Word2Vec Skipgram Model

Raghu Chittersu
Department of Computation and Data Sciences
Indian Institute of Sciences, Bengaluru
chittersuv@iisc.ac.in

1 Introduction

Word embeddings are vector representation of words. They are shown to capture salient features of the words. Word embeddings are very popular as they are used in various as features in many downstream NLP tasks.

In this assignment, word embeddings are generated using Skip-gram model by Mikolov et al. (2013). The impact on results due to various hyperparameters such as window size, negative samples and embedding dimensions is analyzed. The claim that the word embeddings capture relationships between words is verified through analogical reasoning tasks. Finally, general observations and features of generated word embedding are presented.

The core of the skip-gram model is built using tensorflow and code can be found at https://github.com/rv-chittersu/word2vec

2 Data

To train skip-gram model The Reuters Corpus available through NLTK is used. The corpus has 10,788 news documents totaling 1.3 million words. The corpus is divided into 7769 training documents and 3019 test documents. Training documents are further split to generate validation set.

3 Model

The objective of skip-gram model is predicting surrounding words given input word. It is achieved by maximizing the following equation for the given sequence of input words $w_1, w_2..w_T$.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_j)$$

Generally $p(w_{t+j}|w_j)$ is estimated through softmax probability. In this setup softmax is replaced by Noise Contrastive Estimation (NCE). so the $p(w_i|w_j)$ is substituted by

$$\log\sigma(v'_{w_i}v_{w_j}^T) + \sum_{k=1}^{k} E_{w_k\sim p(w)}[\log\sigma(-v'_{w_k}v_{w_j}^T)]$$

So for each input token, label token (which is neighbor token present in the pre-defined window) and k negative samples we compute NCE and maximize it. In the following model d dimensional vectors $\{v_i\}_{i=0}^{N}$ and $\{v'_j\}_{j=0}^{N}$ are parameters that are to be estimated.

4 Experiment Setup

4.1 Pre Processing

Once training, validation and test documents are available the very next step is to generate vocabulary.

From training documents, the vocabulary is generated in the following steps.
(1) Tokenize sentences with NLTK's sent_tokenize
(2) Split the sentences at nonalphabet characters(except periods to preserve tokens like U.S.A)
(3) Trim the tokens(to remove trailing and preceding dots)
(4) Convert tokens to lower case
(3) Discard the tokens that have a length less than 2
(4) Remove stop words.

The generated vocabulary will be used as a source vocabulary for the rest of the procedure. The vocabulary also holds the number of occurrences and unique index for each word.

4.2 Training

During training in each epoch, training documents are sequentially processed to generate inputs to the model. At each step, the following values are

passed through placeholders
(1) input word
(2) context word
(3) list of negative samples
(4) unigram probabilities of negative samples

From the inputs negative of Noise Contrastive Estimation (NCE) is computed as the loss. Tensorflow's GradientDescentOptimizer is used to minimize the loss.

For each epoch, validation loss is computed to track the learning. Once the training is over test loss is computed and the embedding layer is stored in a file.

The model is trained with multiple combinations of hyperparameters(window size(c), negative samples(k), embedding dimension(d)) as part of the experiment.

4.3 Evaluation

The embeddings generated in the training set are used for evaluation of the model.

The correlation between the similarities scores produced by the model and SimLex-999 is used as a metric. The evaluation is done for Nouns, Verbs, and Adjectives separately.

5 Results

The embeddings generated with various hypeparameters are used to calculate the similarity score for words present in SimLex-999. The scores given by model and Simlex-999 are used to generate co-relation scores which are reported in tables below.

It can be seen that the rare words add a lot of noise to similarity score generated by the model which results in poor co-relation score.

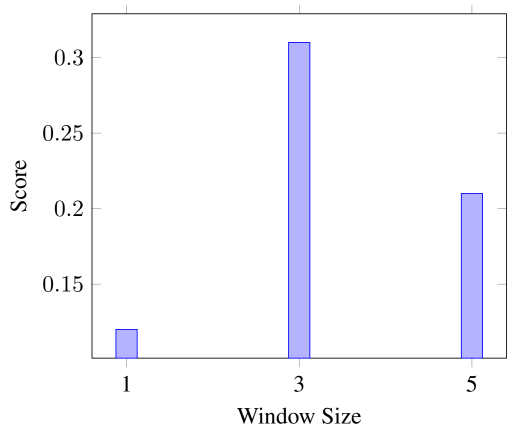
Another set of co-relation scores are also reported by removing rare words(words which occurs less than 100 times).

Further co-relation scores with respect to nouns, verbs, and adjectives are reported separately.

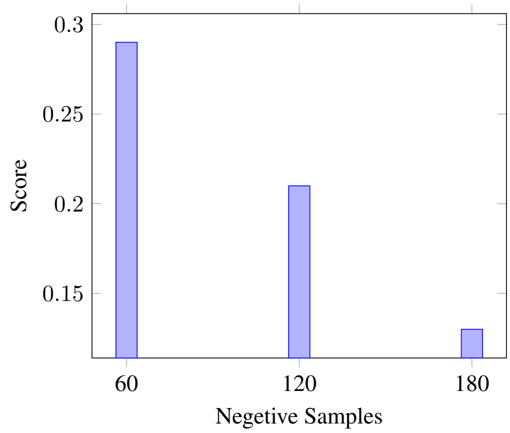
6 Analysis

As it can be seen from [Table 1] that the co-relation scores improve once rare words are filtered while testing. So the rest of the analysis will rely on the words that occur greater than a threshold(in this case 100).

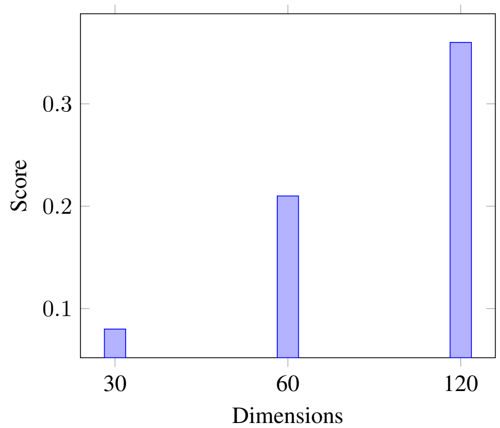
In the experiment to study window size on the result embedding dimensions and negative samples are locked at 60 and 120 respectively. The training is done with window sizes of 1, 3, and 5. From the graph, it can be seen that the best score was achieved at window size 3. probably the model over-fit at window size 5.



A similar test was made with negative samples which were tested with values of 60, 120 and 180 for a window of size 5 and embedding of 60 dimensions A decreasing trend was found with an increase in negative samples. The optimum negative samples count is 60.



The dimension of size 120 is found better than 30 and 60 with a window size of 5 and 120 negative samples.



Similar trends can be seen in case if we consider only nouns.

Table 1: Co-Relation Scores

Embedding Dimensions	**Negative Samples**	**Window size**	**Score**	**Score (threshold = 100)**
60	120	1	0.04711872114	0.1220493397
60	120	3	**0.0505206797**	0.3105621417
60	120	5	0.02206475059	0.2131087133
60	60	5	-0.01793491812	0.2931796434
60	180	5	0.02312335723	0.135010061
30	120	5	0.0383291462	0.08621740159
120	120	5	0.04571867378	**0.3661504363**
120	180	3	0.02075320559	0.1882198335

Table 2: Co-Relation Scores For Nouns

Embedding Dimensions	**Negative Samples**	**Window size**	**Noun Score**	**Noun Score (threshold = 100)**
60	120	1	0.04563564751	0.04563564751
60	120	1	0.04563564751	0.04563564751
60	120	3	**0.1363865401**	0.4282928711
60	120	5	0.05134560941	0.3001244178
60	60	5	0.02397852387	0.4015901274
60	180	5	0.0682602615	0.4238850173
30	120	5	0.1045081084	0.3012992204
120	120	5	0.1441013864	0.3826956967
120	180	3	0.1133083195	**0.458975523**

Table 3: Co-Relation Scores For Verbs

Embedding Dimensions	**Negative Samples**	**Window size**	**Verb Score**	**Verb Score (threshold = 100)**
60	120	1	**0.02475670118**	-0.02421592686
60	120	3	-0.0407206503	0.07494772754
60	120	5	-0.03550466133	-0.06945869851
60	60	5	-0.09172793869	0.02018972524
60	180	5	-0.1097457637	-0.5033712957
30	120	5	-0.08523328442	-0.3574116876
120	120	5	-0.07784494849	**0.1527943869**
120	180	3	-0.09808198607	-0.4895402191

Table 4: Co-Relation Scores For Adjectives

Embedding Dimensions	**Negative Samples**	**Window size**	**Adjective Score**
60	120	1	-0.0146127907
60	120	3	-0.06573593351
60	120	5	-0.007612853526
60	60	5	-0.05354720563
60	180	5	**0.07345874163**
30	120	5	-0.003055926895
120	120	5	-0.08271468455
120	180	3	-0.08531358037

Task	Score	Score threshold = 100
capital-common-countries	0.024	0.118
capital-world	0.006	0.161
currency	0.064	0.192
city-in-state	0.007	0.266
family	0.0207	
gram1-adjective-to-adverb	-0.010	
gram2-opposite	0.001	
gram3-comparative	0.012	0.029
gram4-superlative	0.021	-0.018
gram5-present-participle	-0.005	0.007
gram6-nationality-adjective	0.009	0.025
gram7-past-tense	0.001	0.032
gram8-plural	0.026	
gram9-plural-verbs	0.008	0.032

Table 5: Scores on analogical reasoning task.

No trends were seen in the case of adjectives and verbs. The results obtained with respect to adjectives and verbs are inconsistent as their frequency is very low.

7 Analogical Reasoning task

7.1 Task

It was stated by Mikolov et al. (2013) that the word embeddings generated by Word2Vec can capture the relationship between words. The claim is verified using the Analogical reasoning task. In this task, the relationship between the embedding is captured as the difference vector. It is expected that the cosine similarity between the difference vectors is high if they have the same underlying relationship.

7.2 Data

To accomplish this same dataset used by Mikolov et al. (2013) is used. The data has 14 relationship categories. In each category, there are multiple examples. Each example has 2 pairs of words and has mentioned relationship.

7.3 Evaluation

As mentioned the for each instance the difference between each pair is considered as relationship vector. Since the two pairs have the same relationship the two differences are expected to give high cosine similarity score.

So for each entry in a task the similarity scores are calculated and averaged. The results can be seen in [Table 5].

Similar to that of SimLex-999 evaluation the experiment is done after filtering out the rare words(frequency less than 50). It can be seen that the results are improved after rare word removal.

The adjectives and verbs are seen to have low scores in analogical reasoning task also.

8 conclusion

In this assignment it was shown that Word2Vec can group similar tokens and can also capture various analogies between tokens. Results of corelation with SimLex-999 and analogy tasks are provided to support the claim. The results for adjectives and verbs are poor in general because they aren't found frequently enough in the corpus. With larger dataset, better results are expected.

In the appendix a few examples of similar models and embeddings are visualized with PCA.

References

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.

A Appendices

This appendix provides qualitative results for word embeddings.

A.1 Similar Words

For a few example words list of few similar words ordered based on cosine similarity is presented.

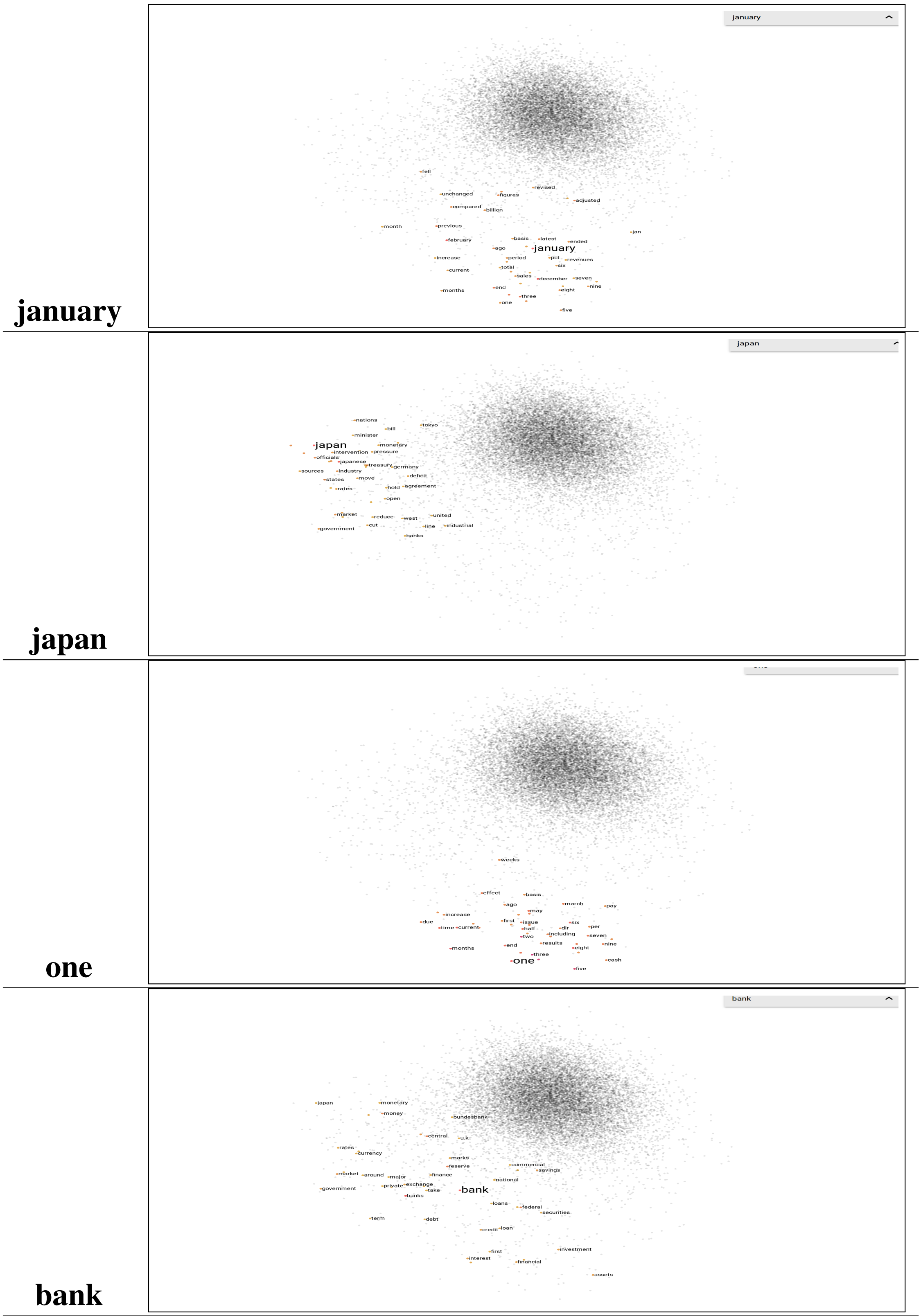
A.2 Embeddings Visualization

The embeddings of the vocabulary generated are visualized on the 2D plane using PCA. This is done using Tensorflow's Embedding Projector.

In the results each word and it's corresponding similar words are visualized.

one	**january**	**bank**	**rice**	**japan**	**minister**
two	december	central	crop	japanese	told
three	february	england	grain	trade	ministry
four	period	banks	cotton	u.s	reagen
five	previous	finance	persist	deficit	industry
seven	november	money	krenzler	major	washington
current	three	banking	usda	monetary	news
months	compared	savings	prepare	nations	president
including	year	loan	soybean	tokyo	house
half	ago	liquidity	acid	must	appleton
six	end	foreign	water	britain	reporters
making	eight	regard	bag	france	saying
march	nine	bundesbank	cargo	congress	congress
time	october	federal	program	bill	economic
may	ended	france	enquiries	timing	press
eight	adjusted	campaign	covered	foreign	officials

Similar Words - List of 15 words with the highest similarity score for a given word.



Embedding Visualization - PCA of the word embeddings with the word and its most similar words highlighted.