Veremyev *et al. Applied Network Science* (2019) 4:104
https://doi.org/10.1007/s41109-019-0228-y

Applied Network Science

RESEARCH Open Access

Graph-based exploration and clustering analysis of semantic spaces

Alexander Veremyev[1], Alexander Semenov[2], Eduardo L. Pasiliao[3] and Vladimir Boginski[1]*

*Correspondence:
vladimir.boginski@ucf.edu
[1]Department of Industrial Engineering and Management Systems, University of Central Florida, 12800 Pegasus Drive, Orlando, FL, USA
Full list of author information is available at the end of the article

Abstract

The goal of this study is to demonstrate how network science and graph theory tools and concepts can be effectively used for exploring and comparing semantic spaces of word embeddings and lexical databases. Specifically, we construct semantic networks based on *word2vec* representation of words, which is "learnt" from large text corpora (Google news, Amazon reviews), and "human built" word networks derived from the well-known lexical databases: WordNet and Moby Thesaurus. We compare "global" (e.g., degrees, distances, clustering coefficients) and "local" (e.g., most central nodes and community-type dense clusters) characteristics of considered networks. Our observations suggest that human built networks possess more intuitive *global* connectivity patterns, whereas *local* characteristics (in particular, dense clusters) of the machine built networks provide much richer information on the contextual usage and perceived meanings of words, which reveals interesting structural differences between human built and machine built semantic networks. To our knowledge, this is the first study that uses graph theory and network science in the considered context; therefore, we also provide interesting examples and discuss potential research directions that may motivate further research on the synthesis of lexicographic and machine learning based tools and lead to new insights in this area.

Keywords: Semantic spaces, Graph theory, Word2vec similarity networks, Cohesive clusters, Cliques, Clique relaxations

Introduction

The amount of text data generated in various domains has exploded exponentially over the past few years, and it is estimated that about 80% of all data is unstructured text-heavy data (Schneider 2016; Sumathy and Chidambaram 2013). Therefore, it is increasingly important to develop effective tools and methodologies for handling and analyzing text data. The field of text analytics contains a set of techniques for extracting valuable knowledge from the text, such as the use of natural language processing tools to convert unstructured text-rich data into structured machine-understandable form of data. Typical text analytics applications include finding/extracting relevant information from the text, text categorization, document summarization, text clustering, sentiment analysis, concept extraction, and others (Gandomi and Haider 2015). Many of these tasks are addressed using various machine learning techniques.

Text data is one of the most underused sources of data (Bengfort et al. 2018; Kasch 2014). A significant challenge for text analytics is understanding language organization

principles, rules, and definitions, which, contrary to formal languages (i.e., programming languages), are often determined by a *context of use* and encompass current human knowledge and experience (produced by people to be understood by people). In general, natural languages are not domain specific but rather universal in the sense that the same words and organizing principles are used in various domains. Moreover, natural languages are redundant, ambiguous, and quickly evolving as they constantly adapt the inclusion of new symbols (e.g., emoji symbols), definitions, contexts, and usages (Bengfort et al. 2018).

Network representations provide intuitive and useful ways to uncover complex structures of natural languages. In particular, a human lexicon, which is a set of words or meanings and their semantic relationships can be naturally modelled by networks. The global organization and dynamics of such networks, which are referred to as *semantic* networks, have been investigated by a number of studies (Sigman and Cecchi 2002; Steyvers and Tenenbaum 2005; Bales and Johnson 2006; Borge-Holthoefer and Arenas 2010; Choudhury and Mukherjee 2009; Fukś H and Krzemiński 2009; de Jesus et al. 2004; Motter et al. 2002). The majority of this previous work is focused on the analysis of semantic networks constructed using some dictionaries or lexical databases, such as WordNet (Miller 1995; Fellbaum 1998) and Moby Thesaurus (Ward 2002). The existence of semantic relations between words in such networks is judged by lexicographers, which may lead to significant differences among the structures of the corresponding networks (Gaillard et al. 2011). Thus, such "human built" networks may not necessarily reflect the true semantic structure and diversity of the corresponding language.

Another way of constructing semantic networks is based on *word embedding*, which is a popular method of representing words as vectors in a multi-dimensional space. It is capable of capturing the *context* of a word in a document, semantic and syntactic similarity, relations with other words, etc. Recent years have seen rapid development of word embedding methods, the most popular of which are *word2vec* embeddings (Mikolov et al. 2013a, b). Word embedding models map words (or word phrases) in large corpora of text into a multidimensional vector space, where each word is represented by a vector in this space, and semantically similar words are located closer to each other. Word embeddings are created using self-supervised machine learning algorithms. The benefit of semantic spaces generated by word embedding algorithms is that they can be trained on very large text corpora (e.g., texts with 100 billion words from Google News (Google Open Source Project 2013)) and may better reflect the *context of use*, diversity and dynamics of human languages than dictionaries and lexical databases compiled by lexicographers, and, hence, help to improve these human built databases. Nowadays, word embeddings are a key instrument in many natural language processing and machine learning applications. Understanding the structure and organization principles of such semantic spaces is very important for measuring the performance and limitations of word embeddings, which can be done using network representations. An edge between two words in such networks means that their corresponding vectors are similar to each other in the sense that the respective words are used in similar contexts.

In this paper, we compare and analyze two "human built" semantic networks constructed using lexical databases (WordNet (Miller 1995; Fellbaum 1998), Moby Thesaurus (Ward 2002)) and two "machine built" semantic networks constructed using word

embeddings based on Google News (Google Open Source Project 2013) and Amazon Reviews (2017) datasets. The WordNet lexicon groups words into sets of synonyms, which we use to construct a network of synonyms (connect each pair of words in every group of synonyms by edges). On the contrary, Moby Thesaurus contains a set of words (*root* words) followed by the list of synonyms and other related conceptually similar words; in its network representation we connect each *root* word with every word in its corresponding list. As Moby Thesaurus has a broader definition of a synonym, the resulting network is denser than the WordNet network and can be viewed as more *relaxed* synonyms network, which is useful for comparison reasons with word embedding-based networks. Since there are a number of ways to generate and obtain word embeddings, the first one we selected is already trained on very large text corpora publicly available and used as a benchmark in machine learning applications: a pre-trained word2vec embedding of Google News (Google Open Source Project 2013). The second one is word2vec embedding of Amazon Reviews dataset (Amazon Reviews dataset 2017) containing more than 400K customer reviews (text size is roughly equal to 100 typical hardcover books). To generate the corresponding word embedding (which is shown to be able to capture semantic similarity among words for sentiment analysis (Bansal and Srivastava 2018)) we use Gensim (Řehůřek and Sojka 2010) module. Thus, two selected word embedding datasets represent somewhat different styles of language (a more formal language of news written by trained journalists versus a more casual language of reviews written by customers who might use slang, acronyms, words from other languages, emojis, etc.) and their network representations may reveal some interesting insights about the contextual use of words in these domains. Moreover, our findings indicate that although the global characteristics of word embedding-based networks are somewhat similar (all networks exhibit small-world properties), there are significant differences in the nodes (words) which occupy more central network positions. Specifically, the most central words in the networks built on lexical databases tend to be more frequently used in the English language, whereas most central words in the networks based on word embeddings are those that are rarely used.

In addition, we identify *dense clusters* (subsets of words with a relatively high number of edges among them) in the constructed semantic networks. Naturally, dense clusters of nodes in semantic networks should represent groups of words that are very close to each other in the semantic space and share similar meanings. In particular, we first use the concept of a clique (subset of nodes in which each pair of nodes is connected by an edge), which is employed in a number of application areas due to its elegance and inherent ability to logically represent cohesive (well-connected) subgroups of elements in complex systems modeled as graph (Bomze et al. 1999). However, the requirement that every possible edge is present within a clique is very strict and may limit the flexibility of this concept. One way to overcome this issue is to relax a certain clique-defining property and find network clusters satisfying this relaxed property (Pattillo et al. 2013b). In this work we consider a widely used concept of a *quasi-clique* (Abello et al. 1999), which ensures that the considered cluster is dense enough (the percentage of edges in a cluster is above a certain threshold). The problem of finding large dense clusters has been addressed in a number of applications from various domains, including telecommunications (Abello et al. 1999), biology (Hartwell et al. 1999; Spirin and Mirny 2003; Bader and Hogue 2003; Bu et al. 2003; Hu et al. 2005), social network analysis (Crenson 1978;

Wasserman and Faust 1994), finance (Boginski et al. 2005; Boginski et al. 2014; Huang et al. 2009; Sim et al. 2006) and data mining (Tsourakakis et al. 2013; Angel et al. 2012). We implement our recently developed mixed integer programming-based methodologies (Pastukhov et al. 2018; Veremyev et al. 2016) for identifying cliques and quasi-cliques to graph representations of semantic spaces. We note that even though the underlying problems are NP-hard in general, it is still realistic to find exact solutions of these problems due to the size and sparsity of the considered networks, as well as due to significant performance improvements of integer programming solvers over the past decade.

We demonstrate the usefulness of dense cluster analysis in local (ego) networks of semantic spaces, that is, subgraphs induced by the words connected to any given word of interest. Intuitively, in word embedding-based networks, these clusters should be able to capture semantically relevant groups of words according to their meanings and contextual use within the text corpora which the word embedding is trained on. It may allow one not only to measure the quality of word embeddings, but also to improve the existing lexical databases by broadening and refining sets of synonyms currently available in those databases. Moreover, as the language quickly evolves, we find that dense clusters are able to identify semantically similar groups for new, unusual (e.g., acronyms, emojis) or misspelled words in the corresponding word embeddings, which, for example, may help to uncover the perceived meanings of emojis across platforms as it is not well understood how people interpret them (Miller et al. 2016). In addition, we show how information about cliques extracted from semantic networks constructed based on word embeddings can be incorporated into machine leaning algorithms, e.g., sentiment analysis of Amazon Reviews.

As a final remark of this section, we note that network science concepts and approaches have been used in psychological linguistics and cognitive science to gain more insights and deeper understanding of human cognition. For example, they help to address one of the most fundamental questions of how semantic knowledge is absorbed, represented, organized and searched in our brains (Vitevitch 2008; Vitevitch and Goldstein 2014; Vitevitch et al. 2014; Abbott et al. 2015; Ke and Yao 2008), as well as investigate other aspects of language complexity and structure (Siew 2013, 2018; Jia et al. 2018; Cong and Liu 2018). For a comprehensive recent survey on this topic that overviews various studies and applications of networks in cognitive science we refer the reader to Siew et al. (2018). In addition, network-based text representations and algorithms have been successfully applied to information retrieval, keyword extraction, text summarization, document classification, and other problems (Vazirgiannis et al. 2018; Altuncu et al. 2019). Therefore, we believe that network-based approaches are promising in the considered domain, and this study takes a further step towards demonstrating the potential value of network science in the analysis of text data.

Notations and definitions

This section introduces graph-theoretic notations and definitions. Note that although the entries in lexical databases and word embeddings may contain both single words and short phrases (e.g., 'quite a little', 'too bad'), we refer to these entries as *words* or *nodes* in the corresponding networks and use these terms interchangeably.

Network characteristics

Let $G = (V, E)$ be a simple undirected graph with the sets of n nodes (vertices) and m edges denoted by V and E, respectively. Denote by $N(i)$ the set of all neighbors of $i \in V$, where $j \in V$ is a neighbor of $i \in V$ if $(i, j) \in E$, i.e., $N(i) = \{j \in V : (i, j) \in E\}$. Then the *degree* of i in G is defined as $deg(i) = |N(i)|$. Two distinct nodes i and j are *connected* if G contains a path between them. A path between i and j in G is the *shortest* path if it contains the least number of edges among all paths between i and j in G. The length (i.e., number of edges) of a shortest path between i and j in G is referred to as the *distance* between i and j in G and denoted by d_{ij}. The maximum distance between any two nodes in G is referred to as the *diameter* of G, i.e., $diam(G) = \max\{d_{ij} : i, j \in V\}$. The *average distance* is simply the arithmetic mean of distances between all pairs of nodes in graph G. For any subset $S \subseteq V$, $G[S] = (S, \binom{S}{2} \cap E)$ defines the *subgraph* induced by S in G. A *connected component* of G is an induced subgraph in which each node has a path to every other node in the component, but not to any node outside the component.

The *global clustering coefficient* for graph G is the ratio of the number of closed triplets, to the number of all triplets in the graph. The *local clustering coefficient* of a node i in graph G is the ratio of the number of connections among its neighbors to its maximum possible value. The *average clustering coefficient* of a graph is the average of all local clustering coefficients calculated for every node i. The *degree assortativity* is the Pearson correlation coefficient between degrees of linked pairs of nodes. For more details and discussion on these standard structural graph characteristics we refer the reader to Newman (2003, 2018).

In addition, to identify the most important or central nodes in semantic networks we use the concept of node centrality. Specifically, we consider four classical centrality measures (degree, closeness, betweenness, and PageRank), which capture the complimentary aspects of node importance (position) in a network. Their definitions, historical background, as well as intuition behind each type of centrality measure can be found in, e.g., (Boldi and Vigna 2014; Borgatti and Everett 2006; Jackson 2010).

Dense clusters

In order to formally define and analyze community-type dense clusters in the considered networks, we use the graph-theoretic concepts of a *clique* and a γ*-quasi-clique*.

A graph G is *complete* if it has all possible edges, i.e., $(i, j) \in E$ for any $i, j \in V$ $(i \neq j)$. A *clique* C is a subset of V such that $G[C]$ is a complete graph (Luce and Perry 1949). The *maximum clique problem* is to find a clique of maximum cardinality in G (Bomze et al. 1999). This problem is known to be NP-hard (Garey and Johnson 1979).

A γ*-quasi-clique* is an *edge density based* clique relaxation defined as a subset $S \subseteq V$ such that the subgraph $G[S]$ induced by S in G has the edge density of at least γ, that is, $\rho(G[S]) = |\binom{S}{2} \cap E| / \binom{|S|}{2} \geq \gamma$, where $\gamma \in (0, 1]$ is a fixed constant parameter (Abello et al. 2002). Clearly, $\gamma = 1$ corresponds to a clique. The problem of finding a maximum γ-quasi-clique is known to be NP-hard for any fixed $\gamma \in (0, 1]$ (Pattillo et al. 2013a; Holzapfel et al. 2006). Cliques and γ-quasi-cliques will be used in the context of dense cluster analysis in the considered networks. As mentioned further in the paper, despite the NP-hardness of the optimization problems related to cliques and γ-quasi-cliques, we have been able to solve such problems to optimality in the constructed networks.

Methods

This section describes methods used for building word embeddings and their network representations, as well as tools and methodologies applied to analyze the constructed networks.

Word embeddings (word2vec) similarity network construction

Let V be a set of unique words (tokens) in the considered text corpora (a text document or a collection of text documents), i.e., V is a vocabulary, and let $|V| = n$ denote the number of words in it. For each word $w \in V$, the word w embedding is a vector $p^w = (p_1^w, \ldots, p_K^w)$ in a K-dimensional (semantic) space, where each p_k^w $(k = 1, \ldots, K)$ is a real number. A mapping function $\phi : V \to \mathbf{R}^K$ maps any word in the considered text corpora to a vector in a K-dimensional space. The goal of a mapping function is to construct a semantic space such that semantically similar words are mapped to similar vectors in the corresponding space. Word2vec models (Mikolov et al. 2013a, b) appear to be appropriate for this task: these are essentially neural networks that are trained to reconstruct linguistic contexts of words based on large corpora of text as an input.

The similarity score between a pair of words $i, j \in V$ is computed as *cosine similarity* between the corresponding vectors p^i and p^j, which is equal to

$$sim(i,j) = \frac{\sum_{k=1}^{K} p_k^i p_k^j}{\sqrt{\sum_{k=1}^{K} \left(p_k^i\right)^2} \sqrt{\sum_{k=1}^{K} \left(p_k^j\right)^2}}$$

The network representation of a semantic space corresponding to given text corpora is a simple undirected graph $G = (V, E)$ with a set of n vertices (nodes, words) V and a set of edges E such that two nodes i and j have an edge between them if $sim(i,j) \geq \delta$, where δ is a predefined threshold (referred to as the slicing cutoff, which we generally set to be greater than 0.5 in the context of cosine similarity).

Network analysis and dense clusters identification

Network analysis and visualization presented in this paper is handled using iGraph (Csardi et al. 2006) and NetworkX (Hagberg et al. 2008) libraries in Python 3.7. The structural network characteristics (diameter, average distance, clustering coefficients, node centralities) are computed using iGraph library as it is much faster than NetworkX. The visualization is also done using iGraph tools since it generally produces nicer layout (see Zinoviev (2018) for more details on comparison of network analysis tools in Python). All other graph manipulations, as well as calling a linear integer programming solver for finding dense clusters, is done using NetworkX library.

The largest cliques in the considered graphs are identified using a linear integer programming formulation γ-**QC** (Pastukhov et al. 2018) for $\gamma = 1$ and some pre-processing techniques (if necessary) that are described in the same reference. The formulation contains n binary variables and n constraints, which allowed us to solve the maximum clique problem in any considered network in a reasonable time. The maximum density-based γ-quasi-cliques (subgraphs with guaranteed edge density γ) are identified using a linear mixed-integer programming (MIP) formulation **F3** from (Veremyev et al. 2016). All MIP

formulations are solved using Gurobi Optimizer 8.1 (Gurobi Optimization LLC 2019) using Python interface.

Results and discussion

In this section, we present the results on global and local characteristics of the considered human built and machine built (learnt) networks. Specifically, we describe the characteristics of WordNet and Moby Thesaurus networks, as well as the word2vec similarity networks constructed using Google News and Amazon Reviews datasets. Further, we compare dense clusters in ego networks and show that such clusters obtained in the word2vec similarity networks appear to produce consistent and meaningful results.

Structural characteristics of human built semantic networks

WordNet network characteristics

WordNet is a large lexical database of English terms (words) developed by George Miller and colleagues (Miller 1995; Fellbaum 1998) in which words are grouped into sets of cognitive synonyms (*synsets*), each representing a distinct concept. The dataset that we analyze was accessed using NLTK (natural language toolkit) module (Bird et al. 2009) in Python 3.7; it contains approximately 117K synsets and 148K words. Although this database includes various semantic relations among words and concepts (e.g., hyponym, meronym, entailment), we construct and analyze network based on the main relation among words in WordNet, which is synonymy. In this case, two words are connected by an edge if they share the same meaning or concept and interchangeable in many contexts.

The constructed network has approximately 35K isolated nodes, i.e., words without synonyms (e.g., 'abacus', 'abbreviation', 'absorber', 'dine'). The remaining 113K nodes form 29K connected components, in which the largest one contains 32611 words and the second largest one contains only 43 words. Hence, the network includes the connected component that spans about 22% words, roughly 23% of the words are isolated nodes, and the remaining half of the words form very small components of sizes not greater than several dozen nodes.

Table 1 reports the basic structural characteristics of the largest connected component of the constructed WordNet synonyms network and Fig. 1 illustrates its degree distribution. Since there are two common ways of representing degree distributions in the

Table 1 Basic characteristics of the largest connected component of WordNet synonyms and Moby Thesaurus networks

	WordNet	Moby Thesaurus
Number of nodes	32611	103306
Number of edges	119463	1783357
Average degree	7.32	34.52
Largest degree	152	1486
Diameter	23	9
Average distance	6.89	3.81
Global clustering coefficient	0.36	0.19
Average local clustering coefficient	0.62	0.66
Degree assortativity	0.26	0.03
Largest clique size	34	68

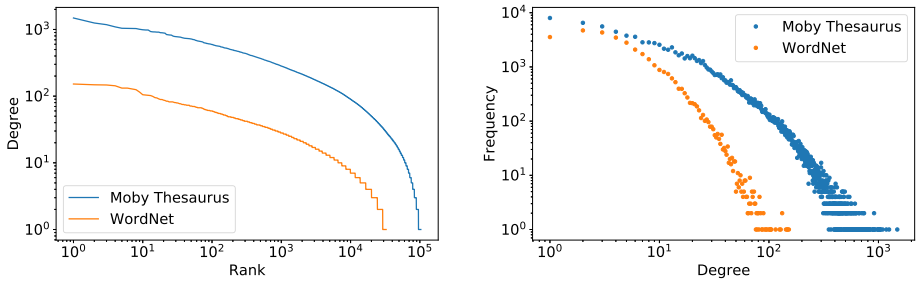


Fig. 1 Degree distributions of largest connected component in WordNet synonyms and Moby Thesaurus networks

literature, we provide two respective figures. In the first one the nodes are ranked according to their degree and are plotted on the corresponding rank-degree curve. In the second one, for each degree value we plot the number of words (frequency) with that degree in the network. The global network characteristics are somewhat similar to the ones calculated for other real-life networks in various domains (Newman 2003) and indicate that this network is small-world (Watts and Strogatz 1998) (that is, it has a high clustering coefficient, small diameter, and small average distance).

Moreover, we have identified the largest clique in this network which contains 34 words (Table 2). The synset with the largest number of words, however, contains 28 synonyms. By definition, all words in one synset form a clique in the constructed synonym network. Hence, although all pairs of nodes (words) in a clique are synonyms, they may correspond to different meanings (concepts).

We have also identified the most important or central nodes in these networks. Specifically, since in the English language (and many other languages as well) some words

Table 2 Largest cliques in WordNet and Moby Thesaurus networks, as well as in word2vec Google News and Amazon Reviews networks containing words from WordNet

Network	Size	Words
WordNet	34	batch deal flock good_deal great_deal hatful heap heaps lot lots mass mess mickle mint mountain muckle passel peck pile piles plenty pot quite_a_little raft rafts sight slew slews spate stack stacks tidy_sum wad wads
Moby Thesaurus	68	abominable arrant atrocious awful base beastly beneath_contempt blameworthy brutal contemptible deplorable despicable detestable dire disgusting dreadful egregious enormous fetid filthy flagrant foul fulsome grievous gross hateful heinous horrible horrid infamous lamentable loathsome lousy monstrous nasty nefarious noisome notorious obnoxious odious offensive outrageous pitiable pitiful rank regrettable reprehensible repulsive rotten sad scandalous schlock scurvy shabby shameful shocking shoddy sordid squalid terrible too_bad unclean vile villainous woeful worst worthless wretched
Google News (threshold 0.7)	14	Amelanchier Clethra Euonymus Eupatorium cotoneaster deciduous_holly flowering_quince marsh_marigold monarda scabiosa silky_dogwood snowberry trumpet_honeysuckle winterberry
Amazon Reviews (threshold 0.8)	13	ante dinero embargo ese falla haber hoy leer lento pas persona saber sus

are used more frequently than others and their usage follows Zipf's law (Powers 1998) (given a large sample of words used, the frequency of any word is inversely proportional to its rank in the frequency table), it is interesting to see if more common or frequently used words have more central positions in the semantic spaces as well. To quantify the word position in the semantic network we used four classic centrality measures (degree, closenness, betweenness and PageRank) which capture the complementary aspects of node importance based on their connectivity, location and proximity to other nodes within the network. For more details on various centrality measures see, e.g., Boldi and Vigna (2014).

Table 3 lists the 15 most central words in the largest connected component of WordNet synonyms network ranked according to four selected centrality measures along with the corresponding centrality scores. Thus, the word 'pass' has the highest degree 152. It also has the highest betweenness score 0.026, which roughly means that 2.6% of all shortest paths in this connected component go through this word. It should be also noted that all top 15 words have *closeness* centrality scores that are very close to each other. All top 15 scores span a small range 0.22-0.23, which is an inverse of the average distance and means that all these words have roughly 4.5 average distance to all other nodes (nodes with highest score have the smallest average distance to other nodes). This is a common issue with the closeness centrality score observed in many networks, and it is also present in this semantic network: its values tend to span a rather narrow range from the smallest to the largest (see, e.g., Newman (2018) for more discussion on centrality score distributions in real-world networks). Note that the scores of the most central nodes of other centrality measures span a much wider range, which makes them more suitable for word ranking. As the PageRank centrality score is essentially a probability distribution of some random walk over the network, a PageRank score of 0.00042 of the word 'pass' means that a person doing random walk (defined by PageRank algorithm) in the connected component of WordNet synonyms network with 32K nodes can be found at the word 'pass' with

Table 3 The most central nodes (words) in the largest connected component in WordNet synonyms network according to four classic centrality measures: degree, closeness, betweenness, and PageRank centrality

	Degree		Closeness		Betweenness		PageRank	
Rank	Word	Score	Word	Score	Word	Score	Word	Score
1	pass	152	get	0.23	pass	0.026	pass	0.00042
2	break	148	take	0.23	get	0.025	break	0.00040
3	get	147	make	0.23	take	0.023	hold	0.00036
4	take	143	takings	0.22	break	0.023	check	0.00036
5	make	132	getting	0.22	check	0.022	take	0.00036
6	hold	132	taking	0.22	go	0.021	get	0.00034
7	check	128	break	0.22	make	0.021	run	0.00033
8	go	125	taken	0.22	run	0.020	go	0.00032
9	run	115	making	0.22	hold	0.019	make	0.00030
10	deal	105	made	0.22	draw	0.017	line	0.00030
11	see	104	go	0.22	charge	0.015	cut	0.00029
12	beat	103	broken	0.22	cover	0.012	passing	0.00028
13	set	102	draw	0.22	broken	0.012	charge	0.00028
14	passing	99	run	0.22	clear	0.012	set	0.00028
15	cut	95	pass	0.22	place	0.012	see	0.00026

0.042% chance. If any word is chosen randomly then the chances would be roughly 0.003% (1/32K) or 14 times smaller.

As a final remark we note that the most central words in this synonyms network are indeed very common and can be found in the list of 1000 most frequent words in the English language. We believe that this is an important observation that may have many practical applications and may need to be investigated further to understand the reasons behind it. One possible explanation might be the fact that people naturally try to avoid the usage of the same words many times within a small window of context and attempt to substitute them with the synonyms, which leads to the need of creating more synonyms for the more frequently used words. Another plausible explanation might be that lexicographers working on lexical databases tend to spend more time finding synonyms for more frequent words. Also, one may conjecture that words which are used more often are inclined to have more meanings (homonyms) which results in more synonyms as well. The are some earlier studies that support the hypothesis that "synonym representation covaries with the frequency of word use" (Lepley 1950; Lepley and Kobrick 1952). Hence, more frequently used words are more likely to occupy more central positions in the corresponding semantic networks and more thorough analysis of this fact can be a good direction of further research.

In contrast, we do not observe this pattern in word embedding-based (machine built) semantic networks. This is discussed in more detail below.

Moby thesaurus network characteristics

The project Moby Thesaurus II (Ward 2002) has a publicly available thesaurus dictionary in which each entry has a list of words that are conceptually similar to the entry word. Specifically, the dataset contains a file with 30260 lines, each line starts with a *root* word followed by a set of conceptually similar words. In order to construct a network, for each entry, we connect by an (undirected) edge the *root* word with all its similar words. The resulting network is connected and contains 103306 nodes and 1.7M edges with average degree of approximately 34.5 (Table 1), which is roughly 4.7 times larger than the average degree of the largest connected component of WordNet synonyms networks. This is due to the fact that in Moby Thesaurus synonyms are interpreted in a broader sense than in WordNet. Table 1 reports the basic characteristics of the resulting Moby Thesaurus network, and Fig. 1 illustrates its degree distribution in comparison with the same statistics of the WordNet synonyms network.

As it can be expected, its diameter and average distance are much smaller than that of WordNet. The largest clique in Moby Thesaurus network (Table 2) is exactly twice as large (68 vs. 34) as the largest clique in WordNet network. The highest degree of a node is almost 10 times larger than the highest degree in the WordNet network. Interestingly, average local clustering coefficients of both networks are very similar, which means that each node in both networks has on average a little over 60% of pairs of its neighbors being connected by an edge. The global clustering coefficient (transitivity) of a WordNet network is almost twice as large as global clustering coefficient of Moby Thesaurus network. This can be explained by the fact that Moby Thesaurus has nodes with larger degrees and in real-life networks large degree nodes normally have very small clustering coefficients. The global clustering coefficient weights the contribution of larger degree nodes more heavily as it measures the density of triangles in a network (Newman 2003, 2018).

We have also identified the most central nodes according to four aforementioned centrality measures (Table 4). The most central nodes also seem to be quite common (almost all of them are in the list of 1000 most frequently used words in the English language) and overlap with the most central words in WordNet network. It indicates that in both networks the words which occupy the most central network positions, are also frequently used in English texts. The range of closeness centrality score for most central nodes is also very small, similarly to the observation made in WordNet. However, the most central nodes are on average much closer to other nodes and have the average distance roughly 2.6 (1/0.38) from other nodes.

Structural characteristics of machine built semantic networks

Google news word embedding-based network

The semantic network of Google News word embedding is constructed based on publicly available pre-trained vectors trained on part of Google News dataset (Google Open Source Project 2013) (about 100 billion words) using word2vec algorithms (Mikolov et al. 2013a, b). The dataset contains 300-dimensional vectors for 3 million words and phrases. In our study, for comparison reasons, we consider only words or phrases included in WordNet lexicon. There are 64278 of such terms (words). To get a network representation (backbone) of this semantic space we construct similarity-based networks using cosine similarity and slice it at various threshold levels, i.e., for any given threshold, only pairs of nodes with cosine similarity higher than this threshold are included in the sliced network.

Figure 2 illustrates one of the connected components (the third largest one) in this network sliced at 0.6 cosine similarity cutoff containing 109 nodes and 159 edges. Clearly, the edges do connect the words with similar meanings, e.g., 'confirm' - 'verify', 'stop' - 'halt', 'calculate' - 'compute', etc. This slicing threshold seems to be reasonable to capture the semantic similarity among words. Hence, for further analysis, we consider threshold values around 0.6.

Table 4 The most central nodes (words) in the Moby Thesaurus network according to four classic centrality measures: degree, closeness, betweenness, and PageRank

	Degree		Closeness		Betweenness		PageRank	
Rank	Word	Score	Word	Score	Word	Score	Word	Score
1	cut	1486	set	0.38	cut	0.009	language	0.00084
2	set	1250	cut	0.38	set	0.009	cheese	0.00065
3	turn	1180	turn	0.37	light	0.008	english	0.00057
4	run	1093	run	0.37	color	0.008	magpie	0.00045
5	line	1042	line	0.37	language	0.007	color	0.00041
6	check	1037	point	0.37	turn	0.006	wine	0.00034
7	break	1035	cast	0.37	right	0.006	pigment	0.00031
8	color	1032	light	0.37	head	0.006	fish	0.00030
9	pass	1004	head	0.37	close	0.006	cut	0.00030
10	light	990	measure	0.37	run	0.005	philosopher	0.00030
11	point	981	mark	0.37	line	0.005	parts	0.00027
12	close	980	pass	0.37	flat	0.005	silver	0.00027
13	flat	928	check	0.37	cross	0.005	set	0.00026
14	charge	920	break	0.37	mean	0.005	device	0.00026
15	cast	918	round	0.36	point	0.005	turn	0.00025

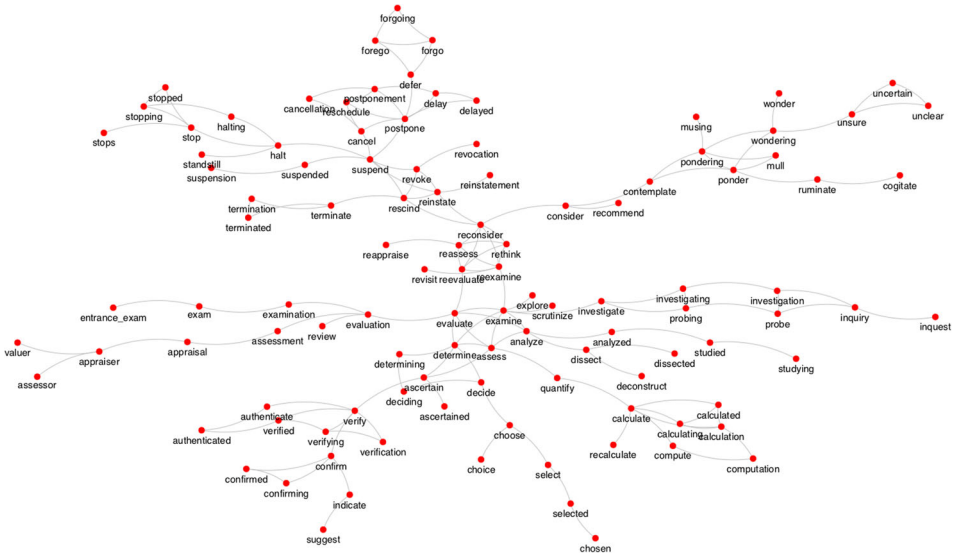


Fig. 2 One of the connected components in the network constructed based on word2vec embedding of Google News containing words form WordNet sliced at 0.6 cosine similarity threshold. The component contains 109 words and 152 edges

Table 5 reports the basic characteristics of the resulting networks (and their largest connected components) sliced at five thresholds (0.5, 0.55, 0.6, 0.65, and 0.7) and Fig. 3 illustrates the degree distributions in the largest connected components of the respective networks. Note that in the resulting networks, by construction, each node has at least one neighbor, so the networks may have less than 64278 nodes, which means that the remaining nodes are isolated (for such words, no other word has similarity higher than the considered threshold). In terms of the average degree, observe that each time the slicing threshold increases by 0.05, the average degree in the largest connected component drops almost twice (for the whole network, the drop is even higher). It suggests that the

Table 5 Basic characteristics of the Google News word embedding-based similarity networks obtained for various slicing thresholds on cosine similarity among the corresponding vectors

Only the words that are also in WordNet are considered.					
Cosine similarity threshold	0.5	0.55	0.6	0.65	0.7
Number of nodes	58186	50576	39363	26509	14694
Number of edges	2033297	801085	373458	78227	19557
Average degree	69.88	31.67	13.84	5.9	2.66
Largest connected component characteristics					
Number of nodes	57102	46717	29374	11363	1739
Number of edges	2032530	798363	263731	62834	5885
Average degree	71.19	34.17	17.95	11.05	6.76
Diameter	21	27	45	67	23
Average distance	5.75	7.89	11.45	18.09	8.65
Global clustering coefficient	0.43	0.43	0.41	0.39	0.28
Average local clustering coefficient	0.37	0.36	0.36	0.36	0.32
Degree assortativity	0.43	0.41	0.40	0.38	0.11
Largest clique size	245	155	89	37	14

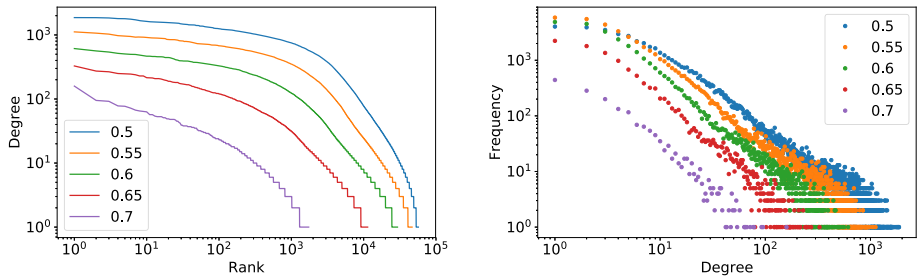


Fig. 3 Degree distributions of the largest connected component of the Google News word2vec embedding similarity networks (containing words from WordNet lexicon) sliced at five different thresholds (0.5, 0.55, 0.6, 0.65, and 0.7) on cosine similarity between the corresponding vectors

dependence of average degree on the slicing cutoff may exhibit a power law behavior. Interestingly, the diameter of the largest connected component with 0.65 threshold containing 11363 nodes is 67, which is unusually large for a real-life network. It indicates that this network topology may have some non-typical structural features. Another interesting property is that for almost all thresholds (except 0.7) global clustering coefficients are greater than average local clustering coefficients. Moreover, these values are almost the same for all thresholds except 0.7 as well. This observation clearly distinguishes these networks from the aforementioned WordNet and Moby Thesaurus networks.

In addition, we have identified the largest cliques in all sliced networks. Similarly to the average degree, their size quickly drops as the slicing threshold increases. However, the largest cliques seem to be formed by unusual and rarely used words (unlike largest cliques in lexical databases). For example, the largest clique with 0.7 cutoff contain 14 words (Table 2), which are the names of flowers and shrubs.

Another interesting observation is that nodes with high degrees are rarely used words in texts. For example, five nodes with the largest degrees for 0.5 slicing cutoff are 'glomerular' (degree: 1866), 'leiomyoma' (1862), 'lichen_planus' (1842), 'eccrine', (1800), 'peroxidase' (1794), whereas the top 5 nodes with highest degrees in WordNet network (Table 3) have substantially smaller degrees in Google News word2vec similarity network with this 0.5 cutoff: 'pass' (degree: 1), 'brake' (34), 'get' (10), 'take' (4), 'make' (2). Note that in WordNet and Moby Thesaurus networks we observe that their most central words are the words which appear relatively frequently in English language (most of them are the list of 1,000 most frequent words). To investigate this observation in more detail we computed the average degrees of most common words in this network sliced at 0.5 cutoff (and other considered networks for comparison purposes). The lists of the most frequent words in the English language are obtained from Moby Thesaurus project (most frequent 1000) and the list of most common 1/3M words (Norvig 2009) (most frequent 3000, 5000, and 10,000). Table 6 reports the average degree of nodes in the considered networks which appear in the corresponding lists of most frequent words. We also report the ratios between the average degrees of most frequent words and all words in the corresponding networks to better illustrate the differences.

Clearly, in WordNet and Moby Thesaurus networks more frequently used words on average have higher degrees. For example, the words which appear in the 1000 most frequent words have on average degree 3 and 7 times larger, respectively, than the average

Table 6 Average degree of all nodes vs. average degree of words appearing in the lists of 1000, 3000, 5000 and 10000 most frequent words (from Moby Thesaurus II and Beautiful Data, Natural Language Corpus data book (Norvig 2009)) in WordNet synonyms network, Moby Thesaurus and Word2Vec embedding of Google News and Amazon Reviews containing WordNet words (cosine similarity threshold = 0.5)

	Lexical databases		Word2Vec embeddings	
Number of words	WordNet	Moby thesaurus	Google news	Amazon reviews
Average degree of most frequent words				
All Words	7.32	34.52	69.89	31.63
1000	22.48	241.33	10.72	8.16
3000	17.58	175.71	11.08	15.88
5000	16.02	157.61	11.50	17.91
10000	14.10	134.01	14.23	21.80
Ratio of average degree of most frequent words to average degree of all words				
1000	3.06	6.99	0.15	0.25
3000	2.39	5.08	0.16	0.50
5000	2.19	4.57	0.17	0.57
10000	1.92	3.88	0.20	0.69

word degree in these networks. The ratios decrease as we consider larger and larger sets of the most frequent words (include less and less frequent words), but nevertheless remain quite high. Even the average degrees of words which appear in the 10,000 most frequent words have on average degree almost 2 and 4 times larger, respectively, than the average word degree in these networks. Hence, more frequent words do tend to occupy more central positions in these networks.

However, the situation for word embedding based networks is exactly the opposite. For example, the average degree of words in word2vec embedding network of Google News (sliced at 0.5 threshold) which appear in 1000 most frequent words is almost 7 times smaller (10.7 vs. 69.9) than the average degree of all words. Moreover, the average degree increases as more and more most frequent words are considered. We find this observation to be very interesting and worth exploring in more detail, since word embeddings are heavily used in various text mining applications. As it will be discussed below, a similar pattern is observed in another word2vec embedding based-network obtained from the Amazon Reviews dataset (although its size and other characteristics are rather different from the Google News dataset).

Amazon reviews word embedding-based network

The semantic network of Amazon Reviews is generated using the dataset containing more than 400,000 reviews (Amazon Reviews dataset 2017) from Amazon's unlocked mobile phone category. Customer reviews have become a ubiquitous and influential part of many people's everyday lives; therefore, constructing and analyzing the network corresponding to these text corpora would be an interesting task. The choice of the reviews category subject (in this case, unlocked mobile phones) is not critical in the context of this study, but the choice of this dataset was mostly motivated by the fact that, unlike other datasets considered above, it consists of text entries written in more casual English language, which may contain errors, misspellings, abbreviations, incomplete phrases, foreign-language words, emojis, etc. Thus, such a dataset may be rather challenging to analyze and it would be interesting to test our proposed graph-based approaches on this dataset.

To generate the word embeddings we first use sentence tokenizer and tweet tokenizer (as it preserves emojis as separate tokens) from NLTK module (Bird et al. 2009) to split the reviews into sentences and sentences into words (tokens). It produced approximately 1.1M sentences, 19M words and 30K unique words. Then we feed the resulting list of sentences split into words into word2vec function in Gensim library (Řehůřek and Sojka 2010) in Python, which returns the word embeddings as vectors. For comparison reasons with other networks, we have generated 300-dimensional vectors keeping other word2vec parameters as their default values (window = 5, number of negative samples = 5, algorithm used = CBOW, number of epochs = 5, min_count=5) and consider only words or phrases included in WordNet lexical database. There are only 8547 such words. However, we will use other words for analysis of clusters of new words and emojis later. Then we construct similarity-based networks using cosine similarity and slice it at 0.5 threshold level.

We observe that since the reviews are not always written in proper English, but occasionally contain misspellings, abbreviations, foreign-language entries, incomplete sentences, grammatical errors, etc., the analysis of global characteristics of the resulting network may not have any meaningful information. For example, the words with the largest degrees ('consecutively', 'wyatt', 'trouble-free', 'wrest', 'asl') and the largest clique with 0.8 threshold containing 13 words (Table 2) do not seem to have any practical interest. In addition, we observe that the words, which are the most central in WordNet synonyms network have very small degrees in this word embedding-based network, e.g., 'brake' (degree 1), 'get' (9), 'take' (1).

Similarly to results for the Google News semantic network mentioned above, the average degree of the most frequent English words in this network is also very small in comparison to the total average degree (Table 6). For example, the words that appear in the 1,000 most frequent words have degrees on average four times smaller (8.16 vs. 31.63) than the average node degree in the network. Although we observed this effect only for two machine built semantic networks, one may hypothesize that this property might be common for semantic spaces corresponding to word embeddings. Verifying this hypothesis on a larger variety of text datasets may be one of the potential future research directions.

Dense clusters in ego networks

In the previous sections, we presented results on the structural characteristics and largest cliques in all considered networks. Although this information might be useful to better understand global topological properties of the semantic networks, analyzing ego networks (networks around certain words or word phrases (Everett and Borgatti 2005; Newman 2018)) and finding dense clusters (cliques or clique relaxations) in these networks may provide more insights into the local structure of semantic space and have many practical applications. Specifically, we demonstrate the advantages of this approach in networks constructed based on word embeddings.

Cliques in ego networks

In practice, one might not necessarily need to identify a large cluster (i.e., clique) in the entire network of words, but to find a large cluster that contains a *given* word of interest: this can be interpreted as a comprehensive set of synonyms for that word sharing the same meaning. For example, consider the word 'happy'. In the Google News word embedding

network sliced at 0.5 cutoff this word has 31 neighbors (Table 7). Note that the list of neighbors has not only words with positive sentiment, such as 'pleased' or 'glad', but also words with negative sentiment (e.g., 'sad', 'unhappy', 'anxious'). This is due to the way the word embeddings are generated: words which appear in similar context tend to be closer to each other in the corresponding semantic space. The largest clique in the subgraph of this network induced by these 31 neighbors contains 11 words. Note that all words in that clique have positive attitude and appear to be synonyms of the word 'happy'. We observe the similar trend in the word2vec embedding of Amazon Reviews network (with only WordNet words) sliced also at 0.5 cutoff. The neighborhood of the word 'happy' contains 28 words and some of them have negative sentiment as well (e.g., 'displeased', 'dissatisfied'). The largest clique in the induced by this neighborhood subgraph has 14 words; however, there are still some words with a negative sentiment in that clique. This might be due to the fact that this word embedding is based on much smaller text corpora and indicates the importance of training word embeddings on large datasets (or parameter tuning is required). Moreover, higher slicing cutoffs might need to be used.

Therefore, the structure of an ego network of a particular word in word embedding-based networks provides much richer information about its semantic neighborhood in addition to semantic distances (similarity scores) to other words. We believe that this is an important observation that can be used to measure the quality of word embeddings. One way to measure this quantitatively would be to use some benchmark ego network clusters, for instance, constructed for the most frequent 1000 words (which would comprise most of the text), and quantify the deviation of these clusters for a given word embedding, which would measure the distance from that benchmark.

Table 7 Neighborhood of the word 'happy' and the largest cliques in the corresponding neighborhoods in the considered networks

Cluster type	Size	Words
Word2vec embedding of Google News network (threshold = 0.5)		
Neighborhood	31	anxious appreciative chuffed confident delighted disappointed eager ecstatic elated excited fortunate glad good grateful hopeful lucky nice okay optimistic overjoyed pleased proud relieved sad satisfied sorry sure surprised thankful thrilled unhappy
Clique	11	appreciative delighted ecstatic elated excited glad grateful overjoyed pleased proud thrilled
Word2vec embedding of Amazon Reviews network (threshold = 0.5)		
Neighborhood	28	delighted disappointed disgusted displeased dissatisfied ecstatic excited familiar frustrated glad glade grateful impressed infatuated mad obsessed optimistic picky pleased proud relieved sad satisfied thrilled unhappy unimpressed unsatisfied upset
Clique	14	delighted disappointed disgusted displeased dissatisfied grateful impressed pleased proud satisfied thrilled unhappy unsatisfied upset
WordNet and Moby Thesaurus networks		
WordNet Neighborhood	3	felicitous glad well-chosen
Moby Thesaurus Clique	34	advantageous advisable appropriate becoming befitting congruous convenient decent desirable expedient favorable feasible felicitous fit fitting fructuous good likely meet opportune politic profitable proper recommendable right seasonable seemly suitable timely 'to be desired' useful well-timed wise worthwhile

The number of neighbors in Moby Thesaurus is 233 (words not reported)

For comparison reasons, we have also reported in Table 7 neighbors of the word 'happy' and the largest cliques in its ego networks in WordNet and Moby Thesaurus networks, respectively. It seems that these lexical databases can be improved by including more relevant words or synonyms using such word embedding-based networks. As it has been discussed in Gaillard et al. (2011), the list of synonyms in the lexical databases is based on a lexicographer's judgment, which might be subjective and may not fully reflect the general language use.

Incorporating cliques in ego networks into machine learning algorithms

In this section we provide an example illustrating how information about cliques in ego networks extracted from semantic networks constructed based on word embeddings can be used in machine learning applications. Specifically, we consider sentiment analysis task which we conducted on Amazon Reviews dataset. Intuitively, as we have observed in the previous section, all words in a clique in the ego network tend to share similar meaning with the given ego (word). Therefore, the performance of a machine learning algorithm should be better if it "knows" that all words from a given clique and its ego have the same meaning.

Specifically, we selected reviews from the Amazon Reviews dataset with positive and negative sentiments and considered a review to be positive if it has rating 4 or 5, and negative if it has rating 1 or 2. Note that the Amazon Reviews dataset contains customer rating for each review in the respective column. Totally, out of 413840 reviews, there are 382075 (approximately 92.3%) positive and negative reviews, from which approximately 75% (286556) are positive and the other 25% (95519) are negative. Note that the remaining reviews (7.7% of the total number) have rating 3 and are considered to be neutral: these reviews are not included into the sentiment analysis example discussed here.

To perform data split, text vectorization (transforming reviews into high-dimensional feature representations), classification and evaluation tasks, we use *scikit-learn* (Pedregosa et al. 2011) module in Python. First, we split the dataset (reviews and corresponding sentiments) into training and test sets using the respective function with default parameters (0.75/0.25 split) and random state set to 0. As a result, the training set (on which a classifier will be trained) contains 286556 reviews (213794 positive and 72762 negative) and the test set (on which the performance of a classifier will be evaluated) contains 95519 reviews (71203 positive and 24316 negative).

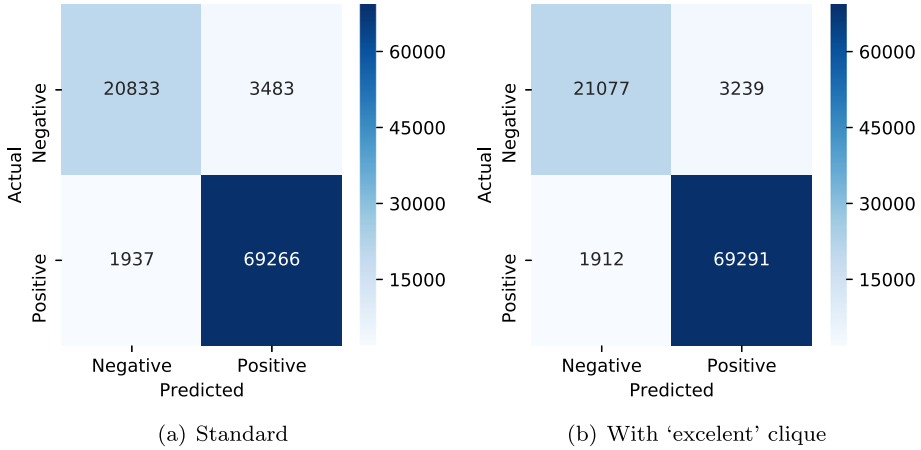
To vectorize the reviews (in order to feed them into a machine learning algorithm) and illustrate how to incorporate the information about cliques in ego networks into vector representations of reviews we use the standard *bag-of-words* approach, as it is known to be very effective in document classification tasks, and serves as a simple and good starting point (baseline) for performing more complex methods (Bengfort et al. 2018). Moreover, it does not consider any order among word appearances in the texts and, hence, does not encode any relations (contextual use) among words. Essentially, this approach represents each review as a vector whose length is equal to the vocabulary of the whole training set of reviews. We use frequency vector representations of reviews, which simply fill in the vector with frequency of each word as it appears in a given review.

Next, we perform two vectorization processes using *CountVectorizer* function in scikit-learn. To be consistent with previous computational experiments on constructing word embedding based on Amazon Reviews dataset, we also feed tweet tokenizer from NLTK

module (Bird et al. 2009) into this function to split the reviews into tokens and set the minimum word count parameter to 5 (min_df=5). The rest of parameters are kept at their default values. The first process uses all tokenized reviews as is and the second one substitutes words in a clique by some unique word (say, 'clique1'). For illustration purposes we identified the maximum clique containing the word with one of the largest coefficient in the logistic regression performed on the training set, as it may have high impact on the classifier performance when compressed into a single word. Interestingly this word is *'excelent'*, which turned out to be the misspelled word 'excellent', and the largest clique of size 7 containing this word in Amazon Reviews network constructed from the respective word embedding described above (cosine similarity cutoff is 0.5 and the neighbourhood of top 10 similar words is considered) also seems to contain misspelled words ('excelent', 'exelent', 'excellente', 'exellent', 'exccelent', 'ecellent', 'excente').

As a result, in the first setup the vocabulary (and the corresponding feature space dimension) contains 23530 unique tokens (which appear at least 5 times in the training set of reviews) and the vocabulary in the second setup contains 23524 unique tokens (since all 7 words in a clique are treated as the same word). Figure 4 reports the confusion matrices of the predictions of a classifier (logistic regression) on a test set when it was trained on a training set of reviews without any modifications (Fig. 4a) and when all words in a clique containing the word 'excelent' in the respective word embedding based network were substituted by one unique word (Fig. 4b). As it can be observed, a classifier trained when information about clique is introduced showed slightly better performance for both predicting true positive and true negative reviews, and have smaller number of false positive and false negative predictions.

Although we have illustrated how to incorporate information about a single clique into machine learning algorithms from the respective semantic networks and demonstrated its benefits, this methodology can be further generalized by incorporating any number of cliques containing a given set of words, for example, those that might be important in classification tasks. It should also be noted that in practice cliques may overlap and their



(a) Standard (b) With 'excelent' clique

Fig. 4 Confusion matrices for (**a**) standard classification and (**b**) classification when words included in a maximum clique around word 'excelent' from the respective word embedding are treated as words with the same meaning. Accuracy scores of a classifier on a training and test instances are equal to (**a**) 0.9483, 0.9433 and (**b**) 0.9521, 0.9461 respectively

structure depends heavily on the chosen similarity threshold level (existence of an edge between two words based on cosine similarity value). Therefore, future research should address methods for clique selection and incorporation into machine learning algorithms in more detail.

In addition, we would like to point out a few other comments and observations related to the illustrated technique and its potential use. First, since the clique in the aforementioned illustrative example was extracted from the semantic network constructed form word embedding trained on the entire dataset (which included both the training and the test set), it should be done more carefully in practice. Ideally, one should use word embeddings trained on a set of reviews that does not include the test set, or on other text corpora (preferably, with a similar style). For example, in order to test this approach using word embedding trained on other text corpora, we conducted additional computational experiments where the incorporated clique was obtained from the word embedding of Google News dataset containing the word 'happy' ('appreciative', 'delighted', 'ecstatic', 'elated', 'excited', 'glad', 'grateful', 'overjoyed', 'pleased', 'proud', 'thrilled') from Table 7. For these experiments, we still observed a slight improvement in the accuracy of sentiment analysis performed on the original training and test datasets: 0.9490 vs. 0.9483, and 0.9439 vs. 0.9433 with and without incorporating this clique, respectively.

Second, many studies indicate that using word embeddings instead of bag-of-words is more preferable in machine learning and usually leads to better performance, so it might be interesting to see if the same technique can be applied in this case as well. Although we believe that this is a promising research direction, it should be approached carefully as it is not clear how to represent documents using a more complex semantic space and if there is any intuitive explanation for that. For example, we found that using most common word embedding representations such that each review is represented by an average vector of words in it (which might be hard to justify and interpret the results) shows worse performance than standard bag-of-words representations. For example, the training/test scores of the same classifier are approximately 0.91 and 0.92 when word embeddings of Google News and Amazon Reviews (considered in the paper) are used, respectively (vs. 0.94-0.95 with standard bag-of-words representations). Note that similar lower accuracy scores are reported in Bansal and Srivastava (2018) for the sentiment analysis of the same dataset when word embeddings are used. Although in certain NLP tasks using distributed representation of words (word embeddings) shows better performance than bag-of-words representations, we attribute this observation to the fact that in our case (sentiment analysis of short reviews) the polarity of a review might be heavily dependent on a small set of very influential words (e.g., love, great, fantastic, etc.) whose effect might be diminished after the vector averaging process. The bag-of-words model does not have this drawback as every feature represents a certain word and its impact on the polarity of a review can be captured better. Moreover, incorporating cliques into bag-of-words representation helps mitigate one of its main weakness, namely, the lack of semantic similarity among words, without increasing complexity. Hence, we believe that incorporating information about cliques into bag-of-word representations allows extracting only local and most important information from word embeddings and, hence, infuse bag-of-word representations with semantic information and take advantages of both representations. Moreover, the resulting feature space is still quite simple and intuitive and the results can be interpreted in the same manner as with standard bag-of-word

representations, which may help further improve the performance of machine learning algorithms.

Quasi-cliques in ego networks

As it is mentioned in Zinoviev (2018), game developers and creative writers are often in need of a collection of adjectives that characterize a particular property in a range from 'very good' to 'very bad'. Although, such word rankings can be based on a survey data, the amount of that data is very limited and is based on responses of relatively few respondents. The semantic spaces constructed from word embeddings intuitively should be good sources of that information as they are usually trained on large text corpora written by many people and reflect the *use* of these words in a natural language.

For example, consider the word 'amazing'. There are 52 neighbors of this word (Table 8) in the Google News network sliced at 0.5 threshold, which appear to be semantically similar (synonyms) to this word. One may note that some of them are better than others. For example, 'fantastic' seems to be better than 'nice' and closer to the word 'amazing'. The largest clique in the ego network of 'amazing' contains 11 words, all of which seem to be very close to the meaning of the original word.

Intuitively, if we start relaxing the edge density requirement and find the largest clusters with a given edge density, these clusters should include more and more words with a lower level of perceived "excitement" than words in denser clusters. Table 8 reports the largest γ-quasi-cliques for $\gamma = 0.9, 0.8, 0.7, 0.6$ (sizes 15, 18, 23, 28), which means that the edge densities of these clusters are at least 90%, 80%, 70% and 60%, respectively (in fact, they are roughly equal to these values). Clearly, the cluster sizes increase as γ decreases, and the sets of words contained in the respective clusters appear to support the aforementioned intuition.

In addition, Table 8 reports the neighbors and the largest cliques in the ego networks of the word 'amazing' in other semantic networks. Interestingly, in word embedding of Amazon Reviews network almost all the neighbors are also present among neighbors in word embedding of Google News network; whereas the largest cliques in both ego networks have almost the same size and large overlap. It indicates that the contextual use of these words in Google News text corpora and customer reviews is very similar.

New words and emojis clusters

The language that people use in everyday life constantly evolves and more and more new words are being added to the lexicon. For example, emojis and various abbreviations (e.g., 'fb'), which have become an integral part of many texts nowadays, can be treated as words as well. In the final set of experiments in this study, we have generated word embeddings based on Amazon Reviews and consider only relatively frequently used words (set minimum word count parameter to 20). Then we constructed the corresponding semantic networks and identified the maximum cliques (Fig. 5) containing the words 'twitter', and the 'grinning face with smiling eyes' emoji (note that slicing thresholds are different). As one can observe, the respective similar words are naturally clustered together, which is true even in the case of emojis. This observation may help to uncover the perceived meanings of emojis, which are found to have diverse interpretations across platforms and it is not well understood how people interpret the meaning of emojis (Miller et al. 2016).

Table 8 Neighborhood of the word 'amazing', largest clique and quasi-cliques in its neighborhood in word2vec embedding of Google News network with 0.5 similarity cutoff

Cluster type	Size	Words
Word2vec embedding of Google News network		
Neighborhood	52	amazed amazingly astonishing astounding awesome awful beautiful breathtaking brilliant captivating dazzling ecstatic excellent exceptional exciting exhilarating extraordinary fabulous fantastic fascinating gorgeous gratifying great humbling impressive incomparable incredible indescribable inspiring interesting lovely magical magnificent marvelous mesmerizing miraculous nice phenomenal remarkable spectacular stunning stupendous superb terrific tremendous unbelievable unforgettable unreal weird wonderful wondrous wow
Clique	11	brilliant excellent fabulous fantastic great incredible marvelous phenomenal superb terrific wonderful
0.9-quasi-clique	15	awesome brilliant excellent fabulous fantastic great incredible magnificent marvelous phenomenal remarkable superb terrific unbelievable wonderful
0.8-quasi-clique	18	awesome brilliant excellent fabulous fantastic great impressive incredible magnificent marvelous phenomenal remarkable spectacular superb terrific tremendous unbelievable wonderful
0.7-quasi-clique	23	astonishing awesome breathtaking brilliant dazzling excellent fabulous fantastic great impressive incredible lovely magnificent marvelous phenomenal remarkable spectacular stunning stupendous superb terrific unbelievable wonderful
0.6-quasi-clique	28	astonishing awesome beautiful breathtaking brilliant dazzling fabulous fantastic gorgeous great impressive incredible lovely magnificent marvelous mesmerizing nice phenomenal remarkable spectacular stunning stupendous superb terrific unbelievable unforgettable wonderful wondrous
Word2vec embedding of Amazon Reviews network		
Neighborhood	31	astounding awesome awful beautiful brilliant excellent exceptional exquisite fabulous fantastic good gorgeous great impressive incredible insane lovely magnificent nice outstanding phenomenal remarkable spectacular stellar stunning superb terrific unbeatable unbelievable unreal wonderful
Clique	12	awesome excellent exceptional fabulous fantastic great outstanding phenomenal stellar superb terrific wonderful
WordNet network		
Neighborhood	23	amaze astonish astonishing astound awe-inspiring awesome awful awing baffle beat bewilder dumbfound flummox get gravel mystify nonplus perplex pose puzzle stick stupefy vex
Clique	16	amaze baffle beat bewilder dumbfound flummox get gravel mystify nonplus perplex pose puzzle stick stupefy vex
Moby Thesaurus network		
Neighborhood	36	astonishing astounding awesome breathtaking confounding dazzling exciting extraordinary eye opening eye-opening fabulous good incredible marvellous marvelous mind-boggling miraculous overwhelming phenomenal portentous prodigious remarkable sensational shocking spectacular staggering startling strange striking stunning stupendous superhuman surprising tremendous wonderful wondrous
Clique	16	astonishing astounding extraordinary fabulous incredible marvelous miraculous phenomenal prodigious remarkable sensational strange striking stupendous wonderful wondrous

Neighborhoods and largest cliques in these neighborhoods in other networks are also reported

Extensions and future research

As mentioned above, the main goal of this study is to illustrate that network science and graph theory are potentially useful tools for exploring semantic spaces; however, there are many possible ways to develop related network-based techniques beyond the scope of this paper. In this section, we discuss potential extensions and future research in the context of the results of this study. Below we outline several directions that we believe are worth exploring in detail in subsequent studies.

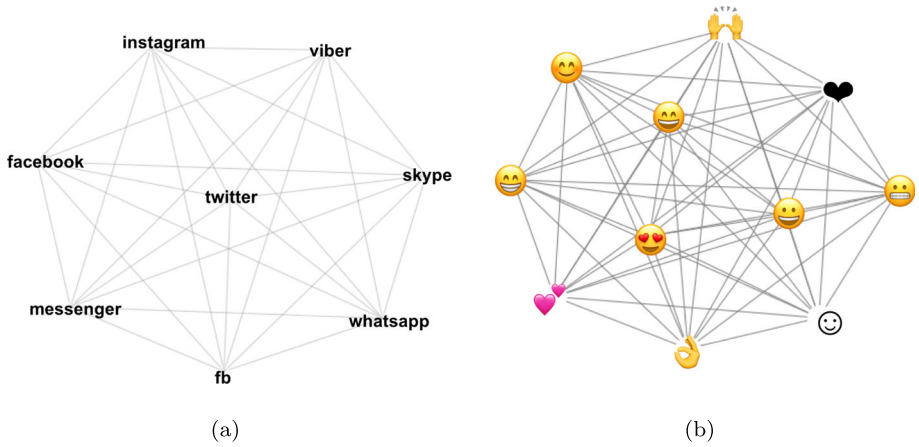


Fig. 5 Maximum cliques containing in an ego network of **a**) word "twitter" (similarity cutoff=0.65), **b**) "grinning face with smiling eyes" emoji (similarity cutoff=0.55). The network of words is constructed using the word embedding of Amazon Reviews dataset

Similarity measures

One of the important aspects that is worth exploring further is the choice and the comparison of similarity measures between word2vec representations of words. Clearly, the topological properties of the constructed networks of words would depend on the choice of similarity measure, and it is possible that the results of a similar graph-theoretic study of word embeddings would be different if another similarity measure was used for constructing the respective graphs. Thus, besides the popular cosine similarity measure that we chose in this study, it would be interesting to conduct similar studies by constructing graphs based on other well-known similarity measures (i.e., Euclidean distance, Pearson correlation coefficient, Jaccard similarity, etc.) In particular, it would be interesting to see if the discrepancy in global connectivity patterns of human built and machine built word networks would still be observed when these networks are constructed using other similarity measures. If this is indeed the case, this may indicate the need for an in-depth study of tuning the parameters of word2vec embedding techniques so that they would produce word networks with more intuitive and consistent node degree distributions (i.e., more frequently used words having higher degrees).

Datasets

Another important aspect that naturally follows from this work is the choice of datasets (text corpora) on which the proposed graph-theoretic approaches can be tested. In this study, we chose two datasets that are easily accessible in the public domain, with one dataset (Google News) mostly containing text written by journalists in proper English, and the other one (Amazon Reviews) containing text entries that may contain errors, typos, occasional words from other languages, emojis, etc. Rather than cleaning up the Amazon Reviews dataset, we deliberately decided to test the proposed graph-based techniques on this dataset as is, in order to see whether our tools would be able to extract interesting and intuitively consistent patterns from such text corpora. It turned out that the identified dense clusters were consistent and meaningful on both of the considered datasets, which indicates that these graph-based tools are rather robust with respect to

possible impurities in text entries. Moreover, as the above example suggests, the information about dense clusters may help to reveal and correct misspelled words as well as improve the performance of machine learning algorithms. One may also argue that the usage of slang, informal acronyms, words from other languages, and emojis (which could also be treated as a separate language) is an essential part of the modern casual written English language; therefore, it is important that the developed techniques work consistently not only on texts written in proper English, but also on "mixed" text corpora such as Amazon Reviews considered here.

Clearly, the methods proposed in this paper, as well as text analytics tools developed in other studies, should be tested on a variety of text corpora. We believe that the datasets used in this study are a good representative sample that illustrates the applicability of our techniques; however, it would certainly be of interest to test them on a much larger selection of other text corpora, such as various discipline-specific texts, modern vs. classical literature, etc.

Network community structures

From the perspective of network science and graph theory, there are also a lot of possible extensions of the proposed techniques that can be addressed in future research. For instance, in addition to cliques and γ-quasi-cliques that were considered here, one may employ other community-type clusters in word networks and investigate the respective results. In particular, there are many other types of clique relaxations (see Pattillo et al. (2013b) for a comprehensive review) that can be used in the context of identifying clusters of similar data points. One may also consider partitioning of the constructed word networks into modularity-based communities (i.e., by maximizing modularity, which characterizes the difference between the number of intra-cluster and inter-cluster links). It would be interesting if the identified modularity-based communities exhibit any patterns related to the meanings of the words within each community.

Integration with other natural language processing techniques

Last but not least, it is important to point out that there exists a large body of work on natural language processing and text analytics using machine learning techniques (e.g., sentiment analysis, syntactic parsing, recommender systems) which use various text vectorization approaches to produce underlying input representations of words. Hence it would be interesting to investigate in a greater detail how to boost the performance of such natural language processing tasks by incorporating network-based features extracted from the respective semantic networks (similar to the example presented in this paper on using cliques in sentiment analysis).

In addition, it might be interesting to analyze how to integrate word embeddings with lexical databases using network science tools and concepts. Specifically, a semantic network for a given word embedding can be enhanced by adding edges which are present in the semantic network constructed from some lexical database but not in the word embedding-based network, i.e., connecting true semantically similar words which are not close to each other in the word embedding. For example, as we have observed, frequently used words, which have high degrees in WordNet or Moby Thesaurus networks, tend to have small degrees in word embeddings-based semantic networks. Hence, such missing links can be added to the word embedding-based networks and the resulting network

can, again, be embedded in some geometric space preserving small distances among connected nodes. Intuitively, this word embeddings should capture not only contextual similarity among words, but also true semantic similarity; therefore, performance comparisons of natural language processing tasks with original word embedding and the new one would be of interest and could potentially be studied in the future by the machine learning research community.

Concluding remarks

In this study, we have shown potential benefits of applying network science and graph-theoretic techniques to the analysis of semantic spaces. The comparison of machine built and human built semantic networks reveals interesting observations, which highlight potential advantages and disadvantages of the respective network models. Specifically, human built networks exhibit global characteristics that are more consistent with the frequency of word usage in the English language, whereas machine built networks lack such global characteristics. However, we observed that local properties of machine built networks (specifically, dense clusters) allow one to produce meaningful and consistent sets of synonyms for given words, which could enhance the existing lexical databases and improve the performance of machine learning algorithms. Moreover, the considered models of community-type dense clusters exhibit inherent flexibility in the sense that as one relaxes the edge density of a cluster (starting from a clique and moving on to γ-quasi-cliques with decreased values of γ), one may increase the size of the synonyms set and control the level of semantic similarity of the words in this set. On the other hand, since the existing word embedding techniques appear to produce networks with somewhat counter-intuitive global connectivity patterns, there may be a potential for further research in order to synthesize human built and machine built semantic networks. We should also note that although this study used primarily English-language text corpora, the proposed approach may work on texts in other languages with similar organization principles (for instance, in our study of Amazon reviews dataset, occasional Spanish-language entries did not affect the quality of the identified clusters), thus potentially producing text data analytics tools applicable to some other languages besides English. We hope that this work could initiate promising research directions that would be of interest to both machine learning and network science communities.

Abbreviations

MIP: Mixed-integer programming; NLTK: Natural language toolkit; NP-hard: Non-deterministic polynomial-time hard

Acknowledgements

This material is based upon work supported by the AFRL Mathematical Modeling and Optimization Institute.

Authors' contributions

AV and AS conducted the experiments. All authors contributed to analyzing the results and writing the paper. All authors read and approved the final manuscript.

Funding

The work of V. Boginski and A. Veremyev was supported in part by the U.S. Air Force Research Laboratory (AFRL) award FA8651-16-2-0009. The work of A. Semenov was supported in part by the U.S. Air Force Research Laboratory (AFRL) European Office of Aerospace Research and Development under Grant FA9550-17-1-0030.

Availability of data and materials

The datasets analyzed in this study are publicly available online: the respective sources cited in the References section.

Competing interests

The authors declare that they have no competing interests.

Author details
[1]Department of Industrial Engineering and Management Systems, University of Central Florida, 12800 Pegasus Drive, Orlando, FL, USA. [2]Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland. [3]Air Force Research Laboratory, Eglin AFB, FL, USA.

Received: 11 March 2019 Accepted: 22 October 2019
Published online: 13 November 2019

References

Abbott JT, Austerweil JL, Griffiths TL (2015) Random walks on semantic networks can resemble optimal foraging. Psychol Rev 122(3):558–569
Abello J, Pardalos PM, Resende MGC (1999) On maximum clique problems in very large graphs. In: Abello J, Vitter J (eds). External Memory Algorithms and Visualization. American Mathematical Society, Boston. pp 119–130
Abello J, Resende MGC, Sudarsky S (2002) Massive quasi-clique detection. In: Rajsbaum S (ed). LATIN 2002: Theoretical Informatics. Springer-Verlag, London. pp 598–612
Altuncu MT, Mayer E, Yaliraki SN, Barahona M (2019) From free text to clusters of content in health records: an unsupervised graph partitioning approach. Applied Network Science 4(1):2
Amazon Reviews dataset (2017) Unlocked Mobile Phones. https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones. Last accessed 15 Feb 2019
Angel A, Sarkas N, Koudas N, Srivastava D (2012) Dense subgraph maintenance under streaming edge weight updates for real-time story identification. Proc VLDB Endowment 5(6):574–585
Bader GD, Hogue CW (2003) An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics 4(1):2
Bales ME, Johnson SB (2006) Graph theoretic modeling of large-scale semantic networks. J Biomed Inf 39(4):451–464
Bansal B, Srivastava S (2018) Sentiment classification of online consumer reviews using word vector representations. Procedia Comput Sci 132:1147–1153
Bengfort B, Bilbro R, Ojeda T (2018) Applied Text Analysis with Python: Enabling Language-aware Data Products with Machine Learning. O'Reilly Media, Inc., Sebastopol
Bird S, Klein E, Loper E (2009) Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc, Sebastopol
Boginski V, Butenko S, Pardalos PM (2005) Statistical analysis of financial networks. Comput Stat Data Anal 48(2):431–443
Boginski V, Butenko S, Shirokikh O, Trukhanov S, Lafuente JG (2014) A network-based data mining approach to portfolio selection via weighted clique relaxations. Ann Oper Res 216(1):23–34
Boldi P, Vigna S (2014) Axioms for centrality. Internet Math 10(3-4):222–262
Bomze IM, Budinich M, Pardalos PM, Pelillo M (1999) The Maximum Clique Problem. In: Handbook of Combinatorial Optimization. vol. 4. Kluwer Academic Publishers, Amsterdam. pp 1–74
Borgatti SP, Everett MG (2006) A graph-theoretic perspective on centrality. Soc Netw 28(4):466–484
Borge-Holthoefer J, Arenas A (2010) Semantic networks: Structure and dynamics. Entropy 12(5):1264–1302
Bu D, Zhao Y, Cai L, Xue H, Zhu X, Lu H, et al. (2003) Topological structure analysis of the protein-protein interaction network in budding yeast. Nucleic Acids Res 31(9):2443–2450
Choudhury M, Mukherjee A (2009) The structure and dynamics of linguistic networks. In: Dynamics on and of Complex Networks. Springer. pp 145–166
Cong J, Liu H (2014) Approaching human language with complex networks. Phys Life Rev 11(4):598–618
Crenson MA (1978) Social Networks and Political Processes in Urban Neighborhoods. Am J Polit Sci 22(3):578–594
Csardi G, Nepusz T, et al. (2006) The igraph software package for complex network research. InterJournal Complex Syst 1695(5):1–9
de Jesus HA, Pisa IT, Kinouchi O, Martinez AS, Ruiz EES (2004) Thesaurus as a complex network. Phys A Stat Mech Appl 344(3-4):530–536
Everett M, Borgatti SP (2005) Ego network betweenness. Soc Netw 27(1):31–38
Fellbaum C (ed) (1998) WordNet: An electronic lexical database. MIT press, Cambridge
Fukś H, Krzemiński M (2009) Topological structure of dictionary graphs. J Phys A Math Theor 42(37):375101
Gaillard B, Gaume B, Navarro E (2011) Invariants and variability of synonymy networks: Self mediated agreement by confluence. In: Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing. Association for Computational Linguistics. pp 15–23
Gandomi A, Haider M (2015) Beyond the hype: Big data concepts, methods, and analytics. Int J Inf Manag 35(2):137–144
Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman. https://doi.org/10.1137/1024022
Google Open Source Project (2013) word2vec. https://code.google.com/archive/p/word2vec/. Last accessed 15 Feb 2019
Gurobi Optimization LLC (2019) Gurobi Optimizer Reference Manual. http://www.gurobi.com
Hagberg A, Swart P, S Chult D (2008) Exploring network structure, dynamics, and function using NetworkX. Los Alamos National Lab.(LANL), Los Alamos
Hartwell LH, Hopfield JJ, Leibler S, Murray AW (1999) From molecular to modular cell biology. Nature 402:C47–C52
Holzapfel K, Kosub S, Maaß MG, Täubig H (2006) The complexity of detecting fixed-density clusters. Discret Appl Math 154(11):1547–1562
Hu H, Yan X, Huang Y, Han J, Zhou XJ (2005) Mining coherent dense subgraphs across massive biological networks for functional discovery. Bioinformatics 21(suppl 1):i213–i221
Huang WQ, Zhuang XT, Yao S (2009) A network analysis of the Chinese stock market. Phys A Stat Mech Appl 388(14):2956–2964
Jackson MO (2010) Social and economic networks. Princeton University Press, Princeton
Jia ZY, Tang Y, Xiong JJ, Zhang YC, et al. (2018) Quantitative learning strategies based on word networks. Phys A Stat Mech Appl 491:898–911

Kasch N (2014) Text Analytics and Natural Language Processing in the Era of Big Data. Pivotal Data Labs. Accessed: 6 June 2019. https://content.pivotal.io/blog/text-analytics-and-natural-language-processing-in-the-era-of-big-data
Ke J, Yao Y (2008) Analysing language development from a network approach. Journal of Quantitative Linguistics 15(1):70–99
Lepley WM (1950) An hypothesis concerning the generation and use of synonyms. J Exp Psychol 40(4):527
Lepley WM, Kobrick JL (1952) Word usage and synonym representation in the English language. J Abnorm Soc Psycholo 47(2S):572
Luce RD, Perry AD (1949) A method of matrix analysis of group structure. Psychometrika 14(2):95–116
Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. ICLR
Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013b) Distributed Representations of Words and Phrases and their Compositionality. In: Advances in Neural Information Processing Systems. pp 3111–3119
Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
Miller HJ, Thebault-Spieker J, Chang S, Johnson I, Terveen L, Hecht B (2016) "Blissfully Happy" or "Ready to Fight": Varying Interpretations of Emoji. In: Tenth International AAAI Conference on Web and Social Media. AAAI Press, Palo Alto
Motter AE, De Moura AP, Lai YC, Dasgupta P (2002) Topology of the conceptual network of language. Phys Rev E 65(6):065102
Newman ME (2003) The structure and function of complex networks. SIAM Rev 45(2):167–256
Newman M (2018) Networks. Oxford university press, Oxford
Norvig P (2009) Natural language corpus data. In: Beautiful Data, In Beautiful Data, edited by Toby Segaran and Jeff Hammerbacher. O'Reilly, Sebastopol. pp 219–242
Pastukhov G, Veremyev A, Boginski V, Prokopyev OA (2018) On maximum degree-based-quasi-clique problem: Complexity and exact approaches. Networks 71(2):136–152
Pattillo J, Veremyev A, Butenko S, Boginski V (2013a) On the maximum quasi-clique problem. Discret Appl Math 161:244–257
Pattillo J, Youssef N, Butenko S (2013b) On clique relaxation models in network analysis. Eur J Oper Res 226(1):9–18
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. (2011) Scikit-learn: Machine Learning in Python. J Mach Learn Res 12:2825–2830
Powers DM (1998) Applications and explanations of Zipf's law. In: Proceedings of the joint conferences on new methods in language processing and computational natural language learning. Association for Computational Linguistics, Stroudsburg. pp 151–160
Řehůřek R, Sojka P (2010) Software Framework for Topic Modelling with Large Corpora. ELRA, Valletta
Schneider C (2016) The biggest data challenges that you might not even know you have. IBM Watson. https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/. Accessed: 6 June 2019
Siew CS (2013) Community structure in the phonological network. Front Psychol 4:553
Siew CS (2018) The orthographic similarity structure of English words: Insights from network science. Appl Netw Sci 3(1):13
Siew CS, Wulff DU, Beckage NM, Kenett YN (2018) Cognitive Network Science: A review of research on cognition through the lens of network representations, processes, and dynamics. PsyArXiv. https://doi.org/10.31234/osf.io/eu9tr
Sigman M, Cecchi GA (2002) Global organization of the Wordnet lexicon. Proc Natl Acad Sci 99(3):1742–1747
Sim K, Li J, Gopalkrishnan V, Liu G (2006) Mining Maximal Quasi-Bicliques to Co-Cluster Stocks and Financial Ratios for Value Investment. In: Proceedings of the Sixth International Conference on Data Mining. ICDM '06. IEEE Computer Society, Washington. pp 1059–1063
Spirin V, Mirny LA (2003) Protein complexes and functional modules in molecular networks. Proc Natl Acad Sci 100(21):12123–12128
Steyvers M, Tenenbaum JB (2005) The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. Cogn Sci 29(1):41–78
Sumathy K, Chidambaram M (2013) Text mining: concepts, applications, tools and issues-an overview. Int J Comput Appl 80(4)
Tsourakakis C, Bonchi F, Gionis A, Gullo F, Tsiarli M (2013) Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp 104–112. https://doi.org/10.1145/2487575.2487645
Vazirgiannis M, Malliaros FD, Nikolentzos G (2018) GraphRep: Boosting Text Mining, NLP and Information Retrieval with Graphs. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM. pp 2295–2296. https://doi.org/10.1145/3269206.3274273
Veremyev A, Prokopyev OA, Butenko S, Pasiliao EL (2016) Exact MIP-based approaches for finding maximum quasi-cliques and dense subgraphs. Comput Optim Appl 64(1):177–214
Vitevitch MS (2008) What can graph theory tell us about word learning and lexical retrieval?. J Speech Lang Hear Res 51(2):408–422
Vitevitch MS, Goldstein R (2014) Keywords in the mental lexicon. J Mem Lang 73:131–147
Vitevitch MS, Goldstein R, Siew CS, Castro N (2014) Using complex networks to understand the mental lexicon. In: Yearbook of the Poznan Linguistic Meeting. vol. 1. De Gruyter Open. pp 119–138. https://doi.org/10.1515/yplm-2015-0007
Ward G (2002) Moby thesaurus II. Project Gutenberg Literary Archive Foundation. Available from: http://onlinebooks.library.upenn.edu/webbin/gutbook/lookup?num=3202
Wasserman S, Faust K (1994) Social Network Analysis. Cambridge University Press, New York
Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world'networks. Nature 393(6684):440
Zinoviev D (2018) Complex Network Analysis in Python: Recognize-Construct-Visualize-Analyze-Interpret. Pragmatic Bookshelf, Raleigh

Publisher's Note