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.* [1, 2] 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.* [3] 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[4, 5], 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

[3] 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
family	2467	brother : sister mother : father
adjective-to-adverb	506	amazing : amazingly calm : calmly
opposite	992	acceptable : unacceptable aware : unaware
comparative	813	bad : worse big : bigger
superlative	1331	bad : worst big : biggest
present-participle	1122	code : coding dance : dancing
nationality-adjective	1056	Albania : Albanian Argentina : Argentinean
past-tense	1599	dancing : danced decreasing : decreased
plural	1560	banana : bananas bird birds
plural-verbs	1332	eat : eats generate : generates

3.2 Results

3.2.1 Measuring linguistic regularity via analogies

3.2.2 Visualizing low dimensional approximations of embedding space

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. We use clustering techniques to attempt to find analogies and discover that analogies that were absent in the original test set are picked up my the method. 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.

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 2 displays the results of accuracy for varying k.

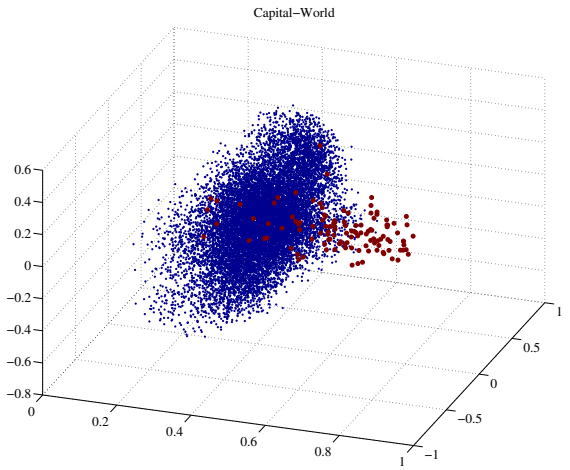
Table 2: Top k Accuracy on Analogical Reasoning Test

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%

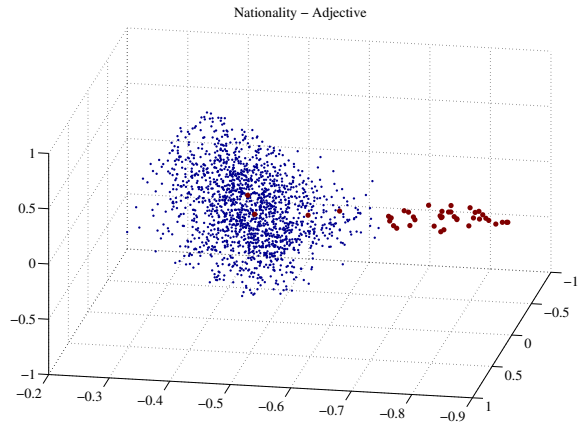
As we can see, the vectors trained on Google's Dataset outperform the models we train by atleast 13% in terms of accuracy on the analogical reasoning test. This likely has to do with the vast differences in training data available to us versus Google. Between the two models, the skip-gram architecture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

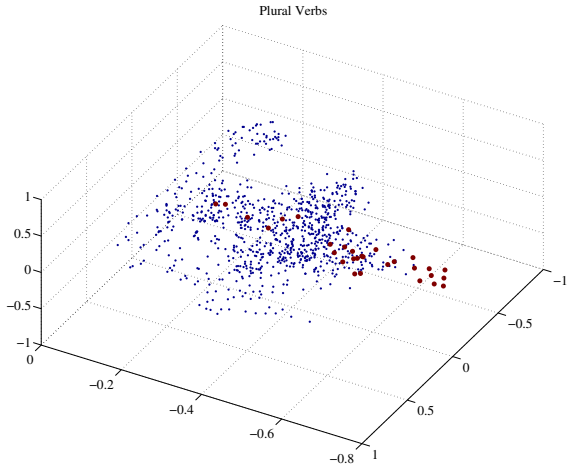
One of the goals of this project is to automatically discover relationships between words pairs. We hypothesized that valid word pairs have vector offsets that lie on a lower dimensional subspaces. To test this hypothesis, we run the following experiment. Using the analogical reasoning task, which gives 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 3(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 1: 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 2 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).

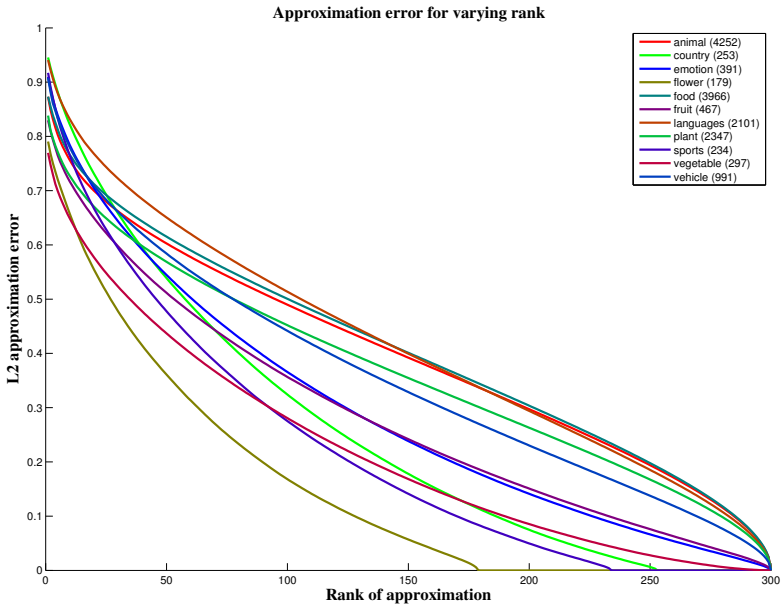


Figure 2: Rank vs. $L2$ error for different sets of word vectors

5.4 Exploring planar structure of analogical pairs

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

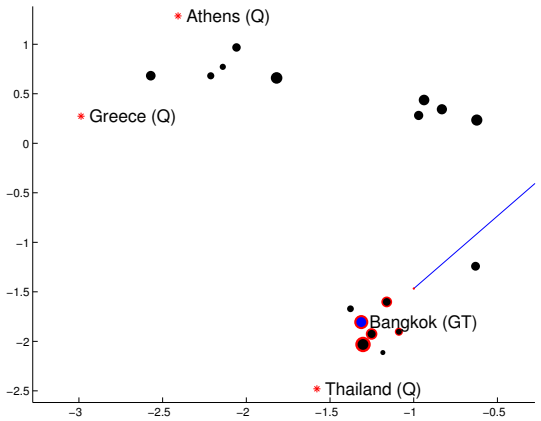
1. Take 3 words from one of the cases in the analogical reasoning test set and compute the subspace that best fits the corresponding word vectors.

2. Project all word vectors onto this plane.

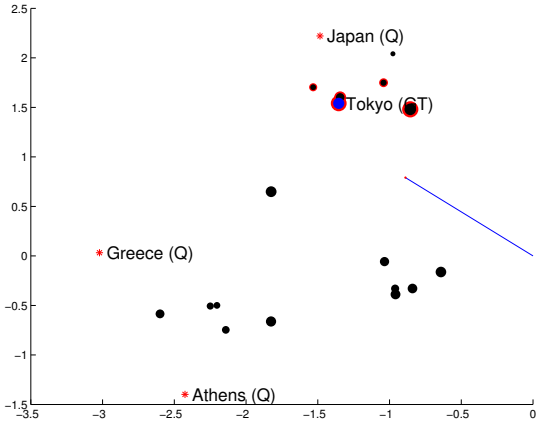
3. Throw away all vectors greater than some threshold away from the plane (in the examples plotted below we kept only the closest 20 vectors)

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.

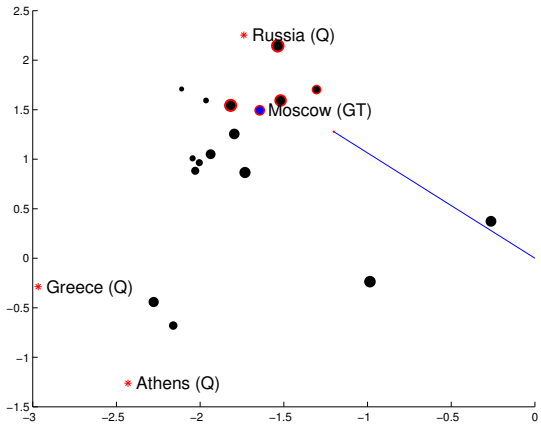
5. Highlight points that would be predicted (using the projected vectors) within the top k (we used $k = 5$) using the vector offset method.



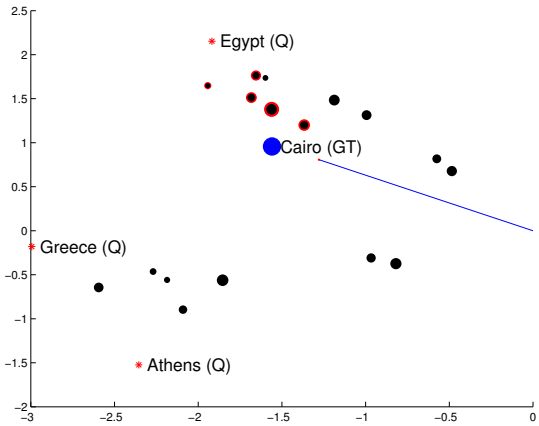
(a) Greece : Athens - Thailand : Bangkok



(b) Greece : Athens - Japan : Tokyo



(c) Greece : Athens - Russia : Moscow



(d) Greece : Athens - Egypt : Cairo

Figure 3: 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

References

[1] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. ICLR Workshop (2013)

[2] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. NIPS (2012)

[3] Mikolov, T., tau Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. HLT-NAACL (2013)

[4] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. JMLR (2003)

[5] Mikolov, T., Karafit, M., Burget, L., Cernock, J., Khudanpur, S.: Recurrent neural network based language model. INTERSPEECH (2010)

[6] Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. CoRR **abs/1309.4168** (2013)