Graph Analysis of Word Networks

Lars G. Bagøien Johnsen
Department of Research
National Library of Norway
lars.johnsen@nb.no

Keywords: graphs, clusters, meaning, cliques

Abstract

We discuss two ways of clustering words from an underlying graph structure with the aim of uncovering semantic distinctions between words, where formal relationships between words are constructed from co-occurrences within sentences. The method illustrates how geometric relations of closeness and connectedness in graphs correspond to closeness and groupings of concepts in the real world. Graphs are analyzed using methods of network analysis taken from the social sciences. In particular, the concepts of cliques and centrality of graph structures as well as partitioning methods, are put to use. These methods find communities in graphs as sets of words, which we interpret as reflecting a grouping of the meaning. Word clusters and relationships between clusters are visualized on two layers where one is the graph itself, and the second consists of a derivative graph of the subset relation between word groups. The basic graph is rendered using the force layout algorithm, while the relationship between sets and groups of words in the graph are rendered as trees.

Introduction

In this paper we present one way in which word networks constructed from word vectors (e.g. Turney and Pantel (2010)) lend themselves to semantic analysis using concepts and methods from graph theory (Chakrabarti and Faloutsos (2012)). The methods and concepts are taken from graph theory as it is used in studies of social networks, as in (Hanneman and Riddle 2005), and employed and reinterpreted for relationships between words.

Themes to be discussed are clustering and disambiguation of words, based on so called bag of words or vector models, which are transformed into network structures. We will demonstrate a particular way of generating graphs from those. Typical representatives are vectors made using

word2vec algorithm (Mikolov & Zweig 2013), or produced within the application LancsBox (Brezine et.al 2016).

Our approach to disambiguation from word vectors differs from the stochastic approach in (Bartunov et.al 2017), since word clusters and sets are constructed out of edges and nodes in the network structure, using the topology of the network, in addition to any weighting of the connections themselves. So, from a formal point of view, the method aligns itself with the formal treatment of networks as found in analyses of social networks.

The word vectors studied here are constructed from collocations computed from trigrams of coordinations (see below), and can therefore be considered a subtype of the above word vectors. Even though the vectors are constructed differently, the formal treatment of the network structures will be the same.

Research questions

One question that is addressed is how textual raw data can be transformed into structures that represent knowledge of language, while at the same time reflect its external significance. The algorithms themselves have no access to the external world, so the correspondence lies in how closeness (or groupings) of words matches the closeness of their corresponding concepts. For example, words like *apple* and *pear* go together as vegetables and *jazz* and *pop* as genres of music.

This feature of graph analysis as a source of groupings and connections can then be used to analyze the semantics of language and literary works, ranging from the disambiguation of particular words to semantic fields and frames, where these objects are represented as nothing more than a collection of words.

To be specific, the meaning and reference of words are taken to be external to language, while word vectors represent those meanings internally, within language, by associating a word with a vector (bag of words). These vectors are taken to stand proxy for meaning, so that operations and combinations of meaning can be performed on the vector representations, which corresponds to meaning operations externally, see e.g. (Mikolov & Zweig 2013).

Our aim here is not to provide an algebra, or a combinatorial system, of meanings to be applied to these representations, but rather see in which ways word clusters can represent aspects of word meaning and make distinctions between shades of meaning.

Methods

Graphs may be constructed from word vectors, and vectors from graphs. For example, a selection of word pairs, taken from skip bigrams for example, may simply be viewed as a graph of word to word pairs, where the words are the vertices and the pairs are edges of the graph.
From a graph, a word vector (or bag of words) is constructed by collecting all the words a given word *w* pairs up with. Conversely, a word *w* with an associated bag of words W, form a natural graph structure by pairing the words in W with *w.*

The construction is illustrated here with vectors made from coordinative construction in Norwegian like *ost og kjeks* (cheese and biscuits), which at the same time introduce a semantics to the word vector – two words are coordinated if they share something in the context in which they are uttered.

Each coordination is assigned a weight in the form of pointwise mutual information (PMI), computed from the collection of all texts, ensuring to a certain degree that the conjunctions selected are full phrasal words, and not the end or start of a phrase. For example, blindly selecting edges X → Y from coordination instances of "X og Y" results in many pairs that are not true coordinations, since X may just be the end of a phrase coordinated with a phrase that starts with Y. By using frequency and PMI almost all these instances are eliminated, resulting in a graph that contains less noise, so that an edge (X,Y) is in fact a coordination of X and Y. So, the whole process goes like this. The graphs used to represent relations are constructed from a subset of trigrams on the form "*X og Y",* conjunction in the middle. Words in the trigram form edges in a graph X→ Y.

The actual construction of the graph iterates the formation of nodes and edges. A graph from a given word, say Norwegian *jordbær (strawberry),* is constructed by considering the set of edges X→ *jordbær* and *jordbær*→ Y, resulting in a directed graph. In the analysis below, the directedness is not taken into account; the graph is converted to an undirected graph before undergoing further analysis.

The iteration of the process gives the graphs complexity. For each X and Y above, the process of collecting words is continued, and so on up to a certain level, creating the necessary complexity in the graph for network analysis, where the complexity comes from the recurrence of nodes, and from nodes pointing back to other nodes.

The graphs shown here are made in three steps. In the case of *is (ice)*, the first step collects words connected to it, like *jordbær,* then the second step adds edges for *jordbær* like *jordbær*

(strawberry) → *moreller (cherries)* and *jordbær* → *blåbær (blueberry)*. The third step collects edges for *moreller* and *blåbær.* When expanding these words, a link is created between them; as well as new common node *bringebær* (raspberry), as illustrated in the following diagram, which shows part of the graph generated by *jordbær*:

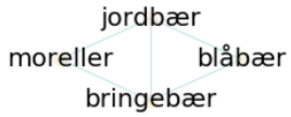


Figure 1; Small graph for jordbær

This graph also illustrates the typical cluster between nodes in a network, namely that of a clique, a set of nodes which are all connected to each other. The set of nodes {jordbær, blåbær, bringebær} is a subgraph in which all the nodes are connected, and thus defines a clique. Another clique is formed by {jordbær, bringebær, moreller}.

Below, we describe the process of forming clusters using cliques within the development platform provided by the module networkx (NetworkX (2016)) for the Python programming language.

Main findings

Consider the graph shown below, constructed as described above from the word form *kirsebær* (cherry) and displaying most of the words related to it through the coordination construction. The layout of the graph is of the force-directed type (Fruchterman & Reingold 1991) as implemented in networkx. This type of layout provides a visual set of clusters of words based on the weights between nodes, and the topology of the graph structure. Implementations of force-directed layouts will, in general, give different outputs on different runs. However, the overall grouping will be the same, although rotational orientation will vary, as well as spatial relationships.

The graph has been amended with colors marking a partitioning of the nodes according to the Louvain method (Blondel et.al. 2008), to which we will return below.

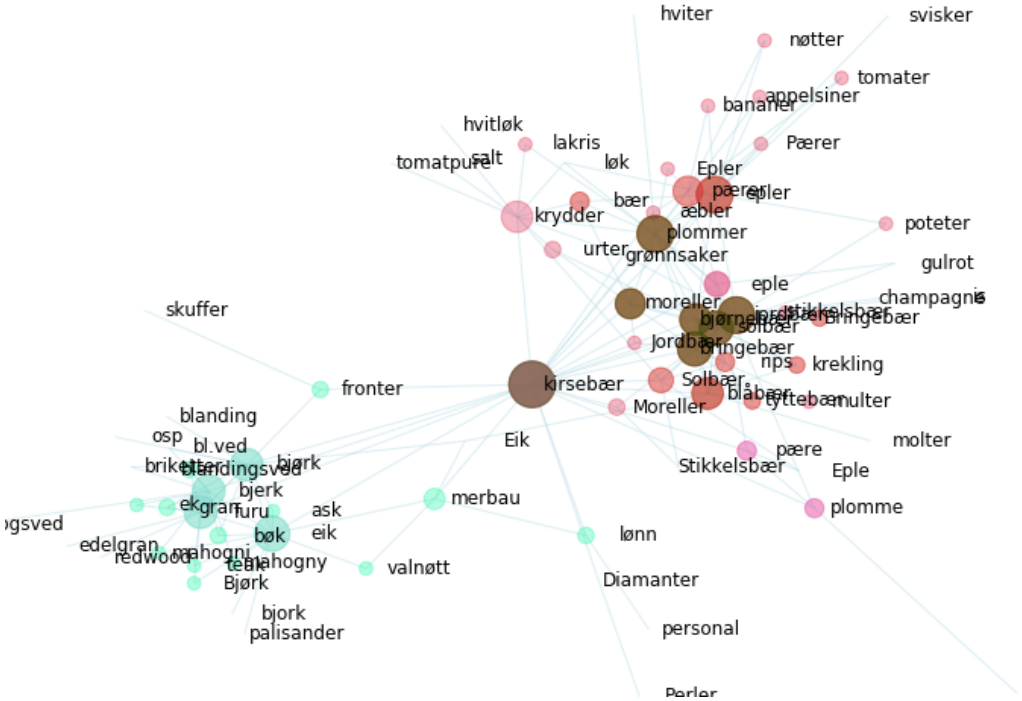


Figure 2: Graph for "kirsebær"

On visual inspection, a couple of word groups are immediately apparent. There is one area with interconnected elements containing berries, and another area containing wood types, in addition to other accidental readings and connections for some of the words, which can be attributed to food relations like spices, salt etc.

Words are clustered in terms of k-cliques (Chakrabarti and Faloutsos 2012) showing how different readings or meanings of *kirsebær* (cherry) can be extracted from the set structure. A k-clique cluster is made such that first a clique with k members is selected, then the cluster gets new nodes from another clique if the latter only differs from the starting clique in only one node. So, in the case of Figures 1, {jordbær, blåbær, bringebær} and {jordbær, bringebær, moreller} are joined together to from {jordbær, blåbær, bringebær, moreller}, and so on.

For reference, we show the whole cluster structure for *kirsebær* here, together with a pair of numbers (k, s) such that k is the size of the clique from which the group is created, and s is an arbitrary sequence number. For the groups (4, 1), (4, 3) and (7, 1) on which further commentary follows below, see the English translations in italic:

(3, 2) merbau, furu, gran, kirsebær, teak, fronter, blandingsved, bøk, ask, bjerk, eik, lønn, bjørk, ek, Bjørk, mahogni, valnøtt, mahogny
(4, 3) furu, kirsebær, eik, bjerk, bjørk (*pine, cherry, oak, birch, birch)*

(5, 1) plommer, kirsebær, epler, bringebær, bjørnebær, moreller, blåbær, jordbær, solbær

(4, 2) rips, tyttebær, krekling, plommer, kirsebær, Bringebær, pærer, eple, epler, bringebær, bjørnebær, Solbær, moreller, bær, blåbær, jordbær, solbær

(6, 1) plommer, kirsebær, epler, bringebær, bjørnebær, moreller, blåbær, jordbær, solbær

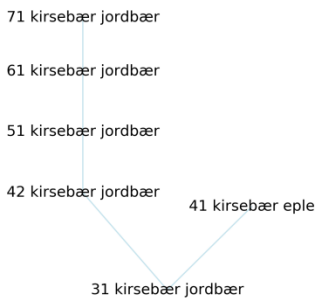
(3, 1) pære, stikkelsbær, Jordbær, solbær, æbler, krydder, plommer, kirsebær, pærer, bananer, krekling, epler, Bringebær, Epler, eple, bringebær, Solbær, tomater, moreller, Pærer, jordbær, rips, multer, lakris, appelsiner, poteter, Moreller, bjørnebær, bær, plomme, blåbær, tyttebær, nøtter, urter

(4, 1) pære, kirsebær, eple, plomme (*pear, cherry, apple, plum*)

(7, 1) bringebær, bjørnebær, moreller, plommer, kirsebær, jordbær, solbær (*raspberries, blackberries, cherries, plums, cherries, strawberries, black currants*)

Each group is a cluster generated as a k-clique cluster from the graph. An overview of the relationships between them can be obtained by looking at the subset relation, which is visualized using tree structures depicting relationships between clusters.

In the two trees below, the labels are built to correspond with the (k, s) pairs in the clusters, and the labelling is taken from the two most central nodes in the graph that are also members of the cluster. There are two main branches, one starting from (3, 1) and one starting from (3, 2):



Figures 3 Subset for berry and fruit reading

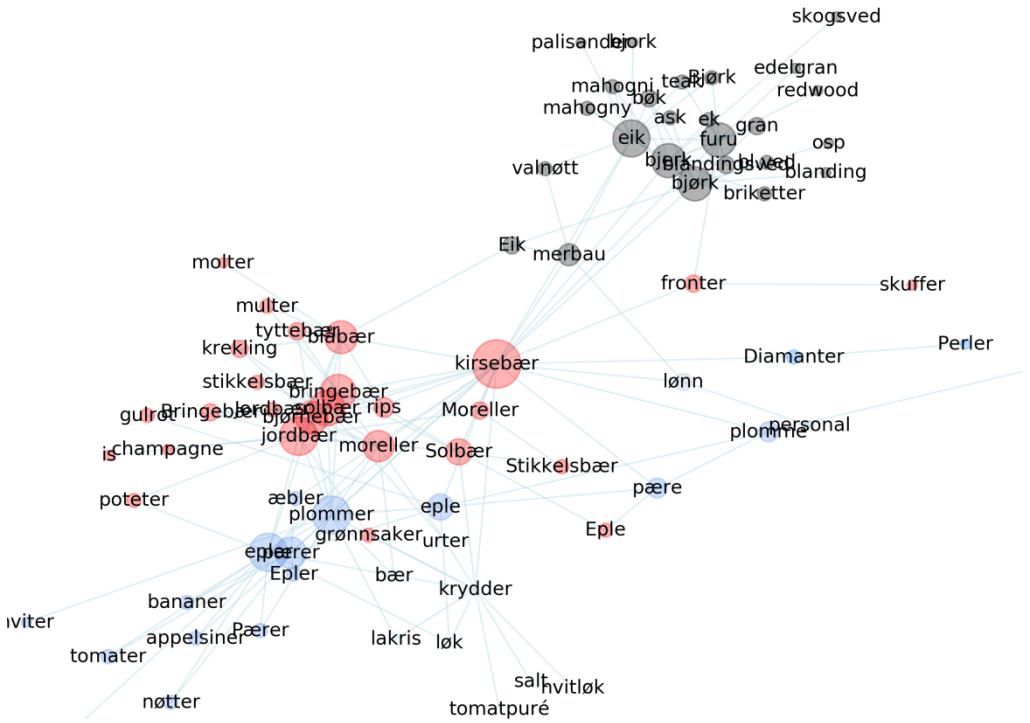
Figures 4 Subset for wood reading

The three clusters that appear with a translation, (7, 1), (4, 1) and (4, 3) are the topmost and the smallest sets within their branch. (7, 1) represents the reading "berry", which, together with the associated fruits in (4, 1) constitute the clusters emanating from the larger set (3, 1), which also contains words for vegetables, like the word for potato. Then, on the other side is the reading "wood" or "tree", which is separate from the rest, although *kirsebær* is still a member of the sets. So, all in all, the graph represents three variations of meaning for one word, one as a tree (cherry tree), and one as an edible substance, the berries, which also go together with fruits such as apple and pears (the words *eple* and *pære* in the clusters).

The topology of the differences between the trees may be interpreted along different axes of polysemy: *cherry* is used both for berry and tree (a kind of homonymy by metonymy), while the readings "fruit" and "berry" are brought closer together.

Additional structure

In addition to the overlapping clustering produced by k-cliques, there is also the option of partitioning the graph into non overlapping clusters. A community analysis (Chakrabarti & Faloutsos 2012) will also generate sets that contain words that may be interpreted as belonging together semantically. For example, coloring the graph using communities computed using the Louvain method (Blondel et.al. 2008), it is apparent that the coloring adds information and aids the visualization of the clusters. Note the slight difference between this rendering of the graph compared to the one above. The visual graph structure fits well with the structures from the clique analysis. The nodes representing trees in the upper right corner are colored grey, while the fruits and berries (as well as spices) occupy the lower left corner.



Figures 5 kirsebær colored by communities

Communities produce a partition of the nodes as a collection of mutually disjoint sets. Therefore, our main word *kirsebær* is only member of one set, making it harder to detect ambiguities within the sets, in contrast to the k-clique clustering which created sets that overlapped. However, the community analysis is broader and groups together words that are missed out by the k-clique clustering.

Future plans for this research is to try and group together these two ways of structuring the data, so that both high precision and subset structure can be used to arrive at a description of multiple meanings for a word.

The principal finding is that both k-clique clusters and community detection may be used to find different levels of meaning for words, and that k-cliques are in general more conservative with high precision, while community detection, in general, creates partitions that covers the graphs entirely.

REFERENCES

Bartunov, S., Kondrashkin, D., Osokin, A. and Vetrov, D. (2017). Breaking Sticks and Ambiguities with Adaptive Skip-gram. [online] Arxiv.org. Available at: https://arxiv.org/abs/1502.07257

Blondel, Vincent & Jean-Loup Guillaume & Renaud Lambiotte & Etienne Lefebvre (2008) Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment.*

Breder Birkenes, Magnus & Johnsen, Lars G. & Lindstad, Arne Martinus & Ostad, Johanne, 2015. From digital library to n-grams: NB N-gram in *Proceedings of the 20th Nordic Conference of Computational Linguistics*, pp 293—295, Linköping University Electronic Press, Sweden.

Brezina, V., McEnery, T., & Wattam, S. (2015). Collocations in context: A new perspective on collocation networks. *International Journal of Corpus Linguistics*, 20(2), 139-173.

Chakrabarti, Deepayan and Faloutsos, Christos 2012. *Graph Mining*, Morgan & Claypool Publishers.

Fruchterman, Thomas M. J. & Reingold, Edward M. (1991), Graph Drawing by Force-Directed Placement, *Software – Practice & Experience*, Wiley, 21 (11): 1129–1164, doi:10.1002/spe.4380211102

Hanneman, Robert A. and Mark Riddle. (2005). *Introduction to social network methods*. Riverside, CA: University of California, Riverside. Online text: http://faculty.ucr.edu/~hanneman/nettext/

Mikolov, T., Yih, W. & Zweig, G. Linguistic regularities in continuous space word representations. In *NAACL HLT*, pp. 746–751, 2013b.

NetworkX (2016). https://networkx.github.io/

Turney P.D. and Pantel P. 2010. From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Acknowledgements

We are grateful to the National Library for providing the facilities for carrying out the work described here. The data in this study are all available from the URLs described in Breder Birkenes et.al. 2015.