to vectorize ("index") the sentences:

```
./bin/vectorize.py author-homer-gutenberg
```

Again, it is quite likely you will have missing Python modules and/or you do not have Ollama installed. Do your best to install Ollama. You will also need to install two large language models: 1) nomic-embed-text, and 2) llama2. First, nomic-embed-text:

```
ollama pull nomic-embed-text:latest
```

And then llama2:

```
ollama pull llama2:latest
```

Repeat the previous steps, and please be patient; these steps are computationally expensive.

Once you get this far, you can query the database of vectorized sentences. The following command queries the study carrel named "author-homer-gutenberg" for the word "hector" and returns thirty-two sentences:

```
./bin/search.sh author-homer-gutenberg hector 32
```

The result ought to be a long paragraph thirty-two sentences in length. Each sentence ought to allude to Hector in some way, shape, or form.

One way to make more sense of the long paragraph is to divide it into smaller paragraphs, like this:

```
./bin/format.sh
```

Another way to make more sense of the long paragraph is to use a large-language model to summarize it:

```
./bin/summarize.sh
```

Finally, you can use the following command to actually submit a question to be addressed by the system. For example:

```
./bin/elaborate.sh 'who killed hector'
```

## Usage

This suite of software is made up of many little Python scripts and Bash front-ends. This first list of scripts are the most used:

- **./bin/carrel2sentences.py** - given the name of study carrel, extract and cache each of the sentences in each of the carrel's items
- **./bin/vectorize.py** - given the name of a study carrel, vectorize ("index") the cached sentences
- **./bin/search.py** - given a study carrel, a query, and an integer (N), search the carrel's database and return N sentences while simultaneously caching the results in the /etc directory
- **./bin/format.py** - takes the cached result of ./bin/search.py, compares each sentence to its subsequent sentence, and (usually) outputs many smaller paragraphs instead of just one
- **./bin/summarize.py** - takes the cached result of ./bin/search.py, and uses a large-language model to summarize the cache
- **./bin/elaborate.py** - given a query in the form of a question, uses the cached result of ./bin/search.py to address the given question; as such, this script is a simple implementation of a retrieval-augmented generation (RAG) application

The following scripts are front-ends to their Python equivalents, and all they really do is add some formatting to the outputs:

- **./bin/search.sh** - a front-end to ./bin/search.py; simply reformats the results into a single paragraph
- **./bin/format.sh** - a front-end to ./bin/format.py; simply reformats the results to include a few blank lines for readability
- **./bin/summarize.sh** - takes the cached result of ./bin/search.py, and uses a large-language model to summarize the cache
- **./bin/elaborate.sh** - a front-end to ./bin/elaborate.py; simply adds a few blank lines to the output for readability purposes

Queries can be of just about any length and require zero syntax. That said, it is oft-times difficult to articulate useful, meaningful, or comprehensive queries. The scripts below use extracted features from the given study carrel to create more expressive queries for you:

- **./bin/search-with-unigrams.sh** - given the name of a study carrel, an integer (N), and another integer (D), identifies the

N-most frequent unigrams in the given carrel, uses them as the query for `./bin/search.py`, and returns D sentences

- **`./bin/search-with-nouns.sh`** - just like `./bin/search-with-unigrams.sh` but identifies the given carrel's N-most frequent nouns instead of unigrams
- **`./bin/search-with-keywords.sh`** - just like `./bin/search-with-unigrams.sh` but identifies the given carrel's N-most frequent keywords instead of unigrams
- **`./bin/search-with-entities.sh`** - given the name of a carrel, a value of "PERSON" or of "ORG", an integer (N), and another integer (D), identify the given carrel's N-most frequent persons or organizations, uses them as the query for `./bin/search.py`, and returns D sentences
- **`./bin/search-with-semantic.sh`** - given a carrel, a word, an integer (I), and other integer (D), identify the I-most semantically related words to the given word, uses the given word and the related words as the query to `./bin/search.py`, and returns D sentences

In natural language processing, a set of stop words is a list of words with no or little importance. Examples usually include the words "the", "a", "an", "of", etc. Conversely, one might articulate a list of very useful words -- words of great significance. Such a set of words is sometimes called a "lexicon". If you create a file named `./etc/lexicon.txt` within your study carrel(s), then the following scripts will use that file as they query part of the input:

- **`./bin/search-with-lexicon.sh`** - given a study carrel and an integer (D), use the carrel's lexicon as the query for `./bin/search.py` and outputs D sentences
- **`./bin/search-with-modals.sh`** - given a study carrel, reads the carrel's lexicon and outputs all sentences where the lexicon words are the subject of the sentence, and the verb of the sentence is a modal verb; good for identifying very assertive sentences
- **`./bin/search-with-verb.py`** - given a carrel and a lemmatized verb (think "root" word), find all sentences whose subject is a lexicon word and whose verb is a form of the verb
- **`./bin/search-with-verb.sh`** - a front-end to `./bin/search-with-verb.py`, and merely adds some formatting to the output

The curation of lexicons is a thing all unto itself. See Reader Lexicons for ways to create more expressive lexicons.

The following two scripts help you to define words. They do not output *the* definitions of words but rather *plausible* definitions:

- **`./bin/define.py`** - given a carrel and a words, finds all sentences containing the given word, uses the Lesk Algorithm to predict the word's definitions, and outputs possible definitions of the word and their frequencies
- **`./bin/concordance.sh`** - given the name of a study carrel and a word/phrase, output a list of sentence-like things containing the word/phrase

The following are miscellaneous scripts:

- **`./bin/pose-a-question.py`** - given the name of a carrel, randomly select a question from it's database of sentences
- **`./bin/cites.py`** - given the word "human", "csv", or "json", output bibliographic information describing whence the sentences came; good for learning what study carrel item to do close reading against

The following scripts are just for fun. They employ a Markov modeling technique to pseudo-randomly generate sentences. Use these scripts to become familiar with the common bigrams (two-word phrases) in the given carrel.

- **`./bin/markov2sentences.py`** - given the name of a carrel, a two-word phrase, and an integer, parse the text of the given carrel, and output the given phrase and common two-word phrases that **may** follow it; the integer denotes how many times the process should be repeated.
- **`./bin/tell-a-story.py`** - given the name of a carrel, randomly select an item from it, model the text, and output two paragraphs of pseudo-sentences
- **`./bin/tell-a-story.sh`** - a front-end to `./bin/tell-a-story.py`, and merely adds some formatting to the output

## Case study: Who killed Hector?

First, you *must* ask yourself some sort of question. Given a study carrel, what do you want to know? The questions you ask can range from the mundane to the sublime, but you *must* ask yourself a question. In this case, my question will be, "In Homer's Iliad, who is Hector?" And to begin with, I will download a study carrel, cache it's sentences, and vectorize them:

```
rdm download author-homer-gutenberg
./bin/carrel2sentences.py author-homer-gutenberg
./bin/vectorize.py author-homer-gutenberg
```

Again, please be patient. The vectorizing process is computationally expensive. On my Macintosh laptop with eight cores, the vectorizing process takes about five minutes. Your mileage will vary.

The next step is to begin querying the study carrel, and I always suggest starting out very small. Consequently I will query the study carrel for a single word and request a single sentence:

```
./bin/search.sh author-homer-gutenberg hector 1
```

I get the following result, it is merely the sentence which is most like the query. Apparently Hector, who ever he it, dies:

*The death of Hector.*

I then increase the number of sentences to return, and I suggest you always double the value. Thus, the following command identifies the two sentences most closely matching the query:

```
./bin/search.sh author-homer-gutenberg hector 2
```

My result:

*Hector did as his brother bade him. The death of Hector.*

Apparently, Hector, who ever he is, has a brother.

Get some more sentences, and again, I suggest doubling the desired number of sentences:

```
./bin/search.sh author-homer-gutenberg hector 4
```

*Hector," said he," where is your prowess now? " Hector did as his brother bade him. Hector," said he," you make a brave show, but in fight you are sadly wanting. The death of Hector.*

Apparently, Hector is brave.

Yet more:

```
./bin/search.sh author-homer-gutenberg hector 8
```

*Hector," said he," where is your prowess now? Hector did as his brother bade him. But tell me, and tell me true, where did you leave Hector when you started? Hector was angry that his spear should have been hurled in vain, and withdrew under cover of his men. Is it not Hector come to life again? Hector saw him fall and ran up to him; he then thrust a spear into his chest, and killed him close to his own comrades. Hector," said he," you make a brave show, but in fight you are sadly wanting. The death of Hector.*

Apparently, Hector is angry and he uses spears.

Even more:

```
./bin/search.sh author-homer-gutenberg hector 16
```

*Then Hector upbraided him. Then Sarpedon rebuked Hector very sternly. Hector," said he," where is your prowess now? Hector did as his brother bade him. Hector had named him Scamandrius, but the people called him Astyanax, for his father stood alone as chief guardian of Ilius. Hector the son of Priam rages with intolerable fury, and has already done great mischief. " Nestor replied," Most noble son of Atreus, king of men, Agamemnon, Jove will not do all for Hector that Hector thinks he will; he will have troubles yet in plenty if Achilles will lay aside his anger. "I, Hector," said he," Will to the ships and will exploit them. But tell me, and tell me true, where did you leave Hector when you started? Hector was angry that his spear should have been hurled in vain, and withdrew under cover of his men. Is it not Hector come to life again? Hector saw him fall and ran up to him; he then thrust a spear into his chest, and killed him close to his own comrades. Hector sprang also from his chariot to the ground. Nevertheless the end of Hector also was near. Hector," said he," you make a brave show, but in fight you are sadly wanting. The death of Hector.*

Hector seems to be the son of Priam, has fury, does mischief, and rides chariots.

Search for many sentences:

```
./bin/search.sh author-homer-gutenberg hector 32
```

By this time the results are long and somewhat difficult to consume, but you can reformat the results by transforming the big paragraph into smaller ones. The result is easier to read:

```
./bin/format.sh
```

*Then Hector upbraided him. And Alexandrus answered," Hector, your rebuke is just.*

*Then Sarpedon rebuked Hector very sternly. Hector," said he," where is your prowess now? " Hector made him no answer, but rushed onward to fall at once upon the Achaeans and kill many among them. " Hector did as his*

brother bade him. " Then Hector left her, and forthwith was at his own house. Hector 's darling son, and lovely as a star.

Hector had named him Scamandrius, but the people called him Astyanax, for his father stood alone as chief guardian of Ilius. Hector was greatly grieved at the loss of his charioteer, but let him lie for all his sorrow, while he went in quest of another driver; nor did his steeds have to go long without one, for he presently found brave Archeptolemus the son of Iphitus, and made him get up behind the horses, giving the reins into his hand. Hector was greatly grieved at the loss of his charioteer, but for all his sorrow he let him lie where he fell, and bade his brother Cebriones, who was hard by, take the reins. Hector the son of Priam rages with intolerable fury, and has already done great mischief. " Nestor replied, " Most noble son of Atreus, king of men, Agamemnon, Jove will not do all for Hector that Hector thinks he will; he will have troubles yet in plenty if Achilles will lay aside his anger.

"I, Hector," said he, " Will to the ships and will exploit them. Then he took a pointed javelin, and left the camp for the ships, but he was not to return with any news for Hector. But tell me, and tell me true, where did you leave Hector when you started?

" Hector looked fiercely at him and said, " Polydamas, I like not of your reading. Hector was angry that his spear should have been hurled in vain, and withdrew under cover of his men. He found Hector no longer lying upon the ground, but sitting up, for he had just come to himself again. Is it not Hector come to life again? Hector killed Stichius and Arcesilaus, the one, leader of the Boeotians, and the other, friend and comrade of Menestheus.

Hector then cried out to the Trojans, " Forward to the ships, and let the spoils be. Hector saw him fall and ran up to him; he then thrust a spear into his chest, and killed him close to his own comrades. Hector sprang also from his chariot to the ground. Nevertheless the end of Hector also was near. Hector had stripped Patroclus of his armour, and was dragging him away to cut off his head and take the body to fling before the dogs of Troy. Hector," said he, " you make a brave show, but in fight you are sadly wanting. Meanwhile Hector called upon the Trojans and declared that he would fight Achilles. The death of Hector. As a mountain falcon, swiftest of all birds, swoops down upon some cowering dove— the dove flies before him but the falcon with a shrill scream follows close after, resolved to have her— even so did Achilles make straight for Hector with all his might, while Hector fled under the Trojan wall as fast as his limbs could take him. Thus was the head of Hector being dishonoured in the dust. Priam ransoms the body of Hector—Hector's funeral.

Not incidentally, the format command computes the cosine distance between each sentence and its subsequent sentence. If the distance is great, then we can assume the subsequent sentence is not like the given sentence, and therefore a new paragraph is begun.

Moreover, sentences are *not* returned in any sort of relevancy ranking nor closeness value. Instead, sentences are sorted by the name of the study carrel from whence they came and then by the ordinal number of the sentence. Consequently, sentences which are closer together in the output are more likely to appear in the same study carrel item.

One could now closely read the formatted output, and one can begin to answer the question, "Who is Hector?" On the other hand, you can supplement the process by summarizing it with the use of a large-language model:

```
./bin/summarize.sh
```

*In this context, "Then Hector upbraided him." refers to a situation where Hector, a character in the Trojan War, is reprimanding or scolding someone. The sentence is followed by Alexandrus' response, "Hector, your rebuke is just," indicating that he agrees with Hector's criticism.*

*Later in the passage, Sarpedon rebukes Hector sternly, and Hector makes no response but continues to fight. This suggests that Hector is focused on his battle tactics and not paying attention to Sarpedon's criticism.*

*The context then shifts to Hector's grief over the loss of his charioteer, which he mourns by giving him a proper burial. However, he quickly sets aside his sorrow to find a new driver for his chariot.*

*Throughout the passage, Hector is portrayed as a fierce and determined warrior who will stop at nothing to defend his city and family. Despite setbacks and criticisms from others, he remains resolute in his determination to fight for Troy.*

Given enough context, as per above, one can now elaborate on an idea and ask a specific question:

```
./bin/elaborate.sh "who is hector"
```

*In the context of the passage, Hector is a major character and a hero of Troy. He is known for his bravery and prowess in battle, as well as his love for his family and his city. Hector is the son of Priam and the brother of*

Paris, and he is the leader of the Trojan army during the Trojan War. He is also the father of a young son named Scamandrius, who is known as Astyanax.

Throughout the passage, Hector is depicted as a strong and courageous warrior who is determined to defend his city and his people against the Greeks. He is shown to be grief-stricken after the loss of his charioteer, but he quickly recovers and continues to fight bravely in battle. Despite his efforts, Hector ultimately meets his fate at the hands of Achilles, who kills him during a fierce battle.

The passage also highlights Hector's relationship with his family and his city. He is shown to be deeply loved and respected by his father Priam and his brother Paris, and he is described as being "greatly grieved" at the loss of his charioteer. Additionally, Hector's love for his young son Astyanax is evident in the passage, as he is depicted as being deeply concerned about his son's safety during the battle.

Overall, the passage portrays Hector as a noble and heroic character who is fiercely loyal to his family and his city, and who will stop at nothing to defend them against their enemies.

At the very beginning of this reading, it was brought to our attention that Hector dies. Well, let's ask how:

```
./bin/elaborate.sh 'how did hector die'
```

*In the context of the story, Hector died after being attacked by Achilles during a battle. Hector was rebuked by Alexandrus for his criticism, but he remained undeterred and continued to fight. However, he was eventually killed by Achilles, who chased him down and struck him down with a spear. The death of Hector marked a significant turning point in the conflict between the Greeks and Trojans.*

## Case Study: What is the relationship between Ulysses and Penelope?

Let's now try to figure out how Ulysses and Penelope are related. Like before start out small:

```
./bin/search.sh author-homer-gutenberg 'ulysses penelope' 1
```

*Penelope was moved still more deeply as she heard the indisputable proofs that Ulysses laid before her; and when she had again found relief in tears she said to him, " Stranger, I was already disposed to pity you, but henceforth you shall be honoured and made welcome in my house.*

Apparently they know each other.

Get more context:

```
./bin/search.sh author-homer-gutenberg 'ulysses penelope' 2
```

*Penelope was moved still more deeply as she heard the indisputable proofs that Ulysses laid before her; and when she had again found relief in tears she said to him, " Stranger, I was already disposed to pity you, but henceforth you shall be honoured and made welcome in my house. PENELOPE EVENTUALLY RECOGNISES HER HUSBAND—EARLY IN THE MORNING ULYSSES, TELEMACHUS, EUMAEUS, AND PHILOETIUS LEAVE THE TOWN.*

She recognizes him.

Get even more context:

```
./bin/search.sh author-homer-gutenberg 'ulysses penelope' 4
```

Apparently Penelope is admirable and has an excellent nature, and if you read closely Ulysses calls Penelope "wife":

*Not that your wife, Ulysses, is likely to murder you, for Penelope is a very admirable woman, and has an excellent nature. " Penelope was moved still more deeply as she heard the indisputable proofs that Ulysses laid before her; and when she had again found relief in tears she said to him, " Stranger, I was already disposed to pity you, but henceforth you shall be honoured and made welcome in my house. PENELOPE EVENTUALLY RECOGNISES HER HUSBAND—EARLY IN THE MORNING ULYSSES, TELEMACHUS, EUMAEUS, AND PHILOETIUS LEAVE THE TOWN. On this, Ulysses rose from his comfortable bed and said to Penelope, " Wife, we have both of us had our full share of troubles, you, here, in lamenting my absence, and I in being prevented from getting home though I was longing all the time to do so.*

Jump ahead and get a lot of context, reformat the results, and summarize:

```
./bin/search.sh author-homer-gutenberg 'ulysses penelope' 32
```

```
./bin/format.sh
./bin/summarize.sh
```

*In this context, Penelope is talking to a stranger who claims to be her husband Ulysses, but Minerva has transformed him into an old man to prevent the suitors from recognizing him. Penelope is skeptical at first, but eventually recognizes Ulysses through his words and stories. She is overjoyed to see him and they share a romantic moment together. The ghost of Agamemnon also appears and praises Ulysses for his faithful wife. The scene ends with Ulysses and Penelope preparing for bed, with Ulysses expressing gratitude for having a wife as understanding and loyal as Penelope.*

Finally, ask a specific question:

```
./bin/elaborate.sh 'what is the relationship between ulysses and penelope'
```

*In this passage, Ulysses and Penelope have a deep emotional connection as they reunite after many years of separation. Penelope recognizes Ulysses as her husband and is overjoyed to see him, while Ulysses is grateful to be back in his home with his wife. The goddess Minerva plays a crucial role in their reunion, helping them to recognize each other despite their physical transformations. The passage also highlights the loyalty and devotion of Penelope, who has remained faithful to Ulysses even after many years of his absence. Overall, the relationship between Ulysses and Penelope is one of love, trust, and mutual respect.*

## Next Steps

A few possible next steps include:

- download Jane Austen's *Emma* and ask the question, "Who is Emma?"; the Distant Reader study carrel for Austen's Emma is `title-emma_by_austen-gutenberg`
- download Henry David Thoreau's *Walden* (`title-walden_by_thoreau-gutenberg`), and learn about the importance of beans
- download John Locke's *An Essay Concerning Human Understanding* (`title-essay_by_locke-gutenberg`), and address the question, "What are knowledge, truth, and understanding?"

## Summary

This suite of software is as tool for reading. Create and/or download Distant Reader study carrels ("data sets"). Extract and cache all of the sentences in all of the items in the carrel. Vectorize (index/model) them. Use the scripts in this distribution to query the sentences and thus become familiar with the carrel's content. Compare and contrast the results from one study carrel with the results from another study carrel. Use the outputs of these scripts as points of discussion with other people.

Finally, while the system uses various techniques to riff on the search results, do your best to closely read the search results before you summarize or elaborate. It is your responsibility to figure out the degree any of the underlying large-language models are hallucinating. Again, the system does not output *the* answer but instead it outputs *plausible* answers.

---

Creator: Eric Lease Morgan <[eric\\_morgan@infomotions.com](mailto:eric_morgan@infomotions.com)>

Source: This document is rooted in the REAME file of the corresponding GitHub repository at <https://github.com/ericleasemorgan/reader-sentences/>.

Date created: 2025-10-15

Date updated: 2025-10-17

Subject(s): Distant Reader; readings; indexes;

URL: <https://distantreader.org/blog/reader-sentences/>