Analysis of Dota 2 as a Language using Word Embeddings

Samarth Desai

*Da-iict, Near Indroda Circle,Gandhinagar-382007.*

[201301456@daiict.ac.in](mailto:201301401@daiict.ac.in)

*Prof. Asim Banerjee,Da-iict*

***Abstract* – *Dota 2 is a very complex game, a single 40 min game encompassing countless choices and the many possible paths related to those choices leading to one of two possible outcomes.Players are constantly trying to outplay and outsmart each other and the mind games begin even before the game starts.More often than not among players of the same skill caliber the draft is what tilts the game to favour one side, and choosing heroes to play from a pool of 112 while keeping in mind your current picks and your opponents picks is a daunting task.We aim to visualize dota 2 as a language and view it solely on the basis of the characteristics that emerge due to it being looked upon as a language.We assume each team to be a sentence where a hero represents a word and a team of 5 represents a valid sentence , and with a sufficiently large database we aim to create word embeddings for each hero which can then be used to simulate or generate possible choices of teams given any situation.We also analyze the merits and demerits of viewing dota 2 as so.***

***Keywords*** – **Word Embeddings, Dota 2, Language Analysis, Machine Learning**

1. Introduction

Drafting is the core of any strategy while going into any game of Dota 2. The way that different heroes interact with each other creates countless different possibilities and it is the job of the drafter (The Captain) and the players to know how they gel with or fare against each other at different stages of the game. More often than not there are clear cut counters to certain heroes that can be core to effectively shutting down many strategies and due to the vast number of variables involved it is sometimes very difficult to come up with the right pick, processing the different variables presented up till then while including the individual player skills. We attempted to develop a method of suggesting picks, bans and counterpicks while removing the uncertainty of the variables involved, while looking at Dota 2 objectively as a language and analysing hero picks in situations only on the basis of how the heroes have interacted with one another in the past. We view every team of a match as a valid sentence and each hero as a word belonging to that valid sentence.We operate on the teams (sentences) and create word vectors which operate under the distributional hypothesis.

1. Hero Vectors(Word Vectors)

.To analyze dota 2 as a language we create hero vectors and perform queries on them. Hero vectors are basically n-dimensional vectors (For adequate and efficient representation we have taken n to be 100) where each dimension or variable denotes a vector index. These hero vectors operate under a set of assumptions about the language (in our case Dota 2) known as the “Distributional Hypothesis”.

1. *The Distributional Hypothesis*

The Distributional Hypothesis states that “there is a correlation between distributional similarity and meaning similarity, which allows us to utilize the former to estimate the latter” [1].

Simply put “words which are similar in meaning occur in similar contexts” [2].

For example (Shadow Fiend) and Storm Spirit are heroes played in the same position (Mid) and their scaling abilities are very similar so according to the distributional hypothesis they should more or less be interchangeable in teams.

ShadowFiend-: [0.006069395691156387,0.003811658127233386,0.006546148099005222,0.00031248945742845535,-0.00879650376737117….]

StormSpirit-: [-0.06813218,-0.00902375,0.10162564,...]

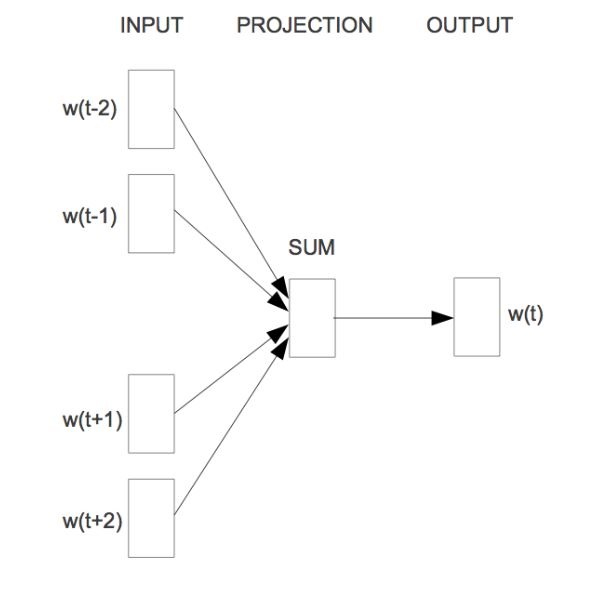
These are the vector representations of Storm Spirit and Shadow Fiend. As it turns out the cosine similarity between the two vectors is 0.9170020818710327 which indicates a strong similarity between the heroes Shadow Fiend and Storm Spirit which Dota 2 experience dictates is common sense but helps us to determine that the application of the distributional hypothesis to Dota 2 as a language yields meaningful results.

We use the Word2Vec model available in the gensim library of python to create our Hero Vectors. Word embeddings are relatively new and the word2vec model is the most widely accepted and used model to generate word vectors.

1. *The Word2Vec Model*

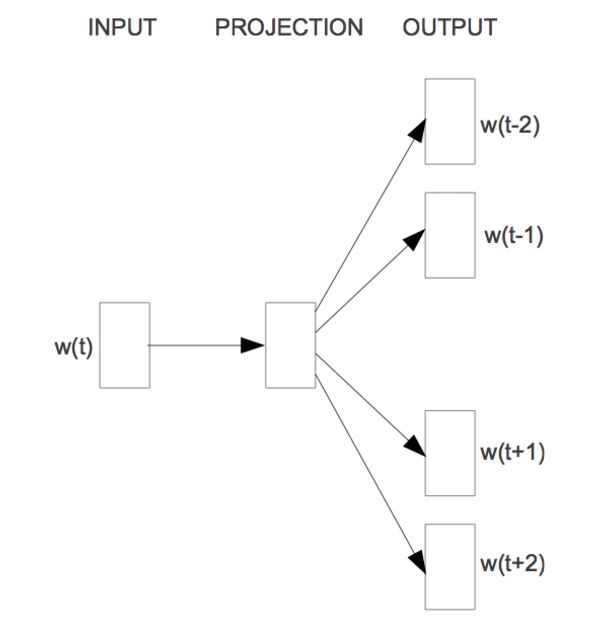
The word2vec model involves the uses of the following architecture-:

1. *CBOW (Continuous Bag of Words)*-: The word2vec model receives a continuous bag of words i.e a window of n words before and after the word wt which it then uses to predict the word wt . It is known as a continuous bag of words because of the use of continuous representations whose order is of no importance.



*Fig-1 CBOW Model*

1. *Skip-Gram* -: Unlike Cbow which uses the neighbouring words to predict the center word, Skip-Gram uses the center word to predict the neighbouring words.

**

*Fig-2 Skip-Gram Model*

1. *DataSet*

The data set we used to obtain the hero vectors was created by parsing the massive 450 gb json data dump obtained from “http://academictorrents.com/collection/yasp-data-dumps’’.

The data was parsed in python using a json streamer where teams were saved if they passed the criteria. The criteria being the matches must have been played after and before a certain patch (6.83 to 6.88) and the average Match Making Rank (MMR) must be above 5k. These constraints were put in so as to obtain a sufficiently large relevant quality data set, quality as in player quality and relevance as in the currentness of the meta as dota 2 shifts as patch changes occur. We ended up with 100000 teams (approx.) over 50 k matches. The constraint of the patch was also due to the introduction of talent trees in patch 7.00 which changed the idea of hero stereotypes and so led to a lot of experimentation leading to the data being unreliable. The period between 6.83 and 6.88 being the most stable period was chosen as the period for observation for relevant data.

The data was then fed to the word2vec module as a corpus instead of the english language. The domain size was 100, window size as 5 because we have a team of 5 heroes and so a sentence will only have 5 words, the train\_epoch was taken as 100 with alpha the learning rate being -0.02 [These are all self explanatory Doc2Vec parameters.]

III.OBSERVATIONS AND ANALYSIS

We experimented with different queries to understand the extent to which the distributional hypothesis was being confirmed. The following is a sample basic query for finding a hero (word ) similar to a certain hero and to find a hero that acts as if that heroes opposite. The input is in format of a string containing 1 to 5 indices. And we can query for similarities to the hero, as well as a negative query results in a hero which is the most drastically different from the current one, in effect a hero opposite to the current one in many cases. Below are a few examples-:

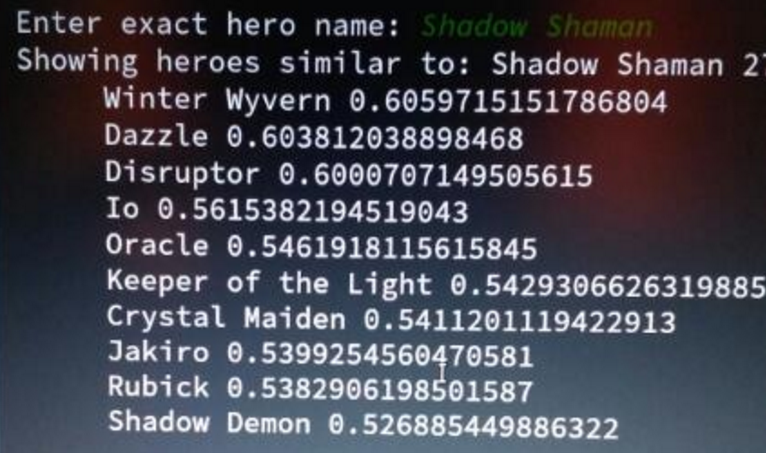


Fig.3 Querying Shadow Shaman

Shadow Shaman is a popular support and intelligence type hero and the heroes similar to it are also popular support and intelligence type heroes.

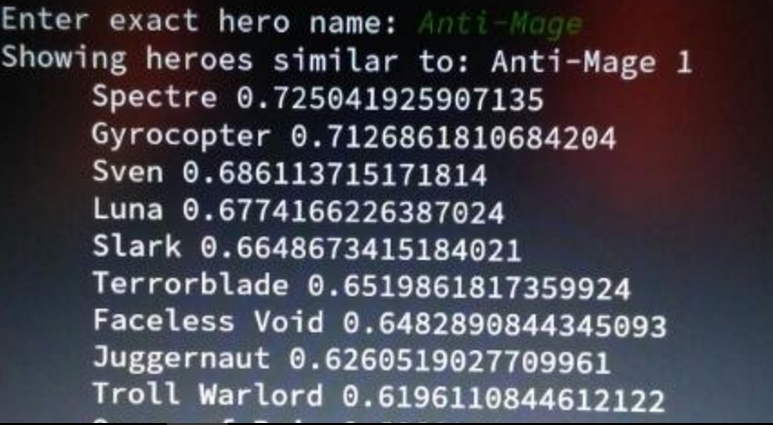


Fig.4 Querying Anti-mage

The same is seen for anti-mage which is a hard carry agility type hero and has the same type of heroes similar to it.

These are some of the cases where the model was correctly able to identify similar heroes as we the dota community see them but in some cases that does not happen. For such cases we believe that as the hero is played in many different positions and ways the system is not able to pick up on the clear similarities, because maybe too many or too few similarities exist. We chalked these failures to the trend being lost in translation. The model works well in most cases, but of these cases which cause confusion for the model; some have been mentioned as a point of discussion as to why the model fails.

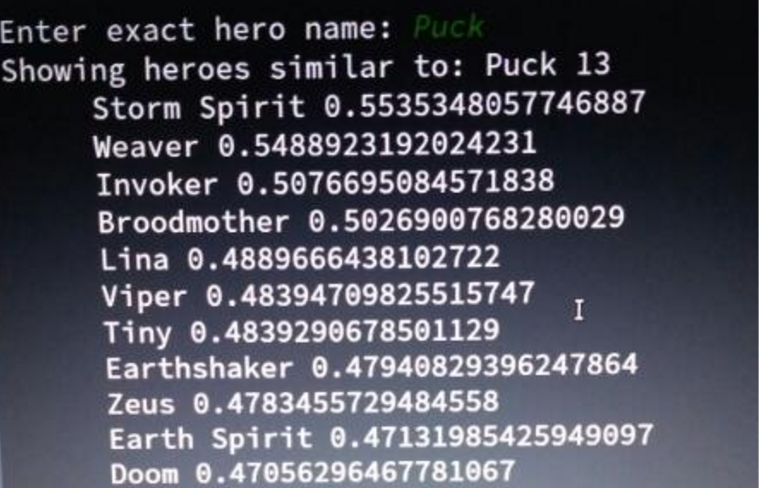


Fig.5 Querying Puck

Puck is a primarily mid hero and a nuker but due to its varying play styles and positions there are many similar heroes so the model may be getting confused and hence we aren’t able to get a proper hero similar to puck. The same confusion happens below for the hero riki and we believe in this case it is because in itself riki is quite a unique hero and so the model fails to pick up on the similarities or dissimilarities on the basis of the corpus provided.

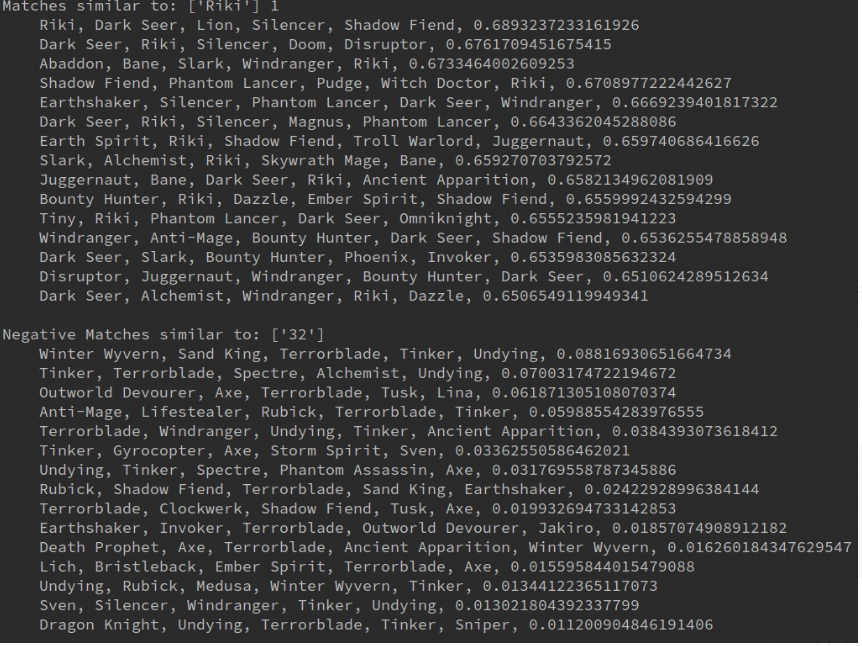
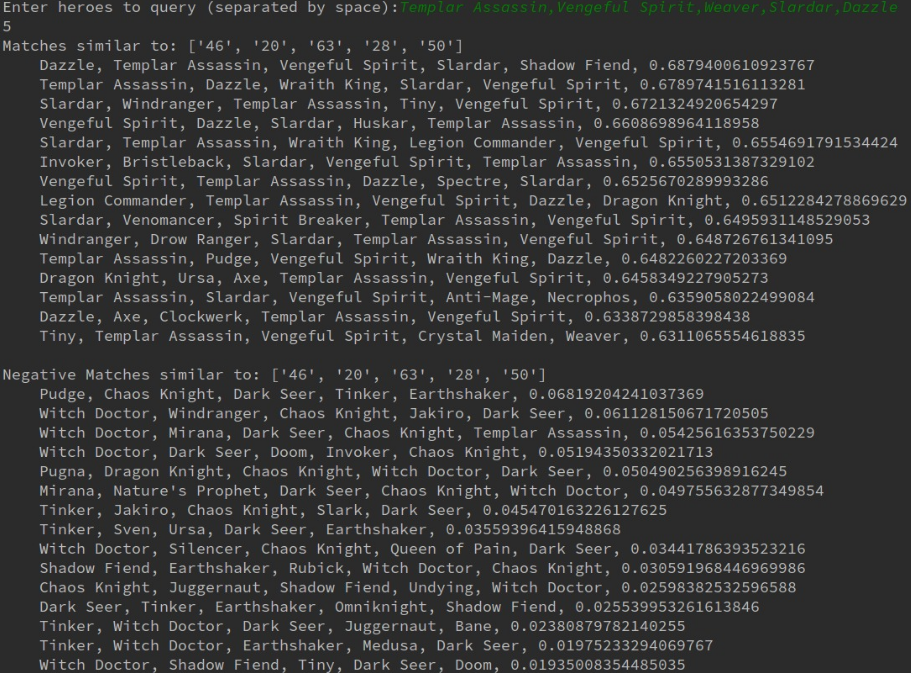


Fig.6 Querying Riki and its negative

Fig 7. Querying a team 

Querying teams gives us possible team scenarios which function in the same way or possess the same dynamics i.e what the model determines it to be. And similarly negative queries give us possible team scenarios which are the most different from the string we request . Negative queries serve as a counter picking mechanism and the similar queries function

for similar scenario picks. Amazingly based on the trends the model is able to identify strategies (strats) and the similar suggestions consist of the same strategies or similar strategies employed albeit with some junk mixed in. The model is able to counter unique, less used strategies just on the basis of the hero properties, characteristics it depicts and we can effectively use it to suggest a pick/ban at any phase of the picking/banning.

The team above Templar-Assassin, Vengeful-Spirit, Weaver ,Slardar and Dazzle is a popular strategy relying on armour reduction and we see that the suggestions include teams that functions on the same premise or focus on physical damage. In the same way this draft is weak against lockdown and attack miss or being unable to hit, illusions and magic damage which is shown in its negative matches. Queries were performed on a wide variety of strats and as shown above the

suggestions vary on how adequate they are but in most cases there is adequate representation of the team characteristics in the suggestions and adequate counter representation. There are anomalies where it doesn't work or doesn’t provide adequate suggestions and we believe its is due to not much data existing for that field or too much garbage data existing which ends up causing garbage values as results. Moreover these results are not complete garbage and so it may be that we as experienced as we are are not experienced enough to understand the significance of those results. The queries also work for groups of heroes as such but the string size cannot cross 5 because a team consists of only 5 heroes.

IV. CONCLUSIONS

This model is able to provide adequate scenarios for picking and counter-picking and helps a lot in the picking phase. We have tried simulating it in our matches and we believe for the somewhat less experienced players (less than 6000 hrs) it is able to provide sufficient insight coupled with their experience to provide a better gaming experience as well as help in tournaments and in increasing MMR.

V. FUTURE SCOPES

This model is a framework and as we have applied it to Dota 2 in the same way it can be applied to any collection of data to make word embeddings and compute on it using that. For example, given a corpus of collected synopsis (or complete books for that matter) of different books we should be able to suggest similar books to a query request. The same can be done for movies, songs, poems or any literary work for that matter.

Acknowledgment

We would like to acknowledge and thank Sairam Pillai, VGEC for providing his informative thoughts on both the model and Dota 2.

References

1. http://sebastianruder.com/word-embeddings-1/
2. https://www.diva-portal.org/smash/get/diva2:1041938/FULLTEXT01.pdf
3. http://federicov.github.io/word-embeddings-and-dota2.html
4. https://www.tensorflow.org/tutorials/word2vec
5. https://radimrehurek.com/gensim/models/word2vec.html
6. https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
7. http://sebastianruder.com/secret-word2vec/