Machine Learning and Computational Statistics: Project Report

Emily Denton (eld297) & Rahul Gopalkrishnan (rg2451)

1 Introduction

Neural network based models for computing continuous vector representations of words have gained popularity. In particular, Mikolov *et al.* [**?**, **?**] have proposed two new models, called the Continous Bag-of-Words (CBOW) model and the Skip-gram model, for learning a continuous embedding space for words from raw text data. Such representations are useful as inputs to NLP applications since the embedding space learned may contain many interesting properties. Recent work*et al.* [**?**] showed that the embedding space learned from these models has an interesting linear structure that can be exploited to finding analogical relationships between words. They show that analogical questions of the form 'King is to man as Queen is to ____' can be solved by algebraic operations on the word vectors. In this case, they compute the vector (King - man + Queen) and then search for its nearest word vector. The result should be woman. This could potentially be useful in tasks such as Information Extraction where given a set of known analogies, one would like to discover new vector pairs that have a similar relationship.

Compared to related work in learning language models[**?**, **?**], the neural language models are log linear, and thus have a lower computational complexity. To our knowledge, given how recent the work by Mikolov et. al. is, it is unclear if anyone has managed to successfully discover relationships automatically from a set of word vectors. Our aim is a first attempt at doing so.

Our approach uses techniques in unsupervised learning such as clustering and SVD to analyze the structure of the word relationships. We separate our results into several sections. Since we do not have access to the amount of text data that Google possesses, we present results that attempt to quantify the effect of training data size of the quality of resulting vector representations. We investigate the differences between the CBOW and the Skip-Gram models with respect to their performance on the analogical reasoning dataset. Next, we visualize the lower-dimensional representation of the analogies to attempt to discover the underlying structure. Finally, we attempt to learn the analogies automatically.

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

Mikolov *et al.* [**?**] 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 **??**) 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 **??**. 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. Mikolov *et al.* [**?**] 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 top-k accuracy is defined as the number of times D appears in the set of k closest words to v. 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 **??** illustrates this intuition.

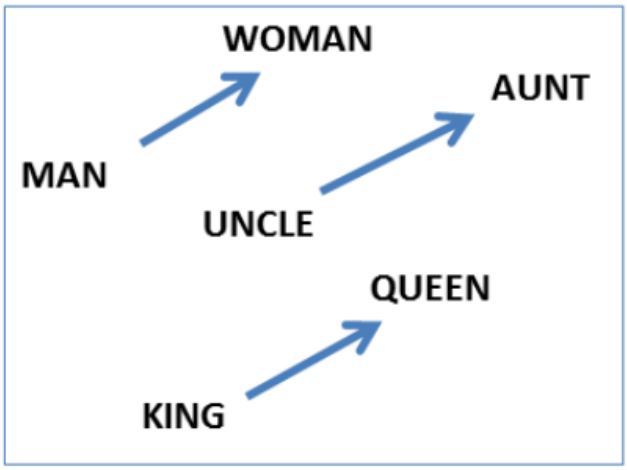


Figure 1: Vector offsets for three analogous word pairs.

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 that we trained. The Google vectors outperform both of our trained models for all k. This is to be expected since the Google vectors were trained on far more data than we have access to.

Our CBOW model performs consistently worse than our Skip-Gram model. We hypothesize that this is a factor of the amount of training data we used to train our models. 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 might 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 **??** plots the top-k accuracy as a function of k for the three models split up by analogy types. Figure **??** 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 (2) whereas the CBOW model performs better on the questions of type (2). We hypothesize 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%

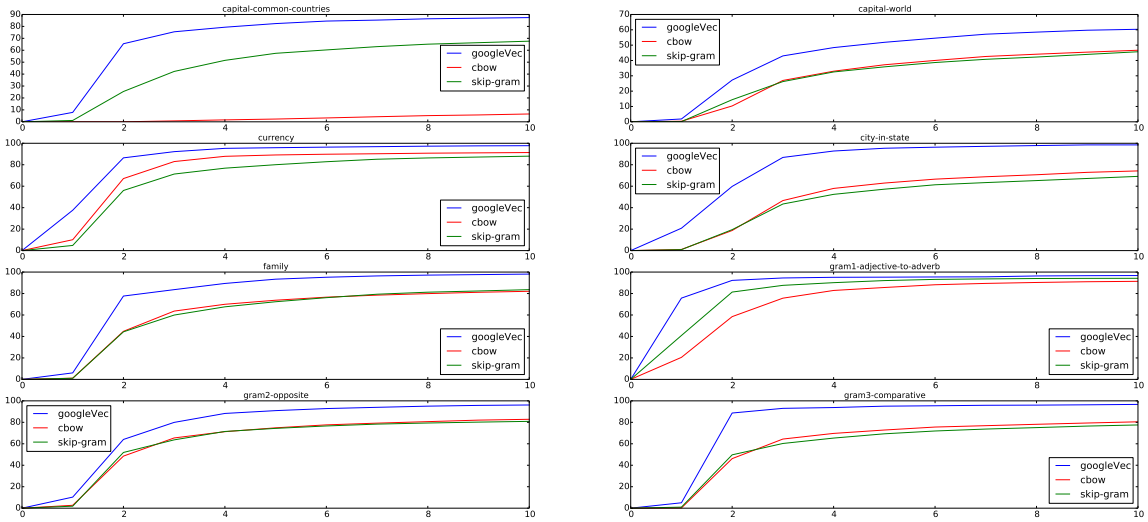


Figure 2: Top K accuracy for increasing K.

3.2.2 Visualizing low dimensional approximations of embedding space

We explored the low dimensional structure of different words used in the analogical reasoning set. In an attempt to gain some insight into the embedding space we did the following:

1. Take 3 words from one of the quartets in the analogical reasoning test set and compute the subspace that best fits the corresponding word vectors. (For example, fine the plane that best approximates $\phi(Paris), \phi(France)$ and $\phi(London)$.

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 based on Euclidean distance from the plane (the radius of the point is proportional to the distance, so larger implies farther away).

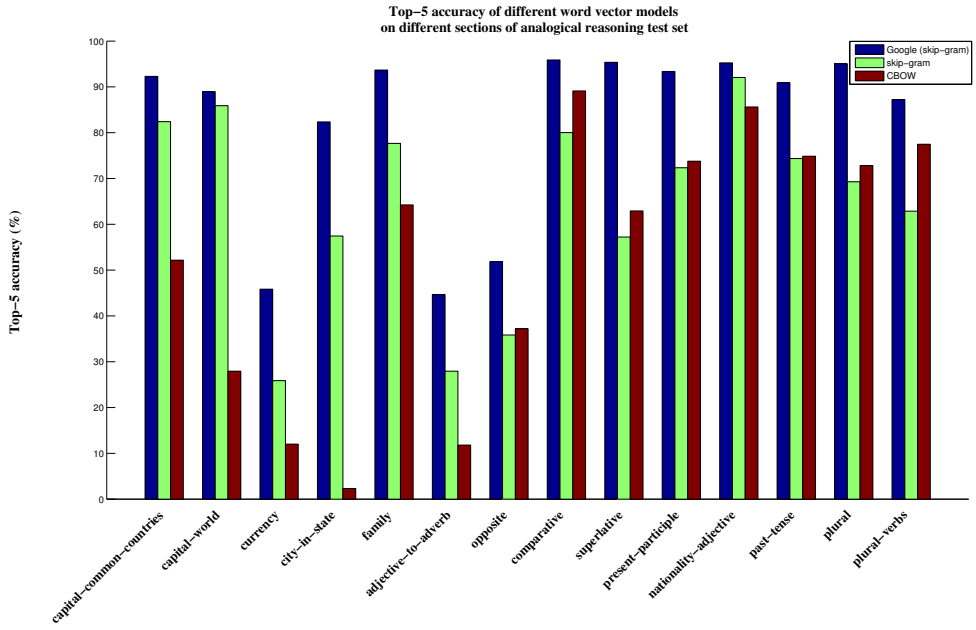
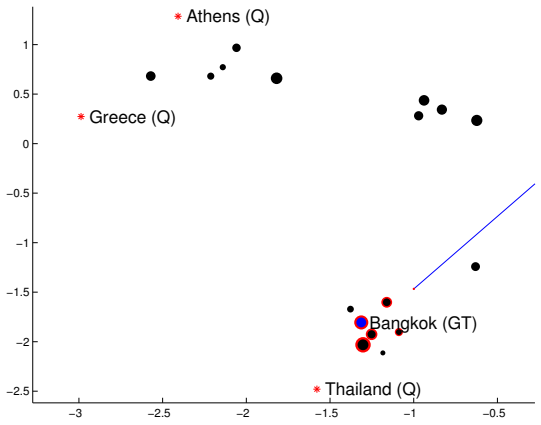


Figure 3: Top-5 accuracy per analogy question type.

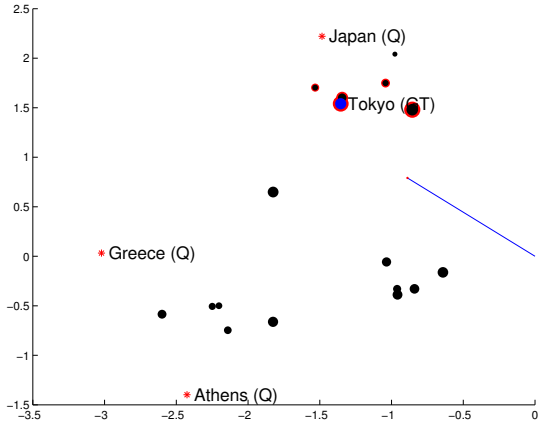
5. Highlight points that would be predicted (using the projected vectors) within the top k (we used $k = 5$) using the vector offset method.

Figure **??** shows for of these such plots using country-capital analogies. In the plots, the red $*$'s denote the analogical query vectors, $\phi(A), \phi(B), \phi(C)$, projected onto the plane of best fit. The red $\cdot$ denotes the vector computed with the vector offset method $(\phi(B) - \phi(A) + \phi(C))$ projected onto the same plane. A blue line is drawn format he origin to this point to elucidate the direction of the vector (recall we only care about the direction of the vector, not it's magnitude). The true solution vector, $\phi(D)$, projected onto the same plane, is plotted with a blue circle. The radius of the circle indicates the distance the point lies form the plane. Finally, the black points denote the 20 closest word vectors to the plane, where again, the size of the point indicates distance front he plane. The points that would be predicted as being in the top-5 solution set are highlighted in red. Figure **??** (a)-(c) are examples where the correct answer is found int he top-5 (as indicated by the red circle around the blue point). Figure **??** (d) shows an examples where the correct word is not found in the top-5 set. This case is interesting because the true solution is very close (based on cosine distance) to $(\phi(B) - \phi(A) + \phi(C))$, however it was too far away from the plane to be found.

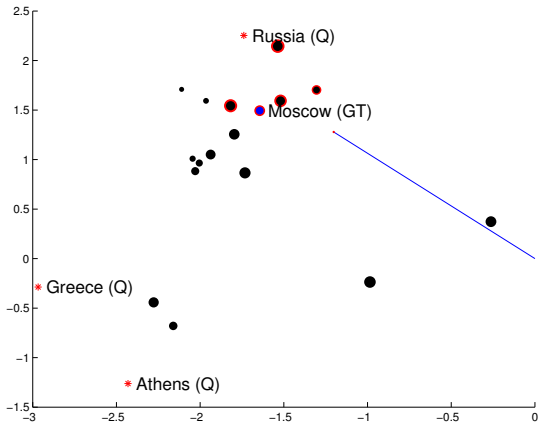
One could imagine constructing a new method of answering the analogical query questions using this information by first restricting the set of possible solution vectors to only the p closest ones to the plane and then doing the top-k search for a solution. We tried this experiment for a variety of k and p and did not find it to perform better than the original method. This is to be expected given the huge amount of information that is being thrown away. However, it is interesting that the method does allow some questions to be answered correctly. We achieved a 30% top-5 accuracy for $p = 20$, which means that a significant amount of information is retained after projecting



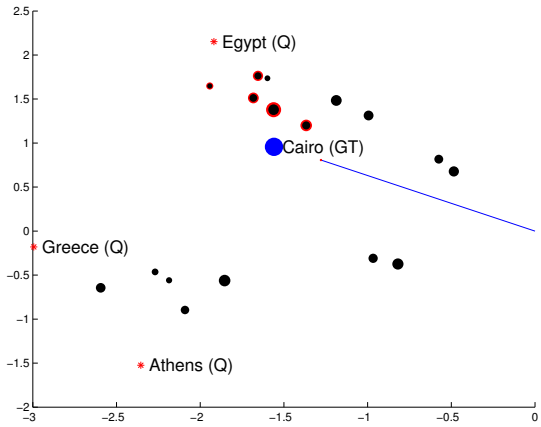
(a) Greece : Athens - Thailand : Bangkok



(b) Greece : Athens - Japan : Tokyo



(c) Greece : Athens - Russia : Moscow



(d) Greece : Athens - Egypt : Cairo

Figure 4: Analogical reasoning word vectors projected onto 2D plane

down into 2 dimensions.

3.2.3 Finding analogical relations

4 Conclusion

In conclusion, this report details our investigations into neural language models and the properties of the continuous vector representations that result from training these models. We show that having more data results in vectors whose linear properties are more amenable for use in finding analogies. We conjecture that the vast amounts of data that Google trains these models on allows the vectors to consistently outperform any models that we train when evaluated on the analogical reasoning dataset. We visualize the underlying structure of the linear properties and see that there exists some noisy linear structure in two and three dimensions. Our results from clustering finds analogies absent in the original test set. Such a model could be used by asking which cluster a new vector offset pair fits best in. Most of our work was done using the vectors provided by Google. It would be interesting

to see if we could still find interesting analogies when the models are trained with limited data. While clustering analogies does indeed let us find interesting analogies. We are currently evaluating supervised linear models to predict different kinds of analogies when given a pairs of vector offsets.

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 **??** displays the results of accuracy for varying k.

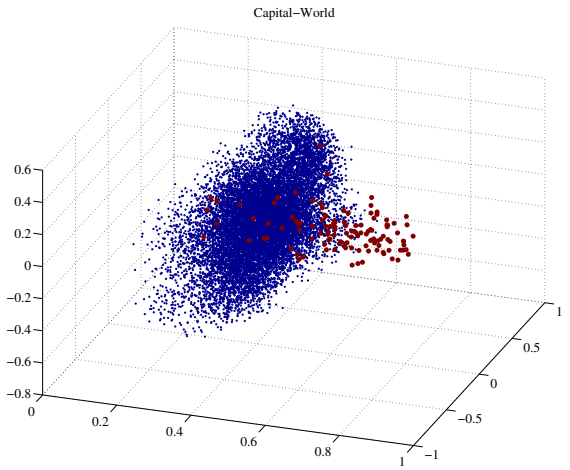
Table 3: Top k Accuracy on Analogical Reasoning Test

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%

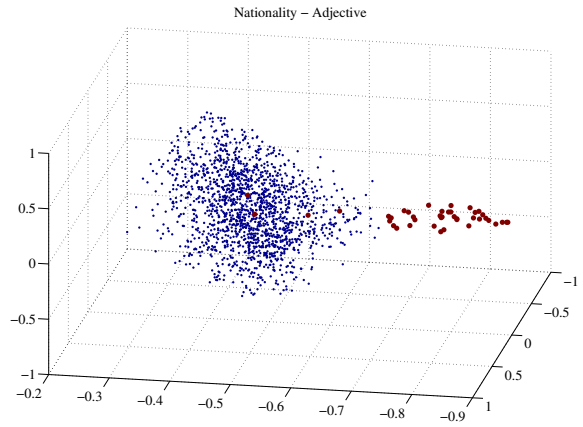
As we can see, the vectors trained on Google's Dataset outperform the models we train by atleast 13% in terms of accuracy on the analogical reasoning test. This likely has to do with the vast differences in training data available to us versus Google. Between the two models, the skip-gram architecture 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

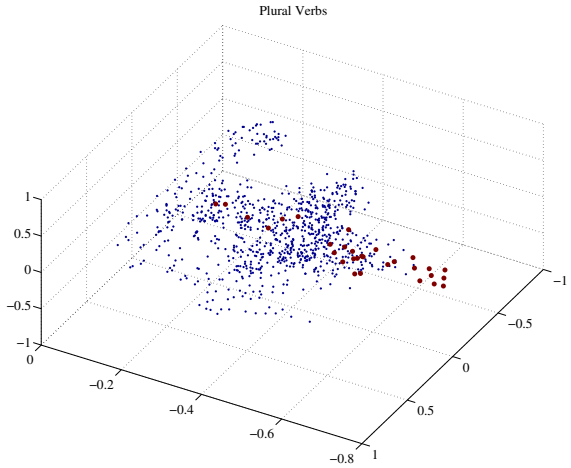
One of the goals of this project is to automatically discover relationships between words pairs. We hypothesized that valid word pairs have vector offsets that lie on a lower dimensional subspaces. To test this hypothesis, we run the following experiment. Using the analogical reasoning task, which gives 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 **??**(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 5: 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 **??** 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).

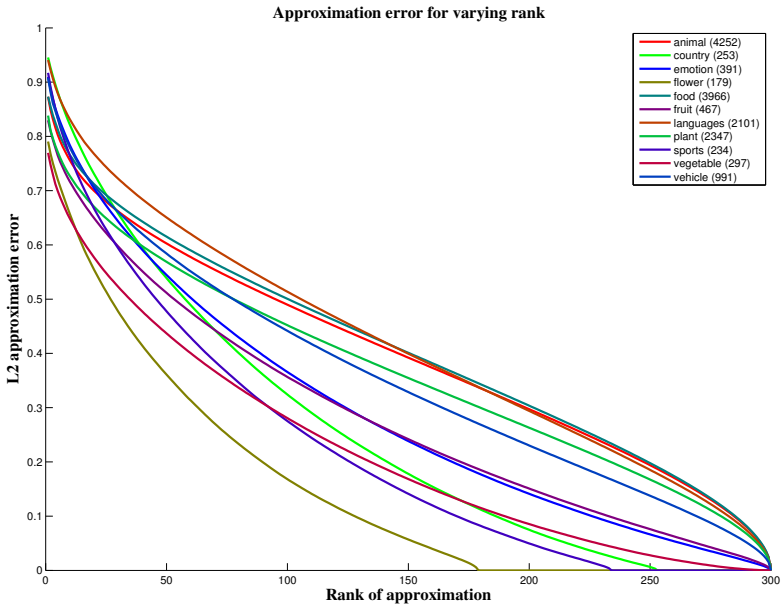


Figure 6: Rank vs. $L2$ error for different sets of word vectors

5.4 Exploring planar structure of analogical pairs

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

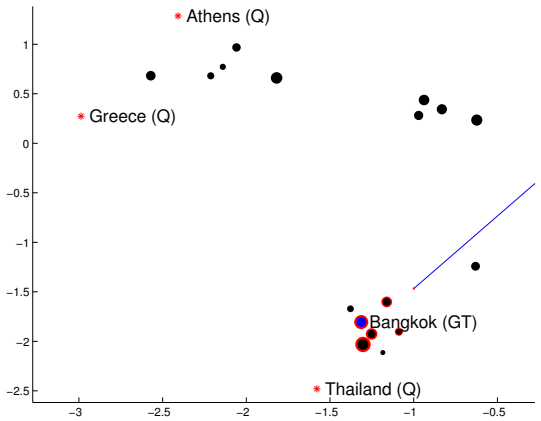
1. Take 3 words from one of the cases in the analogical reasoning test set and compute the subspace that best fits the corresponding word vectors.

2. Project all word vectors onto this plane.

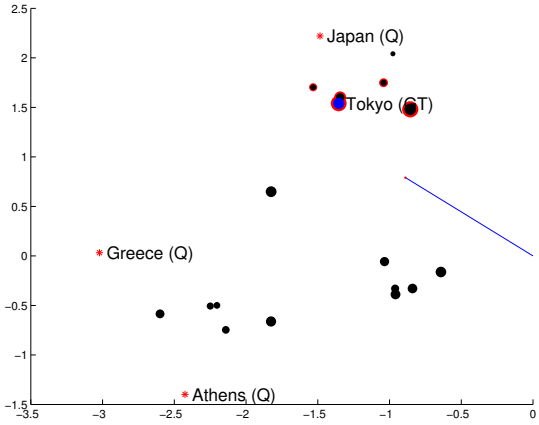
3. Throw away all vectors greater than some threshold away from the plane (in the examples plotted below we kept only the closest 20 vectors)

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.

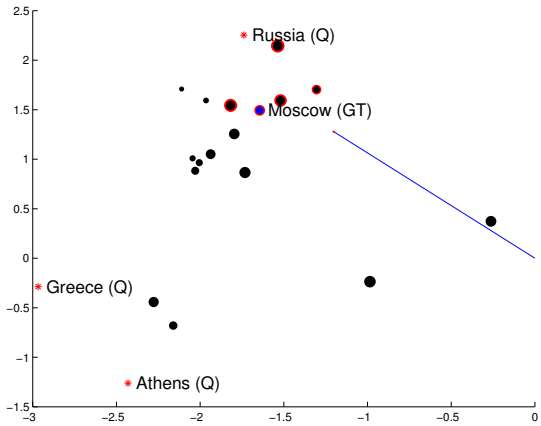
5. Highlight points that would be predicted (using the projected vectors) within the top k (we used $k = 5$) using the vector offset method.



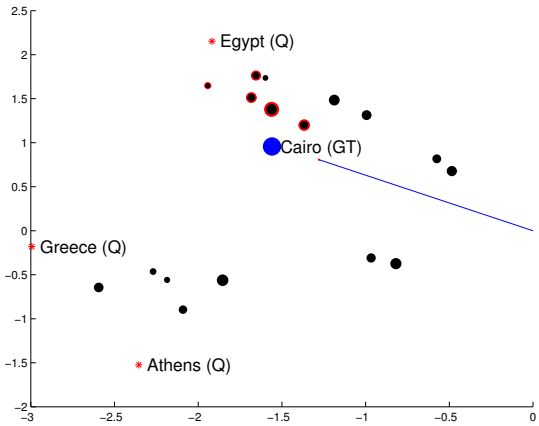
(a) Greece : Athens - Thailand : Bangkok



(b) Greece : Athens - Japan : Tokyo



(c) Greece : Athens - Russia : Moscow



(d) Greece : Athens - Egypt : Cairo

Figure 7: 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