Clustering High Dimensional Vector Space with Graphs Case Study using Word Vectors

Urmas Pitsi 192028IAPM
TalTech

Abstract

Clustering high dimensional vector space is an active research area. It touches all disciplines wherever there something to describe with real-valued vectors. It is especially hot topic in Natural Language Understanding where researchers are continuously trying to discover novel ideas of how extract meaning representations of words, sentences and concepts. In this work we use graphs as a tool to approach the problem of clustering in high dimensional vector space. Our hypothesis is that graphs offer a competitive alternative to traditional approaches in clustering and analysis of high dimensional data. By converting original problem into a graph problem we get all the benefits of graph analysis as granted. Not that our hypothesis is either groundbreaking or novel, but we feel that graphs are quite underutilized given the power and expressiveness they possess. We conclude that graphs are indeed an excellent and easy to implement tool for high dimensional data analysis and clustering. We feel that there are a multitude of various avenues for further investigation using graphs and our work is just a tip of an enormous iceberg.

1. Introduction

Modelling high dimensional vector space is a challenging task. Most real life data mining tasks deal with inputs that are very high dimensional. One could imagine tabular data where observables are in rows and attributes in columns. It could be modelling of clients' behaviour, sensor data, images, text, recommender system etc. One intuitive way of representing high dimensional data is by graph data structures. Graphs are very expressive and powerful for representing complex information. In addition to their expressive power we get thorough analytics from the field of graph theory. In this work we explore how to use graphs for clustering high dimensional input data using word vectors as an example as word vectors are an excellent example of a high dimensional real-valued vector space. There are multiple different implementations of word vectors along with the precalculated embeddings that have been published by their authors. Some of the most popular embedding methods for generating word vectors include word2Vec [1], GloVe [2], fastText [3]. Despite the fact that word vectors are generated through a statistical co-occurrence basis, they seem to capture semantic representations of the words. This particular property of word vectors has been fascinating the field of Natural Language Understanding since 2013 when Mikolov et al. [1] published their paper on word representations in vector space. Hence the huge popularity of word vectors in the field.

We would like to highlight that analysis and methods presented in the current work are not entirely word vector specific. Similar analysis can be performed on any high dimensional data. Word vectors serve as a good example of such data because words and their relations among each other along with semantics is inherently very intuitive and understandable to humans.

1.1. Motivation

Why bother? In many cases it is very difficult or impossible to cluster high dimensional data with "standard" methods and approaches. Whether because of high dimensionality of input data or poor scalability of the chosen method. Graph analysis is an excellent additional tool in our toolbox for solving these kinds of problems. Graphs are very good at representing high dimensional data that has some kind of inherent built-in community structure.

1.2. Goals of the Paper

The main goals of the paper is to show how to:

(i) convert high dimensional real-valued vector space into a graph.
(ii) apply graph clustering on a problem that is represented as high dimensional real-valued vector space.
(iii) visualize high dimensional space, similarities etc more intuitively using graphs.

1.3. What the paper is NOT about

The current work is not a comparative analysis of different clustering methods. Rather we focus solely on graph construction and graph analysis with a single aspect of trying to preserve similarities between items in their closest neighborhoods.

2. Data

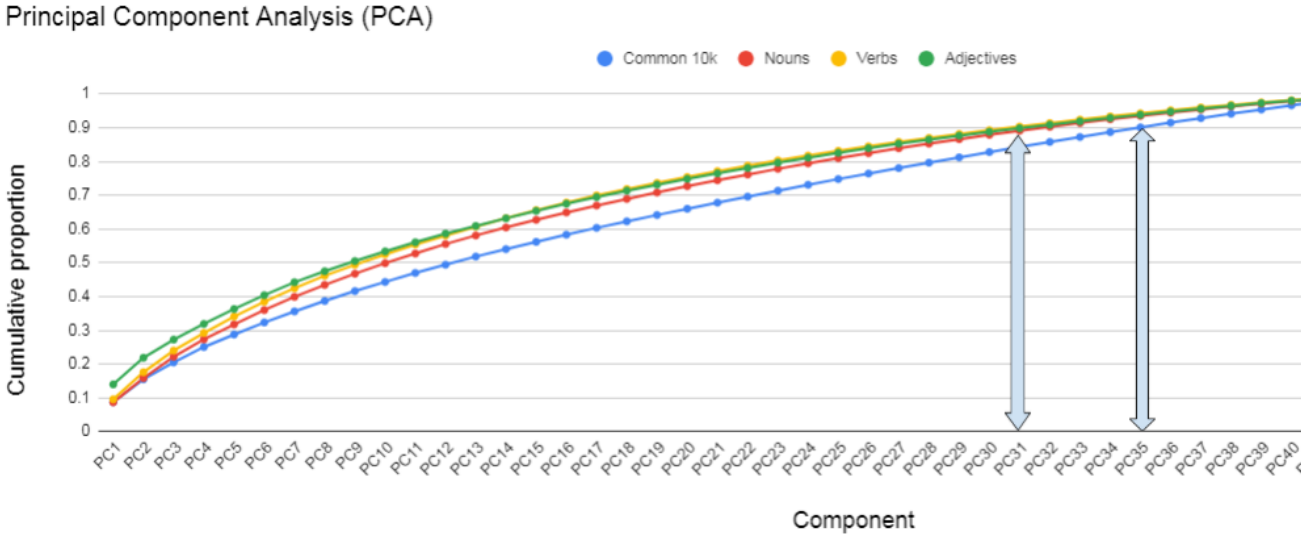
In the analysis we use 50-dimensional GloVe word vectors [2] as a case study for our purpose. The analysis should be quite agnostic of which exact set of word vectors are used. However it could be also a possible future task to compare how different word vectors lend itself to this kind of cluster analysis. We chose GloVe because it has proven to be a pretty robust input in language modelling tasks. Glove word vectors contain 400,000 words/tokens with 50 dimensions each. We extract 4 subsets as follows:
Subset 1: 1000 most common nouns [4].
Subset 2: 1000 most common verbs [4].
Subset 3: 1000 most common adjectives [5].
Subset 4: 10000 most common english words [6],[7].

One reason for extracting subsets is to make calculations more lightweight. The other reason is to compare the properties of grammatically different types of words. So that we can compare nouns with verbs with adjectives. There are very few works recently, if any, that analyze word vectors by splitting the data into different subsets based on grammatical properties. For any other task it would be quite natural to experiment with an innate bias if there is an obvious choice - grammar in case of words. In our case we split the data by nouns, verbs and adjectives. By doing so we can find clusters of similar words inside each group. As word vectors have been generated on the basis of statistical co-occurrence, it might make sense to analyze word types separately. Because some frequently co-occurring verb-noun or adjective-noun pairs do not necessarily share their meaning.

2.1. Dimensionality reduction with Principal Component Analysis (PCA).

As a preliminary step we explore how well does the PCA work on the raw data. Running PCA on our subsets of word vectors we can see that in order to obtain 90% explainability we need at least 31 (of 50) components in case of Nouns, Verbs and Adjectives. In case of Common 10k words we need 35 (of 50) components. We can conclude that standard dimensionality reduction on raw word vector data doesn’t offer us much help. Let’s see if we can improve on that with graph clustering techniques.

Figure 1: Principal Component Analysis.



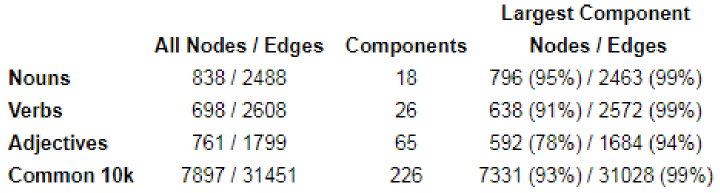
3. Results

3.1. Graph construction and centrality analysis

For graph construction we use k-Nearest Neighbours (kNN) method. For similarity distance we use cosine similarity. One could use euclidean or any other distance metric. Our short experiments showed that euclidean distance produces quite similar results. In this work we don't analyse further the differences between different distance metrics. It could be an interesting direction for future work.

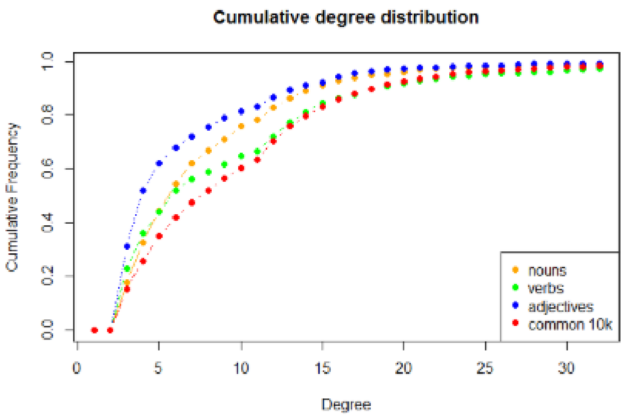
For kNN we used following hyperparameters: k = 10 and minimum similarity = 0.75. Meaning that for each word we calculate the similarities to all other words and take a maximum of 10 of those which have similarity at least 0.75. These hyperparameters produce unconnected graph with one large connected component.

Table 1: Graphs' characteristics.



			Largest Component
	All Nodes / Edges	Components	Nodes / Edges
Nouns	838 / 2488	18	796 (95%) / 2463 (99%)
Verbs	698 / 2608	26	638 (91%) / 2572 (99%)
Adjectives	761 / 1799	65	592 (78%) / 1684 (94%)
Common 10k	7897 / 31451	226	7331 (93%) / 31028 (99%)

Figure 2: Cumulative Degree Distribution.



* All Nodes: words in the subsets that are present in Glove word vectors.

Table 1 shows that Adjectives are more brittle having the highest fragmentation (lowest connectedness). Figure 2 supports this argument: steeper cumulative degree distribution for adjectives means more nodes with fewer connections. If this is truly inherent to Adjectives needs further investigation, because it could be that our sample of adjectives is very unlucky and produces more disconnected graph as it really should.

In the following analysis we focus only on the largest connected component and disregard the rest. We obtain this by extracting induced subgraph containing only the largest component. Reasoning is that in reality there are no isolated words in semantic sense. All words are somehow semantically close with some other words. By taking arbitrary subset from the data we could have destroyed some links causing unconnectedness. We could re-link unconnected words to largest connected component. However as the largest connected component covers over 90% nodes in most cases and over 94% edges in all cases, it seems reasonable to continue analysis with the largest connected component and disregard the rest. As a side note, we think that it is possible to achieve better connectedness by taking bigger and qualitatively better subset. Let's explore the constructed graph. Table 2 has examples of some random neighbourhoods. They seem pretty reasonable, capturing words with similar meanings into close neighbourhoods.

Table 2: Example neighbourhoods from constructed graphs (see Appendix 1 for more examples).

Noun: food -> meat, coffee, fish, milk, drink, tea, sugar, bread, wine, fruit

Verb: drive -> line, run, back, start, break, time, put, go, free, point, hit, last

Adjective: brown -> gray, white, green, black, blue, red, orange, yellow, pink, purple, bright, dark

Let's do centrality analysis in order to find out which words are more central. Table 3 contains comparative centrality analysis between graphs of different subsets. Table 4 shows Top 5 words in each category, Figure 3 shows neighbourhoods. Top 5 words and their neighbourhoods seem very reasonable.

Table 3: Centrality analysis. Close to 0 means less, close to 1 means more (see Appendix 3 for more examples).

	Degree	Closeness	Betweenness
Common 10k	0.008	0.125	0.029
Nouns	0.045	0.168	0.113
Verbs	0.08	0.234	0.071
Adjectives	0.065	0.159	0.117

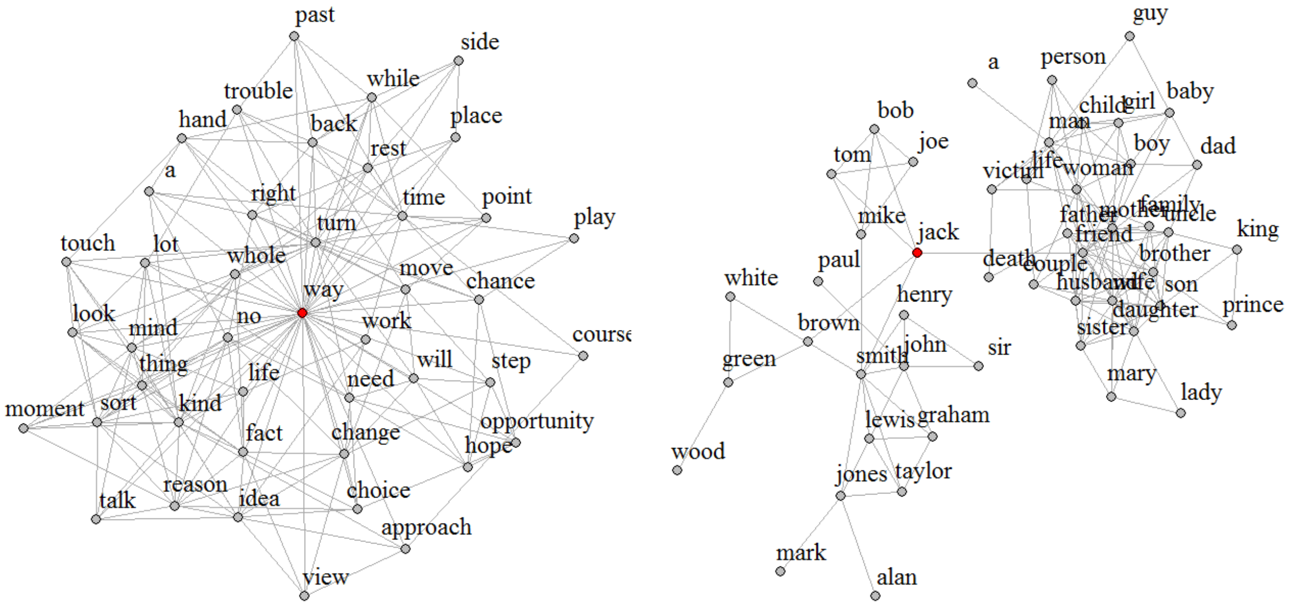
Table 4: Top 5 words in each subset (Degree, Closeness and Betweenness Centrality).

Common 10k	**Degree**	smith	but	really	so	even
	Closeness	but	this	well	for	to
	Betweenness	met	powell	new	state	well
Nouns	**Degree**	way	fact	time	reason	kind
	Closeness	way	fact	work	change	move
	Betweenness	friend	jack	part	man	way
Verbs	**Degree**	take	come	make	want	keep
	Closeness	should	take	make	need	must
	Betweenness	take	continue	should	make	come
Adjectives	**Degree**	even	this	only	same	any
	Closeness	pretty	good	even	kind	crazy
	Betweenness	wonderful	whimsical	delightful	even	amusing

In order to qualitatively estimate the results of our closeness and betweenness calculations, let's visualize neighbourhoods of some top words in each category. In case of Noun 'jack' we can see that it is a connection point of two clusters: male names (bob, tom, etc) and person related words (friend, father etc).

Figure 3: Neighbourhood Visualization (see Appendix 2 for more examples).

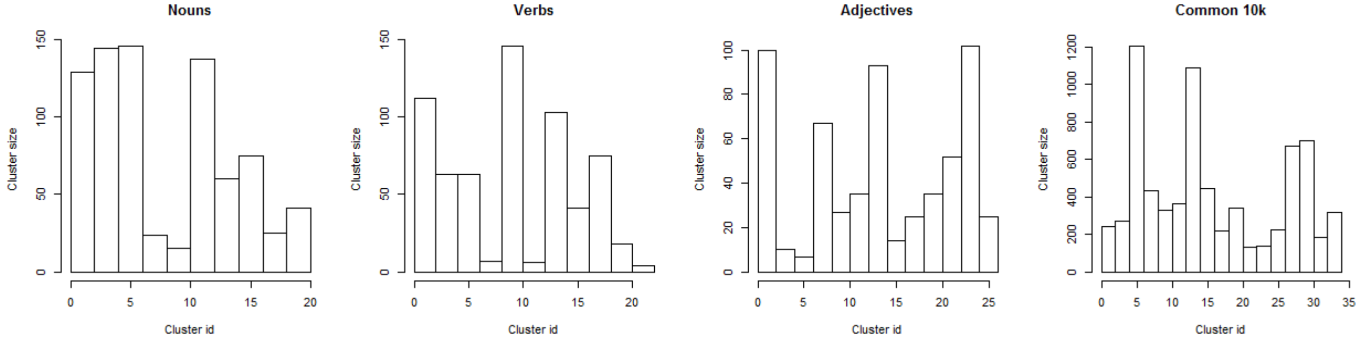
Top close Noun: 'way' **Top between Noun: 'jack'**



3.2. Graph clustering for community detection.

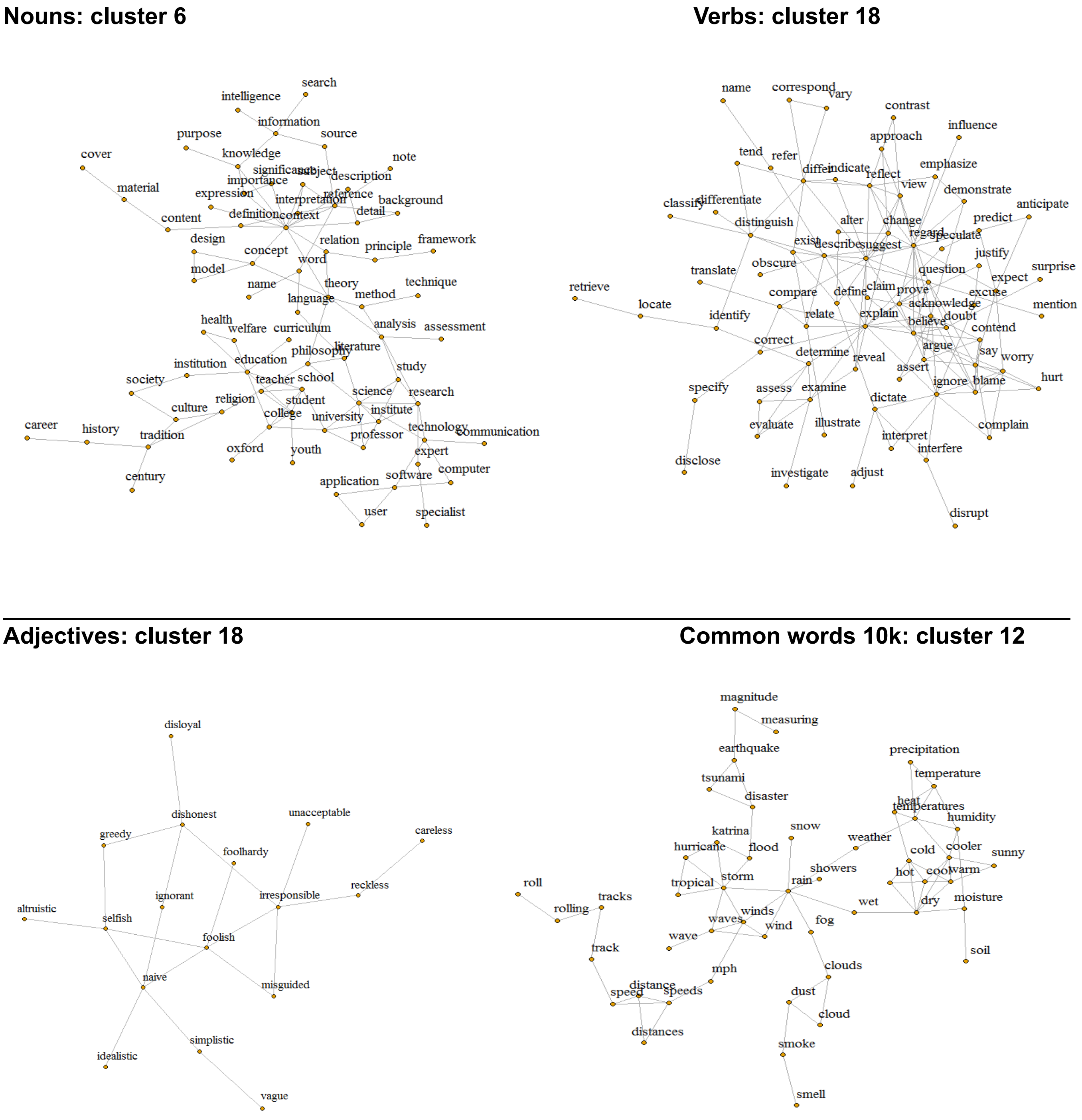
There are multiple methods for detecting communities in graphs. In this case study we use Louvain method [9] as it is fast, highly scalable and has good convergence properties [8]. Below are the results of clustering. Histograms show the distribution of cluster sizes. Number of clusters: Nouns=20, Verbs=21, Adjectives=25 and Common words 10k=35. As we expected graph clustering seems to work very well.

Figure 4: Histograms of clusters.



Some sample clusters for visual inspection. Seems that our graph construction has retained the similarity structure of the vector space pretty well. As can be seen below, the clustering captures semantically similar words into same clusters.

Figure 5: Cluster examples.



Nouns: cluster 6

Verbs: cluster 18

Adjectives: cluster 18

Common words 10k: cluster 12

4. Conclusion and discussion

In this work we presented a blueprint on how to construct a graph from high dimensional data. With graph analysis techniques we clustered and visualized high dimensional real-valued data. By using GloVe word vectors as an example we showed that graphs can be especially useful and powerful for exploring and analyzing high-dimensional data. In case of word vectors we saw that a standard dimensionality reduction techniques such as Principal Component Analysis (PCA) could not help much. However by converting our problem into graph problem, we got pretty good clustering of the data into tight communities. In addition graphs allow us easily inspect the results visually. This work scratches only the surface of what is possible with graph analysis techniques. We truly believe that graph data structures and analysis are highly underutilized in data mining and machine learning fields. However in recent years graphs are gaining more and more popularity because of their expressiveness and universality.
Future work may be extended in various directions.

- Explore different embeddings including the ones that are not based on statistical co-occurrence but extracted from knowledge bases eg WordNet, Wikidata etc.
- Use the obtained graph as input down the line for prediction tasks: eg link prediction, new node classification/cluster attribution etc.

Techniques used in this work include: Principal Component Analysis for dimensionality reduction, Cosine Similarity for distance/similarity calculations, k-Nearest Neighbours for similarity neighbourhood extraction, Graph Clustering (Louvain method) for community detection, Centrality measures (degree, closeness, centrality) for graph analysis etc. As well as Graph Construction as such which is a very powerful concept of mapping a problem in one domain into graph theory domain.

Acknowledgements

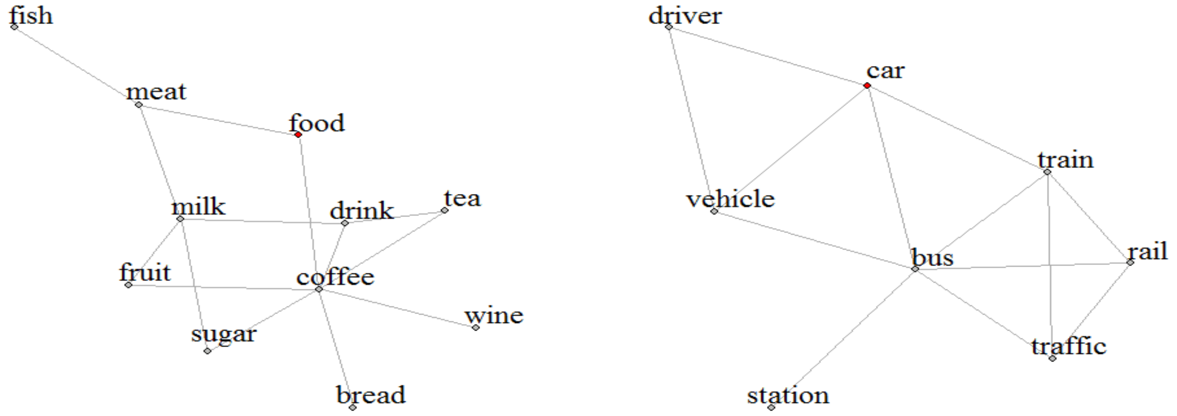
This work has been programmed in R. In addition to standard packages we used following packages: coop (cosine similarity calculations), igraph (all graph tools and plots), RColorBrewer (colour maps).

References

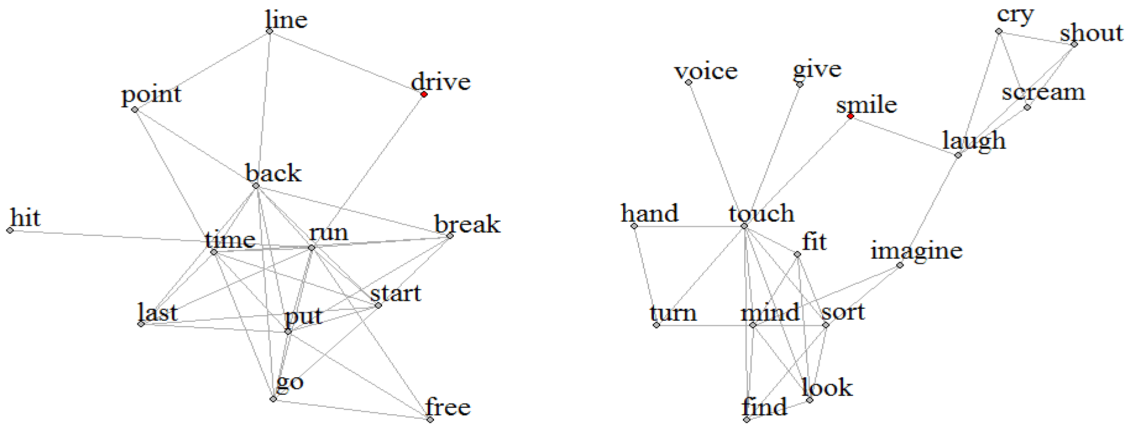
[1] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, 2013. Efficient Estimation of Word Representations in Vector Space.
[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/
[3] Enriching Word Vectors with Subword Information, 2016. Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov.
[4] www.wordexample.com.
[5] List of english adjectives: https://gist.github.com/hugsy/8910dc78d208e40de42deb29e62df913.
[6] https://github.com/first20hours/google-10000-english.
[7] https://norvig.com/ngrams/count_1w.txt.
[8] Jure Leskovec, 2018. http://snap.stanford.edu/class/cs224w-2018/handouts/06-communities.pdf.
[9] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre 2008, https://arxiv.org/abs/0803.0476, https://perso.uclouvain.be/vincent.blondel/research/louvain.html.

Appendix 1: Some random neighbourhood examples.

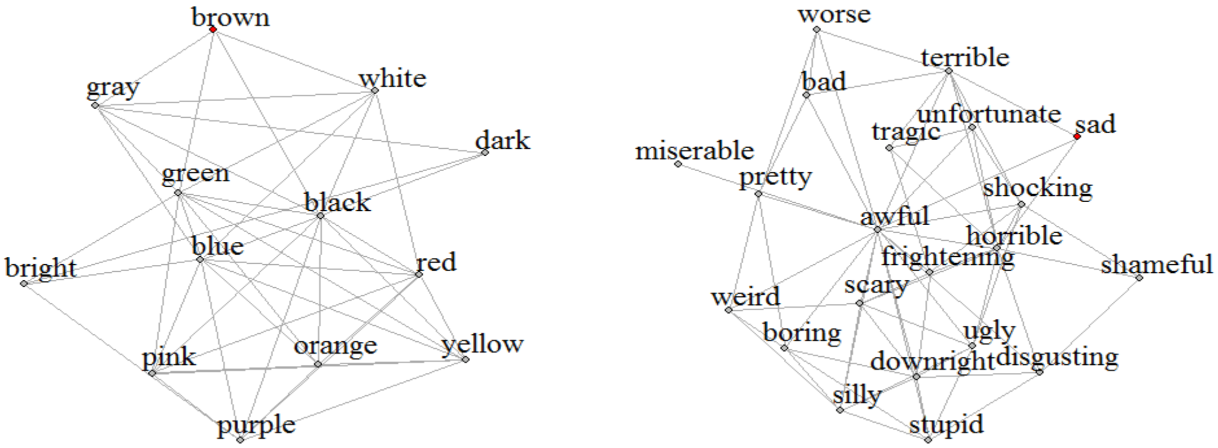
Nouns: neighbourhood around ‘food’ and ‘car’



Verbs: neighbourhood around ‘drive’ and ‘smile’

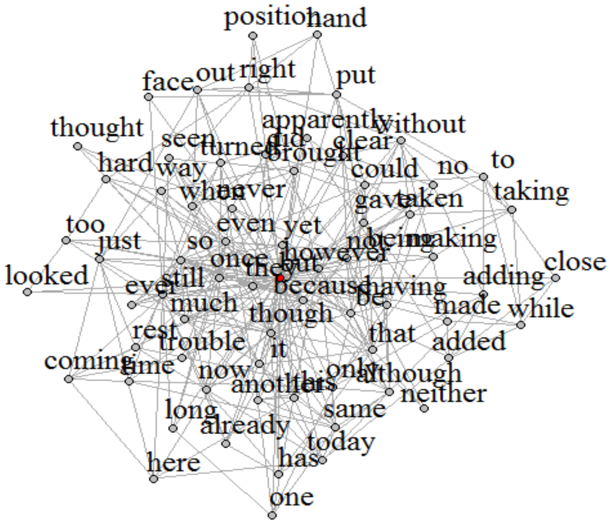


Adjectives: neighbourhood around ‘brown’ and ‘sad’

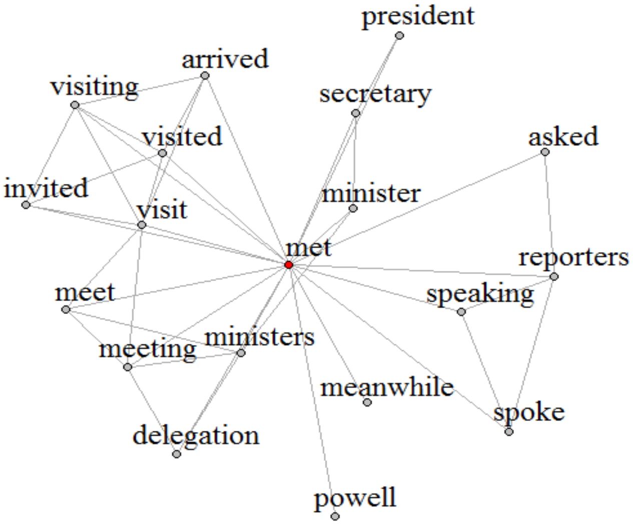


Appendix 2: Some random closeness and betweenness examples.

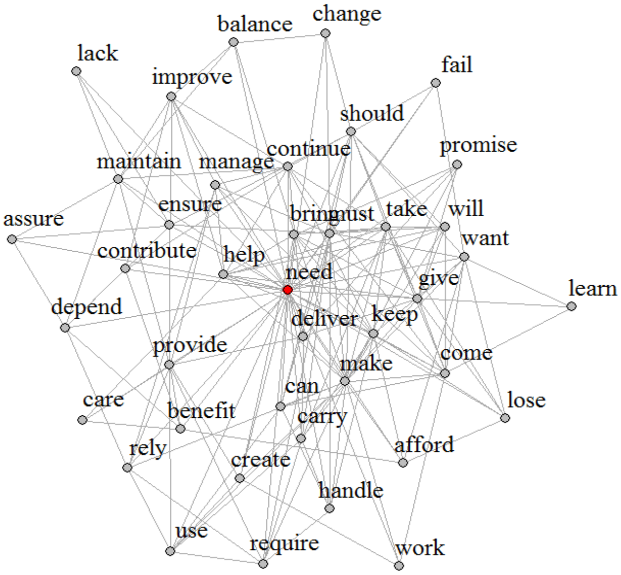
Top close common word: ‘but’



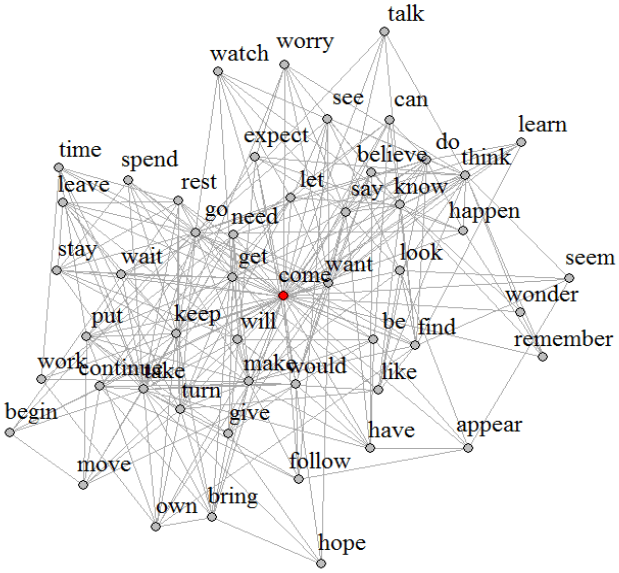
Top between common word: ‘met’



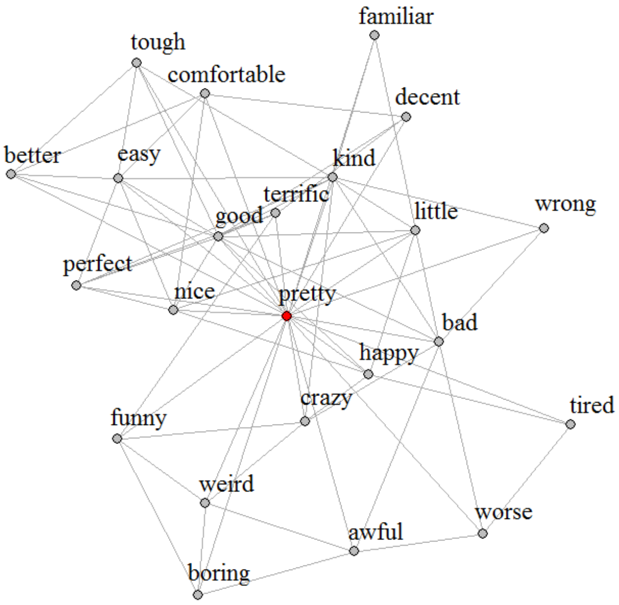
Top closeness verb: ‘need’



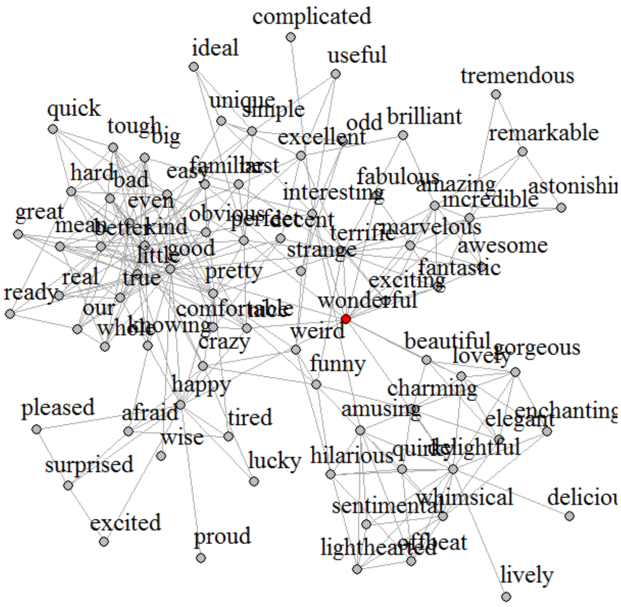
Top between verb: ‘come’



Top closeness adjective: ‘pretty’

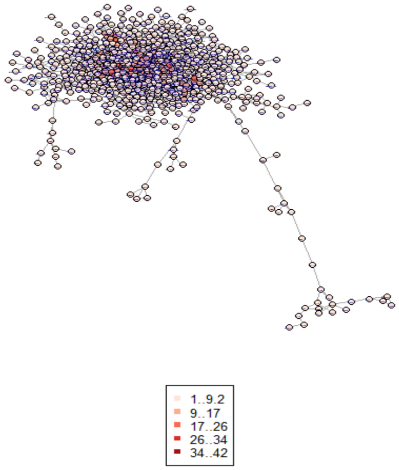


Top between adjective: ‘wonderful’

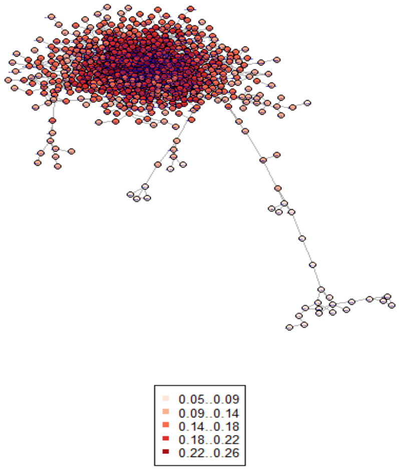


Appendix 3: Degree, Closeness and Betweenness of constructed graphs.

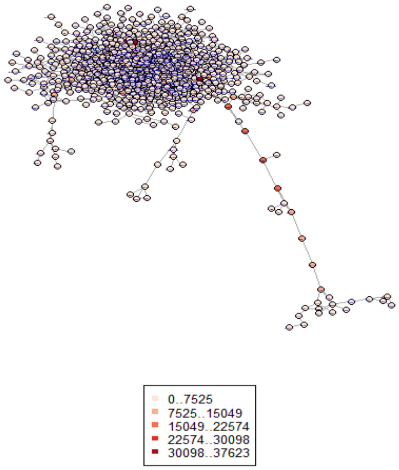
Nouns: Degree = 0.045



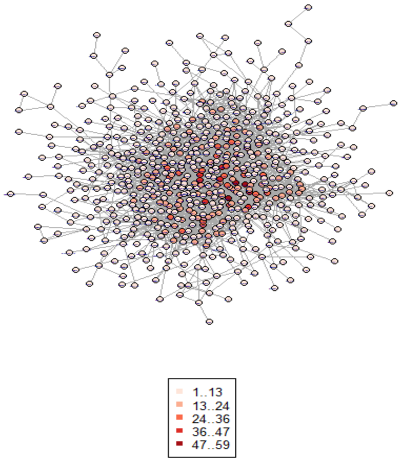
Closeness = 0.168



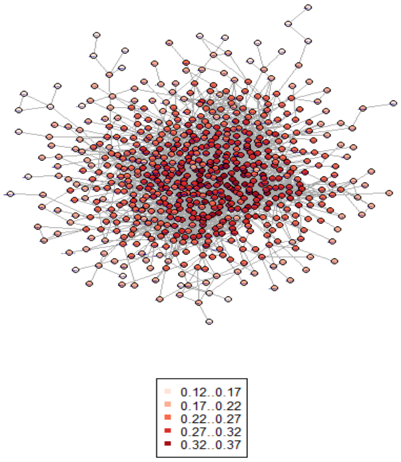
Betweenness = 0.113



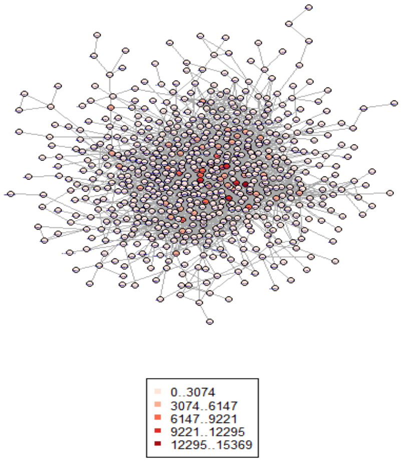
Verbs: Degree = 0.08



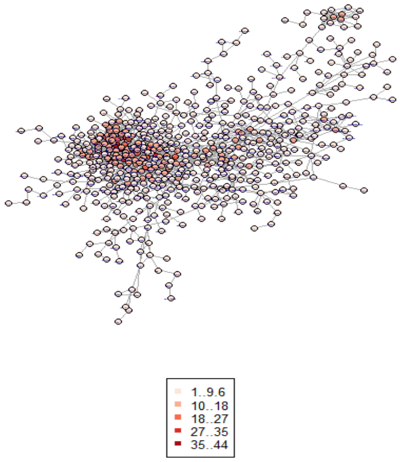
Closeness = 0.234



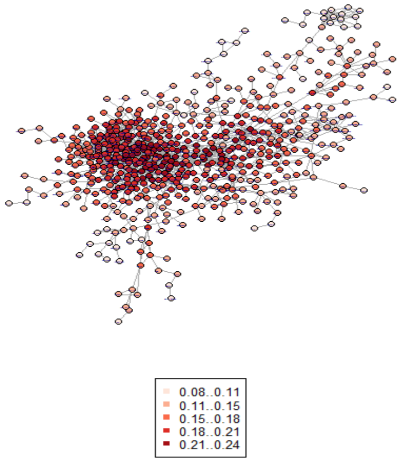
Betweenness = 0.071



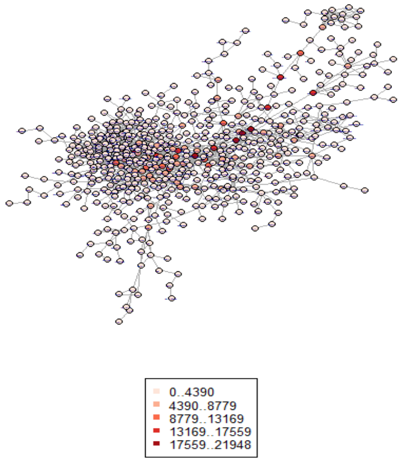
Adjectives: Degree = 0.065



Closeness = 0.159



Betweenness = 0.117


0..4390
4390..8779
8779..13169
13169..17559
17559..21948