Machine Learning and Computational Statistics: Project Report

Emily Denton (eld297) & Rahul Gopalkrishnan (rg2451)

1 Introduction

2 Problem definition

2.1 Learning a word embedding space

2.2 Exploring properties of learned word embedding space

3 Experimental results

3.1 Data

3.1.1 Training data

3.1.2 Test data

[?] propose evaluating the regularities of the learned embedding space with a test set of analogy questions. The questions are of the form ”a is to b as c is to __ ”. The test set contains 14 different types of analogies (see Table 1) relating to semantic concepts and grammatical relations.

Table 1: Analogical reasoning test set

Relation	**# Questions**	**Example**
capital-common-countries	506	Athens : Greece Bangkok : Thailand
capital-world	4524	Abuja : Nigeria Accra : Ghana
currency	866	Algeria : dinar Japan : yen
city-in-state	2467	Chicago : Illinois Houston Texas
family	506	brother : sister mother : father
adjective-to-adverb	992	amazing : amazingly calm : calmly
opposite	813	acceptable : unacceptable aware : unaware
comparative	1331	bad : worse big : bigger
superlative	1122	bad : worst big : biggest
present-participle	1056	code : coding dance : dancing
nationality-adjective	1599	Albania : Albanian Argentina : Argentinean
past-tense	1560	dancing : danced decreasing : decreased
plural	1332	banana : bananas bird birds
plural-verbs	870	eat : eats generate : generates

3.2 Results

3.2.1 Measuring linguistic regularity via analogies

We evaluate the performance of (a) pre-trained Google vectors, (b) our Skip-Gram model, (c) our CBOW model, on the analogical reasoning test set introduced in section 3.1.2. Given three query words (e.g. *Paris, France, London*), the task is to return the answer that fits with the analogy (in this case, *England*). This problem can be solved in many ways, [?] propose a simple solution that relies on the inherent regularities of the embedding space learned by the Skip-Gram and CBOW models. The method uses simple vector algebra in the embedding space to find the solution word given three query words. For example, suppose the analogical relation of interest is A is to B as C is to D. Given three query words, A, B, C, the predicted solution is computed as follows:

1. Compute the vector representations of each word, $\phi(A), \phi(B), \phi(C)$.
2. Let $v = \phi(B) - \phi(A) + \phi(C)$.
3. Do a nearest neighbors search, based on cosine distance, to find the K closest word vectors to v. In other words, solve for the top K solutions to

$$\max_u \frac{v \cdot u}{\|u\| \|v\|} \tag{1}$$

The intuition behind this method is that the cosine distance between $\phi(A - B)$ and $\phi(C - D)$ is small when A and B are analogous to C and D. Figure 1 illustrates this intuition.

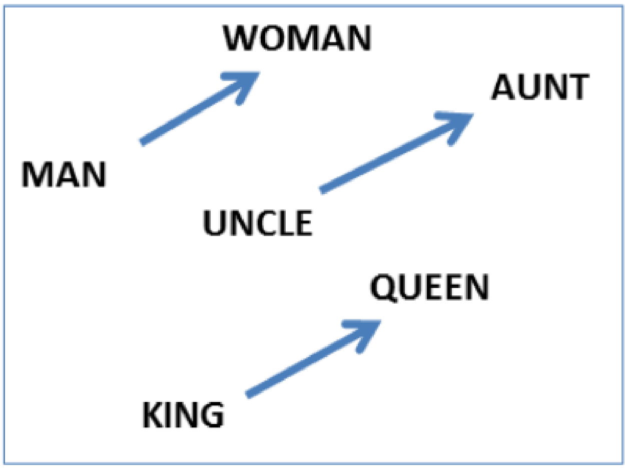


Figure 1: Vector offsets for three analogous word pairs.

Table **??** shows the top-k accuracy on the analogical reasoning dataset of the pre-trained Google vectors (GoogleVec) and the Skip-Gram and CBOW models we trained. The Google vectors outperform both our versions for all k. This is to be expected since the Google vectors were trained on far more data than we have access to. Out CBOW model performs consistently worse than our Skip-Gram model. We hypothesize that this is a factor of the amount of training data we had. The CBOW model throws away a lot of information by ignoring the ordering of words. In the limit of infinite data, the bag-of-words assumption would not matter, however in a limited data setting we believe the CBOW model is hurt more than the Skip-Gram model due to the loss of information. Figure

2 plots the top-k accuracy as a function of k for the three models split up by analogy types. Figure 3 plots the top-5 accuracy of the three models for each of the analogy question types. A very interesting pattern emerges when we consider these results. We can break down the analogy questions into two type: (1) Analogies involving semantic relations between words such as capital-country, currency-country, and family relations (1) Grammatical analogies such as past/present tense. singular/plural terms, and present-participle relations. We notice that the Skip-Gram model performs better on analogical questions of type (1) whereas the CBOW model performs better on the questions of type (2). We hypothesis that this is a function of the number of examples of each kind of relation. The semantic analogies appear in very specific contexts. For example, the word *France* is unlikely to appear in general text but would rather appear in particular contexts. However, words in the grammatical relations are not specific to a particular context and would appear in a very wide variety of sentences. Thus, we hypothesize that the models have, in some sense, more information about grammatical relations than they do about specific semantic relations during training. Thus, since the CBOW model performs better when there is more data available, the CBOW model performs worse on semantic analogical relations than grammatical ones.

Table 2: Top k Accuracy on Analogical Reasoning Test

Top k	**GoogleVec**	**CBOW**	**Skip-Gram**
1	20.185%	3.029%	6.211%
2	68.967%	24.764%	46.986%
3	78.346%	37.919%	59.179%
4	82.424%	43.921%	65.314%
5	84.716%	47.513%	68.967%
6	86.246%	50.332%	71.479%
7	87.285%	52.553%	73.362%
8	88.149%	54.308%	74.631%
9	88.835%	55.879%	75.716%
10	89.352%	57.142%	76.698%

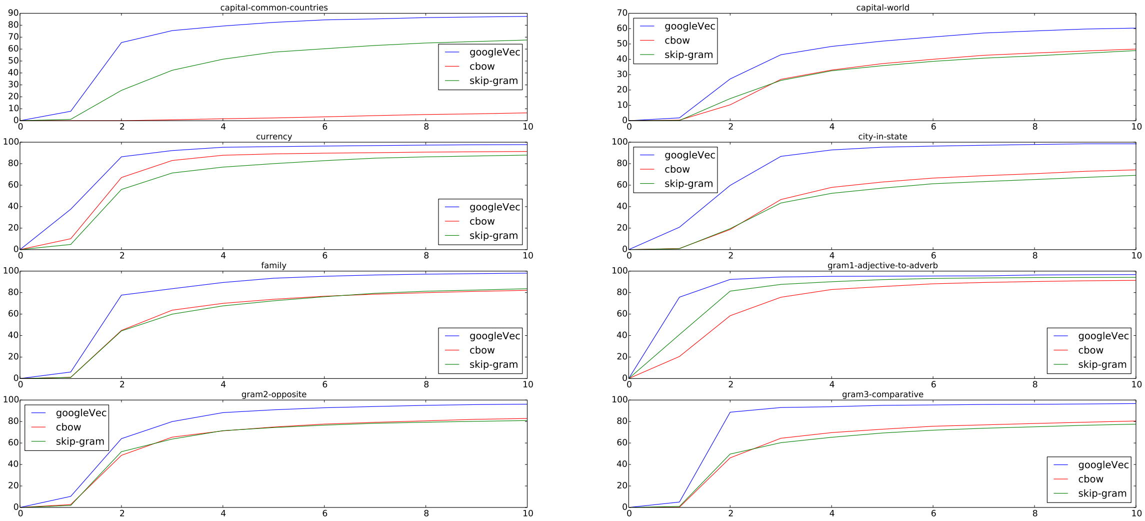


Figure 2: Top K accuracy for increasing K.

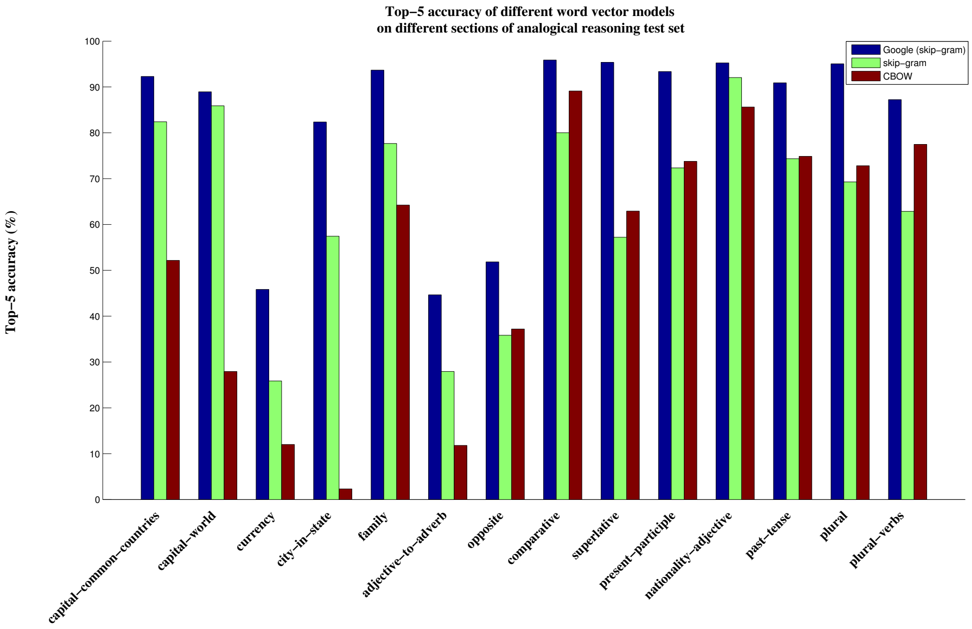


Figure 3: Top-5 accuracy per analogy question type.

3.2.2 Visualizing low dimensional approximations of embedding space

3.2.3 Finding analogical relations

4 Conclusion

5 Experimental Results

In this section, we refer to the 3 million vectors trained on Google's dataset as GoogleVec.

5.1 Evaluating Analogical-Reasoning on GoogleVec

We evaluated the analogical reasoning test on GoogleVec, CBOW and Skip-Gram and see the following performance. Note that the analogical reasoning test comprises lines of quartets. The task is as following, given the first three words, predict the fourth. Consider the case where the four words are A, B, C, D. We predict D using the vector representations by computing the vector $T = vec(B) - vec(A) + vec(C)$ and computing the k closest word vector to T. Accuracy is defined as the number of times $vec(D)$ appears in the set of k closest word vectors. Table 3 displays the results of accuracy for varying k.

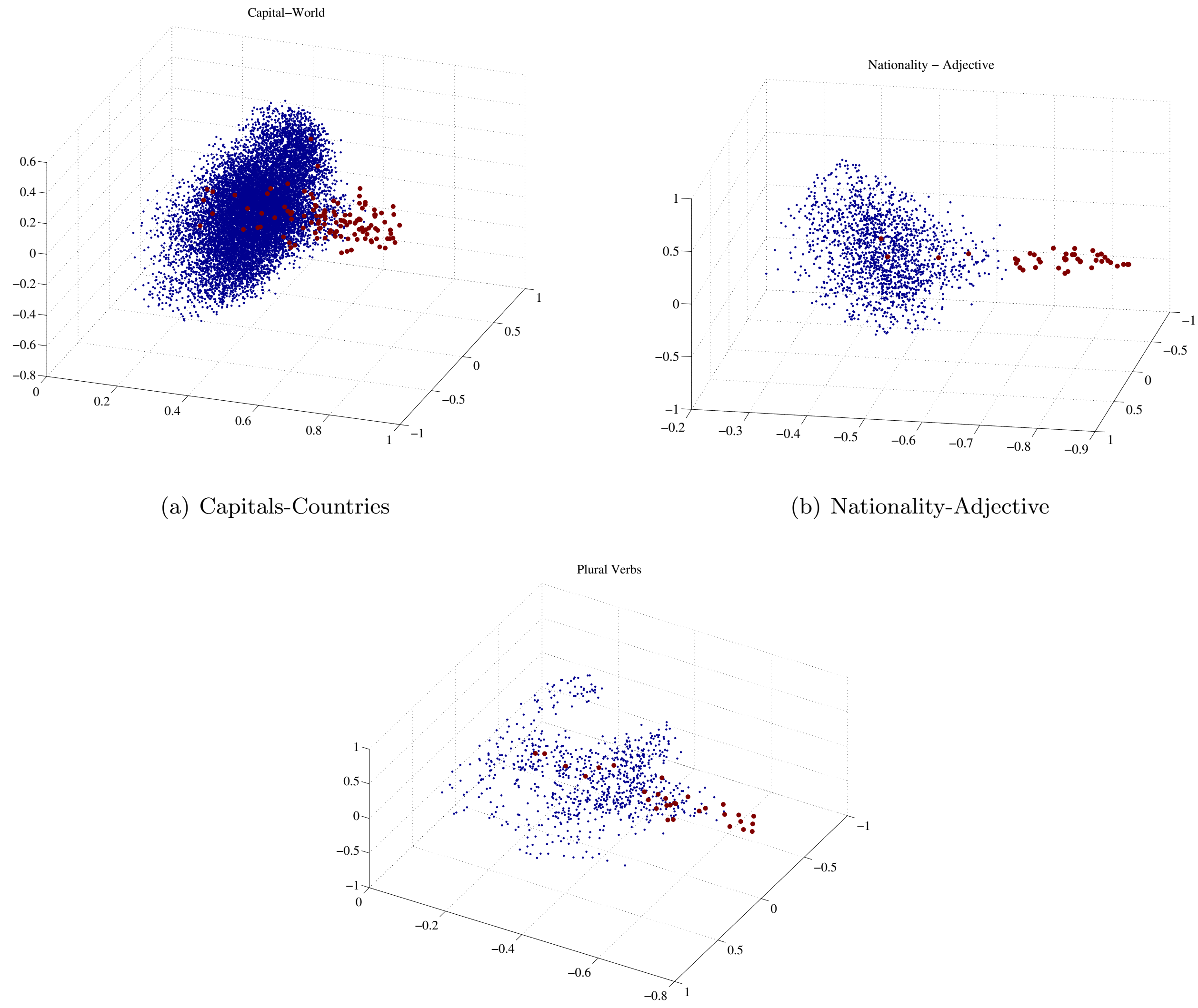
Table 3: Top k Accuracy on Analogical Reasoning Test

Top k	**GoogleVec**	**CBOW**	**Skip-Gram**
1	20.185%	3.029%	6.211%
2	68.967%	24.764%	46.986%
3	78.346%	37.919%	59.179%
4	82.424%	43.921%	65.314%
5	84.716%	47.513%	68.967%
6	86.246%	50.332%	71.479%
7	87.285%	52.553%	73.362%
8	88.149%	54.308%	74.631%
9	88.835%	55.879%	75.716%
10	89.352%	57.142%	76.698%

As we can see, the vectors trained on Google's Dataset outperform the models we train by atleast 13% in terms of accuracy on the analogical reasoning test. This likely has to do with the vast differences in training data available to us versus Google. Between the two models, the skip-gram architecture tends to do better. We conjecture that this has to do with the bag-of-words assumption made by CBOW during the training procedure. This concurs with the findings by Google in their original paper.

5.2 Unsupervised Learning of Word Pair Relationships

One of the goals of this project is to automatically discover relationships between words pairs. We hypothesized that valid word pairs have vector offsets that lie on a lower dimensional subspaces. To test this hypothesis, we run the following experiment. Using the analogical reasoning task, which gives us relations of the form $A \rightarrow B$ where A can be countries and B can be capitals of those countries. For every A and B in the analogical reasoning dataset, we compute $vec(A) - vec(B)$ and thus create a matrix of offsets. Note that only a subset of these correspond to true word relationships. We compute the 3 largest eigenvectors of the resulting offset matrix. We plot the projection of the offset matrix onto the three eigenvectors and highlight the true word pairs (i.e word pairs that we are hoping to find) in red. Some examples are depicted in Figure 6(a)-(d). As we can see, a large number of the red points seem to lie within a lower dimensional subspace even in the space spanned by the three eigenvectors. More specifically, they seem to lie within one line in the depicted three dimensional space.



(a) Capitals-Countries

(b) Nationality-Adjective

(c) Plural-Vebs

Figure 4: Projections of Vector Offsets for different categories of Word-Pairs

5.3 Low rank approximations of word vectors

Ideally, we would like to be able to fit a variety of subspaces to all the word vectors. However, in practice the number of word vectors in our embedding space may be too large to afford running a subspace clustering algorithm on the entire set. To with this problem, we generate sets of words related by a high level concept and explore how well the word vectors associated with the set is approximated by a rank k subspace, for varying k. If the set of vectors is well approximated by a rank k subspace, for k smaller than the original dimension of the space, then we can conclude the set of word vectors does in fact have linear low dimensional structure. Figure 5 shows rank versus the $L2$ approximation error for word vectors in a variety of classes. As the figure shows, some classes are much more amenable to low rank approximations than others. This may partially be a product of our method generating related words (currently being done via a search of the WordNet hierarchy).

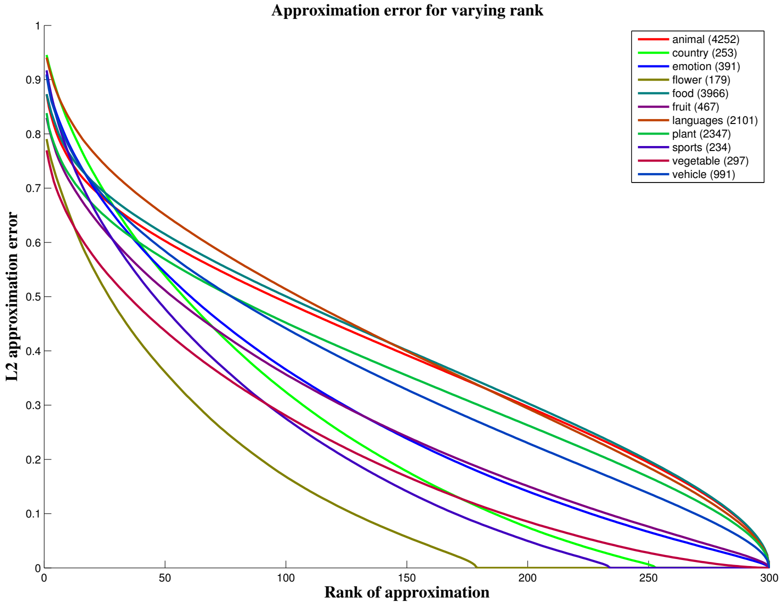


Figure 5: Rank vs. $L2$ error for different sets of word vectors

5.4 Exploring planar structure of analogical pairs

In an attempt to gain some insight into the embedding space we did the following:

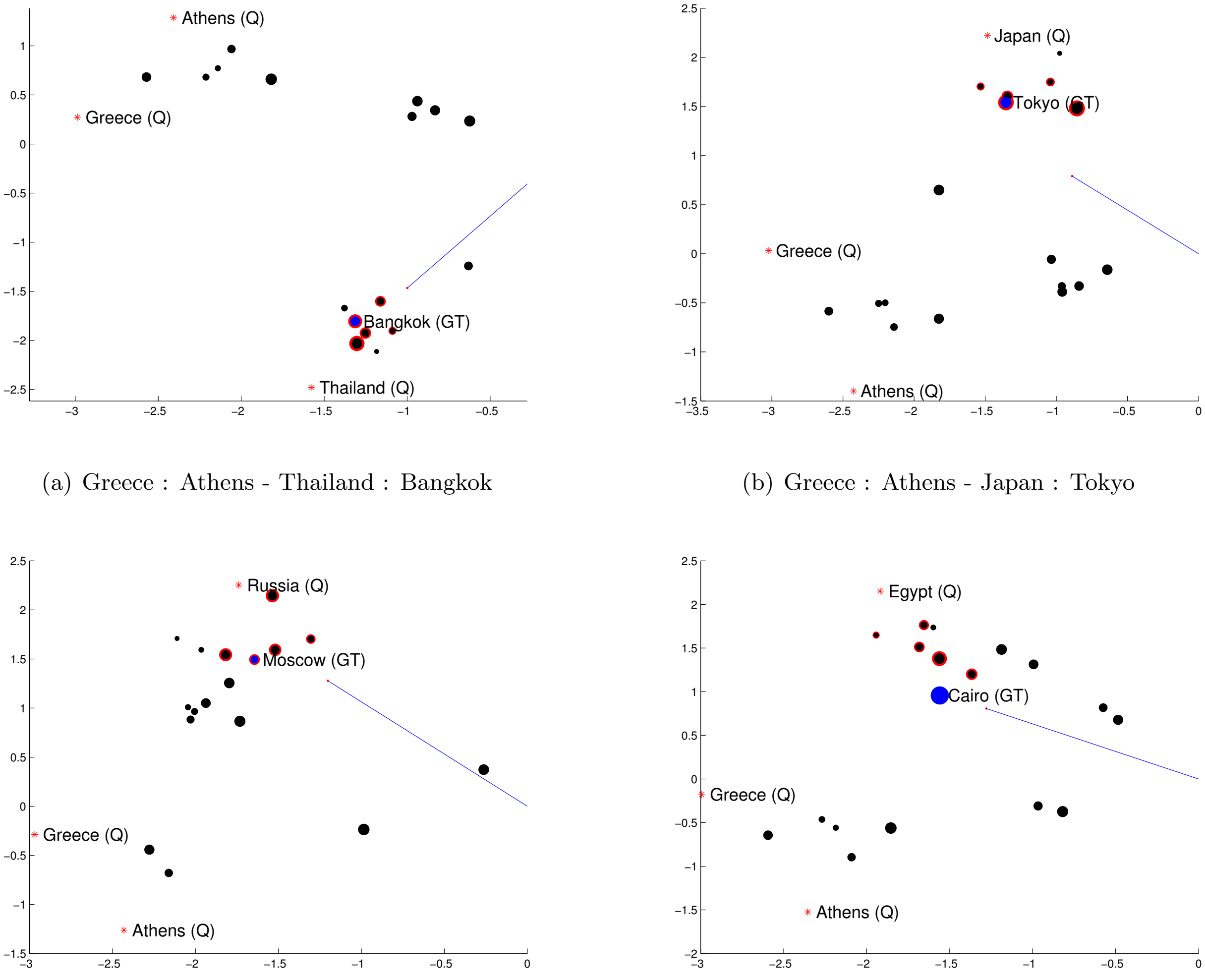
1. Take 3 words from one of the cases in the analogical reasoning test set and compute the subspace that best fits the corresponding word vectors.

2. Project all word vectors onto this plane.

3. Throw away all vectors greater than some threshold away from the plane (in the examples plotted below we kept only the closest 20 vectors)

4. Plot the remaining word vectors projected onto the plane, coded (by size where large indicates a greater distance) based on their distance from the plane.

5. Highlight points that would be predicted (using the projected vectors) within the top k (we used $k = 5$) using the vector offset method.



(a) Greece : Athens - Thailand : Bangkok

(b) Greece : Athens - Japan : Tokyo

(c) Greece : Athens - Russia : Moscow

(d) Greece : Athens - Egypt : Cairo

Figure 6: Analogical reasoning word vectors projected onto 2D plane

5.5 Comparing Models

In this section, we compare the different models based on their top 5 accuracy on the different sections of the analogical reasoning test.

Additionally, we depict the values of the

6 Conclusion