Word2Vec & TransE

Abstract

This report contains 2 main sections: Word2Vec and TransE. Word2Vec is a classic model of word embedding, it models the word similarity based on the contextual information between words. TransE is an energy-based model that produces knowledge base embeddings. It models relationships by interpreting them as translations operating on the low-dimensional embeddings of the entities[1]. The task of the assignment is to implement Word2Vec and TransE based on Text8[2] and Wikidata[3] data sets respectively in a given framework, and using specific examples to understand the meaning of word vectors and entity/relationship vectors.

For the Word2Vec model, The best combination among the values we tested are – vector size: 250, window size: 35, minimum count: 10 and max epoch: 16. The best Spearman's Rank correlation coefficient on the WordSim353 is 0.747125.

After analyzing the impact of embed dimensions, regularization methods, margins and learning rates on the TransE model, we choose “Norm: L2, Embed Dimension: 200, Learning Rate: 1.0, Margin: 1.75 with SGD optimizer. The SGD has momentum 0.9 and weight decay 1e-5” as our final model. For the given questions: <*USA, capital, ?*> and <*USA, ?. North America*>, our final model has successfully understood the “capital”, it could infer capitals of different countries but fail to point out the capital for the United States of America, our given head entity. As for the second question, the ideal answer would be “USA is part of North America”, however, our model fails to calculate relationships that are more closely to “part of”. The inferences that are related to “part of” would be: “applies to jurisdiction”, “located in the administrative territorial entity” and “country”. As “country” is a subproperty of “located in the administrative territorial entity”, we could assume that the model still has not finished learning. In addition, as mentioned in the experiment observations, the political/region related information can be learnt at an early stage of training(with small max epoch). With large epoch amounts, the geographically related information becomes less important, and the model starts to pick up and remember social/society related connections.

1 Word2Vec

A Word2Vec implementation based on gensim has been provided. The task is to train the given model on the Text8 corpus, and test on WordSim353 dataset.

Gensim is an open-source library for unsupervised topic modeling, document indexing, retrieval by similarity, and other natural language processing functionalities, using modern statistical machine learning[4].

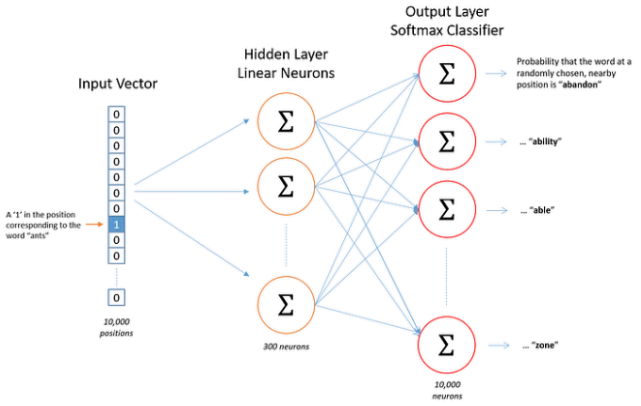
	Word 1	Word 2	Human (mean)
1	love	sex	6.77
2	tiger	cat	7.35
3	tiger	tiger	10.00
4	book	paper	7.46
5	computer	keyboard	7.62
6	computer	internet	7.58
7	plane	car	5.77

WordSim353 is a test collection for measuring word similarity or relatedness, developed and maintained by E. Gabrilovich[5]. The left image is an example of the information contained in this

dataset. The Human(mean) column indicates how similar the pair is with human review.

1.1 Word2Vec Model

It is a shallow 2-layer neural network. The input contains all documents in our training dataset, and these texts are represented in 1-hot encoding of words. The Neurons' amount in the hidden layer is equal to the length of the embeddings we want. That is, if we want all our words to be vectors of length 300, then the hidden layer will contain 300 neurons.



Understanding the neural network training of Word2Vec model

The output layer contains probabilities for a target word (given an input to the model, what word is expected) given a particular input. At the end of the training process, the hidden weights are treated as the word embedding. Intuitively, this can be thought of as each word having a set of n weights (300 considering the example above) "weighing" their different characteristics (an analogy we used earlier)[6].

Based on the document of gensim Word2Vec model, we will, in this section, investigate how vector size, window size, minimum word count and max epochs affect the learning quality.

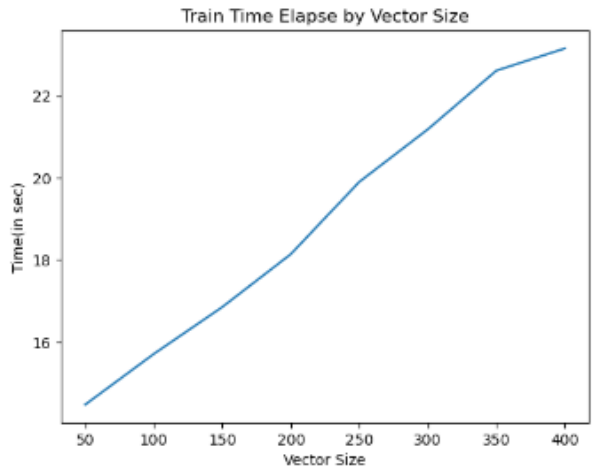
1.2 Vector Size

As mentioned in Section 1.1, the vector size indicates the length of the vector to represent a word. We tested vector sizes: [50, 100, 150, 200, 250, 300, 350, 400].

Both the window size and the minimum count is set to 10. The max epoch is set to 5, which is the default value in the gensim Word2Vec model definition.

The size of the vector can be selected according to the corpus size and the type of project. The entire corpus is scanned, and the vector creation process is performed by determining which words the target word occurs with more often. In this way, the semantic closeness of the words to each other is also revealed[7]. The vector size for Word2Vec model commonly falls between 300-400, and according to research, the quality for vector representations improves as you increase the vector size until you reach 300 dimensions. After 300 dimensions, the quality of vectors starts to decrease. However, this 300 value could also change based on the training dataset amount and the computation power.

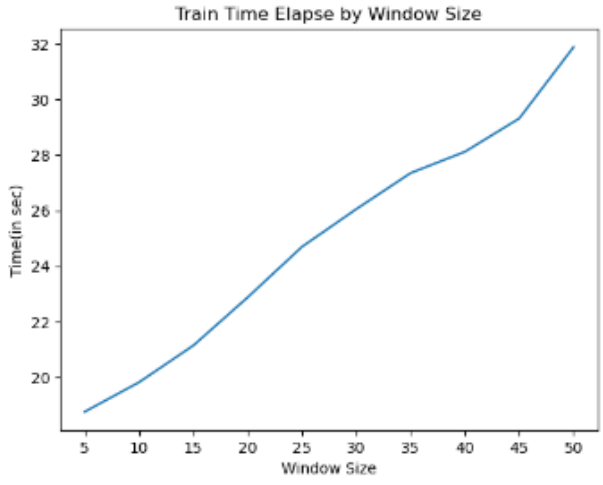
Below are the diagrams of train time and the score on the test set(WordSim353). The score is Spearman's Rank correlation coefficient, a common coefficient to summarize the strength and direction (negative or positive) of a relationship between two variables.

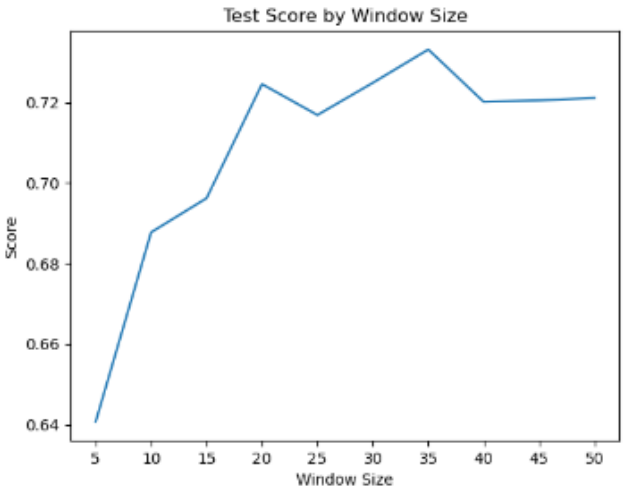


As can be seen from the diagrams, the greater the vector size is, the longer the training will take. The model reaches the highest Spearman's Rank correlation coefficient at the vector size of 250. Then the performance starts to decrease as the vector size gets greater. In the later experiments, we will therefore use vector size 250 for testing other parameters.

1.3 Window Size

It is the number of neighboring words that will directly affect the vector calculations of the target expression. Usually, larger window size is recommended for more topic-specific corpora, whereas a smaller one is more fit for input texts concerning more general topics and has also been proved to encode more syntactical information in the Skip-Gram variation of word2vec[8].





Same as the vector size, larger window size means more training time(complexity). The performance of the model generally follows an upward trend as the window size increases. When the window size reaches 20, small fluctuations happen on the score curve, reaches the highest score, drops down and

then tends to stabilize. The choice of the window size mainly depends on the material you are using for training. If the window size of 2 can capture the context of a word, but 5 is chosen, it will decrease the quality of the learnt model, and vice versa. The best score is when the window size is set to 35.

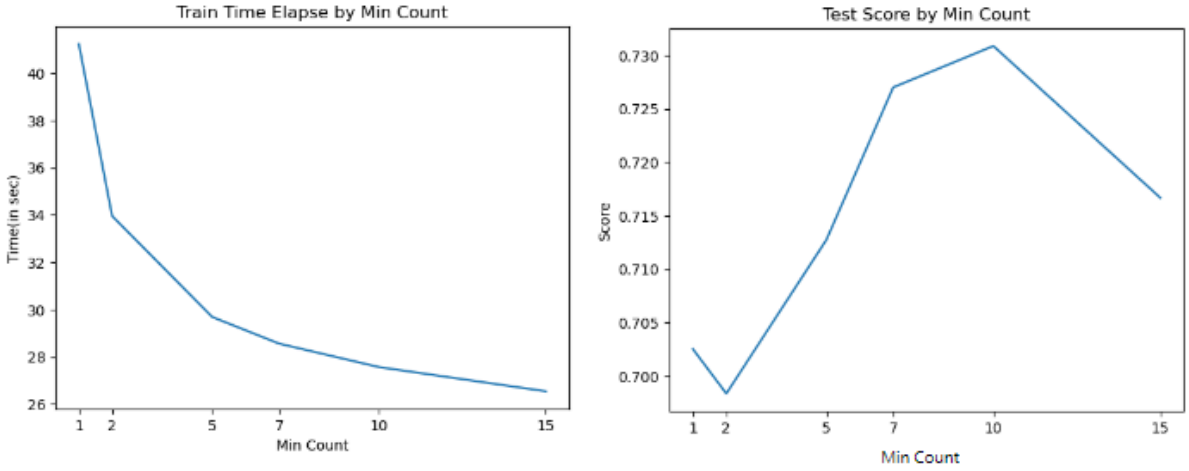
1.4 Min Count

Minimum count is the minimum number of occurrences of the target word in the corpus. Especially for very large corpuses, keeping this limit high increases the success even more. However, it would be more accurate to keep it small for small datasets[7]. Words below the minimum count frequency are dropped before training occurs.

Minimum count values we tested are: [1, 2, 5, 7, 10, 15]. The diagrams of model performances are below.

As the minimum count increases, the training time decreases at the same time, as more words are dropped before the actual training happens.

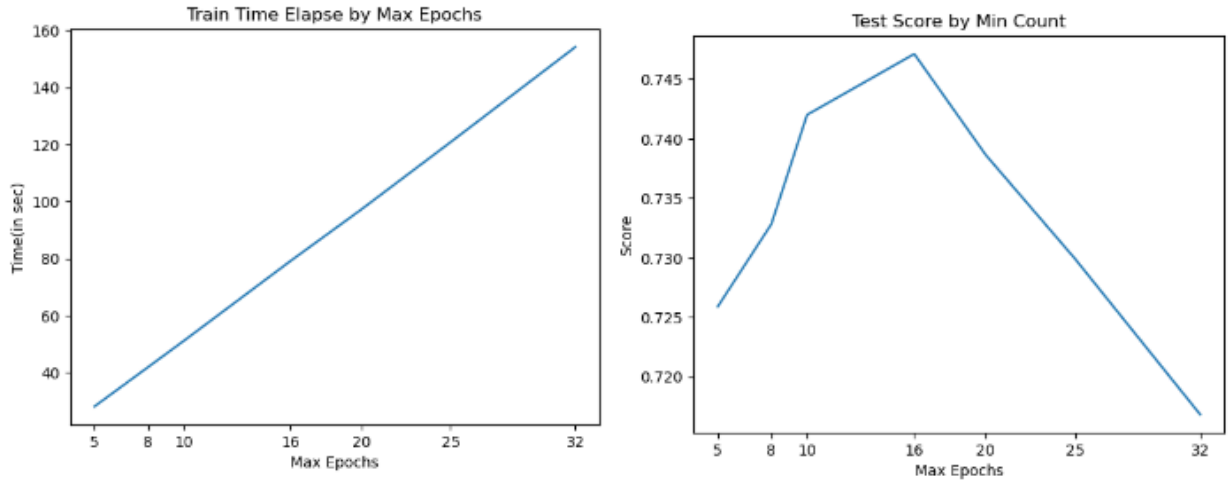
Minimum count equals to 1 will create a much larger-in-memory model, with a lot of low-value, low-quality infrequent words, that takes longer to train – a loss on almost every dimension, except the completist desire to have a vector-for-every-seen-word. On the other hand, a too large minimum count will also lead to a problem. Ignoring too many words is a defensible choice – and most deployed systems need to be tolerant of novel unknown words.



The performance diagram also proves the above conclusion. The Spearman's Rank correlation coefficient is quite low when the minimum count is pretty small, since there are a lot of rare words in the vocabulary. The coefficient keeps climbing up until the minimum count reaches 10, then immediately follows a sharp downward trend as the minimum count gets greater. Too many words have been filtered out with a high minimum count(=15), and potentially leaving too many words unavailable at the inference stage.

1.5 Max Epochs

The hyperparameters for the Word2Vec model now are – window size: 35, vector size: 250, minimum count: 10. We will increase the training epochs and see if the model performance could be improved.



Originally the epoch amounts we tested are: [5, 8, 16, 25, 32]. After drawing the Spearman's Rank correlation coefficient diagram, we noticed that there exists a sharp downward trend when the epoch amount is above 16, We made an assumption that the max epoch amount that could lead to the best model solution should be in the range of [16, 25]. Therefore, we in addition tested a model with 20 epochs. The new diagram suggests that the best performance of the model could be produced when the epoch amount is set to 16. Further increasing the training epoch may result in overfitting, and the performance quality on the test set will be reduced.

1.5 Summary

We investigated the impact of 4 tunable hyperparameters: vector size, window size, minimum count and the max epoch. The best combination among the values we tested are – vector size: 250, window size: 35, minimum count: 10 and max epoch: 16. The best Spearman's Rank correlation coefficient on the WordSim353 is 0.747125.

For 4 parameters we tuned, they all improve the model performance as their values grow up, until reaching a certain optimal solution. Afterwards, the increase on the parameter values will bring down the model performance. Although there are researchers specifying certain optimal values for the Word2Vec model to have the best learning results, it is still important to tune these parameters since the data and the task may differ.

2 TransE

To represent knowledge, knowledge graphs are commonly used. Usually, we use a triple (head, relation, tail) to represent knowledge. Here, head and tail are entities. For example, (sky tree, location, Tokyo). Our task here, link prediction, is a typical downstream task for knowledge representation.

In this report, we will focus on a preliminary method for entity prediction task known as TransE, an energy based model that learns continuous, low-dimensional embedding of entities and relationships by minimizing a margin-based pair-wise ranking loss[9].

Algorithm 1 Learning TransE

input Training set $S = \{(h, \ell, t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\boldsymbol{\ell} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$

  $\boldsymbol{\ell} \leftarrow \boldsymbol{\ell} / \|\boldsymbol{\ell}\|$ for each $\ell \in L$

  $\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

loop

  $\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$

  $S_{batch} \leftarrow$ sample(S, b) // sample a minibatch of size b

  $T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets

  for $(h, \ell, t) \in S_{batch}$ **do**

    $(h', \ell, t') \leftarrow$ sample$(S'_{(h,\ell,t)})$ // sample a corrupted triplet

    $T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell, t'))\}$

  end for

  Update embeddings w.r.t. $\sum_{((h,\ell,t),(h',\ell,t')) \in T_{batch}} \nabla[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h}' + \boldsymbol{\ell}, \boldsymbol{t}')]_+$

end loop

The code implementation on score calculation and loss calculation is inspired by a github project “TransE-PyTorch”[10]. For the loss calculation, we scripted based on the source code and documentation of the torch MarginRankingLoss[11]. Note that when calculating positive and negative scores, we need to normalize the embeddings with the L2 norm. And the final score(distance) is calculated based on whether L1 or L2 norm is specified in the model configuration stage.

Below attached code snippets for score and loss calculation.

```
def _calc(self, h, t, r):
    # TO DO: implement score function
    # Hint: you can use F.normalize and torch.norm functions
    if self.norm_flag: # normalize embeddings with l2 norm
        h = F.normalize(h, p=2, dim=-1)
        t = F.normalize(t, p=2, dim=-1)
        r = F.normalize(r, p=2, dim=-1)
    score = h + r - t #[2, batch_size]
    return torch.norm(score, p=self.p_norm, dim=-1)
```

```
def loss(self, pos_score, neg_score):
    # TO DO: implement loss function
    # Hint: consider margin

    # the loss is calculated by taking the larger value among 0 and (margin+pos_score-neg_score)
    # which is basically a ReLU, so we will directly use ReLU
    if self.margin_flag:
        return F.relu(self.margin + (pos_score - neg_score).mean(), inplace=True)
    return F.relu((pos_score - neg_score).mean(), inplace=True)
```

In the following experiments, we will investigate how hyperparameters, such as epochs, learning rates, embed dimensions, margins and regularization methods can affect the model performance. In addition, we will check if using Adam as the training optimizer could improve the TransE model. Based on the assignment requirements, we take pairs (Q30, P36) and (Q30, Q49)[3] as examples to analyze the model learning quality.

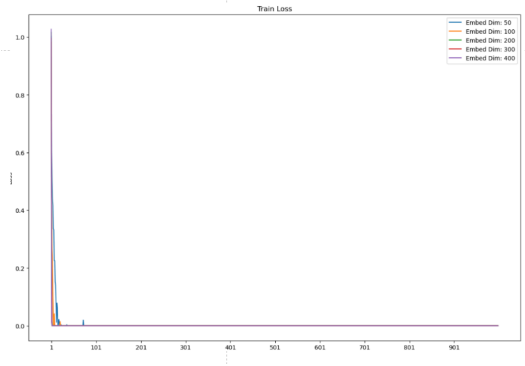
2.1 Embed Dimension and L1/L2 Regularization

We will start with in total 100 batches per epoch, 10 epochs for training, margin 1.0 and learning rate 1.0. The final model will have an early stop mechanism, but this mechanism will not be in the parameter tune stage . SGD without momentum or decay is the model optimizer.

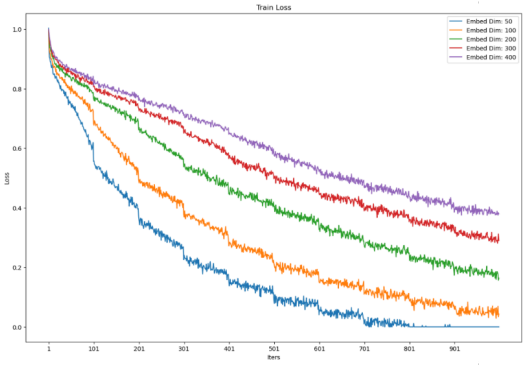
An embedding is a vector representation of a word/relationship in a high-dimensional space. The dimensionality of word embedding is a crucial factor in determining the quality and effectiveness of the embedding. The dimensionality of the word embedding represents the total number of features that are encoded in the vector representation.

Deciding on the appropriate dimensionality for a word embedding depends on several factors, such as the size and nature of the dataset we are using, the specific NLP task we are working on, and the computational resources available to us[12].

Norm: L1 Train Loss per Iter



Norm: L2 Train Loss per Iter



We use the pair [USA(Q30), Capital(P36), Washington D.C.(Q61)] to validate the model performances.

Norm: L1 Q30+P36(USA + Captital)

Embed Dim = 50		Embed Dim = 100		Embed Dim = 200		Embed Dim = 300		Embed Dim = 400	
Q45	Portugal	Q30	United States of America	Q30	United States of America	Q30	United States of America	Q30	United States of America
Q17	Japan	Q142	France	Q183	Germany	Q38	Italy	Q145	United Kingdom
Q31	Belgium	Q148	People's Republic of China	Q148	People's Republic of China	Q145	United Kingdom	Q159	Russia
Q38	Italy	Q145	United Kingdom	Q145	United Kingdom	Q183	Germany	Q16	Canada
Q159	Russia	Q183	Germany	Q38	Italy	Q142	France	Q212	Ukraine
Q36	Poland	Q38	Italy	Q142	France	Q1346044	Jan Weenix	Q29	Spain

Norm: L2 Q30+P36(USA + Captital)

Embed Dim = 50		Embed Dim = 100		Embed Dim = 200		Embed Dim = 300		Embed Dim = 400	
Q90	Paris	Q90	Paris	Q90	Paris	Q220	Rome	Q16666	Nanjing
Q60	New York City	Q656	Saint Petersburg	Q515	city	Q495	Turin	Q3711	Belgrade
Q65	Los Angeles	Q60	New York City	Q60	New York City	Q5826	Xi'an	Q2280	Minsk
Q656	Saint Petersburg	Q65	Los Angeles	Q656	Saint Petersburg	Q216	Vilnius	Q8678	Rio de Janeiro
Q220	Rome	Q220	Rome	Q220	Rome	Q60	New York City	Q216	Vilnius
Q84	London	Q84	London	Q65	Los Angeles	Q2280	Minsk	Q406	Istanbul

Norm: L2 Q61-Q30(USA -WashingtonD.C)

Embed Dim = 50		Embed Dim = 100		Embed Dim = 200		Embed Dim = 300		Embed Dim = 400	
P1038	relative	P463	member of	P2388	office held by head of the organization	P1071	location of creation	P1074	fictional or mythical analog of
P400	platform	P289	vessel class	P287	designed by	P2643	Carnegie Classification of Institutions of Higher Education	P86	composer
P37	official language	P407	language of work or name	P607	conflict	P1552	has characteristic	P1038	relative
P748	appointed by	P366	has use	P530	diplomatic relation	P69	educated at	P306	operating system
P1889	different from	P1308	officeholder	P641	sport	P840	narrative location	P184	doctoral advisor
P26	spouse	P85	anthem	P118	league	P57	director	P417	patron saint

Norm: L2 Q61-Q30(USA -WashingtonD.C)

Embed Dim = 50		Embed Dim = 100		Embed Dim = 200		Embed Dim = 300		Embed Dim = 400	
P36	capital	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity
P150	contains the administrative territorial entity	P36	capital	P36	capital	P36	capital	P463	member of
P2962	title of chess person	P1383	contains settlement	P421	located in time zone	P277	programmed in	P1411	nominated for
P1622	driving side	P735	given name	P37	official language	P463	member of	P134	/
P735	given name	P1622	driving side	P286	head coach	P272	production company	P735	given name
P509	cause of death	P1303	instrument	P463	member of	P735	given name	P734	family name

Blue: USA related; Yellow: Capital related; Green: Capital and administrative territorial related.

Observations:

1. The training loss gets to 0 at an early stage of training with L1 norm; Although there exists extreme small fluctuations before the first epoch finishes, the loss is still close to 0. That is, the model is not learning much.

   The diagrams of tail entity and relationship predictions for L1 norm models can also verify the analysis above. Not much useful/important relationship for the pair [USA(Q30), Capital(P36), Washington D.C.(Q61)] are acknowledged by the model. When we do "*USA - WishingtonD.C.*", not much region or administrative relationships are extracted, instead, the model calculates more about the relationship between people or entities. This is also the reason when we use "*USA + Capital*" to inference WashingtonD.C., most of the outcomes are countries.

2. L2 norm models have better performances. The loss does not get down to 0, and except when the embed dimension is set to 50, other L2 models all still have a downward trend. The model has not fully converged, and later on we will try to apply larger learning rates.

   When doing "*USA + Capital*" inference, the tail entities are mostly cities that are capitals or were once capitals. USA cities also pop up in the top-6 most related list. The learning direction is correct, we just need to tune the model so that the connection between America and capital could be strengthened.

3. TransE models with L1 norm and L2 norm have huge differences. From the above diagrams, we assume that the L2 norm works better with our dataset and task.

   L1 regularization penalizes the sum of absolute values of the weights, whereas L2 regularization penalizes the sum of squares of the weights. The L1 regularization solution is sparse. The L2 regularization solution is non-sparse[13].

   The L1 regularization is a form of feature selection, because when we assign a feature with a 0 weight, we're multiplying the feature values by 0 which returns 0, eradicating the significance of that feature. If the input features of our model have weights closer to 0, our L1 norm would be sparse. L2 regularization on the contrary, combats overfitting by forcing weights to be small, but not making them exactly 0. That is, less significant features for predicting would still have some influence over the final prediction, but it would only be a small influence. This to some extent explains the loss graph above, why L1 norm models get to 0 so quickly and most of the time only the first entity in the inference tail entities has some connections to the given conditions.

4. Since the inferences generated by L1 norm models are almost non-correlated to the given pair, we will mainly analyze the impact of embed dimensions with L2 norm models. The higher the embed dimension is, the slower the model converges. The loss diagrams also suggest that the current learning rate is too small for the model to start actual learning immediately.

   From a personal perspective, the result of "*USA - Washington D.C.*" should be P36(capital) and P150(contains the administrative territorial entity). With embed dimensions 50, 100, 200, and 300, both these two relationships have been inference out. When the embed dimension is set to 400, the model fails to calculate the "capital" relationship, which should be the top-1 correlated answer. One potential reason could be the model still has a long way to go before converging. Another

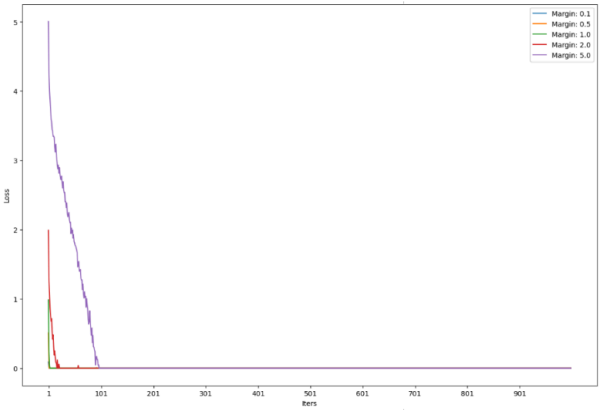
potential reason would be the “curse of dimensionality”, a phenomenon in which the data get too sparse and the model easily gets overfitted and cannot generalize well.

When the embed dimension is set to 200, it has the most inferences that hit some of the closely related entities/relationships. Therefore we will use this value for the later experiments.

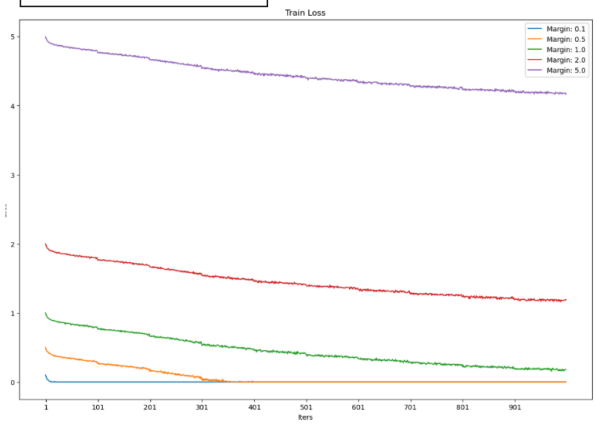
2.2 Margin

Given the fact the L1 norm models have comparatively weaker performances from Section 2.1, we will mainly focus on investigating margin impact on L2 norm models performances.

Norm: L1 Embed Dim: 200



Norm: L2 Embed Dim: 200



Norm: L2 Q30+P36(USA + Captital)

Margin=0.1		Margin=0.5		Margin=1.0		Margin=2.0		Margin=5.0	
Q10887660	Baokang	Q16666	Nanjing	Q90	Paris	Q656	Saint Petersburg	Q90	Paris
Q1412243	25 kV AC railway electrification	Q38	Italy	Q60	New York City	Q515	city	Q656	Saint Petersburg
Q947766	Wallace Reid	Q1085	Prague	Q65	Los Angeles	Q90	Paris	Q60	New York City
Q2367525	Julio	Q1780	Bratislava	Q220	Rome	Q60	New York City	Q220	Rome
Q21097404	mitotic metaphase plate congression	Q1741	Vienna	Q656	Saint Petersburg	Q641	Venice	Q65	Los Angeles
Q1762951	Chalcosyrphus	Q30	United States of America	Q84	London	Q216	Vilnius	Q84	London

Norm: L2 Q61-Q30(USA -WashingtonD.C)

Margin=0.1		Margin=0.5		Margin=1.0		Margin=2.0		Margin=5.0	
P2388	office held by head of the organization	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity
P749	parent organization	P463	member of	P37	official language	P36	capital	P36	capital
P407	language of work or name	P2578	is the study of	P36	capital	P1303	instrument	P37	official language
P205	basin country	P1075	rector	P1196	manner of death	P735	given name	P136	genre
P840	narrative location	P618	source of energy	P136	genre	P1622	driving side	P272	production company
P9	/	P194	legislative body	P735	given name	P463	member of	P463	member of

Norm: L2 Q30+P36(USA + Captital)

Margin=1.0		Margin=1.25		Margin=1.5		Margin=1.75	
Q656	Saint Petersburg	Q656	Saint Petersburg	Q515	city	Q515	city
Q60	New York City	Q60	New York City	Q656	Saint Petersburg	Q90	Paris
Q220	Rome	Q90	Paris	Q90	Paris	Q220	Rome
Q90	Paris	Q220	Rome	Q60	New York City	Q656	Saint Petersburg
Q515	city	Q65	Los Angeles	Q216	Vilnius	Q60	New York City
Q641	Venice	Q84	London	Q163012	Pankow	Q65	Los Angeles

Norm: L2 Q61-Q30(USA -WashingtonD.C)

Margin=1.0		Margin=1.25		Margin=1.5		Margin=1.75	
P150	contains the administrative territorial entity	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity
P36	capital	P463	member of	P36	capital	P36	capital
P1196	manner of death	P36	capital	P21	sex or gender	P735	given name
P37	official language	P37	official language	P463	member of	P37	official language
P178	developer	P735	given name	P37	official language	P463	member of
P1589	lowest point	P135	movement	P2643	Carnegie Classification of Institutions of Higher Education	P31	instance of

Blue: USA related; Yellow: Capital related; Green: Capital and administrative territorial related.

Observations:

1. We could see more details on the learning process for L1 norm models with higher margins. However, all L1 norm models have loss approaches to 0 at the first epoch. We could make an assumption that, same as in Section 2.1, L1 norm model performances would not enhance too much since they stop learning after the first epoch.
2. For both L1 norm and L2 norm models, the greater the margin is, the slower the model converges. In addition, margin values seem also to somehow settle bottom(minimum) loss values after a L2 norm model converges. This is because the L2 regularization tries to approach the loss to 0, rather than set it to 0. And margin is directly added to this positive and negative score difference.
3. Originally, we tested margin values of [0.1, 0.5, 1.0, 2.0, 5.0] on L2 norm models. From the inference diagrams of (USA, capital, Washington D.C.), we could see that the models have comparatively good performances(most top-6 inferences that connect to USA or capital) when the margin is equal to 1.0 or 2.0 or 5.0. In order to get more details on how margin affects the learning results, we tested margin values of [1.25, 1.5, 1.75] in addition.
4. For margin values of [1.0, 1.25, 1.5, 1.75, 2.0], all L2 norm models have successfully answered the relationship between USA and Washington D.C. with the top2 most relative inference. The most frequent model responding to this question is: Washington D.C contains the administrative territorial entity of America, it is the capital of America. However, the relationship “capital” is not the first inference of the model, this is also the reason why the model fails to nominate “Washington D.C.” as the answer of “*USA+capital*”. We will eliminate margin=1.25 at this stage since the “capital” only ranks the thirds in its inference.
5. We will choose 2 L2 norm models for testing learning rates. Both 2 values below have answered “*USA+capital*” with most of the references related to capital or USA.
    - Margin=1.75;
    - Margin=1.0.

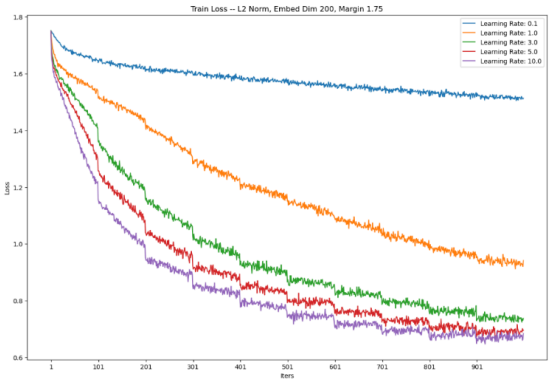
2.3 Learning Rate

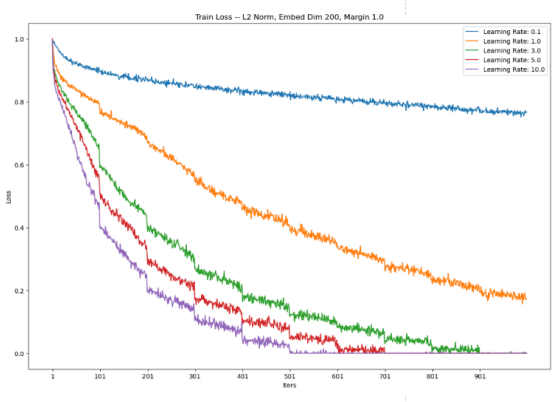
As analyzed above, the current learning rate 1.0 is a bit too small. In this section we will use much larger learning rates and see how the model performs.

The learning rates tested in this section are: [1.0, 3.0, 5.0, 10.0].

Below images are the loss diagrams when the margin is set to 1.75 and 1.0. Given the same learning rate, higher margin values mean slower convergence. The loss curves with different learning rates all follow a similar trend, in each epoch the loss first decreases massively, then fluctuates until entering the next epoch and decreases significantly again.

When the margin is set to 1.0 and the learning rate is 10.0, the loss of the model approaches 0 at the fifth epoch. With the same margin, learning rate 5.0 converges at around epoch 7 and learning rate 3.0 converges at around epoch 9. Here, we only refer to convergence by the train loss curves. This is not quite accurate since in reality, we need also take into account the validation loss, and other model evaluation metrics.





Norm: L2 Embed Dim 200, Margin 1.75 Q30+P36(USA + Captital)

Learning Rate = 1.0		Learning Rate = 3.0		Learning Rate = 5.0		Learning Rate = 10.0	
Q90	Paris	Q90	Paris	Q90	Paris	Q90	Paris
Q60	New York City	Q60	New York City	Q60	New York City	Q60	New York City
Q65	Los Angeles	Q65	Los Angeles	Q65	Los Angeles	Q65	Los Angeles
Q220	Rome	Q656	Saint Petersburg	Q220	Rome	Q220	Rome
Q656	Saint Petersburg	Q220	Rome	Q84	London	Q84	London
Q84	London	Q84	London	Q656	Saint Petersburg	Q656	Saint Petersburg

Norm: L2 Embed Dim 200, Margin 1.75 Q30+P36(USA -Washington D.C.)

Learning Rate = 1.0		Learning Rate = 3.0		Learning Rate = 5.0		Learning Rate = 10.0	
P150	contains the administrative territorial entity	P1383	contains settlement	P1383	contains settlement	P1383	contains settlement
P36	capital	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity	P734	family name
P37	official language	P36	capital	P734	family name	P150	contains the administrative territorial entity
P360	is a list of	P734	family name	P36	capital	P36	capital
P102	member of political party	P1622	driving side	P735	given name	P1589	lowest point
P509	cause of death	P463	member of	P1622	driving side	P35	head of state

Norm: L2 Embed Dim 200, Margin 1.0 Q30+P36(USA + Captital)

Learning Rate = 1.0		Learning Rate = 3.0		Learning Rate = 5.0		Learning Rate = 10.0	
Q90	Paris	Q90	Paris	Q90	Paris	Q90	Paris
Q60	New York City	Q60	New York City	Q60	New York City	Q60	New York City
Q656	Saint Petersburg	Q65	Los Angeles	Q65	Los Angeles	Q65	Los Angeles
Q220	Rome	Q220	Rome	Q220	Rome	Q220	Rome
Q65	Los Angeles	Q656	Saint Petersburg	Q656	Saint Petersburg	Q84	Saint Petersburg
Q220	Rome	Q84	London	Q84	London	Q64	Berlin

Norm: L2 Embed Dim 200, Margin 1.0 Q30+P36(USA -Washington D.C.)

Learning Rate = 1.0		Learning Rate = 3.0		Learning Rate = 5.0		Learning Rate = 10.0	
P150	contains the administrative territorial entity	P150	contains the administrative territorial entity	P1383	contains settlement	P1383	contains settlement
P463	member of	P36	capital	P150	contains the administrative territorial entity	P150	contains the administrative territorial entity
P735	given name	P1383	contains settlement	P36	capital	P36	capital
P1343	described by source	P734	family name	P734	family name	P734	family name
P2363	NMHH film rating	P37	official language	P37	official language	P1622	driving side
P470	Eight Banner register	P735	given name	P30	continent	P37	official language

Blue: USA related; Yellow: Capital related; Green: Capital and administrative territorial related; Purple: Washington as a name.

Above are the inference diagrams of margin 1.75 and margin 1.0. The color code of highlights is also stated above.

Observations:

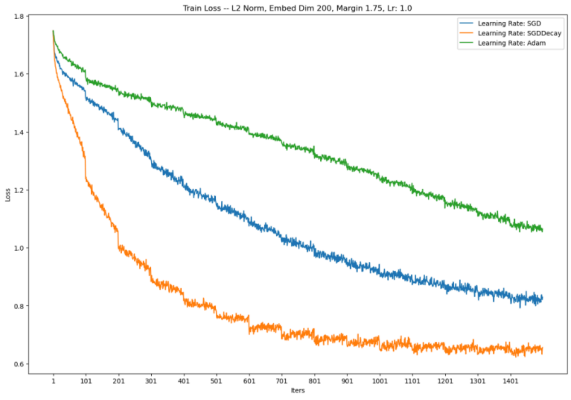
1. When doing the “USA+Capital” inference, the nominate tail entity lists from the models are almost fixed, especially for the top 3(most related) tail entities. An interesting fact is that the model at this point understands what “capital” relationship means, since most of the answers are capitals of different countries. However, the model finds it difficult to find the exact capital for the United States of America.
2. When doing the “USA-Washington D.C.” inference, as the learning rate grows, relationships that related to capital(ideal answer) or other administrative territorials are pushed down to less-importance ranks, whilst the relationships that link to Washington D.C. as a name starts to pop up more in the nomination list. This could be a potential cause of Observation 1 above.

"Washington" as a word has great ambiguity. Our model mainly learned the entities and relationship types associated with it as a person's name, but did not learn the associated knowledge representations when it was used as a city name. Therefore, when we try to find the capital city of the United States of America, Washington D.C is not considered as a city. If we want significant improvements in this TransE model, we need to invoke stronger methods to eliminate word sense ambiguous.

3. We will use "Norm: L2, Embed Dimension: 200, Margin:1.75, learning rate:1.0" and "Norm: L2, Embed Dimension: 200, Margin:1.0, learning rate:10.0". These two have similar inference results in the above diagrams, and their inference entities/relationship are more closely to the ideal answers we expect to have from the model prediction.

   The next section will be a simple test on different optimizers. Researches online have mentioned that a simple switch from using SGD as optimizer to Adam could bring in significant improvements to the TransE algorithm.

2.4 Optimizer



Norm: L2 Embed Dim 200, Margin 1.75 Lr 1.0 Q30+P36(USA + Captital)

SGD		SGD(momentum 0.9, weight decay 1e-5)		Adam	
Q90	Paris	Q90	Paris	Q163012	Pankow
Q60	New York City	Q60	New York City	Q79808	Constanța
Q65	Los Angeles	Q65	Los Angeles	Q38545	Adana
Q656	Saint Petersburg	Q656	Saint Petersburg	Q495	Turin
Q220	Rome	Q84	London	Q33959	Nice (city)
Q515	city	Q220	Rome	Q158876	Reinickendorf

Norm: L2 Embed Dim 200, Margin 1.0 Lr 10.0 Q30+P36(USA + Captital)

SGD		SGD(momentum 0.9, weight decay 1e-5)		Adam	
Q90	Paris	Q90	Paris	Q2467	Nassau
Q60	New York City	Q60	New York City	Q3449655	rue Quincampoix
Q65	Los Angeles	Q65	Los Angeles	Q522398	Kimbolton
Q656	Saint Petersburg	Q84	London	Q2124272	Maasdam
Q220	Rome	Q220	Rome	Q132997	Florianópolis
Q84	London	Q656	Saint Petersburg	Q166382	emir

Norm: L2 Embed Dim 200, Margin 1.75 Lr 1.0 Q30+P36(USA - Washington D.c.)

SGD		SGD(momentum 0.9, weight decay 1e-5)		Adam	
P150	contains the administrative territorial entity	P1383	contains settlement	P263	official residence
P36	capital	P734	family name	P2650	interested in
P735	given name	P36	capital	P177	crosses
P1622	driving side	P150	contains the administrative territorial entity	P1427	start point
P1383	contains settlement	P832	public holiday	P511	honorific prefix
P37	official language	P6	head of government	P2388	office held by head of the organization

Norm: L2 Embed Dim 200, Margin 1.75 Lr 1.0 Q30+P36(USA - Washington D.c.)

SGD		SGD(momentum 0.9, weight decay 1e-5)		Adam	
P1383	contains settlement	P1383	contains settlement	P150	contains the administrative territorial entity
P150	contains the administrative territorial entity	P734	family name	P36	capital
P735	given name	P1552	has characteristic	P832	public holiday
P734	family name	P610	highest point	P463	member of
P36	capital	P208	executive body	P2936	language used
P1589	lowest point	P611	religious order	P1416	affiliation

Observations:

1. Models with Adam optimizer are the slowest to learn and to converge. They also have the weakest inference on the given entities and relation pair. This phenomenon is very likely caused by underfitting.
2. With higher learning rates, we could see that representation of “Washington” as a name is strengthened in the learned experiences, whilst the word as a city is gradually forgotten(less important).
3. We will use combinations below as our final model:
    - Norm: L2, Embed Dimension: 200, Learning Rate: 1.0, Margin: 1.75 with SGD optimizer. The SGD has momentum 0.9 and weight decay 1e-5.

    We choose this model because:

    - It has successfully built understanding over “capital”.
    - It learns the ambiguous meaning of “Washington” as a name and as a city. It also tries to link these learnt representations to historical/political entities/relationships.

2.5 Final Model

We use Norm: L2, Embed Dimension: 200, Learning Rate: 1.0, Margin: 1.75 with SGD optimizer. The SGD has momentum 0.9 and weight decay 1e-5 for our final TransE model. Below are attached diagrams for its inference on <*USA, capital, ?*> and <*USA, ?. North America*>.

Q30+P36	USA + Capital	Ideal Result
Q90	Paris	Q61 Washington, D.C.
Q60	New York City	
Q65	Los Angeles	
Q220	Rome	
Q84	London	
Q656	Saint Petersburg	

Q30-Q49	USA -> North America	Ideal Result
P1532	country for sport	P361 part of
P1001	applies to jurisdiction	
P131	located in the administrative territorial entity	
P159	headquarters location	
P291	place of publication	
P17	country	

Highlighted means closer to the expected answers.

From all the above experiments on TransE models, we could observe a phenomenon that the political/region related information can be learnt at an early stage of training(with small max epoch). With large epoch amounts, the geographically related information becomes less important, and the model starts to pick up and remember social/society related connections.

3 Summary

In this assignment, we mainly focus on parameter tuning for the TransE and word2vec model. The parameter tuning experiments of the TransE model have major flaws, mainly due to: 1. The training set, verification set and test set are not splitted. The quality of the models only relies on the loss during training and human evaluation of certain individual cases. 2. No evaluation indicators/metrics were used; inspired by online resources, we found an indicator called mean reciprocal rank (MRR), but due to the time limitation, it is not implemented in this assignment. In summary, the experimental completeness of TransE needs to be improved. Loss and evaluation indicators on the verification set and test set need to be introduced to make the quality of parameter adjustment more objective.

For the Word2Vec model, The best combination among the values we tested are – vector size: 250, window size: 35, minimum count: 10 and max epoch: 16. The best Spearman's Rank correlation coefficient on the WordSim353 is 0.747125.

For the TransE model, using the L2 norm in our case has better performance than using the L1 norm. The loss gets to 0 at the first training epoch with L1 regularization, and the model stops learning(underfitting) too early. L2 has its nature to force values to be small, but not 0. This allows the TransE model with L2

regularization to learn more. After analyzing the impact of embed dimensions, regularization methods, margins and learning rates on the TransE model, we choose “Norm: L2, Embed Dimension: 200, Learning Rate: 1.0, Margin: 1.75 with SGD optimizer. The SGD has momentum 0.9 and weight decay 1e-5” as our final model. For the given questions: <*USA, capital, ?*> and <*USA, ?. North America*>, our final model has successfully understood the “capital”, it could infer capitals of different countries but fail to point out the capital for the United States of America, our given head entity. As for the second question, the ideal answer would be “USA is part of North America”, however, our model fails to calculate relationships that are more closely to “part of”. The inferences that are related to “part of” would be: “applies to jurisdiction”, “located in the administrative territorial entity” and “country”. As “country” is a subproperty of “located in the administrative territorial entity”, we could assume that the model still has not finished learning. In addition, as mentioned in the experiment observations, the political/region related information can be learnt at an early stage of training(with small max epoch). With large epoch amounts, the geographically related information becomes less important, and the model starts to pick up and remember social/society related connections.

During the process of finishing this report, I had a high fever and a serious cold, and therefore there was not enough time left to implement and test any enhancement in Word2Vec and TransE models. An obvious limitation from our experiments suggests that basic Word2Vec and TransE models both have difficulties in word sense ambiguous elimination. It would be a topic that is worth exploring. Also inspired by an article “Improving Knowledge Graph Embedding Using Affine Transformations of Entities Corresponding to Each Relation”, we could also spend some time later to try out the idea.

4 References

[1] TransE https://paperswithcode.com/method/transe

[2] Text8 https://paperswithcode.com/dataset/text8

[3] Wikidata:WikiProject Datasets https://www.wikidata.org/wiki/Wikidata:WikiProject_Datasets

[4] Gensim
https://en.wikipedia.org/wiki/Gensim#:~:text=Gensim%20is%20an%20open%2Dsource,Gensim

[5] WordSim353 - Similarity and Relatedness http://alfonseca.org/eng/research/wordsim353.html

[6] A Dummy’s Guide to Word2Vec
https://medium.com/@manansuri/a-dummys-guide-to-word2vec-456444f3c673

[7] Word Embedding Techniques: Word2Vec and TF-IDF Explained
https://towardsdatascience.com/word-embedding-techniques-word2vec-and-tf-idf-explained-c5d02e34d08

[8] Lin, Chu-Cheng & Ammar, Waleed & Dyer, Chris & Levin, Lori. (2015). Unsupervised POS Induction with Word Embeddings. 10.3115/v1/N15-1144.

[9] Knowledge Graph Embeddings for Entity, Link Prediction — The Basics

[10] mklimasz/TransE-PyTorch https://github.com/mklimasz/TransE-PyTorch

[11] MARGINRANKINGLOSS https://pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html

[12] Dimensionality of Word Embeddings https://www.baeldung.com/cs/dimensionality-word-embeddings

[13] Fighting Overfitting With L1 or L2 Regularization: Which One Is Better?
https://neptune.ai/blog/fighting-overfitting-with-l1-or-l2-regularization