Network analysis of word embeddings

Supervised by
Aurélien Bellet
aurelien.bellet@inria.fr

Yao FENG
Telecom ParisTech
yao.feng@telecom-paristech.fr

Chen WEI
Telecom ParisTech
chen.wei@telecom-paristech.fr

Abstract

How to adequately represent words as vectors is a long-standing problem in text mining and Natural Language Processing. In this project, we use several ways of constructing a graph over words from the embeddings, and study various properties to evaluate their performance.

1 Introduction

1.1 Word embedding and *mangoes*

Word embedding is an important and fundamental technique in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. The proposition of word embedding opens up a new path for many NLP tasks such as syntactic parsing [1], semantic similarity and sentiment analysis, enabling researchers to apply machine learning methods to linguistic problems.

There are many research groups that work on word embedding both theoretical study and implementation. The most well-known one may be Word2vec[2] created by a team at Google, using continuous bag-of-words and skip-gram architectures. Also Stanford University's GloVe[3].

Mangoes is a python based toolbox for constructing and evaluating word embeddings. For the evaluation, the toolbox provides statistical and intrinsic evaluation methods such as analogy tasks, similarity tasks and outlier detection tasks which are all classic linguistic method to evaluate word embedding system. The objective of our work will be evaluating mangoes with graph based method which will give a novel perspective of this problem.

1.2 Evaluation with network analysis

Graph representation is a new approach to evaluate and visualize the existing word embedding models. In many applications, word vectors can be represented as a graph with various graph construction methods[4]. Such graphs can be useful for word sense induction, word clustering, discovering words centrality in semantic networks, and for visualizing semantic relations between words [5].

In our work we use python library NetworkX[6] to build graph and analyses the graph properties to evaluate the word embedding.

2 Graphs based approach

2.1 Graph construction

There are more than one way to represent word vectors in a graph. The most obvious approach is to create a fully connected graph with words as nodes and weights on edges equal to cosine similarity between word pairs. However, for large corpus (millions of words in vocabulary) this can be incredibly costly in terms of memory and processing time.

Thus there are ways to approximate these semantic similarities without construct a fully connected graph. In our work we use three graph construction methods: nearest neighbor graph (knn-graph), approximate knn-graph and relative neighborhood graph (rng) [4].

2.1.1 knn graph

In knn-graph, nodes are only connected to its k nearest neighbors, which can well show the relation between every words in word embedding. The similarity between two nodes is defined by Gaussian similarity function

$$s_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$$

where σ controls the width of the neighborhoods. In our experiments σ = 1, and K= 5.

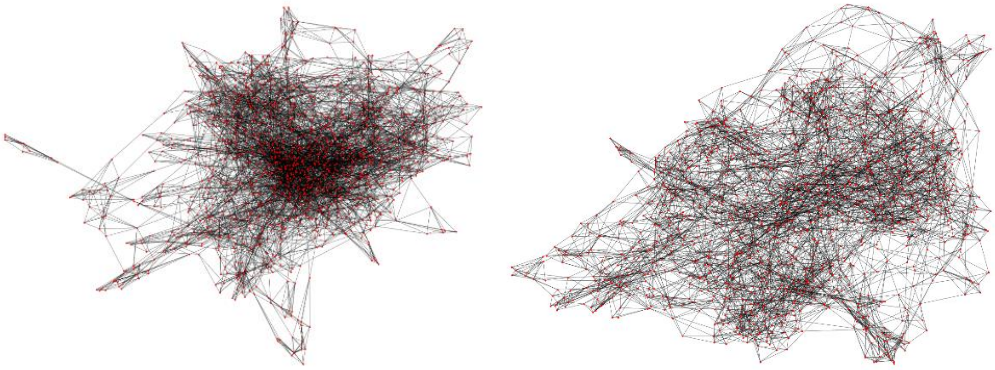
2.1.2 Approximate knn-graph

The complexity of constructing k-nn graph is $O(n^2\log(n))$ which is unacceptably high especially for large corpus. So we try to find closest neighbors of a node by searching the nodes around it instead of checking all, thus greatly reducing the complexity of searching nearest neighbors. For instance KD-Trees, Cover Trees and ball trees are all useful data structures for nearest neighbor searches.

In our work we use Annoy a C++ library with Python bindings to search for approximate nearest points of a given query point in space. In this approach the number of trees n_trees is the parameter to control the approximation degree. A larger value will give more accurate results, but longer runtime.

On top of the word vectors 50 dimension generated by mangoes with window size=2, shift=0 (*ppmi_svd_1500words_win2*), we plot the graph constructed respectively by knn and approximate knn. These graphs will only include 1500 most common words in English to keep it easy to visualize.

We find approximate knn will result in a different graph from standard knn. It tends to create a more even graph whose degree distribution is more uniform than that of knn.



(a) knn graph of 1500 words (b) approximate knn graph of 1500 words, with n_tree=100

Figure 1: knn vs approximate knn with k=5, sigma=1

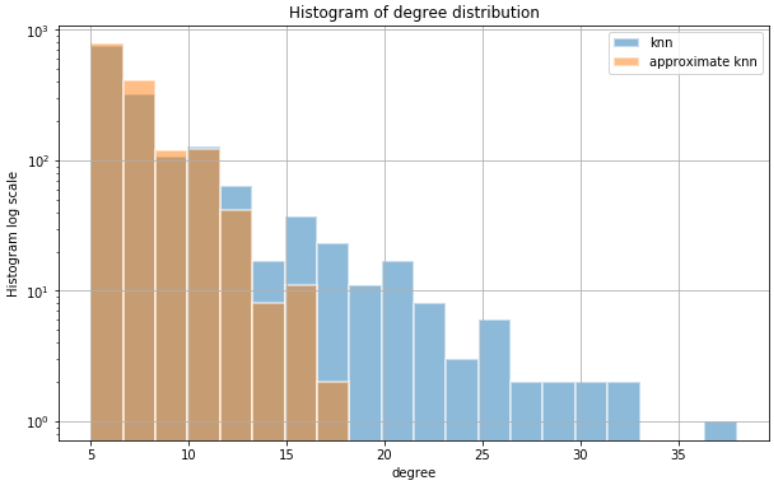


Figure 2: Histogram of degree distribution, graphs constructed by knn and approximate knn

2.1.3 Relative neighbourhood graph

In rng, a node A connects to B if the region between them are empty. The region can be defined as the intersection of two spheres with centers in A and B and radius $d(A, B)$. So rng contains the information about directions while knn-graph only cares about distance. However, as rng ignores many neighbors in the same directions, it is not a good idea to use rng for global graph evaluation. Given k nearest neighbors of a node A, we will be able to build a rng tree rooted in A to show the local direction information.

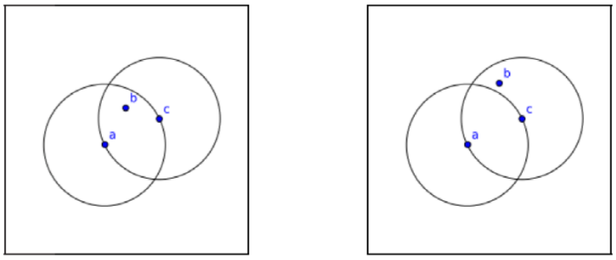


Figure 3: Example of when point b is between point a and c (left), and when it is not (right).

3 Graph properties

3.1 Central nodes with various centrality notions

The most central nodes in the graph are usually the most important words in the vocabulary. However there exist more than one measurement of centrality, and we will focus on four notions of centrality:

- Degree centrality measures how many neighbors a node has.
- Closeness centrality measures the mean distance from a vertex to other vertices.
- Betweenness centrality measures the extent to which a vertex lies on paths between other vertices.
- Eigenvector centrality measures the influence of a node in network. A node is important if it is linked to other important nodes.

Despite different notions, it turns out that more central words always have more importance in the network. The table below shows that various centrality notions give almost the same 10 most central nodes, with the order of one or two words changes.

Table 1: Most central nodes by various centrality notions,
graph constructed on mangoes size = 2, shift = 5, 1500 words, with knn K=5

Centrality notions	*10 most central nodes*
Degree centrality	the, and, a, of, although, only, in, made, both, one
Closeness centrality	the, and, a, of, in, only, although, made, one, should
Betweenness centrality	the, and, a, of, although, only, in, made, one, both
Eigenvector centrality	the, and, a, of, in, only, although, both, one, made

3.2 Diameter

Diameter of a graph is the length of the **longest shortest path** between any two nodes, which is the greatest distance between any pair of vertices. The diameter should be reasonably large. If it is too small, all nodes are very close and we may not get good partitions for communities.

3.3 Clustering coefficient

Clustering coefficient measures how well nodes tend to cluster together. It presents the number of triangles in graph.

For word embedding analysis, if the clustering coefficient is too large, every words will be intimately connected to each other then it will be hard to partition the words into good communities. In this case two communities like ordinal numbers and cardinal numbers may be regarded as one. If the clustering coefficient is too small, the nodes are not well connected. We may not be able to find meaningful partitioning.

Table X will show the four graph properties with varying K in knn. The results tell us the role of parameter K when constructing the graph is very important. It will directly determine the four properties shown in the table. When K increases, the max degree and clustering coefficient will be naturally increase, while the diameter and community modularity will decrease, which means the graph becomes more connected.

Table 2**:** Graph properties,
graph constructed on mangoes size = 2, shift = 5, 1500 words, with knn

	K=3	*K=5*	*K=10*	*K=20*
Max degree	26	38	83	206
Diameter	Not connected	13	8	5
Clustering coefficient	0.295	0.318	0.346	0.365
Community modularity	0.753	0.687	0.594	0.484

3.4 Community

Graphs can be partitioned to several communities. Different from clustering methods like k-means or spectral clustering, we do not need to specify the number of communities. By maximizing the modularity of the graph, it gives a best partition of communities. The modularity measures the strength of the division of a network into communities, i.e. networks with high modularity have dense connections between the nodes within communities but sparse connections between nodes in different communities.

For word embedding analysis, different communities often represent different themes and categories, such as people names, ordinal numbers, cardinal numbers, month, etc. Networkx provides the community.best_partition() which can present the community partitioning at different scales by changing the resolution parameter.

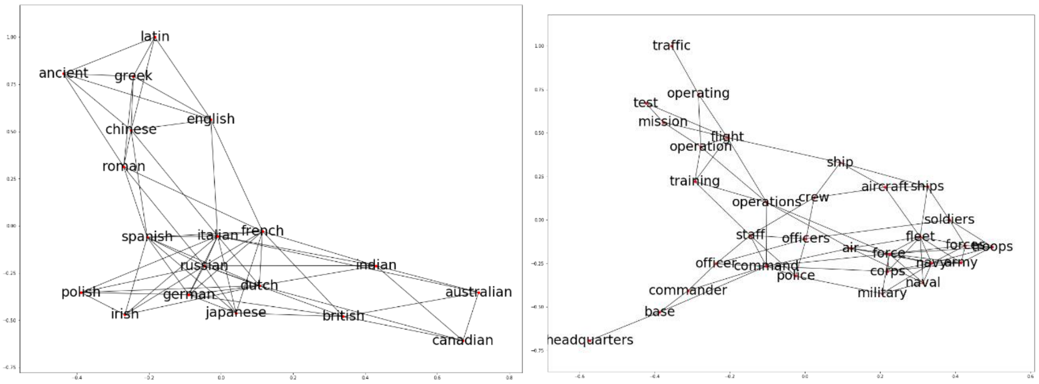


Figure 4: 20th and 21st communities and their member words

By maximizing the modularity, with resolution 0.5, best_partition() will partition the whole vocabulary containing 1500 into 39 communities. In figure X, we present the 20th and 21st community samples in the subgraph and mark the word label representing each node. We find all words share the same topic "language" within 20th community and "military" 21st. This rule also appears in other communities in which the words turn out to share the same topic or same part of speech. Due to this interesting property of community, we will be able to evaluate the word embedding in section 4 by checking if the vocabulary can be properly partitioned.

4 Parameter evaluation

4.1 Evaluation on full vocabulary

In mangoes word embedding there are two parameters which affect the performance, the window size when counting the word co-occurrence and the shift when calculating the ppmi. In order to analyze the two parameters in mangoes, we use the pre-trained model based on large scale corpus (Wikipedia English 2013 tokenized, 3.2G) which contains approximately 120,000 words. Due to the huge number of nodes we'll apply the approximate knn to construct the graph, and the results are shown in the following table.

Table 3**:** Evaluation on window size and shift with full vocabulary

Graph properties		*Window size = 2*	*Window size = 5*
Number of nodes		196,789	123,293
10 most central nodes *(degree centrality)*	Shift = 1	*hole, nakdong, farmersdaughterhotel, patagoniaadventureracing, olifants, seraikis, fullscreenmusic, ferzetti, city, missglobegermany*	*ukcontentframeset, julie's, taylor's, imriel, award', thompson's, laura's, thynne, gonçalves, nightcrawler*
	Shift = 5	*daicos, fenter, olifants, nakdong, ictur, weeks, countrys, skorepa, bring, imriel*	*asthe, thenew, award, awards, ukcontentframeset, simplythe, jewett, triangle, lyman, thefirst*

Clustering coefficient	Shift = 1
	Shift = 5
		0.129
		0.189
			0.199
			0.245

Number of communities	Shift = 1
	Shift = 5
		104
		128
			106
			120

Community modularity	Shift = 1
	Shift = 5
		0.784
		0.840
			0.834
			0.864

4.2 Evaluation on selected vocabulary

We may notice from the central words that there exist many words which are not common English in large text corpus. This may be caused by the inaccurate results of approximate knn. In large corpus containing 120,000 words, this error will be amplified. Therefore, we try to reduce the vocabulary by only focusing on 1500 common English words that we discussed in section 3.

From shift = 2 to shift = 5, the community modularity increases, which implies a better partition. The most central nodes of embedding of shift = 5 are mostly prepositions and conjunctions while those of shift = 1 are nouns, despite the window size.

Table 4: Evaluation on window size and shift with selected vocabulary

Graph properties		*Size = 2*	*Size = 5*
10 most central nodes	*Shift = 1*	*nonfamilies, result, households, householder, cdp, rest, islander, median, capita, addition*	*nonfamilies, householder, and, the, households, cdp, made, subsequent, well, residing*
	Shift = 5	*the, and, a, of, although, only, in, made, both, one*	*only, however, although, of, but, then, since, the, they, which*
Clustering coefficient	*Shift = 1*	0.355	0.340
	Shift = 5	0.294	0.257
Diameter	*Shift = 1*	10	8
	Shift = 5	9	9
Number of communities	*Shift = 1*	38	43
	Shift = 5	39	41
Community modularity	*Shift = 1*	0.511	0.468
	Shift = 5	0.534	0.541

5 Comparison with word2vec, GloVe

The pre-trained word2vec and glove model are based on large text corpus (Wikipedia English 2016 tokenized, 14G). The embedding of word2vec and glove both contains 10,000 words while the embedding of mangoes contains 196,789 words. To balance the scale of embedding, we extract several common words from the 1500 words provided and there are 1464 words in embedding of word2vec and glove and 1494 words in embedding of mangoes. We use embedding of shift 5 and size 2 for mangoes to compare with embedding of size 2 for word2vec and glove.

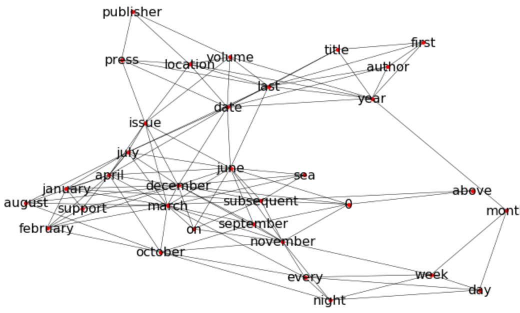
The 10 most central nodes are found by degree centrality. The community are found by best_partition function of NetworkX and the resolution parameter is set to 0.5. The parameter resolution could show community in different scales: small resolution gives small communities (the number of communities is larger).

Table 5**:** Evaluation of 3 word embedding implementations with selected vocabulary

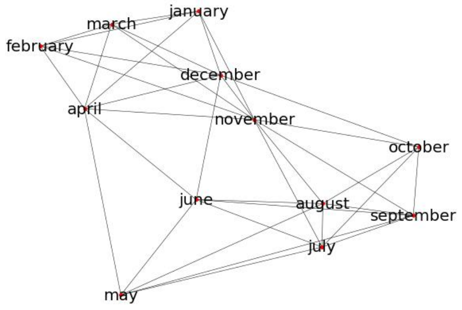
	Mangoes shift = 5	*word2vec*	*Glove*
10 most central nodes	the, and, a, of, although, only, in, made, both, one	and, the, however, therefore, thus, both, in, another, itself, project	and, however, even, both, as,

			addition, rather, while, fact, the
Clustering coefficient	0.294	0.304	0.288
diameter	9	9	8
modularity	0.534	0.731	0.638
Number of communities	39	37	71
Community partitioning	0: titles, family 3: languages 5: cities, geography 15: ordinal numbers 22:cardinal numbers, colors, frequency 25: seasons and time 27: laws 32: months and time 34: country, direction 36: letters 37: names	5: ordinal numbers 7: cardinal numbers 9: months 14: countries, cities 16: languages 17: modal verbs 18: directions 19: family 21: names 25: competition 30: royal titles 33: colors 36: gender	6: ordinal numbers 38: colors 45: directions 23,26: months 32: languages 21,41: names 28: social relations 29: country, cities 34: roman numbers 52: season 35: gender

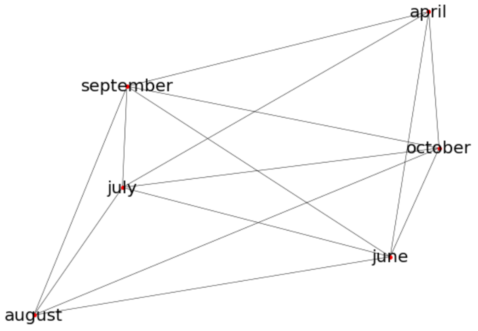
These 3 embeddings have similar clustering coefficients and diameters. If we look at the modularity, word2ve is better than GloVe, better than Mangoes. However, if we look at the communities, word2vec tend to have very large communities (more than 100 words) and very small communities (less than 5 words), which are hardly to identify the topic. GloVe tend to have more and smaller communities, which have been a large one. Mangoes tend to include a few related words in community which should be in other communities. Here are examples of month community.



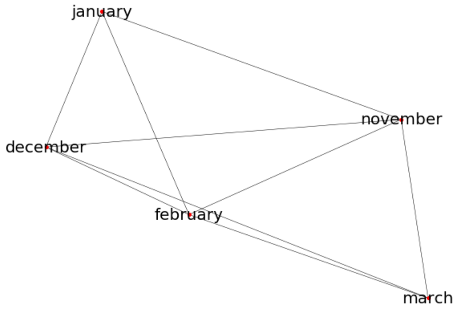
(a)mangoes



(b)word2vec



(c) Glove(23)

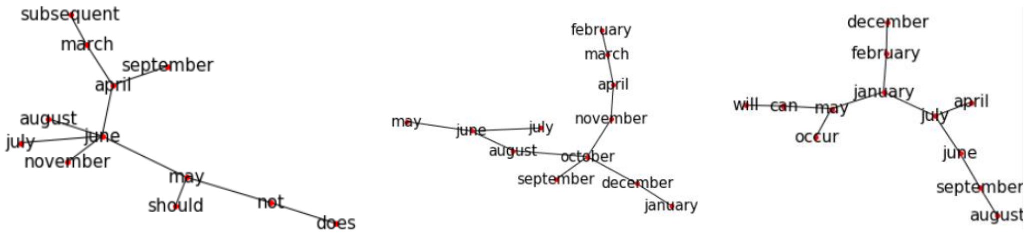


(d)Glove(26)

Figure 5: Communities about “months” of mangoes, word2vec and glove

Word2vec could separate 12 months from other words. Mangoes tend to associate many related words in the same community while Glove divides months into 2 communities. That’s why Glove contains more communities than Mangoes and Word2vec. However, word2vec has very large communities that we can hardly identify the topic

In Mangoes and Glove embeddings, we do not find ‘may’ in community of month. Therefore, we use RNG Tree to present 11 nearest neighbors of word ‘may’.



(a)Mangoes

(b)Word2vec

(c) GloVe

Figure 6: Nearest neighbors of word ‘may’, using rng graph with root word ‘may’

For Word2vec the 11 nearest neighbors are 11 months. For Mangoes, 11 nearest neighbors include months and modal verbs, which are in different directions. The months are in the left side and modal verbs are in the right side. ‘May’ could be divided to community of month or community of modal verb. It depends on which division would increase the modularity.

The results show that ‘may’ is divided in the community of modal verb rather than community of month for Mangoes and Glove, while Word2vec do not regard ‘may’ as a modal verb.

Reference

[1] Socher, Richard; Bauer, John; Manning, Christopher; Ng, Andrew (2013). Parsing with compositional vector grammars (PDF). Proc. ACL Conf.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

[3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation

[4] Gyllensten, A.C. and Sahlgren, M., 2015. Navigating the semantic horizon using relative neighborhood graphs. arXiv preprint arXiv:1501.02670.

[5] Kaspar Beelen, 2015. Visualizing Parliamentary Discourse with Word2Vec and Gephi.

[6] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, “Exploring network structure, dynamics, and function using NetworkX”, in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008