

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

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

Dataset was obtained from an online library, [Nitaiveda](#). 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, cosine similarity is used to assess similarities between 2 word vectors, to answer similarity questions on the 3rd word vector.
- 6) Calculate accuracy.

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 and to benchmark the model I have compiled 144 unique relationship facts, such as father-son, mother-son, siblings and spouse. These 144 unique relationships are used in different combinations to generate 1551 relations. For example,

- 1) Finding Son given Father, based on an example Son-Father relationship
- 2) Finding Father given Son, based on an example Father-Son relationship
- 3) Finding Son given Mother, based on an example Son-Mother relationship
- 4) Finding Mother given Son, based on an example Mother-Son relationship
- 5) Finding Husband given Wife, based on an example Husband-Wife relationship
- 6) Finding Wife given Husband, based on an example Wife- Husband relationship
- 7) Finding Sibling, based on a Sibling example.

Below is a set of a few 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
 .
 .

As Mahabharata dataset is an Indian ancient Sanskrit script, which has been literally translated to English and most of the words in this dataset does not exist in English dictionary makes it extremely complex for word2vec and t-SNE to find relationships, I am not expecting more than 20% accuracy in relationship questions.

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 = 500,
    min_count = 7,
    window = 25,
    sample = 1e-3
)
mahabharata2vec.train(sentences)
```

- 4) The trained word vectors will be in a high dimension, in the above code snippet number of dimensions is equal to 500. 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 implemented using sklearn. Below is a code snippet of the implementation.

```
tsne =sklearn.manifold.TSNE(n_components=3,perplexity=50.0,n_iter=9000,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_3d = 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])
```

```

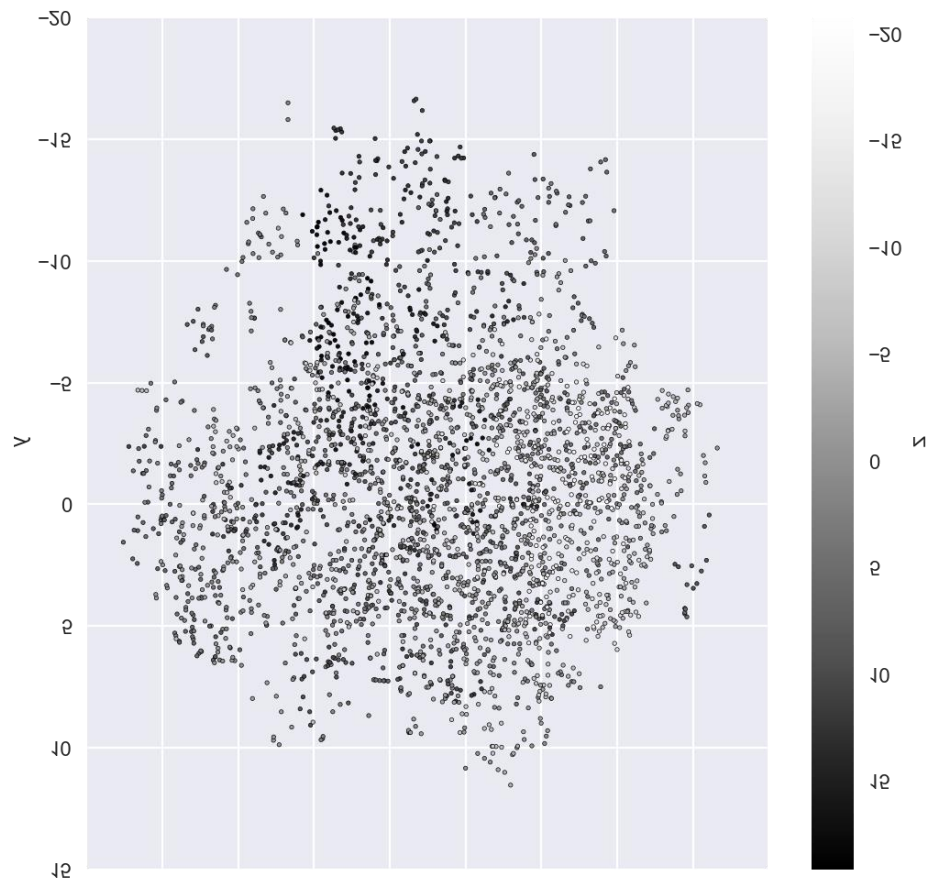
        for word in mahabharata2vec.wv.vocab
    ]
],
columns=["word", "x", "y", "z"]
)

```

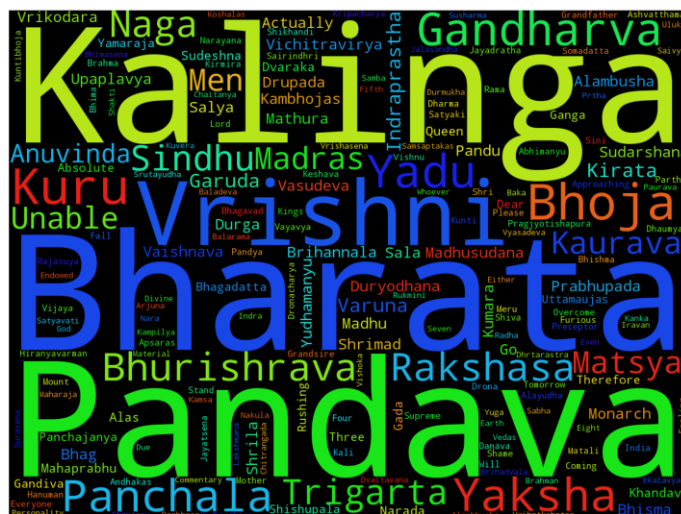
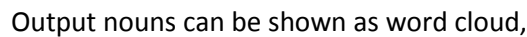
Coordinates of the first few words are as follows,

	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.



- ```
tagged = nltk.pos_tag(tokens)
nouns = [word for word,pos in tagged if (pos == 'NNP') or (pos == 'NNPS')]
```





- 6) For further analysis and to answer the problem statement we use cosine similarity, to assess similarities between 2 word vectors, to answer similarity questions on the 3<sup>rd</sup> word vector.

Firstly, let's see all the similar words, when given a character's name. Let's see all the similar words related to Arjuna, 83% similar word is Partha, which is another name of Arjuna.

```
mahabharata2vec.most_similar("Arjuna")
```

Output:

```
[(u'Partha', 0.8320261240005493),
 (u'Daruka', 0.8209834694862366),
 (u'Satyaki', 0.7644478678703308),
 (u'Pradyumna', 0.7604053020477295),
 (u'Karna', 0.759215772151947),
 (u'Bhima', 0.7575901746749878),
 (u'Kuvera', 0.7552096843719482),
 (u'Shishupala', 0.7415941953659058),
 (u'Subhadra', 0.738818883895874),
 (u'wanting', 0.7378615736961365)]
```

Second part of the problem statement of answering relationship questions is done with the help of Cosine Similarity. Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

```
def nearest_similarity_cosmul(start1, end1, end2):
 similarities = mahabharata2vec.most_similar_cosmul(
 positive=[end2, start1],
 negative=[end1]
)
 start2 = similarities[0][0]
 print("{start1} is related to {end1}, as {start2} is related to {end2}".format(**locals()))
 return start2
```

When the system is asked to provide answer for the below question, i.e. Dhritarastra and Pandu (brothers) and what is for Nakula, system provides Sahadeva, who is brother of Nakula.

```
nearest_similarity_cosmul("Dhritarastra", "Pandus", "Nakula")
```

Output: Dhritarastra is related to Pandu, as Sahadeva is related to Nakula

## Refinement

Refinement was done in 3 stages,

- 1) **word2vec model creation:** word2vec has main 7 parameters, viz. *seed*, *workers*, *size*, *min\_count*, *window* and *sample*. In any dataset high frequency words often provide little information. Words with frequency above a certain threshold may be subsampled to increase training speed, this can be done with the help of *min\_count*. Quality of word embedding increases with higher dimensionality. But after reaching some point, marginal gain will diminish. Typically, the dimensionality of the vectors is set to be between 100 and 1,000. The size of the context window determines how many words before and after a given word would be included as context words of

the given word. According to the authors' note, the recommended value is 5 to 10. As Mahabharata data set has lengthy sentences, I have chosen context window size as 25.

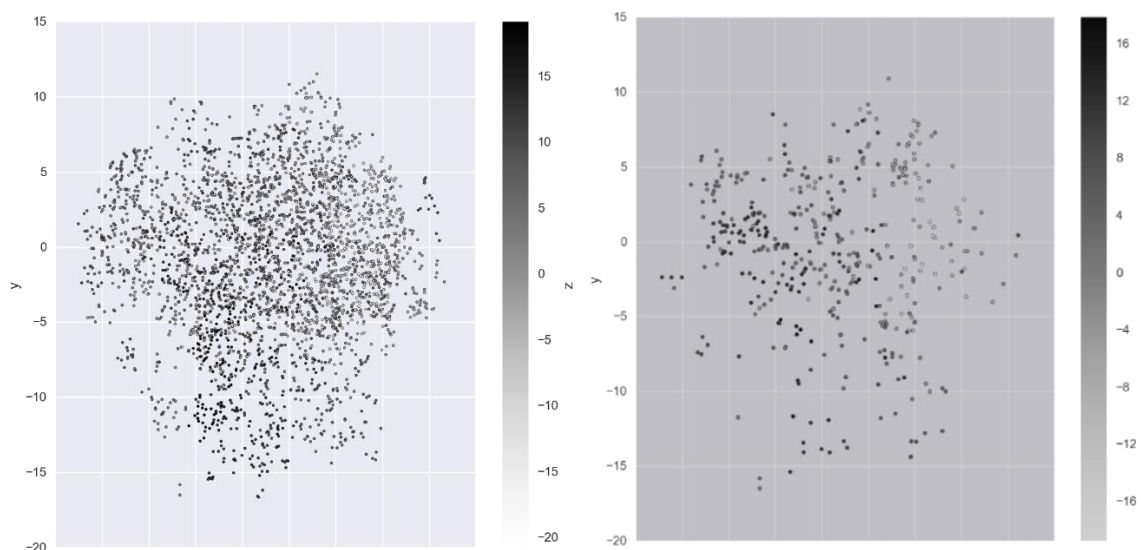
```
mahabharata2vec = w2v.Word2Vec(
 sg=1,
 seed=seed,
 workers=num_workers,
 size=num_features,
 min_count=min_word_count,
 window=context_size,
 sample=downsampling
)
```

- 2) **t-SNE parameters:** t-SNE has 3 important parameters, viz. *n\_iter*, *n\_components* and *perplexity*. Perplexity is a measure for information that is defined as 2 to the power of the Shannon entropy. The perplexity of a fair die with k sides is equal to k. In t-SNE, the perplexity may be viewed as a knob that sets the number of effective nearest neighbors. It is comparable with the number of nearest neighbors' k that is employed in many manifold learners. The performance of t-SNE is fairly robust under different settings of the perplexity. The most appropriate value depends on the density of your data. Loosely speaking, one could say that a larger / denser dataset requires a larger perplexity. Typical values for the perplexity range between 5 and 50.

Ref: <http://distill.pub/2016/misread-tsne/>

```
tsne =sklearn.manifold.TSNE(n_components=3,perplexity=50.0,n_iter=9000,random_state=0)
```

- 3) Next step of refinement is done by extracting only proper nouns from the t-SNE output and using it for further analysis, as our main aim is relationship extraction. This can be seen by the number of points reduction in the below plot (Left is t-SNE for all word and right is t-SNE only for Proper Noun)



## Results

### ***Model Evaluation and Validation***

During development, a validation set was used to evaluate the model. The final architecture and hyperparameters were chosen because they performed the best among the tried combinations. For a complete description of the final model and the training process, refer to ***Implementation*** along with the following list:

- Corpus is 1704913 characters long.
- Size or number of dimensions of word2vec was 500.
- Minimum word count for word2vec was chosen as 7.
- Window size for analyzing each word in word2vec was chosen as 25, based on the length of normal sentences in Mahabharata.
- Word2Vec vocabulary length was 3271.
- Word2vec was trained on training on 1467215 raw words (1013320 effective words).
- 500 dimensions was reduced to 3 dimensions using t-SNE.
- Based on experimentation perplexity was chosen as 15.
- t-SNE was trained for 20000 iterations to get an effective model.
- Total training time of the entire model takes up to 10 mins to crunch the epic Mahabharata.

To evaluate the model, I have compiled 144 unique relationship facts, such as father-son, mother-son, siblings and spouse. These 144 unique relationships are used in different combinations to generate 1551 relations. For example,

- Finding Son given Father, based on an example Son-Father relationship
- Finding Father given Son, based on an example Father-Son relationship
- Finding Son given Mother, based on an example Son-Mother relationship
- Finding Mother given Son, based on an example Mother-Son relationship
- Finding Husband given Wife, based on an example Husband-Wife relationship
- Finding Wife given Husband, based on an example Wife- Husband relationship
- Finding Sibling, based on a Sibling example.

The output results are compared with the correct relationships to calculate the accuracy of the model and it came out to be 13.15% which may seem low, but based on the complexity of Mahabharata and the literal translations of the conversation between characters from Sanskrit to English makes it hard for t-SNE to find relationships.

### ***Justification***

The generated model was tested on a set of 1551 relationship question, out of which on an average 200 relations were correctly predicted, which comes up to 13% accuracy. As explained in the above section, this may seem low, but based on the complexity of Mahabharata and the literal translations of the conversation between characters from Sanskrit to English makes it hard for t-SNE to find relationships.

## Conclusion

### *Reflection*

The process used for this project can be summarized using the following steps:

- 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, cosine similarity is used to assess similarities between 2 word vectors, to answer similarity questions on the 3<sup>rd</sup> word vector.
- 6) Calculate accuracy.

Interesting part of this project was to learn how word2vec works, how it evaluates the sentences, but the most interesting and difficult part was to understand how t-SNE works and how to set all the parameters correctly to get a perfect clustering. The designed model is a general model which can be utilized for relationship extraction for any given corpus of text, this is the greatest advantage of Machine learning, as you don't need to write a different set of code for a different set of input data, i.e. just change the input data and you are good to go. But this was a great challenge to analyze one of the complicated ancient scripts.

### *Improvement*

As seen from Model Evaluation section, accuracy in predicting relations is as low as 13%. This need to be increased by creating an extremely good dataset, thereby helping word2vec and t-SNE to perform better.