Machine Learning Engineer Nanodegree

Capstone Project

"Semantic similarity extraction using word vectors in Mahabharata dataset"

Tilak D

March 23rd, 2017

Definition

Project Overview

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages and, in particular, concerned with programming computers to fruitfully process large natural language corpora.

Challenges in natural language processing frequently involve natural language understanding, natural language generation (frequently from formal, machine-readable logical forms), connecting language and machine perception, managing human-computer dialog systems, or some combination thereof.

The Mahabharata is one of the two major Sanskrit epics of ancient India. The Mahabharata is an epic narrative of the Kurukshetra War and the fates of the Kaurava and the Pandava princes. It also contains philosophical and devotional material, such as a discussion of the four "goals of life" or purusharthas. Among the principal works and stories in the Mahabharata are the Bhagavad Gita, the story of Damayanti, an abbreviated version of the Ramayana, and the Rishyasringa, often considered as works in their own right.

The Mahabharata is the longest known epic poem and has been described as "the longest poem ever written" Its longest version consists of over 100,000 shloka or over 200,000 individual verse lines (each shloka is a couplet), and long prose passages. About 1.8 million words in total, the Mahabharata is roughly ten times the length of the Iliad and the Odyssey combined, or about four times the length of the Ramayana, which makes it a huge dataset for using NLP.

By utilizing NLP, we can organize and structure knowledge of the huge Mahabharata to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis and topic segmentation, which will be helpful for extracting quick, short and concise answers.

Word2vec (Ref[1]) is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes a large corpus of text as input and produces a vector space, typically of

several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space (Ref[2]). Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Problem Statement

In ancient times this knowledge used to pass along generations, but in this fast moving world, everyone needs answers easily and to be in their fingertips. Most of the relationships between characters in lengthy novels are hard to remember for general public, here NLP's Semantic similarities come into play.

This corpus of data, of about 29100 words will be fed as input to the model to create word vectors and with the help of word2vec, we would analyze semantic similarities between characters, i.e. word vector representation will be created with neural network and by measuring similarity between similar words through cosine similarity to find the similarity between words. This will be used to answer questions related to relationships between characters in Mahabharata. For example, Arjuna was the son of Indra- the king of celestials and Krishna was son of Vasudeva. If an input is given as Arjuna, Indra and Krishan, system should be capable to provide an answer as Vasudeva, based on the knowledge learnt using NLP.

Metrics

As explained in the previous section, the result obtained can only be objectively compared with the real facts. An evaluation metrics can only be a percentage of correct semantic obtained, which will be obtained through a sizable number of inputs. The model will be tested with an input containing a list of all the semantics obtained from Mahabharata in the below format.

{A} is related to {B}, as {C} is related to {D}

Here A, B and D would be the inputs and C will the output provided by the model, based on the learning. Below is a subset of the compiled test semantics (Red words are the expected outputs),

```
Dhritarastra is related to Pandu, as Sahadeva is related to Nakula
Bhima is related to Arjuna, as Ambalika is related to Ambika
Pandu is related to Kunti, as Dhritarashtra is related to Gandhari
Bhima is related to Draupadi, as Arjuna is related to Chitrangada
Karna is related to Kunti, as Duryodhana is related to Gandhari
Bhima is related to Draupadi, as Arjuna is related to Subhadra
Yudhisthira is related to Kunti, as Duryodhana is related to Gandhari
Bhima is related to Kunti, as Nakula is related to Madri
Bhima is related to Draupadi, as Arjuna is related to Ulupi
Vichitravirya is related to Ambalika, as Vichitravirya is related to Ambika
Bhima is related to Ghatotkacha, as Arjuna is related to Abhimanyu
Bhima is related to Draupadi, as Arjuna is related to Draupadi
```

The output provided by the model is compared to the actual semantics and a percentage accuracy is calculated, i.e. (Total number of right answers given)/(Test size).

Analysis

Data Exploration

Dataset is a set 18 text file, where in each text file is Parva (Which means book in Sanskrit). Figure 1 shows information of all 18 books.

Parva	Title	Sub-parvas	Contents
1	Adi Parva *(The Book of the Beginning)*	1–19	How the *Mahabharata* came to be narrated by Sauti to the assembled rishis at Naimisharanya, after having been recited at the *sarpasattra* of Janamejaya by Vaishampayana at Takṣaśilā. The history and genealogy of the Bharata and Bhrigu races is recalled, as is the birth and early life of the Kuru princes (*adi* means first).
2	Sabha Parva (The Book of the Assembly Hall)	20–28	Maya Danava erects the palace and court (*sabha*), at Indraprastha. Life at the court, Yudhishthira's Rajasuya Yajna, the game of dice, the disrobing of Pandava wife Draupadi and eventual exile of the Pandavas.
3	Vana Parva *also* Aranyaka-parva, Aranya-parva (The Book of the Forest)	29–44	The twelve years of exile in the forest (*aranya*).
4	Virata Parva (The Book of Virata)	45–48	The year spent incognito at the court of Virata.
5	Udyoga Parva (The Book of the Effort)	49–59	Preparations for war and efforts to bring about peace between the Kaurava and the Pandava sides which eventually fail (*udyoga* means effort or work).
6	Bhishma Parva (The Book of Bhishma)	60–64	The first part of the great battle, with Bhishma as commander for the Kaurava and his fall on the bed of arrows. (Includes the *Bhagavad Gita* in chapters 25[27]-42.[28])
7	Drona Parva (The Book of Drona)	65–72	The battle continues, with Drona as commander. This is the major book of the war. Most of the great warriors on both sides are dead by the end of this book.
8	Karna Parva (The Book of Karna)	73	The continuation of the battle with Karna as commander of the Kaurava forces.
9	Shalya Parva (The Book of Shalya)	74–77	The last day of the battle, with Shalya as commander. Also told in detail, is the pilgrimage of Balarama to the fords of the river Saraswati and the mace fight between Bhima and Duryodhana which ends the war, since Bhima kills Duryodhana by smashing him on the thighs with a mace.
10	Sauptika Parva (The Book of the Sleeping Warriors)	78–80	Ashvattama, Kripa and Kritavarma kill the remaining Pandava army in their sleep. Only 7 warriors remain on the Pandava side and 3 on the Kaurava side.
11	Stri Parva (The Book of the Women)	81–85	Gandhari and the women (*stri*) of the Kauravas and Pandavas lament the dead and Gandhari cursing Krishna for the massive destruction and the extermination of the Kaurava.
12	Shanti Parva (The Book of Peace)	86–88	The crowning of Yudhishthira as king of Hastinapura, and instructions from Bhishma for the newly anointed king on society, economics and politics. This is the longest book of the Mahabharata. Kisari Mohan Ganguli considers this Parva as a later interpolation.'
13	Anushasana Parva (The Book of the Instructions)	89–90	The final instructions (*anushasana*) from Bhishma.
14	Ashvamedhika Parva (The Book of the Horse Sacrifice)[29]	91–92	The royal ceremony of the Ashvamedha (Horse sacrifice) conducted by Yudhishthira. The world conquest by Arjuna. The Anugita is told by Krishna to Arjuna.
15	Ashramavasika Parva (The Book of the Hermitage)	93–95	The eventual deaths of Dhritarashtra, Gandhari and Kunti in a forest fire when they are living in a hermitage in the Himalayas. Vidura predeceases them and Sanjaya on Dhritarashtra's bidding goes to live in the higher Himalayas.
16	Mausala Parva (The Book of the Clubs)	96	The materialisation of Gandhari's curse, i.e., the infighting between the Yadavas with maces (*mausala*) and the eventual destruction of the Yadavas.
17	Mahaprasthanika Parva (The Book of the Great Journey)	97	The great journey of Yudhishthira, his brothers and his wife Draupadi across the whole country and finally their ascent of the great Himalayas where each Pandava falls except for Yudhishthira.
18	Svargarohana Parva (The Book of the Ascent to Heaven)	98	Yudhishthira's final test and the return of the Pandavas to the spiritual world (*svarga*).
khila	*Harivamsa Parva* (The Book of the Genealogy of Hari)	99–100	This is an addendum to the 18 books, and covers those parts of the life of Krishna which is not covered in the 18 parvas of the *Mahabharata*.

Figure 1: Title and contents of all 18 books of Mahabharata

Dataset was obtained from an online library, Nitaaiveda. Figure 2 indicates the statistics of all the books combined, having approximately 291,000 words in total.

Statistics:

Pages	461
Words	291,175
Characters (no spaces)	1,410,985
Characters (with spaces)	1,700,430
Paragraphs	2,585
Lines	18,634

Figure 2: Statistics of Mahabharata

This corpus of data, of about 29100 English words will be fed as input to the model to create word vectors using word2vec (Ref[3]), and with the help of NLTK, we would analyze semantic similarities between characters. Below is a small snippet from the first book of Mahabharata (One of eighteen).

```
According to the historical records of this earth, there once lived a
King named Maharaja Shantanu, the son of Pratipa, who took his birth in the
solar dynasty and was considered naradeva, the manifest representative of the
Supreme Lord on earth. His fame and rule extended to all parts of the world.
The qualities of self-control, liberality, forgiveness, intelligence,
modesty, patience and power always resided this exalted emperor. His neck was
marked with three lines like a conchshell, and his shoulders were broad. In
prowess He resembled a maddened elephant. Above all these qualities, he was a
devoted servant of Lord Vishnu, and therefore he was given the title, "King
of kings".
Once when Maharaja Shantanu, that bull among men, was wandering in the
forest, he came upon a place frequented by the Siddhas and Charanas (a class
of heavenly demigods). There he saw an angelic woman who appeared like the
goddess of fortune herself. In truth, she was the personification of the
river Ganges. She was glancing at the monarch with her youthful longing eyes,
and Maharaja Shantanu became attracted to her. He then approached her
inquiring, "O beautiful woman, are you from the race of the Gandharvas,
Apsaras, Yakshas, Nagas or the human race? As yet I have no queen, and your
birth appears divine. Whatever your origin, O celestial beauty, I request you
to become my wife."
```

Algorithms and Techniques

As described above the corpus of words will be used as an input to create word vectors (Ref[4]) using word2vec, with the help of t-SNE (Ref[5]), reduce the dimensions of the word vectors and finally use cosine similarity to analyze semantic similarities, i.e. to answer relationship questions based on the learning. The end solution of this project will be to analyze relationships and logics in the dataset.

Skeleton of the approach will be,

1) Create a dataset by converting corpus into sentences in turn into a bag of words.
2) Improve the dataset by removing the words and symbols that does not have meanings.
3) Build model by training word2vec and build a vocabulary.
4) The trained word vectors will be in a high dimension, example more than 200 dimension. Using t-distributed stochastic neighbor embedding or t-SNE to reduce this higher dimension to a feasible, analyzable dimension size. Train the above dimensionality reduction algorithms to create a lower dimension dataset. Plot and analyze it for semantics.
5) For further analysis and to answer the problem statement, I am planning to use cosine similarity, to assess similarities between 2 word vectors, to answer similarity questions on the 3rd word vector.

Benchmark

The problem which is being solved can only benchmarked based on the real info based on the book. As described in an example in Problem Statement, Arjuna was the son of Indra- the king of celestials and Krishna was son of Vasudeva. If an input is given as Arjuna, Indra and Krishna, system should be capable to provide an answer as Vasudeva, based on the father son relation knowledge learnt using NLP. This result can only be objectively compared with the real facts.

The real facts about the data set already exists, to benchmark the model I have compiled 23 relationship facts and will be adding few more as I build the model. For example, below are a few of the real data used to benchmark the model.

```
Dhritarastra is related to Pandu, as Sahadeva is related to Nakula
Bhima is related to Arjuna, as Ambalika is related to Ambika
Pandu is related to Kunti, as Dhritarashtra is related to Gandhari
Bhima is related to Draupadi, as Arjuna is related to Chitrangada
Karna is related to Kunti, as Duryodhana is related to Gandhari
.
.
```

Methodology

Data Preprocessing

Initial part of the data preprocessing is done by removing all the *Stop words* from the corpus. Stop words are extremely common words, such as the, at, a, an; which would appear to be of little value to the meaning of the sentence, so that we can focus on the important words instead.

For example, consider the first sentence from first book of Mahabharata,

```
According to the historical records of this earth, there once lived a King
named Maharaja Shantanu, the son of Pratipa, who took his birth in the solar
dynasty and was considered naradeva, the manifest representative of the
Supreme Lord on earth.
```

This can be reduced to a sentence as shown below and still carry majority of the intended meaning.

```
 According historical records earth lived King named Maharaja Shantanu son
Pratipa took birth solar dynasty considered naradeva manifest representative
Supreme Lord earth.
```

Next part of the preprocessing step is done by *tokenizing* all the sentences from 18 books into words for further word analysis. Tokenization is the task of chopping sentences up into pieces, called tokens, also at the same time throwing away certain characters, such as punctuation. Figure 3 shows the input and tokenized output of an example sentence.

```
Above all these qualities, he was a devoted servant of Lord Vishnu, and therefore he was given the title,
 "King of kings".
[u'Above', u'all', u'these', u'qualities', u'he', u'was', u'a', u'devoted', u'servant', u'of', u'Lord',
 u'Vishnu', u'and', u'therefore', u'he', u'was', u'given', u'the', u'title', u'King', u'of', u'kings']
```

Figure 3: Tokenizing

Implementation

The main steps involved in the developed model are listed below along with code snippets,

1) Corpus to bag of words – All the 18 books of Mahabharata is merged to make a huge corpus. Paragraphs are split into sentences and in turn this corpus is chopped into words with the help of split() function.

2) Stop word and punctuation removal - Improve the dataset by removing the stop words and symbols that does not have meanings. This is done using stopwords and punkt packages of NLTK library.

3) Train word2vec - Train word2vec on this processed corpus, thereby converting words into vectors. This is done using Gensim's word2vec library. Below is a code snippet.

```
mahabharata2vec = w2v.Word2Vec(
    sg = 1,
    seed = 1,
    workers = multiprocessing.cpu_count(),
    size = 300,
    min_count = 3,
    window = 7,
    sample = 1e-3
)
mahabharata2vec.train(sentences)
```

4) The trained word vectors will be in a high dimension, in the above code snippet number of dimentions is equal to 300. With the help of t-distributed stochastic neighbor embedding or t-SNE reduce this higher dimension to a feasible, analyzable dimension size. T-SNE is impemented using sklearn. Below is a code snippet of the implementation.

```
tsne =sklearn.manifold.TSNE(n_components=3,perplexity=50.0,n_iter=5000,random_state=0)

all_word_vectors_matrix = mahabharata2vec.wv.syn0
```

Train the above dimensionality reduction algorithms to create a lower 3 dimension dataset.

```
all_word_vectors_matrix_2d = tsne.fit_transform(all_word_vectors_matrix)
```

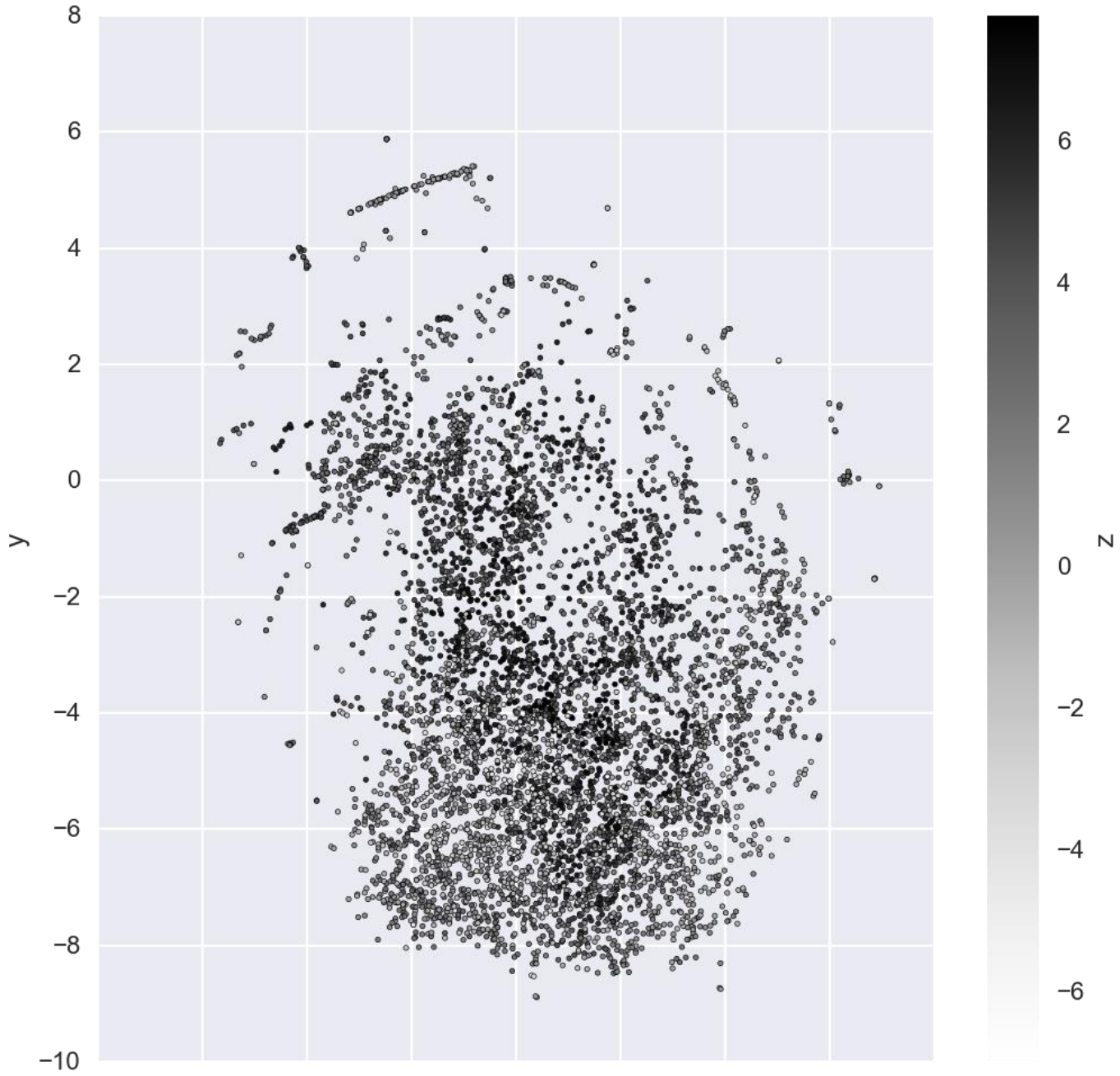
The x, y and z coordinates can be extracted for plotting purpose and semantic analysis.

```
points = pd.DataFrame(
    [
        (word, coords[0], coords[1], coords[2])
        for word, coords in [
            (word, all_word_vectors_matrix_2d[mahabharata2vec.wv.vocab[word].index])
            for word in mahabharata2vec.wv.vocab
        ]
    ],
    columns=["word", "x", "y", "z"]
)
```

Coordinates of the first few words are a follows,

	word	x	y	z
0	raining	0.781272	-4.57775	6.318752
1	yellow	0.704123	0.076027	2.908227
2	four	-3.05614	-1.01975	2.592682
3	woods	2.339273	-4.82267	4.904764
4	hanging	-0.47257	-3.03956	6.286031
5	looking	2.380621	-1.55693	3.994533
6	granting	3.470153	-5.0568	3.453314
7	eligible	-1.1368	-7.32454	1.005169
8	Kundadahara	1.235221	-3.27123	7.61231
9	lord	4.241032	-3.02878	0.147951
10	sinking	0.687666	-2.61176	5.224991

Based on the extracted coordinates, plot scatter plot for visual analysis.



5) For further analysis and to answer the problem statement to use cosine similarity, to assess similarities between 2 word vectors, to answer similarity questions on the 3rd word vector.

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions

should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

- _Has an initial solution been found and clearly reported?_

- _Is the process of improvement clearly documented, such as what techniques were used?_

- _Are intermediate and final solutions clearly reported as the process is improved?_

Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

- _Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?_

- _Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?_

- _Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?_

- _Can results found from the model be trusted?_

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

- _Are the final results found stronger than the benchmark result reported earlier?_

- _Have you thoroughly analyzed and discussed the final solution?_

- _Is the final solution significant enough to have solved the problem?_

Conclusion

(approx. 1-2 pages)

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

- _Have you visualized a relevant or important quality about the problem, dataset, input data, or results?_

- _Is the visualization thoroughly analyzed and discussed?_

- _If a plot is provided, are the axes, title, and datum clearly defined?_

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

- _Have you thoroughly summarized the entire process you used for this project?_

- _Were there any interesting aspects of the project?_

- _Were there any difficult aspects of the project?_

- _Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?_

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

- _Are there further improvements that could be made on the algorithms or techniques you used in this project?_

- _Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?_

- _If you used your final solution as the new benchmark, do you think an even better solution exists?_

Before submitting, ask yourself. . .

- Does the project report you've written follow a well-organized structure similar to that of the project template?

- Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?

- Would the intended audience of your project be able to understand your analysis, methods, and results?

- Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?

- Are all the resources used for this project correctly cited and referenced?

- Is the code that implements your solution easily readable and properly commented?

- Does the code execute without error and produce results similar to those reported?