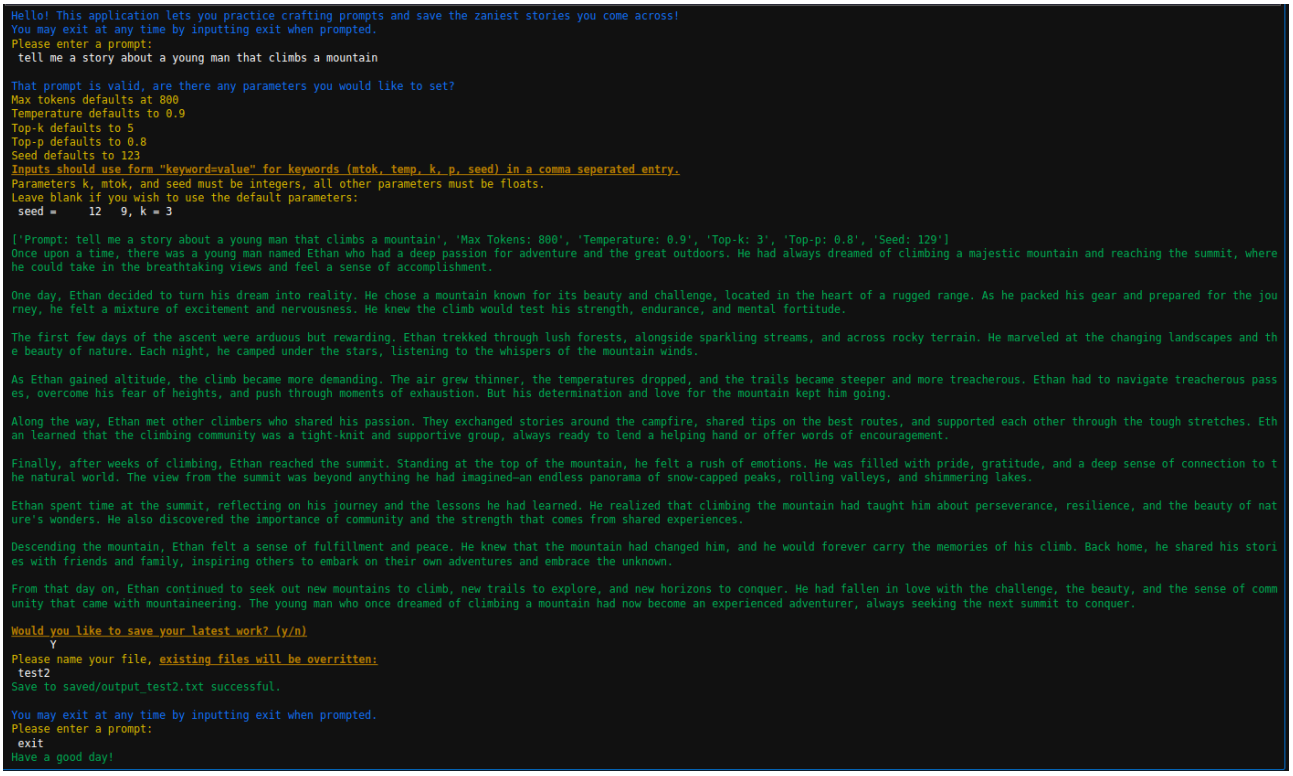
Capstone Project: AI-Powered Text Completion
Repository Link: ___

Intended Usage and Features:

This program asks the user to enter a prompt for cohere model “command-r-plus” to use in generating text. The highlight of this program is allowing the user to tweak parameters in a safe and constructive manner. Any issues are sorted before the API call and the user is given instruction on what went wrong and how it can be resolved. The application also includes a saving functionality where a user is enabled to keep record of the model inputs and generation output in a .txt file saved to their working directory.

Features:
- Color coded responses that allow for easier reading of the interface.
- Inputs are pared down to more universal forms, space and case-sensitivity issues are mitigated.
- Order of parameters specified is not important.
- Users are informed of the exact parameter that is invalid at the moment it is parsed.
- Program accounts for invalid keys, value types, and even defining the same parameter twice.
- Program loops indefinitely until user chooses to exit.
- Program offers two chances to save, once when text is generated and again when exited should the user have opted out of the first chance.
- File names specified by the user are denied any special character input and do so without interrupting the program.

Program Output Examples:



```
Hello! This application lets you practice crafting prompts and save the zaniest stories you come across!
You may exit at any time by inputting exit when prompted.
Please enter a prompt:
 tell me a story about two cats cheating on their bar exam

That prompt is valid, are there any parameters you would like to set?
Max tokens defaults at 800
Temperature defaults to 0.9
Top-k defaults to 5
Top-p defaults to 0.8
Seed defaults to 123
Inputs should use form "keyword=value" for keywords (mtok, temp, k, p, seed) in a comma seperated entry.
Parameters k, mtok, and seed must be integers, all other parameters must be floats.
Leave blank if you wish to use the default parameters:
 tempera=0.9, seed=122, seed=124, p=0.4, k=3.5, mtok=799.9
Argument "tempera=0.9" not valid, please try again:


That prompt is valid, are there any parameters you would like to set?
Max tokens defaults at 800
Temperature defaults to 0.9
Top-k defaults to 5
Top-p defaults to 0.8
Seed defaults to 123
Inputs should use form "keyword=value" for keywords (mtok, temp, k, p, seed) in a comma seperated entry.
Parameters k, mtok, and seed must be integers, all other parameters must be floats.
Leave blank if you wish to use the default parameters:
 temp=0.9, seed=122, seed=124, p=0.4, k=3.5, mtok=799.9
Each parameter can be defined only once.
Argument "seed=124" not valid, please try again:


That prompt is valid, are there any parameters you would like to set?
Max tokens defaults at 800
Temperature defaults to 0.9
Top-k defaults to 5
Top-p defaults to 0.8
Seed defaults to 123
Inputs should use form "keyword=value" for keywords (mtok, temp, k, p, seed) in a comma seperated entry.
Parameters k, mtok, and seed must be integers, all other parameters must be floats.
Leave blank if you wish to use the default parameters:
 temp=0.9, seed=122, p=0.4, k=3.5, mtok=799.9
Parameter value must be of type int.
Argument "k=3.5" not valid, please try again:


That prompt is valid, are there any parameters you would like to set?
Max tokens defaults at 800
Temperature defaults to 0.9
Top-k defaults to 5
Top-p defaults to 0.8
Seed defaults to 123
Inputs should use form "keyword=value" for keywords (mtok, temp, k, p, seed) in a comma seperated entry.
Parameters k, mtok, and seed must be integers, all other parameters must be floats.
Leave blank if you wish to use the default parameters:
 temp=0.9, seed=122, p=1, k=3, mtok=799.9
Parameter value must be of type int.
Argument "mtok=799.9" not valid, please try again:


That prompt is valid, are there any parameters you would like to set?
Max tokens defaults at 800
Temperature defaults to 0.9
Top-k defaults to 5
Top-p defaults to 0.8
Seed defaults to 123
Inputs should use form "keyword=value" for keywords (mtok, temp, k, p, seed) in a comma seperated entry.
Parameters k, mtok, and seed must be integers, all other parameters must be floats.
Leave blank if you wish to use the default parameters:
 temp=0.9, seed=122, p=1, k=3, mtok=799

['Prompt: tell me a story about two cats cheating on their bar exam', 'Max Tokens: 799', 'Temperature: 0.9'
Two cats, Whiskers and Mittens, were studying for their upcoming bar exam. They had been working tirelessly
Leave blank if you wish to use the default parameters:
 temp=0.9, seed=122, p=1, k=3, mtok=799

['Prompt: tell me a story about two cats cheating on their bar exam', 'Max Tokens: 799', 'Temperature: 0.9', 'Top-k: 3', 'Top-p: 1.0', 'Seed: 122']
Two cats, Whiskers and Mittens, were studying for their upcoming bar exam. They had been working tirelessly for months, reviewing legal textbooks and practicing mock trials. However, as the exam dr
ew closer, they began to feel overwhelmed and unsure of their ability to pass.

One day, while studying in the library, Whiskers stumbled upon a hidden room filled with forbidden knowledge. He discovered a stack of ancient scrolls that contained secrets to passing the bar exam
without actually having to study. Whiskers knew that using these scrolls would be cheating, but the temptation was too great. He convinced Mittens to join him in his scheme, and together they began
to use the forbidden knowledge to their advantage.

At first, everything seemed to be going according to plan. Whiskers and Mittens aced their practice exams and felt more confident than ever. But as the real bar exam drew closer, their guilt began
to weigh on them. They knew that what they were doing was wrong and that if they were caught, they could face serious consequences.

On the day of the exam, Whiskers and Mittens entered the testing room with their heads held high. They had memorized the forbidden knowledge and were certain that they would pass with flying color
s. However, as they began the exam, they realized that the questions were nothing like what they had prepared for. The forbidden knowledge had given them a false sense of security, and they soon re
alized that they were in way over their heads.

As the exam progressed, Whiskers and Mittens struggled to answer the questions and their panic grew. They knew that they were going to fail, and their cheating would be exposed. In a desperate atte
mpt to save themselves, they tried to copy answers from other students, but their efforts were futile.

In the end, Whiskers and Mittens failed the bar exam miserably. Their cheating had been discovered, and they were expelled from the law school. They learned a hard lesson that day — that cheating m
ay provide temporary gains, but in the end, honesty and hard work are always the best policy.

Would you like to save your latest work? (y/n)
 n
You may exit at any time by inputting exit when prompted.
Please enter a prompt:
 exit
Would you like to save your latest work? This is the last chance.(y/n)
 y
Please name your file,existing files will be overritten:
 #@..   t est /1
Save to saved/output_test1.txt successful.

Have a good day!
```

Effects in Each Parameter

Experimented parameters include max tokens, temperature, top-k, and top-p. Tweaking max tokens only changes if the generated text truncates earlier than intended so that will be absent from this evaluation.

Temperature:

This metric distributes the probabilities of the next token different from how softmax will naturally do so. Higher temperature values lead to a more even distribution. This translates to more entropy within a given text, as even the naturally unlikely tokens have heightened and fairer consideration when compared to more 'relevant' ones. The model used caps this value off at 1.0, but other models allow for higher degrees. The general rule of thumb is that any temperature value greater than 2.0 will generate nonsense with most models.

Things to note:

While coherency has not been affected much between the mid to high range of temperature values within a prompt, it has helped to improve the depth of the generated response. We can see this in our creative outputs; higher temperature values led to a more thorough story, albeit being a direct copy of *Alice in Wonderland*. Mid ranged temperature values did not make the story any less understandable, though lower ranged values would have this effect, instead they just shortened the story. Interestingly enough, a high temperature value paired with a low k or p value has the same conciseness but much less detail than a low temperature would have in this scenario.

Top-K:

K and p values look at the future token probabilities and samples them, either by their ordinal rank or by the sum of their probabilities. Both start from the 'top', the highest probability, and work down. Top-k is the former, picking the greatest k-tokens and choosing from those rather than the extended list. When paired with higher temperature values, the distribution is more flattened but top-k is pretty resilient against this. When used in joint with top-p, top-k is considered first and then further trimmed by top-p.

Things to note:

There wasn't much of a conciseness factor in the instructional outputs that experimented with top-k, but lower values did offer more relevant responses than higher values. When prompted to 'explain like I am five', the models response had a stronger motif with lower k values than higher. I liked this output more as the general response should be comparing whatever subject to some experience a child is likely to have had, lower k values did so strongly whereas mid and high values did not.

Top-P:

Top-p is the latter in the previous explanation. It begins with the highest probabilities and sums them up so long as it doesn't exceed p. All tokens within that definition are then considered for use. When used with top-k, this summation of p relative to a perfect 1.0 probability is altered. The 1.0 it now considers its probabilities against is that of the total sum of top-k token probabilities. Using both can be tricky to consider but they act similarly enough.

Things to note:

Tested with the informational texts, I pulled a few paragraphs from a Wikipedia page and asked the model to summarize with both a high and a low p value. As suspected, a lower p value does indeed lead to more shallow responses. The depth I got from a 0.3 response was only two thirds as long as the 1.0 response. But this was done at no grand detriment to cohesiveness, naturally some detail was lost but having a model use lower p or k values for summaries can be very valuable for this specific task.

Please look to the Git Repository for the saved outputs, input parameters are listed at the top.

Evaluation:

The model performs decently on instructional and informational tasks, but a clear pattern can be noticed on vague creativity tasks. The model will just regurgitate popular culture references in this case. This is less than ideal but can be countered with a stronger prompt. I haven't noticed it struggle with very much outside of toying with parameters. Discovering that a combination of temperature and k/p values will change how well it considers a context when responding makes the model seem a little flimsier than it really is. For example, the instructional prompts inquired about inodes and asked the model to explain them in simple terms. Lower k values had a stronger argument than higher ones even with a flatter distribution of probabilities. The model has a robustness I could not expect had there been no experimentation. I can speculate that using a lower temperature value in that specific experiment would have likely kept up the overarching context a little better. A lower temperature would give more bias to likelier tokens and the k value would have mattered less. All in all, the model used had performed quite well but struggles when it comes to creativity.