Under consideration for publication in Knowledge and Information Systems

Domain-Agnostic Discovery of Similarities and Concepts at Scale[1]

Olof Görnerup[1], Daniel Gillblad[1] and Theodore Vasiloudis[1]

[1]Swedish Institute of Computer Science (SICS), Kista, Sweden

Abstract. Appropriately defining and efficiently calculating similarities from large data sets are often essential in data mining, both for gaining understanding of data and generating processes, and for building tractable representations. Given a set of objects and their correlations, we here rely on the premise that each object is characterized by its context, i.e. its correlations to the other objects. The similarity between two objects can then be expressed in terms of the similarity between their contexts. In this way, similarity pertains to the general notion that objects are similar if they are exchangeable in the data. We propose a scalable approach for calculating all relevant similarities among objects by relating them in a correlation graph that is transformed to a similarity graph. These graphs can express rich structural properties among objects. Specifically, we show that concepts – abstractions of objects – are constituted by groups of similar objects that can be discovered by clustering the objects in the similarity graph. These principles and methods are applicable in a wide range of fields, and will here be demonstrated in three domains: computational linguistics, music and molecular biology, where the numbers of objects and correlations range from small to very large.

Keywords: Similarity discovery; Concept mining; Distributional semantics; Graph processing

1. Introduction

As stated by Firth (1957) and further popularized in the computational linguistics community by Church and Hanks (1990), "You shall know a word by the company it keeps". Based on this principle, underpinned by Harris' distributional hypothesis (Harris, 1954), there have been substantial efforts to infer semantic and syntactic meaning from words through their effective usage in text (Harispe

[1] This paper is an extended version of (Görnerup et al., 2015).

Received Nov 12, 2015
Revised Jun 30, 2016
Accepted Jul 16, 2016

et al., 2015). Although the same principle has been applied in different and seemingly distinct domains, such as bibliometrics (Kessler, 1963) and bioinformatics (Ravasz et al., 2002), generalizing the notion of characterizing objects through their contexts into a broader fundamental principle for similarity discovery is so far largely unexplored.

Generalizing Harris' distributional hypothesis, we argue that the effective semantics of any object, with respect to observed data, are characterized by the context in which it occurs, or in other words, by how it is related (or correlated) to all other objects. The *similarity* between two objects may therefore be formulated in terms of their contexts, or how similar their relations to all other objects are. A benefit of this is that we can omit the specific functionality or underlying workings of objects, but only observe and consider their context patterns. This is highly attractive from a data-driven machine learning perspective since it requires very few assumptions about the objects.

With this as a starting point, we propose a graph-based method for discovering similarities from large data sets. An *object* is intentionally left vague since it can be many different things, such as music tracks in a playlist, people in a social network, tokens in a text or states in a stochastic process. We narrow down the scope slightly by only considering objects that exhibit pairwise relations, e.g. in terms of spatial, temporal or social correlations, which allows us to represent a collection of objects and their inter-dependencies as a graph. Our approach, which we call *contextual correlation mining* (CCM), involves two main steps: First, we create a *correlation graph* that describes the pairwise correlations between all objects. A correlation may here be any relationship measure such as the frequency of co-occurrence, a transition probability in a stochastic process, a correlation measure such as mutual information or a weighted edge in a graph. Second, we transform the correlation graph to a *similarity graph* by comparing the set of correlations of each object to the sets of correlations of all other objects – the more similar sets of correlations, the higher the weighted edge in the similarity graph.

The correlation graph is either given at the outset, as a Markov model or co-occurrence network for example, or built from data. Since there already exists a multitude of approaches for achieving this, see e.g. (Albert and Barabási, 2002), we will here focus on the second step, which we also view as the main technical contribution of this paper. Transforming a correlation graph to a similarity graph is conceptually straightforward, but as an "all-to-all" similarity problem, it is highly challenging in practice. However, since we are considering pairwise correlations, we can utilize that similar objects always occur in proximity in the correlation graph (at most one neighbour apart to be specific), which means that it is sufficient to compare objects locally in the graph. This not only drastically reduces the number of necessary comparisons, but also facilitates parallelization. Moreover, given that the correlation graph is sparse[2] – which is the case e.g. for gene co-expression (Jordan et al., 2004), semantic (Steyvers and Tenenbaum, 2005), word co-occurrence (Cancho and Solé, 2001) and social networks (Mislove et al., 2007), as well as for many other graphs of interest (Albert and Barabási, 2002) – we can also prune the correlation graph substantially prior to

[2] That is, most objects are either completely unrelated or at most negligibly correlated. Two randomly selected persons in a large social network, for instance, most likely do not know each other.

transforming it to a similarity graph while keeping the approximation error low and controllable.

In comparison, related methods are either limited to specific domains or do not scale well with growing number of objects, while the approach presented here is both scalable and agnostic with respect to objects and correlation measures. These are merely seen as vertices and edges in a graph, and CCM is therefore applicable in a broad range of domains as well as in mixed-data scenarios where several different correlation measures may be considered. In this way, we propose a powerful and efficient scheme that distills the essence in many related, and seemingly distinct, methods by using the core principle that objects can be characterized by the contexts in which they occur.

Furthermore, since CCM does not require any intermediate representations of objects and their correlations, such as sparse vectors or neural networks, it is also interpretable and transparent. This enables us to calculate well-understood notions of similarity and error among other things. Representing objects, correlations and similarities as graphs will also allow us to capture rich higher-scale structures among objects – e.g. without being constrained by geometric properties such as the triangle inequality – including ambiguity, hierarchies and ontologies, both in terms of correlations and similarities. Rather than representing data in terms of its raw constituents, a central task then is often to discover appropriate levels of abstraction of objects, both for gaining insights about data and by computational necessity. We will here demonstrate that CCM can be used for this purpose. Specifically, we will show that *concepts* – abstract generalizations of objects – are constituted by groups of inter-similar objects that play analogous roles in the data, and that we can discover these by clustering the objects in the similarity graph.

Similarities and concepts are both general notions, but discovering these from data in an unsupervised manner has several concrete applications. An immediate use of similarity discovery, for example, is in recommendation systems, where sensible recommendations of similar music, products, services etc. may be given based on contextual information. Concepts can also be used to overcome the curse of dimensionality in machine learning, where generalizations reduce the dimensionality of the state space that needs to be explored. This could be of value in classification tasks for instance, where annotated examples are expressed in terms of concepts rather than raw objects.

This paper is based on (Görnerup et al., 2015), with the following additions: an extended analysis of approximation error bounds; a description and demonstration of an improved vertex clustering algorithm for concept discovery; an extended evaluation, including a comparison with a state-of-the-art word embedding method and a gold standard for word similarity; and a parameter sensitivity analysis with regard to approximation errors and relevant graph measures.

1.1. Outline

The remainder of the paper is outlined as follows: Next we will put the paper in context by giving an overview of the related state-of-the-art. A background with preliminaries is presented in Sec. 3, followed by a description of proposed methods in Sec. 4, including theoretical investigations on error bounds and scalability. In Sec. 5 we demonstrate the versatility of the method by applying it in three distinct domains: computational linguistics, music and molecular biology.

We also evaluate acquired word similarities against a gold standard and compare the result with current the state-of-the-art in word vector embedding; demonstrate the applicability of the concept discovery method; perform a parameter sensitivity analysis; and experimentally evaluate scalability properties. The paper is concluded in Sec. 6 with a summary of our findings and a discussion on possible future directions.

2. Related work

The principle of relating objects with respect to contextual information is employed in several different areas, including ontology learning, computational linguistics, bioinformatics and bibliometrics. To our knowledge, the method that is closest in spirit to ours is SimRank (Jeh and Widom, 2002), which is a general approach for obtaining similarities between vertices in a graph. SimRank is an iterative method that uses the graph structure to derive similarities between objects by relating "objects that are related to similar objects" (Jeh and Widom, 2002). The main drawback with their approach, however, is that it is not scalable due to a cubic time complexity with the number of vertices in the graph. This has partly been remedied in improved versions of the algorithm, such as the one by Yu et al. (2012), but these are still too computationally demanding in order to be applicable on very large graphs. In comparison, we can comfortably run our algorithm on substantial graphs, doing only a single pass over the data. Leicht et al. (2006) propose a similarity calculation method which deals with another limitation of SimRank, namely that similarities are only calculated for nodes connected by paths of even length. The authors propose an alternative iterative method, but it suffers from much of the same scalability problems as with SimRank.

In molecular biology, Ravasz et al. (2002) propose an approach for finding similar vertices using so called topological overlap measures, which they apply on metabolic networks. Zhang and Horvath (2005) generalized this approach for use on weighted gene co-expression networks. As in our case, these methods relate vertices by assigning higher similarity scores between vertices that share many neighbors, but since their approaches are primarily tailored for bioinformatics tasks, they lack the generality of SimRank and the method presented here.

In computational linguistics, distributional analysis – where linguistic items are characterized by their relative distributional properties in the data – has become a fundamental approach (Harris, 1954). We use similar assumptions as a starting point, and when applied to text, the approach can be seen as transforming a graph over syntagmatic similarities to one describing paradigmatic similarities (Sahlgren, 2006), in which concepts are discovered through clustering. A large number of methods to find semantic similarities have been developed – see (Harispe et al., 2015) for a recent review – from the seminal work by Church and Hanks (1990), and Brown et al. (1992), to more recent approaches such as GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013). These methods, however, generate vector embeddings of words, and to calculate all similarities among these scales quadratically with the number or words at worst. Our method, in contrast, calculates *all* relevant similarities at scale, and is not limited to the natural language processing domain. Another important difference is that our method builds similarity graphs without using any dimensionality reduction or intermediate representations, such as high-dimensional vectors

or difficult-to-interpret neural networks. The advantage of using a direct graph representation is that it allows us to understand and reason about higher-scale structures among objects and concepts, such as hierarchical organization, in a straightforward manner using established graph and network methods. Although graph representations are used in natural language processing to relate similar words and documents (Mihalcea and Radev, 2011), these approaches have several limitations in comparison to our approach, e.g. by expecting existing similarity graphs as input, using ad hoc word relations (such as linking words separated by *and* or *or*), requiring part-of-speech tagged data, or by using human curated datasets, such as WordNet (Miller, 1995).

Another related area is ontology learning (Wong et al., 2012), which aims to infer taxonomies from corpora and other data sources. While one can draw parallels between our work and this field, the latter is often limited by exclusively considering a specific type of basic building blocks, such as nouns, where these are related in hierarchies with respect to specific relations, such as *is a* and *part of*. Similarly, context-based similarity discovery can also be viewed as a generalization of methods in bibliometrics, where citation patterns among a set of documents, such as scientific papers, are studied. Using so called bibliographic coupling to relate papers (Kessler, 1963) – i.e. the similarity between two papers is based on the number of citations they share – is a special case of our approach for relating two objects in the correlation graph. Another resemblance is that these and similar measures are used to cluster scientific papers (Small, 1973) as well as web pages (Larson, 1996). The method presented here could be employed in the very same way – where binary correlations are given by citations – to efficiently relate a large number of documents.

3. Background

3.1. Preliminaries

We begin by specifying the terminology used in this paper. Due to the transdisciplinary character of the method, we choose to use general rather than domain-specific terms.

Let $C = \{i\}_{i=1}^{n}$ be a set of *objects*, where each object has a correlation, $\rho_{i,j}$, to each other object. This relation can be expressed in terms of real values, booleans or something else that, for instance, represent a correlation measure, binary or weighted neighbourhood relation in a graph, co-occurrence probabilities in a corpus, or transition probabilities in a Markov chain. An object can for example be a word in text, and the correlations between words can be their co-occurrence probabilities. In another example, objects constitute people, and the correlation between two persons is their strength of acquaintance.

The *context* of an object i is considered to be its vector of relations to every other object, $\rho_i = (\rho_{i,j})_{j=1}^{n}$. In our word example, the context of a word is therefore its correlations to all other words. Analogously, in the people example, the context of a person is all that person's acquaintances.

Under the assumption that an object is characterized by its context, we can formulate the similarity between two objects i and j, denoted $\sigma_{i,j}$, in terms of a similarity measure between their respective contexts. Here we define $\sigma_{i,j}$ to be 1

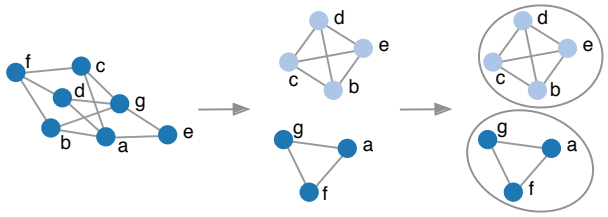

Fig. 1. A correlation graph is transformed to a similarity graph in which clustering is performed.

subtracted by the relative L_1-norm of the difference between ρ_i and ρ_j:

$$\sigma_{i,j} = 1 - \frac{|\rho_i - \rho_j|_1}{|\rho_i|_1 + |\rho_j|_1}, \tag{1}$$

where

$$|\rho_i|_1 = \sum_{k \in C} |\rho_{i,k}| \tag{2}$$

and

$$|\rho_i - \rho_j|_1 = \sum_{k \in C} |\rho_{i,k} - \rho_{j,k}|, \tag{3}$$

denoted $L_1(i,j)$ for short. That is, we normalize the absolute L_1-norm of the difference between i and j:s context vectors with the maximum possible norm of the difference, as given by $|\rho_i|_1 + |\rho_j|_1$, and then subtract the result from one in order to transform it to a similarity measure bounded by 0 and 1, $\sigma_{i,j} \in [0,1]$.

Since objects are discrete and have pairwise relations, we can represent C and $\rho_{i,j}$ as a directed graph (it it directed since the correlations are not necessarily symmetric), $\mathcal{R} = (C, R)$, where vertices constitute objects, and where edges $r_{i,j} \in R$ have weights $\rho_{i,j}$. We term this the *correlation graph* of C with respect to $\rho_{i,j}$. In principle this is a complete graph since every vertex has a relation to every other vertex (including itself) through $\rho_{i,j}$. However, we define the graph such that there is only an edge between two vertices i and j if their corresponding objects have a degree of similarity, i.e. when $|\rho_i - \rho_j|_1 < |\rho_i|_1 + |\rho_j|_1$ and $i \neq j$. In our people example, the correlation network is simply a acquaintance network.

Analogously, the *similarity graph* of C with regard to $\rho_{i,j}$, denoted $\mathcal{S} = (C, S)$, is defined to be an undirected graph where weights of edges $s_{i,j} \in S$ instead are given by $\sigma_{i,j}$.

By *concept* we mean a group of objects that are approximately similar – forming a cluster in the similarity graph – and therefore approximately interchangeable in their respective contexts. In the word example this may correspond to a group of semantically and/or syntactically similar words (e.g. termed *semantic community* or *topic* in the natural language processing community), whereas in the people example, a concept is a group of people that have similar circles of acquaintances, such as a group of colleagues.

3.2. Example

As a simple stylized example, consider the set of objects $C = \{a, b, c, d, e, f, g\}$ with the symmetric, binary correlation graph shown to the left in Fig. 1. Transforming this correlation graph to the similarity graph shown in the same figure using Eq. 1, the pairwise similarities become positive when two objects have overlapping contexts. Each of the two clusters in the similarity graph is then identified as a concept.

Note that in the case of the binary relationship graph, the L_1-norm between two objects, i and j, is given by the number of neighbours that they do not share:

$$|\rho_i - \rho_j|_1 = |n_i \cup n_j| - |n_i \cap n_j| = |n_i| + |n_j| - 2|n_i \cap n_j|, \tag{4}$$

where n_i and n_j are the neighbourhoods of i and j. Since the maximum possible norm of the difference is $|n_i| + |n_j|$, the similarity between i and j becomes

$$\sigma_{i,j} = 1 - \frac{|n_i| + |n_j| - 2|n_i \cap n_j|}{|n_i| + |n_j|} = \frac{2|n_i \cap n_j|}{|n_i| + |n_j|}, \tag{5}$$

which is known as the Sørensen-Dice coefficient (Dice, 1945; Sørensen, 1948), that, in turn, is analogous to the commonly used Jaccard coefficient (Jaccard, 1912) through a monotonic transformation.

4. Methods

4.1. Similarity calculations

In order to efficiently and scalably transform a correlation graph into a similarity graph, we utilize two observations concerning the correlation graph with regard to locality and sparseness. Firstly, according to our definition of similarity, an object only has a degree of similarity to its second-order neighbours (its neighbours' neighbours) in the correlation graph $\mathcal{R}$. Let n_i and n_j be the neighbouring vertices of i and j respectively, and $\rho_{i,k} = 0$ if $k \notin n_i$. Then

$$\begin{aligned} L_1(i,j) &= \sum_{k \in n_i} |\rho_{i,k}| - \sum_{k \in n_i \cap n_j} |\rho_{i,k}| + \sum_{k \in n_j} |\rho_{j,k}| - \sum_{k \in n_i \cap n_j} |\rho_{j,k}| + \sum_{k \in n_i \cap n_j} |\rho_{i,k} - \rho_{j,k}| \\ &= |\rho_i|_1 + |\rho_j|_1 + \Lambda_{i,j}, \end{aligned} \tag{6}$$

where

$$\Lambda_{i,j} = \sum_{k \in n_i \cap n_j} (|\rho_{i,k} - \rho_{j,k}| - |\rho_{i,k}| - |\rho_{j,k}|) \tag{7}$$

When calculating Eq. 1 it is therefore sufficient to compare differences between weights $\rho_{i,k}$ and $\rho_{j,k}$ of edges from i and j to neighbours k that i and j have in common, given that we have the weight sums of outgoing edges of i and j. In practice, we generate a similarity graph by first summing weights of outgoing edges per vertex, and then building an intermediate undirected two-hop multigraph of $\mathcal{S}$, where an edge (i, j) that corresponds to a hop through k in $\mathcal{S}$ has weight $|\rho_{i,k} - \rho_{j,k}| - |\rho_{i,k}| - |\rho_{j,k}|$. The L_1-norm between i and j is then calculated by summing the weights of all edges between i and j in the multigraph according to Eq. 7, and adding this to the edge weight sums of i and j.

Just as our approach is applicable for different correlation measures, it is not strictly limited to the L_1-norm. Using other distance measures is also possible, given that these can be decomposed in a way akin to Eq. 6.

4.1.1. Approximations

Even though we only need to consider shared neighbours when calculating the similarities between objects, these calculations still scale unfavorably as the sum of the square of in-degrees per vertex, since we consider all pairs of incoming edges of vertex k when generating two-hop edges. We therefore need to approximate the similarity measure by reducing in-degrees. To be able to determine whether a certain object distance with respect to a distance measure D is relevant or not, typically we would like to ensure that the error $E_D(i,j)$ in any specific distance approximation is less than a fixed level θ_D,

$$E_D(i,j) \leq \theta_D \tag{8}$$

and more specifically for the L_1-norm approximated by $\tilde{L}_1$,

$$E_1(i,j) = |L_1(i,j) - \tilde{L}_1(i,j)| \leq \theta_1. \tag{9}$$

If we would like to remove terms by approximating by zero while keeping the total approximation error $E_1(i,j)$ as small as possible, we should remove the smallest absolute correlation terms $|\rho_{i,k}|$ in Eq. 6. Put differently, we discard the edges with the smallest weights in the correlation graph. Let τ_i be a threshold value below which absolute correlations of object i are approximated by zero, and $|\check{\rho}_i|_1$ the norm of discarded correlations:

$$|\check{\rho}_i|_1 = \sum_{|\rho_{i,k}|<\tau_i} |\rho_{i,k}|. \tag{10}$$

The upper bound of the error is then given by

$$E_1(i,j) \leq |\check{\rho}_i|_1 + |\check{\rho}_j|_1, \tag{11}$$

where $E_1(i,j) = |\check{\rho}_i|_1 + |\check{\rho}_j|_1$ when the edges of discarded relations of i and j do not share any destination vertex k. When calculating the object similarity based on the L_1-norm, we can therefore reduce the number of terms we need to compare by removing low correlation values with predictable errors. Lowering the number of terms in Eq. 7 while guaranteeing an error $E_1(i,j) \leq \theta_1$ is then a matter of sorting absolute correlations $|\rho_{i,k}|$ and, starting with the smallest one, removing correlations until the cumulative sum reaches $\theta_1/2$, which is one of the terms of the bound in Eq. 11.

This brings us to our second observation, which is that in most correlation graphs of interest, a substantial fraction of the correlations from one object to others are, if not zero, very small or even magnitudes smaller than its largest relations, as exemplified in Fig. 2. We can therefore effectively prune a large fraction of the links while keeping the cumulative discarded weight (and error) comparatively low.

Moreover, if reducing terms in Eq. 6 has priority over accuracy, we may start at the other end by specifying a maximum in-degree per vertex, and keep the corresponding number of incoming edges with the largest weights. Doing so we utilize that the main bulk of vertices have low in-degrees and are therefore not affected by the pruning. This situation is illustrated in Fig. 3. By calculating and

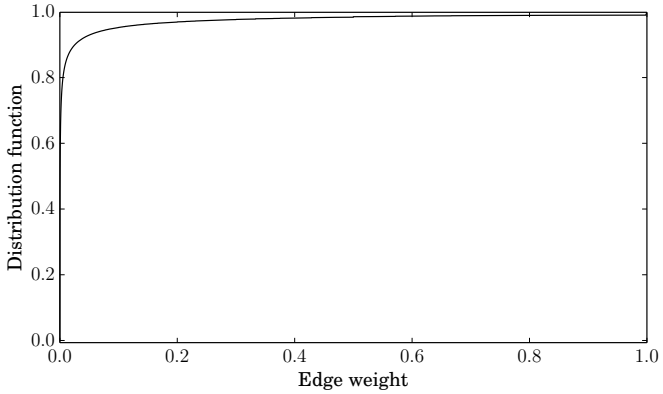

Fig. 2. The cumulative distribution function of edge weights in the Billion word correlation graph described in Sec. 5.1.1 shows that a large fraction of edges with low weights can be pruned. For example, approximately 90% of the edges are discarded when considering edges with weights ≥ 0.01.

storing the sums of discarded weights of outgoing edges per vertex, we can then readily calculate the error bound per object pair according to Eq. 11.

Alternatively, since $\psi_{i,j} = |\rho_i|_1 + |\rho_j|_1$ is known, we can approximate the L_1-norm by solely approximating $\Lambda_{i,j}$ in Eq. 6, with an analogous term $\tilde{\Lambda}_{i,j}$ for the edges in the pruned graph:

$$\tilde{L}'_1(i,j) = \psi_{i,j} + \tilde{\Lambda}_{i,j}. \tag{12}$$

Note that each term in Eq. 7 is at most 0 due to the triangle inequality, and that $\Lambda_{i,j} \leq \tilde{\Lambda}_{i,j} \leq 0$ since the terms in $\tilde{\Lambda}_{i,j}$ constitute a subset of the terms in $\Lambda_{i,j}$. The approximation error for $\tilde{L}'_1(i,j)$ then becomes

$$E'_1(i,j) = |L_1(i,j) - \tilde{L}'_1(i,j)| = |\Lambda_{i,j} - \tilde{\Lambda}_{i,j}| = \tilde{\Lambda}_{i,j} - \Lambda_{i,j} \tag{13}$$

and hence

$$0 \leq E'_1(i,j) \leq -\Lambda_{i,j}. \tag{14}$$

For the error, $\epsilon_{i,j}$, of the approximate relative L_1-norm and of the approximate similarity $\tilde{\sigma}_{i,j}$, this translates to

$$\epsilon_{i,j} \leq \frac{-\Lambda_{i,j}}{\psi_{i,j}} = 1 - \frac{\psi_{i,j} + \Lambda_{i,j}}{\psi_{i,j}} = \sigma_{i,j}, \tag{15}$$

and so the error is bound by

$$0 \leq \epsilon_{i,j} \leq \sigma_{i,j}. \tag{16}$$

4.2. Discovering concepts

After transforming a correlation graph to a similarity graph, we can use the latter to find interesting structural properties among objects in terms of their similarity relations. Since two objects are similar if they occur in similar contexts, we can

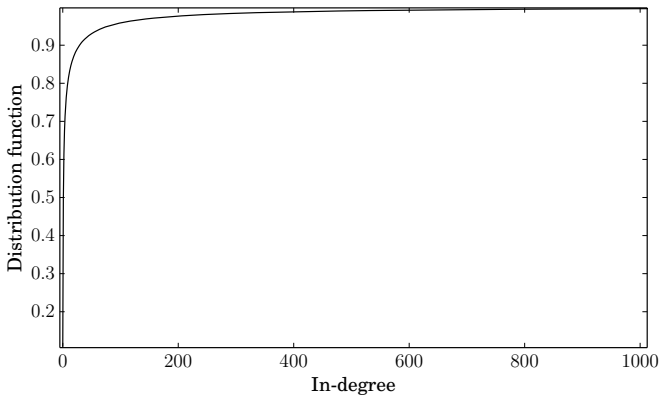

Fig. 3. The cumulative distribution function of in-degrees for the graph referred to in Fig. 2 illustrates that it is possible to apply an in-degree threshold while affecting comparably few vertices. Only a small percentage of the vertices are affected, for instance, when capping the in-degree at 500 edges.

interpret the notion of *similarity* as something that the objects together exhibit if they are approximately exchangeable in their respective contexts, being able to take each other's role. This notion of similarity requires very little, if any, assumptions about the properties of the objects *per se*, since it is completely based on the relations *between* objects.

Given a set of objects and their similarities, we can view a pair of objects i and j that have similarity $\sigma_{i,j} = 1$ as equivalent. This is an equivalence relation in the formal sense since it satisfies reflexivity, symmetry and transitivity. We can therefore partition the set of objects into equivalence classes and interpret each class, in which all objects are interchangeable, as a concept. Due to noise and slight variations, however, full similarity is seldom fulfilled in practical applications. We therefore allow objects that are approximately similar, i.e. $\sigma_{i,j} \geq 1 - \epsilon$ for some small constant ϵ, to belong to the same class. Due to this approximation, transitivity no longer holds, since although i is approximately similar to j and j is approximately similar to k, i is not necessarily approximately similar to k. From this follows that classes can overlap, which reflects that objects indeed may take several different roles (consider for instance proteins with multiple functions, or polysemous words (Palla et al., 2005)). Each concept is then constituted by a group of objects where each object has at least similarity $1 - \epsilon$ to each other object.

From a graph perspective, these groups correspond to cliques (i.e. complete subgraphs) in a similarity graph where $\sigma_{i,j} \geq 1 - \epsilon$ for all edges. Since cliques can contain other cliques – in fact, there is a combinatorial explosion of such sub-cliques – we require that a concept is a clique that is not a subset of another clique, i.e. it is a *maximal clique*. However, finding maximal cliques in graphs is a highly challenging problem, both in theory and practice. We therefore approximate maximal cliques with *communities*, i.e. clusters of vertices in the similarity graph. An approach for finding such communities at scale is described in Sec. 4.3.2.

```
1: ins   = edges.map(((i,j),rij) => (j,(i,rij)))
2: pairs = ins.join(ins).filter((k,((i,rik),(j,rjk))) => i<j)
3: terms = pairs.map((k,((i,rik),(j,rjk))) =>
4:                   ((i,j),abs(rik-rjk)-abs(rik)-abs(rjk)))
5:              .reducebykey((v,w) => v+w)
```

Fig. 4. Pseudo-code of the sum term calculation in Eq. 6. 1) Edge tuples with vertex indices `i` and `j`, and weights `rij` are mapped to key-value pairs keyed by destination vertices. 2) A two-hop graph is generated through self-join, and unique in-edge pairs are extracted through filtering. 3-4) All terms in the sum in Eq. 6 are calculated and 5) summed per two-hop neighbour pair.

4.3. Implementation

4.3.1. Similarities

The calculations of the approximations of the difference norms $|\rho_i - \rho_j|_1$, as formulated in Eq. 6, lend themselves well to functional programming, since they can be implemented as a small number of standard transformations applied on a collection of correlation graph edges. The procedure can be summarized in the following steps:

1. For each vertex i, calculate the norms $|\rho_i|_1$, i.e. the weight sum prior to pruning.
2. Prune the correlation graph by filtering out edges with weights below a given threshold value, τ_i, and/or by keeping a given number of incoming edges with the largest weights per vertex.
3. Calculate $\tilde{\Lambda}_{i,j}$ for each pair of vertices that share at least one neighbour in the pruned correlation graph. This step is described in pseudo-code in Fig. 4 and involves a self-join operation for building a two-hop multigraph that links second-order neighbours, followed by a map transformation for calculating the terms in the sum, which subsequently are summed up per vertex pair by a reduce operation.
4. For each vertex pair in the previous step, calculate the approximate relative L_1-norm, $\tilde{l}_{i,j} = (\tilde{\Lambda}_{i,j} + \psi_{i,j})/\psi_{i,j}$, and the approximate similarity $\tilde{\sigma}_{i,j} = 1 - \tilde{l}_{i,j}$.

The method is implemented in the Scala programming language and uses the in-memory data processing framework Apache Spark (Zaharia et al., 2012), which enables us to employ the method at scale in terms of computing hardware. To facilitate reproducibility, the implementation is available with an open source license in an online repository.[3] Since we are exclusively using standard core primitives in Spark (`map`, `filter`, `join` etc.), implementing the method in other similar frameworks, such as Apache Flink (Alexandrov et al., 2014), is also possible.

4.3.2. Concepts

In order to find object clusters in the similarity graph, we employ a *community detection* method (*community* and *cluster* are used interchangeably from here on). There is a wealth of techniques to choose from and we refer the reader to

[3] https://github.com/sics-dna/concepts

Fortunato (2010) for a thorough review of the area. For our purposes we need an algorithm that allows for overlapping communities and has good scalability characteristics. The ability to detect overlapping communities is important for concept discovery, as objects may exhibit multiple roles within a graph. Another preference is that the algorithm does not require that the number of clusters is predefined, but this number should rather be discovered from data.

Based on these criteria, we employ an algorithm that is akin to the Speaker-listener label propagation algorithm (SLPA) by Xie et al. (2011). In their algorithm, each vertex is assigned a *memory*, constituted by cluster label-frequency pairs, that is initialized with a unique cluster label. Vertices are then updated sequentially and asynchronously: For a vertex i, each of i:s neighbours randomly (proportionally to label frequencies) sample a cluster label from their respective memories. These labels are then sent to i, which adds the most common label to its memory. The procedure is repeated for a given number of iterations, after which vertices are assigned to clusters with frequencies above a specific threshold.

In our implementation, all vertices are updated synchronously and in parallel, and instead of using memories with frequency information, we associate each vertex with a *queue* (i.e., a "first in-first out" data type) of community labels. More specifically, for each vertex i,

1. Initialize a queue $q_i = [i]$.
2. Sample a label uniformly from q_i and send it to all neighbours.
3. Of the received labels, add the one occurring most times to q_i. If q_i has reached a maximum capacity, γ, discard the oldest label in q_i prior to adding the new one.
4. Repeat from step 2 m times.
5. Associate i with clusters with labels that occur in q_i with a frequency above a given threshold.

Since a vertex can be associated with several communities, the communities can overlap, including the case where a community is a subset of another community. It is also possible that several equivalent clusters are found, in which case the redundant ones are removed in a post-processing step.

The reason for storing cluster labels in queues, besides from ease of implementation, is that it has the effect that transient cluster assignments at early iterations are quickly discarded.

Note that the current implementation does not take edge weights into consideration. Instead, edges with weights below a threshold, σ_m, are discarded prior to applying the above steps.

4.4. Scalability characteristics

In order to enable practical use on large tasks in terms of the number of objects, correlations and example data, a key design goal is scalability. Since we are using relational primitives to represent graphs, the scalability of the algorithm can be studied using established results from relational algebra (Chandra and Merlin, 1977; Bitton et al., 1983).

The most computationally demanding component of the algorithm is building the two-hop graph through a self-join operation (the third step in Sec. 4.3.1). Since a self-join is a conjunctive query (Chandra and Merlin, 1977) in relational

algebra terms, we can reason about its computational cost. Specifically for a distributed environment, Koutris and Suciu (2011) define a parallel algorithm as a sequence of parallel computation steps, and define its cost as the number of steps required to complete the algorithm. The authors prove that a join operation can be completed in one parallel computational step using the hash-join algorithm, by using a communication and a computation phase. Just as importantly, they prove that the hash-join operation is load balanced and as such it ensures linear speedup (doubling the server count reduces the load by half) and constant scale-up (when doubling both the size of the data and number of servers, the running time remains the same). Specifically for the Apache Spark platform, on which we implement the algorithm, the self-join operation creates what Zaharia et al. (2012) call a *narrow dependency*. This property allows for pipelined executions of all operations on one node up until the reduction step in Fig. 4, without the need for expensive data shuffles through the network.

5. Experiments

5.1. Examples

In order to demonstrate the broad applicability of our approach, we will showcase it in three distinct domains: computational linguistics, music and molecular biology. Here we prioritize breadth over depth, and more in-depth evaluations of the method's performance with respect to specific applications will be topics in future work.

5.1.1. Words

We begin by relating words in terms of their co-occurrence in text, where two words, i and j, co-occur if they both appear within a window of n words. In the simplest case, for $n = 2$, words co-occur if they are adjacent. There exist many different word association measures, see (Pecina, 2008) for a large number of examples, such as pointwise mutual information (Church and Hanks, 1990) and normalized versions thereof (Bouma, 2009). Here we simply measure the association between i and j as the relative frequency of j occurring in i:s vicinity, or, in other words, as the conditional probability that a randomly selected word in a window that contains i, will be the word j. That is, $\rho_{i,j} \approx c_{i,j}/c_i$, where c_i and $c_{i,j}$ are the number of occurrences of i, and i together with j, respectively. Note that this measure is not symmetric and so $\rho_{i,j} \neq \rho_{j,i}$ may be true.

In this example we use the One billion word corpus (Chelba et al., 2013), which consists of nearly one billion tokens and originates from crawled online news texts. We count the number of occurrences of bigrams (pairs of adjacent words) with words consisting only of alphabetic characters. This results in approximately 8 million unique bigrams and a vocabulary with roughly 0.3 million words. From the bigram counts we relate words by their ordered adjacency.

Despite the comparably modest size of this corpus and the narrow context window, the method manages to discover groups of words that reflect both syntactic and semantic concepts. Examples of such concepts are shown in Fig. 5, where we see that the groups correspond e.g. to specific nouns, (*tablet*, *laptop*, *notebook* etc.), adjectives (*chic*, *trendy*, *fashionable* etc.), or adverbs (*strongly*, *intensely*, *vigorously*, etc.). Note that antonyms, in addition to synonyms, may

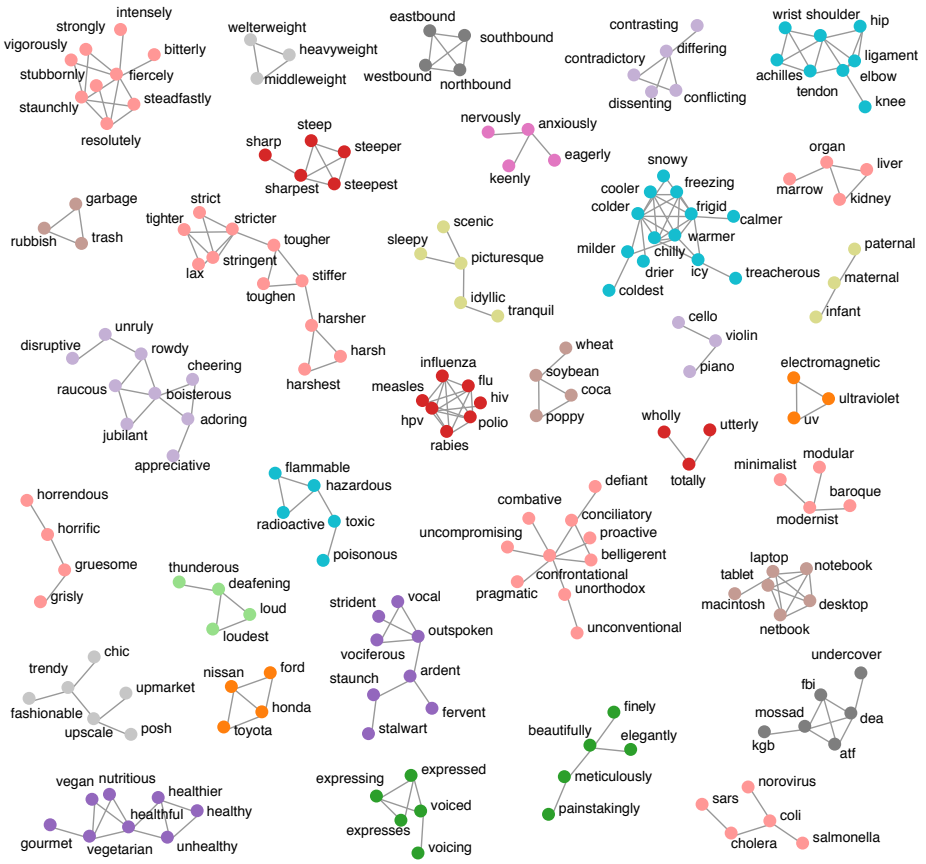

Fig. 5. Examples of groups of words in a word similarity graph based on the Billion word corpus. For sake of clarity, edges with weights $\sigma_{i,j} \geq 0.15$ are shown.

occur in the same group (e.g. *warmer* and *colder*). This highlights that the notion of similarity (here corresponding to what is termed *relatedness* in the NLP field) is very much dependent on the choice of correlation measure. The correlation measure may therefore be both application and domain-specific, whereas the definition of similarity, *given* the correlation measure, is domain-agnostic. Accordingly, antonyms are indeed similar by definition with respect to the correlation measure used in this example. However, for other correlation measures, possibly supporting negative correlations, antonyms may occur in separate concepts.

5.1.2. Artists

In the next proof-of-concept we relate artists by using a dataset that represents the listening habits of users of the *Last.fm* music service.[4] This dataset, provided by Celma (2010), consists of approximately 19 million track plays of 992 users.

[4] http://www.last.fm/

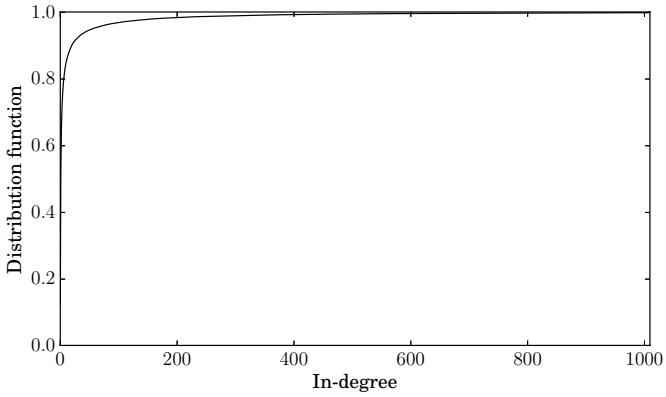

Fig. 6. The cumulative distribution function of in-degrees for the artist correlation graph.

For each user, we extract sequences of played artists – there are roughly 177000 in total – and consider the context of an artist to be defined by the probability distribution of subsequently played artists. Hence, we assume artists are related in a Markov chain, where each artist constitutes a state, and where there is a directed edge from artist i to artist j weighted with the probability that j is played next, given that i is currently playing. This probability is simply estimated as $\rho_{i,j} \approx c_{i,j}/c_i$, where c_i and $c_{i,j}$ are the number of times i, and i followed by j occur in the data set, respectively.

The in-degree distribution of the artist correlation graph resembles those of the word correlation graphs, see Fig. 6, which again means that relatively few vertices are affected by in-degree pruning. Transforming the artist correlation graph to a similarity graph also results in tightly grouped artists that can be clustered, where the resulting clusters appear to represent musical genres as exemplified in Fig. 7. As such, the similarity graph could be used in a music recommendation system to relate similar artists through the listening habits of users, similar to a collaborative filtering system. We could then also provide an intuitive way to incorporate the popularity of artists via their play frequencies in order to mitigate the effect of popularity bias in recommendations (Celma and Cano, 2008).

5.1.3. Codons

Finally, we apply the method in molecular biology, where we consider codons as objects. Codons are triplets of adjacent nucleotides in DNA that translate to amino acid residues that in turn form proteins. These are related through codon substitution dynamics, which is central both for understanding molecular evolution and in applications such as DNA sequence alignment (Anisimova and Kosiol, 2009). Since there are only 64 codons in total, this example differs from the previous two in that we consider relatively few objects.

Codon substitutions are often modeled as Markov processes (Anisimova and Kosiol, 2009), where the substitution probabilities of a codon at a specific location are assumed to be independent of neighbouring codons as well as previous codons at the same location. In this example we use an empirically derived codon

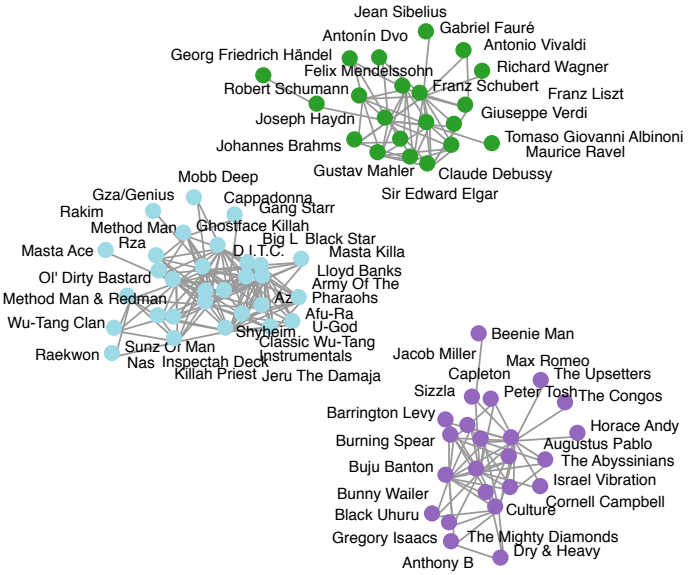

Fig. 7. Examples of components in an artist similarity graph correspond to three distinct music genres. Edges with weights $\sigma_{i,j} \geq 0.5$ are shown.

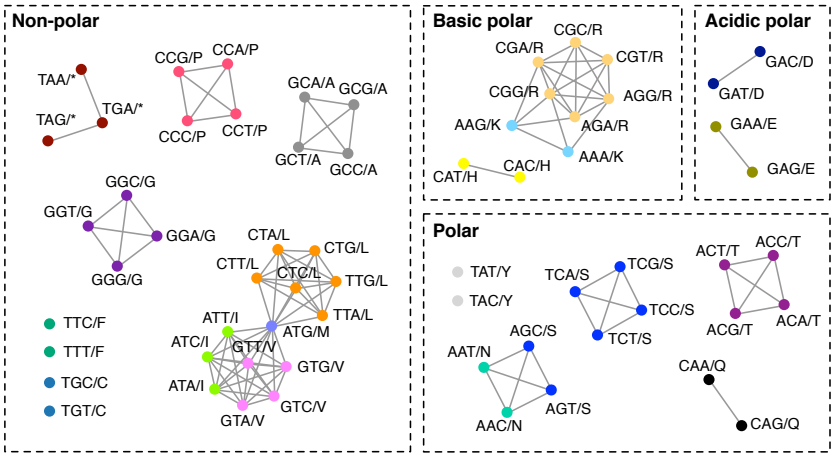

Fig. 8. Codon similarity graph where vertices are labeled with c/a for codon c coding to amino acid a. Edges with weights $\sigma_{i,j} \geq 0.45$ are shown. Vertices are grouped by chemical properties. Note that when the edge weight threshold is lowered, clusters containing several amino acids are split by amino acid. The rare and low mutable amino acid tryptophan is omitted.

substitution matrix provided by Schneider et al. (2005), where we consider the context of a codon i to be given by the relative substitution frequencies $(\rho_{i,j})_{j=1}^{n}$ to other codons j.

As seen in the resulting codon similarity graph in Fig. 8, codons that translate to the same amino acid according to the standard genetic code (Nirenberg et al., 1965) tend to be grouped. This reflects that codons that are highly similar

are commutable – quite literary – since substitutions between these codons are neutral under evolution. These clusters are also present in the correlation graph and therefore preserved through the similarity graph transformation.

We now shift perspective and view *amino acid* as a concept. Again looking at Fig. 8, we see that some of the amino acids are grouped. This can be explained by a higher degree of neutrality within groups than between them, which has been observed in empirical amino acid substitution matrices, such as the accepted point mutation (PAM) matrix by Dayhoff et al. (Dayhoff and Schwartz, 1978). In comparison, Wu and Brutlag derived amino acid substitution groups by group-wise (as opposed to pairwise) statistical analysis of protein databases (Wu and Brutlag, 1996). The groups shown in Fig. 8 ({I, L, M, V}, {K, R} and {N, S}) all agree with their findings. In summary, the codon similarity graph captures both concepts and higher-order concepts: from codons to amino acids, via the genetic code, to collections of amino acids that constitute known substitution groups.

5.2. Evaluation

Due to the general nature of our approach, where several different correlation measures can be used (e.g. co-occurrence probability, co-occurrence existence, pointwise mutual information, normalized pointwise mutual information, Jaccard coefficient, Sørensen-Dice coefficient), an exhaustive evaluation of the method across domains is beyond the scope of this paper. Instead we mainly evaluate our method in the computational linguistics domain, where there is a comparably large body of related work, and leave other domain and application specific evaluations for future work. To broaden the evaluation, we consider different correlation measures and corpora.

5.2.1. Similarity discovery

An established approach to quantitatively evaluate the performance of word similarity methods is to use benchmarks with word pairs that have been manually graded with respect to degree of similarity. Since these benchmarks also contain unassociated words, it is not possible to do a direct comparison between our method and other approaches in terms of benchmark performance, since our method exclusively relates words that have a certain degree of similarity (indeed, this is one of the reasons it is scalable). Instead we compare similarities $\sigma_{i,j}$ with corresponding benchmark similarities for word pairs (i, j) that *do* exist in the similarity graph. For this purpose we use the standard WordSim-353 (WS-353) test collection (Finkelstein et al., 2001), which consists of 353 word pairs that have been graded by human annotators.

In the first experiment we use the English Google Books n-gram dataset (Michel et al., 2010; Lin et al., 2012), which consists of n-gram (contiguous sequences of n tokens) counts derived from a 361 billion token-corpus. We build a correlation graph from co-occurrence windows of size 5 using conditional probabilities as described in 5.1.1, filter out words that occur with a frequency less than 10^{-8} and edges $\rho_{i,j} < 10^{-3}$, and set the maximum in-degree to 200. In the similarity graph, which is built in less than 10 minutes (cf. Fig. 17), 60% of the WS-353 word pairs are present, resulting in a Spearman rank correlation of 0.76. The current state of the art (with respect to the whole dataset) is 0.81 (Halawi et al., 2012; Yih and Qazvinian, 2012). These figures represent the cor-

relation with respect to the average annotator score, and as a comparison the mean performance of individual annotators, with respect to the mean score of the remaining annotators, is in fact also 0.76 (Hill et al., 2014). The preciseness of this agreement, however, is most likely coincidental, although it gives a strong indication of the validity of our approach.

In the next experiment, we compare three different types of similarities: firstly, $\sigma_{i,j}$ using our method, secondly, WS-353 similarities, and thirdly, cosine similarities between vectors generated by GloVe (Pennington et al., 2014), the current state-of-the-art word embedding method. The cosine similarities are calculated using pre-trained word vectors made available by the authors.[5] Four embeddings are provided, with dimensions 50, 100, 200 and 300, that each relates words in windows of size 10. The GloVe vectors are learnt from a concatenation of a Wikipedia dump from 2014 and the Gigaword 5 corpus (Graff, 2003),[6] which together contain approximately 6 billion tokens.

The Gigaword 5 corpus is not freely available, so to enable reproducibility we apply CCM solely on a Wikipedia dump. This dump is more recent (from March 2015) and therefore larger than the Wikipedia corpus used to train the GloVe vectors. The difference in size between the corpora used by GloVe and CCM is therefore smaller than if we would have used the 2014 version of Wikipedia.

We discard all tokens that contain non-alphanumeric characters, which leaves us with approximately 3.6 billion tokens. These are then used to calculate correlations between adjacent words using bigrams that occur at least 20 times in the corpus. Correlations $\rho_{i,j}$ are then given by the pointwise mutual information between i and j (Church and Hanks, 1990):

$$\rho_{i,j} = \log_2 \frac{p_{i,j}}{p_i \; p_j}, \tag{17}$$

where p_i and p_j are the probabilities that i and j are observed in the corpus, and $p_{i,j}$ is the probability that they are observed together (i.e. being adjacent in this case). These probabilities are estimated by $p_i \approx c_i/c_t$, $p_j \approx c_j/c_t$ and $p_{i,j} \approx c_{i,j}/c_t$, where c_i, c_j and $c_{i,j}$ are occurrence counts, and where c_t is the total number of tokens in the corpus. Since $p_i \; p_j$ is the probability that i and j co-occur if they were independent, $\rho_{i,j} = 0$ means that the objects are completely unrelated. If i and j co-occur more frequently than expected from chance, then $\rho_{i,j} > 0$. Similarly, $\rho_{i,j} < 0$ if they are observed together to a lesser extent than expected. Note further that the measure is symmetric, i.e. $\rho_{i,j} = \rho_{j,i}$.

In comparison to associating objects with conditional probabilities, pointwise mutual information has the advantage of being less dominated by very frequent object occurrences (consider for instance the correlation between a relatively infrequent word and a word such as *the*).

After building the correlation graph, we apply an in-degree threshold of 100, and calculate similarities between word pairs that are shared with WS-353. The cosine similarities between the same word pairs are then calculated for the four different word embeddings. To aid intuition and enable a qualitative comparison between acquired similarities, we show scatter plots in Fig. 9 (for 300D vectors in the case of GloVe).

All three types of similarities are inter-correlated. Specifically, the Spearman

[5] http://nlp.stanford.edu/projects/glove/
[6] https://catalog.ldc.upenn.edu/LDC2011T07

Table 1. Uniformly sampled examples of concepts.

achievements, examples	minority, majority
burning, burn	founding, associate
breakfast, lunch	thinking, thought
removing, remove	foreign, overseas
easy, easier	chrysler, inc
colour, color	found, discovered, bodies
loved, liked, enjoy, loves	heard, hear, hearing
liquid, toxic	fiscal, banking, financial, economic, gambling
tuition, tax	solar, wind
britain, nation	impose, enforce, violating, violated, imposed, imposing

rank correlation coefficient between $\sigma_{i,j}$ and WS-353 is 0.65, and the correlations with the cosine similarity depend on the number of dimensions used. For 50D, 100D and 200D, the cosine similarity is more strongly correlated with $\sigma_{i,j}$ (0.71, 0.68 and 0.68, respectively) than with WS-353 (0.57, 0.63 and 0.67, respectively), whereas for 300D, the cosine similarity is more strongly correlated with WS-353 (0.73) than with $\sigma_{i,j}$ (0.69). The latter case is the one shown in Fig. 9.

In summary, despite CCM using a smaller corpus (3.6B versus 6B tokens) and smaller window sizes (2 versus 10), CCM and GloVe generate surprisingly comparable similarities. In particular, we expected that the difference in window sizes would have a larger impact, since substantially more correlations are present beyond the narrow adjacency window CCM employs in this case.

5.2.2. Concept discovery

We will now demonstrate the concept discovery approach by applying the clustering algorithm described in Sec. 4.3.2 on a similarity graph transformed from a word correlation graph. The latter is built from the Billion word corpus using bigram counts, where correlations are given by pointwise mutual information.

The correlation graph consists of vertices with $p_i \geq 10^{-5}$ and edges with $\rho_{i,j} \geq 4$ bits, and has a maximum in-degree of 1000 edges. When clustering the corresponding similarity graph, $\sigma_m = 0.25$, queues have capacity 4, and the iteration described in 4.3.2 is performed 16 times. The resulting cluster assignments are then given by labels that occupy at least 50% per queue. Out of these, the most dominant cluster assignments per vertex are depicted in Fig. 10.

As examples, a random set of clusters is shown in Table 1, and a set of clusters of our choosing in Table 2. Both sets – with some exceptions (e.g. {britain, nation}, perhaps due to that both words are strongly correlated with *great*) – demonstrate that the method is capable of discovering concepts that we perceive as meaningful in that they capture abstract syntactic and semantic notions in the corpus, such as *vehicle*, *US state* (which, incidentally, also demonstrates our bare-bones parsing of the corpus, since *carolina* and *hampshire* – lacking *north*, *south* and *new* – belong to this concept), *color*, *nationality*, *day* and so on.

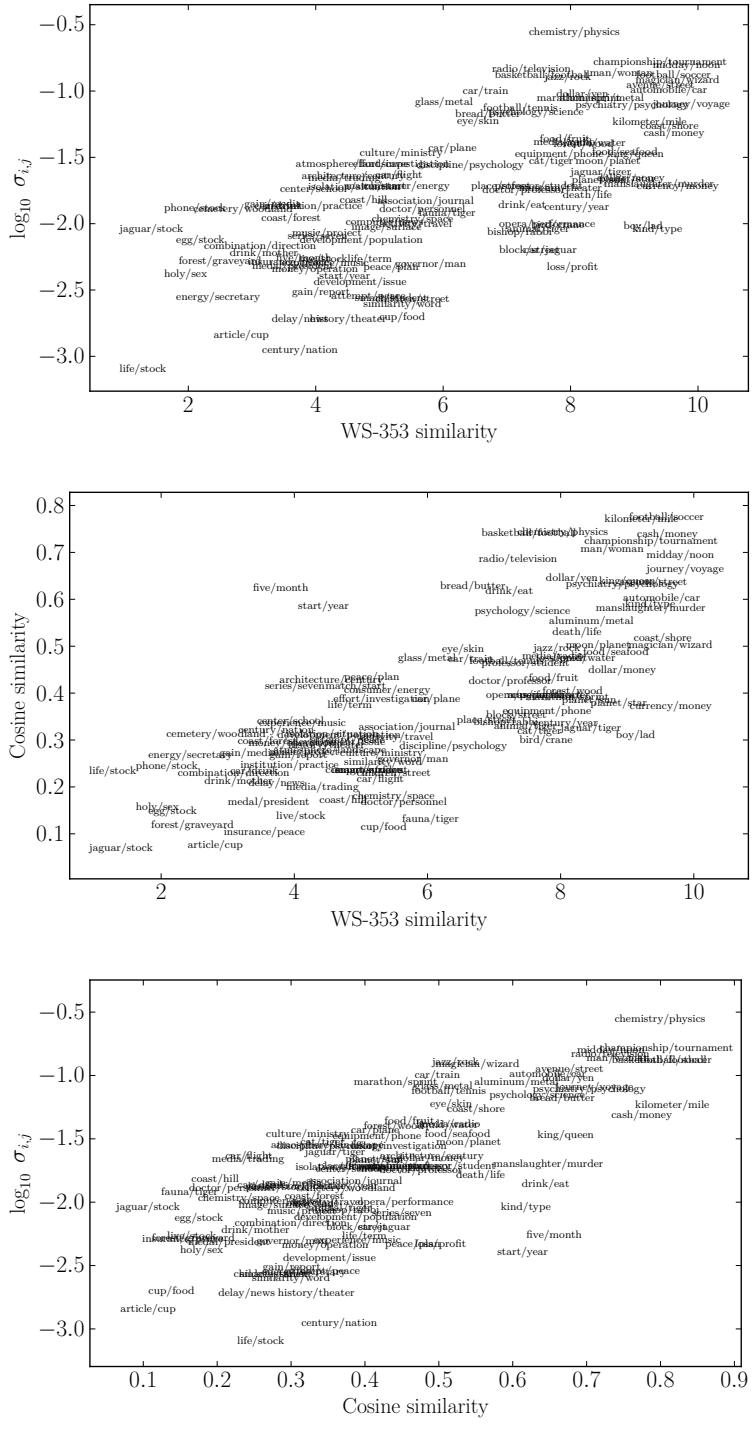

Fig. 9. Scatter plots that compare similarities for labeled word pairs. GloVe is trained on Wikipedia (2014) and the Gigaword 5 corpus using windows of size 10 and 300-dimensional word vectors, whereas CCM uses Wikipedia (2015) with windows of size 2. Top: WS-353 similarity versus $\sigma_{i,j}$. Middle: WS-353 similarity versus GloVe cosine similarity. Bottom: GloVe cosine similarity versus $\sigma_{i,j}$.

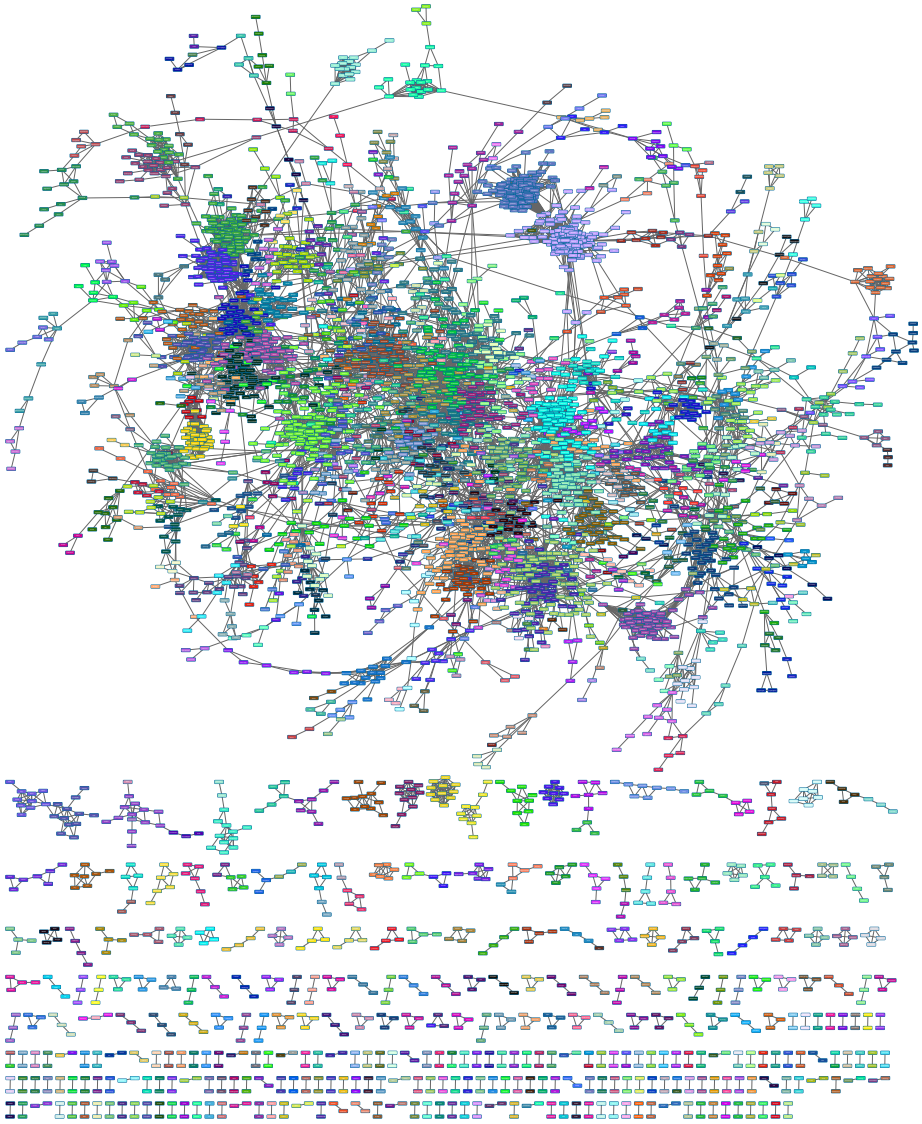

Fig. 10. Word similarity graph. In the electronic version of the paper, concepts are color coded and vertex labels are visible after zooming. The visualization is done with *Cytoscape* (Shannon et al., 2003).

5.3. Parameter sensitivity

In this section we will report how approximation errors and relevant graph structure measures are affected by the correlation graph pruning.

Table 2. Selected examples of concepts.

significant, dramatic, greater, major, enormous, modest, substantial, incredible, sharp, slight, considerable, meaningful, largest, greatest, tremendous, biggest, bigger, great, unprecedented, sudden, huge, rapid, steady, vast, massive, big, genuine, large
dallas, memphis, milwaukee, pittsburgh, detroit, cincinnati, indianapolis, diego, sydney, la, houston, cleveland, chicago, louis, sacramento, oakland, orleans, vancouver, francisco, orlando, angeles, baltimore, seattle, philadelphia, phoenix, buffalo, columbus, atlanta, vegas, denver, boston, montreal, toronto, miami, portland
yale, harvard, duke, oxford, cambridge, school, ucla, stanford, university, college
arkansas, colorado, jersey, delaware, georgia, kansas, florida, mississippi, minnesota, wisconsin, dakota, massachusetts, indiana, california, maine, pennsylvania, illinois, utah, carolina, louisiana, alaska, tennessee, texas, missouri, maryland, oklahoma, iowa, montana, hampshire, oregon, nevada, kentucky, ohio, alabama, connecticut, michigan, virginia, arizona
grey, white, yellow, gray, blue, pink, red, dark, orange, black, green
van, convoy, vessel, aircraft, ship, bus, boat, crews, vessels, cycle, bike, vehicle, trains, boats, helicopters, ships, vehicles, jet, helicopter, truck, buses, car, cars, flights, planes, firefighters, motorcycle, trucks, plane
telegraph, tribune, post, xinhua, times, magazine, newspaper, mirror, observer, herald, guardian
broadcasting, mining, banking, tech, telecommunications, wholesale, utility, retail, telecom, infrastructure
main, principal, key, decisive, vital, precious, helpful, valuable, critical, useful, essential, crucial, necessary, important
appears, sounds, appeared, sound, seemingly, looks, appear, seem, seems, appearing
soccer, tennis, diving, nba, cycling, boxing, hockey, sailing, basketball, football, baseball, rugby, nfl, golf, cricket, nhl, swimming
weeks, years, year, days, month, quarters, hours, hour, contests, moments, decades, holes, centuries, week, months, shortly, primaries, decade, minutes, seconds
tomorrow, wednesday, sunday, today, yesterday, monday, thursday, tuesday, tonight, saturday, friday
footage, photograph, season, images, episode, pictures, episodes, videos, photographs, tape, sessions, session, photos
high, lower, highest, upper, lowest, average, low, median, higher
gold, fuel, ore, electricity, silver, copper, water, ethanol, petroleum, oil, gas, uranium, coal, power, energy, petrol
isolated, remote, wealthy, urban, vulnerable, poor, poorer, poorest, impoverished, rural
ninth, third, first, sixth, fourth, fifth, seventh, second, eighth
venezuelan, british, italian, australian, palestinian, american, spanish, iranian, yemeni, afghan, georgian, swedish, austrian, lankan, lebanese, irish, german, argentine, saudi, indian, brazilian, greek, dutch, serbian, communist, egyptian, cuban, myanmar, pakistani, israeli, colombian, nigerian, tibetan, syrian, mexican, russian, portuguese, korean, somali, thai, soviet, swiss, us, czech, french, polish, chinese, uae, sudanese, japanese, belgian, norwegian, turkish, kurdish, tibet, indonesian, canadian, haitian, iraqi, english, danish, tamil

5.3.1. Approximation errors

We begin by evaluating the degree of approximation errors caused by the in-degree threshold as follows:

1. Build a reference correlation graph $\mathcal{R}$ that represents the full set of unpruned correlations. Here $\mathcal{R}$ is built from co-occurrence frequencies in the Billion word corpus, where correlations are given by conditional probabilities as described in

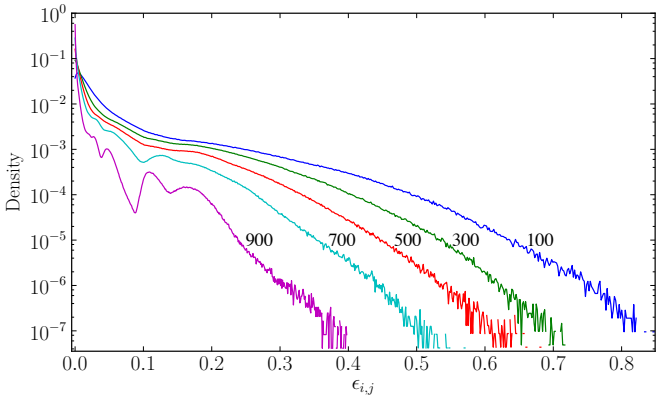

Fig. 11. Densities of absolute similarity errors $\epsilon_{i,j}$ for different in-degree thresholds δ.

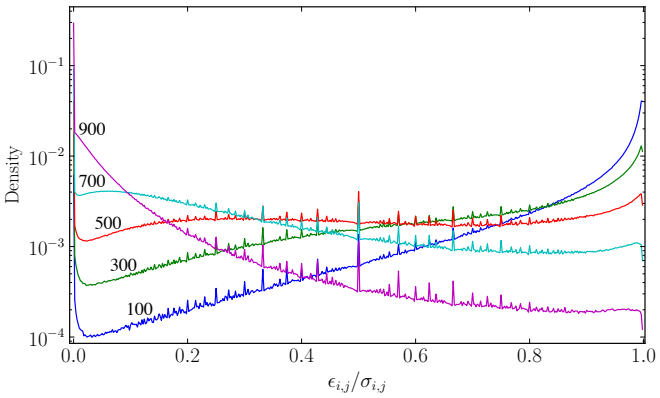

Fig. 12. Densities of relative similarity errors $\epsilon_{i,j}/\sigma_{i,j}$ for different in-degree thresholds δ.

Sec. 5.1.1. Correlations where $\rho_{i,j} \geq 10^{-5}$ are kept and in-degrees are capped at 1000. Furthermore, words occurring with a frequency less than 10^{-5} are discarded.

2. Transform $\mathcal{R}$ to a similarity graph $\mathcal{S}$, constituting the "true" similarity graph.
3. For different in-degree thresholds $\delta \in \{100, 300, ..., 900\}$:
   (a) Prune $\mathcal{R}$ into an approximate correlation graph $\mathcal{R}_\delta$.
   (b) Transform $\mathcal{R}_\delta$ into an approximate (with respect to $\mathcal{R}$) similarity graph $\mathcal{S}_\delta$.
   (c) Calculate the errors $\epsilon_{i,j} = |\sigma_{i,j} - \tilde{\sigma}_{i,j}|$ as the differences between the edge weights in corresponding edges (i, j) in $\mathcal{S}$ and $\mathcal{S}_\delta$.

The result of this procedure is shown in Figures 11 and 12, where we plot the densities (normalized counts of binned errors) of the absolute error $\epsilon_{i,j}$, as well as of the relative error, $\epsilon_{i,j}/\sigma_{i,j}$ with respect to the similarity. In both cases the

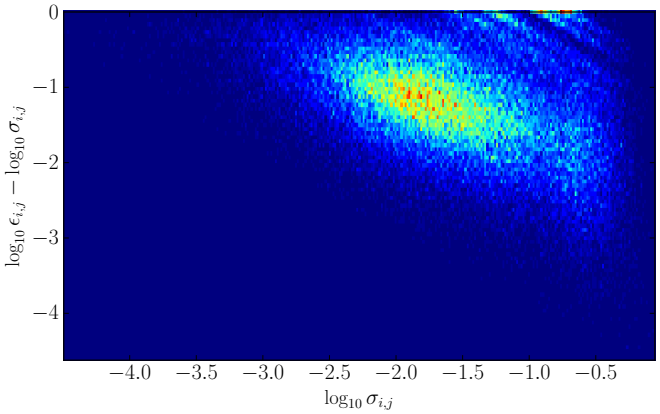

Fig. 13. Heat map of similarities $\sigma_{i,j}$ and relative errors $\epsilon_{i,j}/\sigma_{i,j}$ (log scale) for $\delta = 900$. Color coded in the electronic version of the paper, where red, yellow and cyan indicate a high density, and blue indicates a low density.

approximation errors quickly decrease with growing in-degree thresholds. Note also that the relative error in Fig. 12 is consistent with the error bound in Eq. 16 in being bound by 1.

We hypothesize that the relative errors tend to be smaller for large similarities $\sigma_{i,j}$ than for small ones, since if a correlation $\rho_{i,k}$ between i and some k is sufficiently small to be discarded, the same goes for the correlation $\rho_{j,k}$ as i and j are similar. This has the effect that the number of discarded terms in $\Lambda_{i,j}$, cf. Eq. 13, is comparably small (the number of discarded terms is at most the sum of discarded edges of i and j, which is the case when neither of these edges share a common terminal k). To test this hypothesis, we plot a heat map (i.e., a color-coded 2-dimensional histogram) of the similarities $\sigma_{i,j}$ and the relative errors $\epsilon_{i,j}/\sigma_{i,j}$ (see Fig. 13), and indeed, the relative error appears to be negatively correlated with the similarity.

5.3.2. Graph structure

We also explore how the structures of the correlation and similarity graphs are affected by the in-degree threshold. The same parameters are used as in the previous experiment. We measure the mean local clustering coefficients (Watts and Strogatz, 1998) of the graphs, which is the expected local density of edges in the neighbourhood of a vertex. More specifically, for a given vertex, this measure is given by the ratio of existing edges between the vertex' neighbours, and all possible edges between those neighbours. The mean clustering coefficient can be interpreted as a measure of the degree by which nodes are clustered. This is of interest both for the correlation graph, since it an indication of the sparsity of the graph, and, in particular, for the similarity graph, since it measures the existence of concepts in the form of tightly clustered objects.

As seen in Fig. 14, both graphs are highly clustered compared to random graphs with corresponding numbers of edges and vertices. In the correlation graph, the correlation coefficient is relatively large for small in-degree thresholds and grows substantially as this threshold increases. The situation is different for

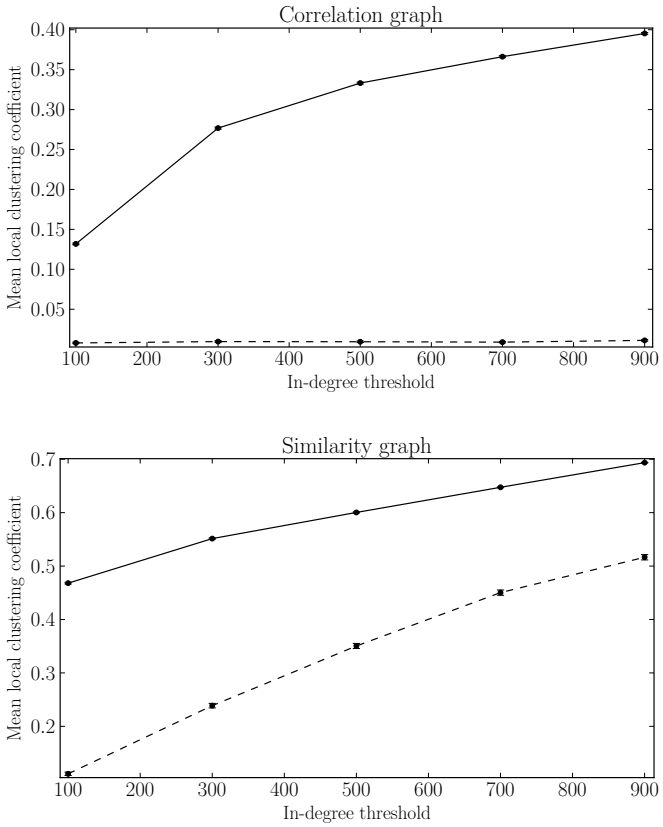

Fig. 14. Mean local clustering coefficients for correlation and similarity graphs (solid lines) for different in-degree thresholds, and for random graphs (dashed lines) with corresponding numbers of vertices and edges. Standard errors given by error bars.

the similarity graph, where although the clustering coefficient grows with the threshold, the relative coefficient with respect to the random graph is slightly decreasing. Naturally, the mean in-degree of the correlation graph decreases with decreasing in-degree threshold, which is translated to a decrease in the similarity graph, cf. Fig.15.

We can conclude that the similarity graph has a clustered structure regardless of in-degree threshold and so contains concepts in the form of grouped objects, and that the structures of both the correlation graph and the similarity graph change substantially with lowered in-degree thresholds. This is expected, but importantly, the change is in both cases smooth, e.g. we do not experience sudden "phase transitions", which tells us that we can apply the in-degree threshold in a predictable and controllable manner.

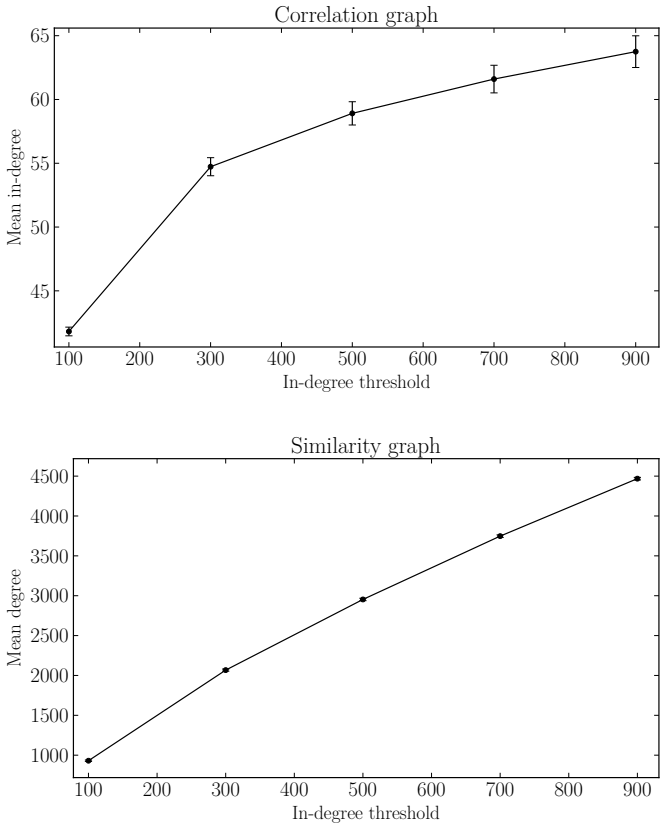

Fig. 15. Mean correlation graph in-degree and similarity graph degree (unweighted, i.e. number of edges) for different in-degree thresholds. Standard errors given by error bars.

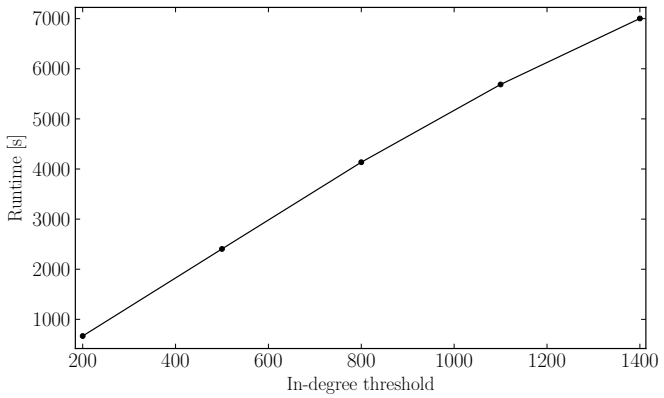

Fig. 16. Runtime for different in-degree thresholds, and $\rho_{i,j} \geq 10^{-5}$. Built from bigrams in the One billion word corpus using a commodity laptop.

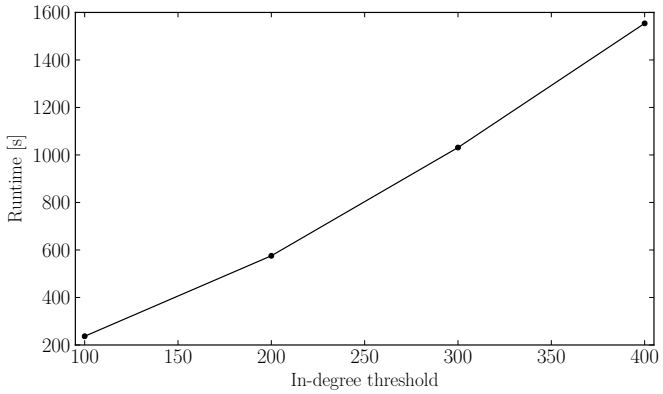

Fig. 17. Runtime for different in-degree thresholds, and $\rho_{i,j} \geq 10^{-3}$. Built from Google Books 5-grams using an Amazon EC2 cluster (see text for details).

5.4. Runtime and scalability

To demonstrate that our approach is applicable at scale in practice, we apply it on a dataset that is based on one of the largest, to our knowledge, text corpora currently available, the Google Books n-gram dataset (Michel et al., 2010; Lin et al., 2012), which corresponds to approximately 4% of all books ever printed. The dataset is publicly available, and in our experiments we use the version that is available through the Amazon S3 service.[7] As described in Sec. 5.1.1, we use the English language corpus which contains approximately 361 billion tokens. When processed into 5-grams, the corpus results in a file with 24.5 billion rows and the total compressed size of the dataset is 221.5 GB. This data is pre-processed to create the correlation graph by retaining only alphabetic characters. The resulting correlation graph before pruning has 706,108 vertices and 94,945,991 edges.

To perform the experiments we employ an Apache Spark cluster created using the Amazon Web Services EC2 service.[8] The cluster consists of 8 nodes (1 master and 7 slaves), where each node has 4 vCPUs and 30.5 GiB of memory (EC2 instance type *r3.xlarge*), such that the total amount of memory available to the cluster is roughly 186 GiB, as reported by Spark.

The experiment results support the theoretical investigation of the computational cost of the algorithm, cf. Sec. 4.4, and together with the pruning described in Section 4.1.1 we are able to transform correlation graphs into similarity graphs in reasonable amounts of time. This also holds true when using more modest computational resources, as shown in Fig. 16, for building similarity graphs using the Billion word corpus as described in Sec. 5.1.1. Analogous results are achieved in the Google 5-gram case, here with runtimes on the order of minutes, as seen in Fig. 17. The experiments were replicated three times, and the runtimes are reported in Table 3. Together with Fig. 11, Fig. 16 and Fig. 17 illustrate the trade-off between accuracy, controlled via the in-degree threshold,

[7] https://aws.amazon.com/datasets/google-books-ngrams/
[8] http://aws.amazon.com/ec2/

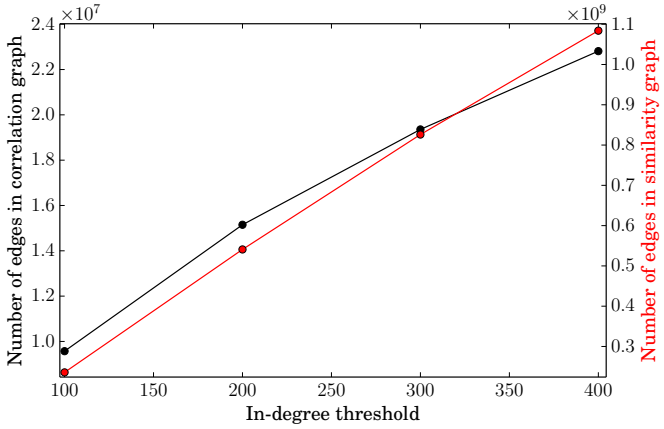

Fig. 18. Number of edges in the correlation- and similarity graph, respectively, for different in-degree thresholds. Built from Google Books 5-grams with the same configuration as in Fig. 17.

Table 3. Runtimes in seconds for Google Books dataset.

In-degree	**Run 1**	**Run 2**	**Run 3**	μ	σ
100	246.7	229.5	236.7	237.6	8.6
200	603.7	573.2	575.4	584.1	16.9
300	1062.4	998.3	1031.2	1030.7	32.0
400	1535.5	1602.5	1554.0	1564.0	34.6

and runtime, where the runtime scales favourably with an increasing in-degree threshold. With respect to the in-degree threshold, we also observe a sublinear scaling of the number of edges in the correlation graph, and a linear growth of the number of edges in the similarity graph, as shown in Fig. 18. This reflects the situation exemplified i n Fig. 3, namely that comparably few vertices are affected by the in-degree threshold.

6. Conclusions

This paper proposes conceptually simple methods for discovering similarities and concepts by transforming a correlation graph to a similarity graph on which clustering is performed. As the approach does not rely on any intermediate representation or dimensionality reduction, or on specific information about objects besides their correlations, it is applicable with few restrictions to any domain in which a correlation graph can be constructed. Our experiments show that the approach not only can detect similarities and concepts in several types of data, but also that it is computationally feasible for large-scale applications with very large numbers of objects and correlations.

Due to the generality of the approach there is a vast number of possible directions to take. For instance, it can potentially be used to discover analogous objects in gene regulatory data or protein interaction networks, to provide recommendations from user data, or in general for detecting higher-order dynamics in discrete-valued stochastic processes. It then remains to quantitatively evaluate the properties of the scheme, for example in terms of application specific benchmark performance, approximation error and runtime.

The main methodological challenge for future work revolves around how to efficiently build hierarchical concept models. The concepts discovered through the methods described in this paper essentially represent OR-relations: All constituent objects of a cluster are commutable, and the concept can be said to be observed if any of its constituents are. Analogously, strong clusters detected in the correlation graph could be considered to represent AND-relations, where the corresponding concept is observed when all of its constituents are. Both these types of concepts can be identified, brought back into the estimation of the correlation graph, and the process iterated, allowing for the discovery of complex higher-order relations. How to reliably and efficiently perform this remains an area of further study.

Acknowledgements. This work was funded by the Swedish Foundation for Strategic Research (*Stiftelsen för strategisk forskning*) and the Knowledge Foundation (*Stiftelsen för kunskaps- och kompetensutveckling*). The authors would like to thank the anonymous reviewers for their valuable comments.

References

Albert, R. and Barabási, A.-L. (2002), 'Statistical mechanics of complex networks', *Rev. Mod. Phys.* **74**(1), 47–97.

Alexandrov, A., Bergmann, R. and Ewen, S. et al. (2014), 'The Stratosphere platform for big data analytics', *The VLDB Journal* pp. 163–181.

Anisimova, M. and Kosiol, C. (2009), 'Investigating protein-coding sequence evolution with probabilistic codon substitution models.', *Molecular Biology and Evolution* **26**(2), 255–271.

Bitton, D., Boral, H., DeWitt, D. J. et al. (1983), 'Parallel algorithms for the execution of relational database operations', *ACM Transactions in Database Systems* **8**(3), 324–353.

Bouma, G. (2009), Normalized (pointwise) mutual information in collocation extraction, *in* 'From form to meaning: Processing texts automatically, Proceedings of the Biennial GSCL Conference', pp. 31–40.

Brown, P. F., deSouza, P. V., Mercer, R. L. et al. (1992), 'Class-based N-gram Models of Natural Language', *Computational Linguistics* **18**(4), 467–479.

Cancho, R. F. and Solé, R. V. (2001), 'The small world of human language', *Proceedings of the Royal Society of London. Series B: Biological Sciences* **268**(1482), 2261–2265.

Celma, Ò. (2010), *Music Recommendation and Discovery in the Long Tail*, Springer.

Celma, Ò. and Cano, P. (2008), From hits to niches? or how popular artists can bias music recommendation and discovery, *in* 'Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition', ACM, p. 5.

Chandra, A. K. and Merlin, P. M. (1977), Optimal implementation of conjunctive queries in relational data bases, *in* 'Proceedings of the Ninth Annual ACM Symposium on Theory of Computing', STOC '77, ACM, New York, NY, USA, pp. 77–90.

Chelba, C., Mikolov, T., Schuster, M. et al. (2013), 'One billion word benchmark for measuring progress in statistical language modeling.', *CoRR* **abs/1312.3005**.

Church, K. W. and Hanks, P. (1990), 'Word association norms, mutual information, and lexicography', *Computational Linguistics* **16**(1), 22–29.

Dayhoff, M. O. and Schwartz, R. M. (1978), Chapter 22: A model of evolutionary change in proteins, *in* 'Atlas of Protein Sequence and Structure'.

Dice, L. R. (1945), 'Measures of the amount of ecologic association between species', *Ecology* **26**(3), 297–302.

Finkelstein, L., Gabrilovich, E., Matias, Y. et al. (2001), Placing search in context: The concept revisited, *in* 'Proceedings of the 10th International Conference on World Wide Web', WWW '01, ACM, New York, NY, USA, pp. 406–414.

Firth, J. R. (1957), A synopsis of linguistic theory 1930–55., *in* 'Studies in Linguistic Analysis (special volume of the Philological Society)', Vol. 1952-59, The Philological Society, pp. 1–32.

Fortunato, S. (2010), 'Community detection in graphs', *Physics Reports* **486**(3-5), 75–174.

Görnerup, O., Gillblad, D. and Vasiloudis, T. (2015), Knowing an object by the company it keeps: A domain-agnostic scheme for similarity discovery, *in* 'IEEE International Conference on Data Mining (ICDM 2015)'.

Graff, D. (2003), 'English Gigaword'.

Halawi, G., Dror, G., Gabrilovich, E. et al. (2012), Large-scale learning of word relatedness with constraints, *in* 'Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, New York, NY, USA, pp. 1406–1414.

Harispe, S., Ranwez, S., Janaqi, S. et al. (2015), 'Semantic Similarity from Natural Language and Ontology Analysis', *Synthesis Lectures on Human Language Technologies* **8**(1), 1–254.

Harris, Z. (1954), 'Distributional structure', *Word* **10**(23), 146–162.

Hill, F., Reichart, R. and Korhonen, A. (2014), 'Simlex-999: Evaluating semantic models with (genuine) similarity estimation', *CoRR* **abs/1408.3456**.

Jaccard, P. (1912), 'The distribution of the flora in the alpine zone', *New Phytologist* **11**(2), 37–50.

Jeh, G. and Widom, J. (2002), Simrank: A measure of structural-context similarity, *in* 'Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '02, ACM, New York, NY, USA, pp. 538–543.

Jordan, I. K., Mariño Ramírez, L., Wolf, Y. I. et al. (2004), 'Conservation and Coevolution in the Scale-Free Human Gene Coexpression Network', *Molecular Biology and Evolution* **21**(11), 2058–2070.

Kessler, M. (1963), 'Bibliographic coupling between scientific papers', *American Documentation 14* pp. 10–25.

Koutris, P. and Suciu, D. (2011), Parallel evaluation of conjunctive queries, *in* 'Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems', PODS '11, ACM, New York, NY, USA, pp. 223–234.

Larson, R. (1996), 'Bibliometrics of the World Wide Web: An exploratory analysis of the intellectual structure of cyberspace', *Ann. Meeting of the American Soc. Info. Sci.* .

Leicht, E. A., Holme, P. and Newman, M. E. J. (2006), 'Vertex similarity in networks', *Physical Review E* **73**, 026120.

Lin, Y., Michel, J., Aiden, E. L. et al. (2012), Syntactic Annotations for the Google Books Ngram Corpus, *in* 'Proceedings of the ACL 2012 System Demonstrations', ACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 169–174.

Michel, J., Shen, Y. K., Aiden, A. P. et al. (2010), 'Quantitative analysis of culture using millions of digitized books', *Science* .

Mihalcea, R. and Radev, D. (2011), *Graph-based natural language processing and information retrieval*, Cambridge University Press.

Mikolov, T., Sutskever, I., Chen, K. et al. (2013), Distributed representations of words and phrases and their compositionality, *in* 'Advances in Neural Information Processing Systems', pp. 3111–3119.

Miller, G. A. (1995), 'Wordnet: a lexical database for english', *Communications of the ACM* **38**(11), 39–41.

Mislove, A., Marcon, M., Gummadi, K. P. et al. (2007), Measurement and analysis of online social networks, *in* 'Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement', IMC '07, ACM, New York, NY, USA, pp. 29–42.

Nirenberg, M., Leder, P., Bernfield, M. et al. (1965), 'RNA Codewords and Protein Synthesis, VII. On the General Nature of the RNA Code', *Proceedings of the National Academy of Science* **53**, 1161–1168.

Palla, G., Derenyi, I., Farkas, I. et al. (2005), 'Uncovering the overlapping community structure of complex networks in nature and society', *Nature* **435**(7043), 814–818.
URL: *http://dx.doi.org/10.1038/nature03607*

Pecina, P. (2008), A machine learning approach to multiword expression extraction, *in* 'Proceedings of the LREC 2008 Workshop Towards a Shared Task for Multiword Expressions', European Language Resources Association, pp. 54–57.

Pennington, J., Socher, R. and Manning, C. (2014), Glove: Global Vectors for Word Represen-

tation, *in* 'Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)', Association for Computational Linguistics, pp. 1532–1543.

Ravasz, E., Somera, A. L., Mongru, D. A. et al. (2002), 'Hierarchical organization of modularity in metabolic networks', *Science* **297**(5586), 1551–1555.

Sahlgren, M. (2006), The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces., PhD thesis, Stockholm University.

Schneider, A., Cannarozzi, G. and Gonnet, G. (2005), 'Empirical codon substitution matrix', *BMC Bioinformatics* **6**(134).

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B. and Ideker, T. (2003), 'Cytoscape: A software environment for integrated models of biomolecular interaction networks', *Genome Research* **13**(11), 2498–2504.

Small, H. (1973), 'Co-citation in the scientific literature: A new measure of the relationship between two documents', *Journal of the American Society for Information Science* **24**(4), 265–269.

Sørensen, T. (1948), 'A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons', *Biol. Skr.* **5**, 1–34.

Steyvers, M. and Tenenbaum, J. B. (2005), 'The large-scale structure of semantic networks: statistical analyses and a model of semantic growth.', *Cognitive science* **29**(1), 41–78.

Watts, D. J. and Strogatz, S. H. (1998), 'Collective dynamics of 'small-world' networks', *Nature* **393**(6684), 409–10.

Wong, W., Liu, W. and Bennamoun, M. (2012), 'Ontology learning from text: A look back and into the future', *ACM Comput. Surv.* **44**(4), 20:1–20:36.

Wu, T. D. and Brutlag, D. L. (1996), Discovering empirically conserved amino acid substitution groups in databases of protein families, *in* D. J. States, P. Agarwal, T. Gaasterland, L. Hunter and R. Smith, eds, 'Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology, St. Louis, MO, USA, June 12-15 1996', AAAI, pp. 230–240.

Xie, J., Szymanski, B. K. and Liu, X. (2011), SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, *in* 'ICDM 2011 Workshop on DMCCI'.

Yih, W. and Qazvinian, V. (2012), Measuring word relatedness using heterogeneous vector space models, *in* 'Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies', NAACL HLT '12, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 616–620.

Yu, W., Zhang, W., Lin, X. et al. (2012), 'A space and time efficient algorithm for simrank computation', *World Wide Web* **15**(3), 327–353.

Zaharia, M., Chowdhury, M., Das, T. et al. (2012), Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing, *in* 'Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)', San Jose, CA, pp. 15–28.

Zhang, B. and Horvath, S. (2005), 'A general framework for weighted gene co-expression network analysis', *Statistical applications in genetics and molecular biology* **4**, Article17.

Author Biographies

Olof Görnerup is currently a senior researcher at the Swedish Institute of Computer Science (SICS) in Stockholm. He received a MSc, a PhLic and a PhD in complex systems from Chalmers University of Technology in Sweden in 2003, 2007 and 2008 respectively. During 2002–2004 he was a Graduate fellow at the Santa Fe Institute in New Mexico, USA. His current research interests revolve around fundamental machine learning, data mining and data analytics, and their interdisciplinary applications.

Daniel Gillblad holds a MSc in electrical engineering and a PhD in computer science, both from the Royal Institute of Technology (KTH) in Sweden. He is currently the director of the Decisions, Networks and Analytics (DNA) laboratory at the Swedish Institute of Computer Science (SICS). The laboratory performs research within machine learning, data analytics, networked intelligence, scheduling and optimisation, and their applications. His research interests are currently focused around graph- and probabilistic methods for large-scale data analytics and machine learning, network management, diagnostics, and mobility modeling.

Theodore Vasiloudis is a researcher at the Swedish Institute of Computer Science (SICS) and a PhD candidate at the Royal Institute of Technology (KTH) in Stockholm. His main research interests include large-scale machine learning, graph processing and natural language processing. He is also a contributor to the machine learning library for Apache Flink, FlinkML.

Correspondence and offprint requests to: Olof Görnerup, Swedish Institute of Computer Science (SICS), SE-164 29 Kista, Sweden. Email: olof@sics.se.