Knowing an Object by the Company It Keeps: A Domain-Agnostic Scheme for Similarity Discovery

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

Daniel Gillblad
Swedish Institute of Computer Science (SICS)
SE-164 29 Kista, Sweden
Email: dgi@sics.se

Theodore Vasiloudis
Swedish Institute of Computer Science (SICS)
SE-164 29 Kista, Sweden
Email: tvas@sics.se

***Abstract*—Appropriately defining and then efficiently calculating similarities from large data sets are often essential in data mining, both for building tractable representations and for gaining understanding of data and generating processes. Here we rely on the premise that given a set of objects and their correlations, each object is characterized by its context, i.e. its correlations to the other objects, and that the similarity between two objects therefore can be expressed in terms of the similarity between their respective contexts. Resting on this principle, we propose a data-driven and highly scalable approach for discovering similarities from large data sets by representing objects and their relations as a correlation graph that is transformed to a similarity graph. Together these graphs can express rich structural properties among objects. Specifically, we show that concepts – representations of abstract ideas and notions – are constituted by groups of similar objects that can be identified by clustering the objects in the similarity graph. These principles and methods are applicable in a wide range of domains, and will here be demonstrated for three distinct types of objects: codons, artists and words, where the numbers of objects and correlations range from small to very large.**

I. Introduction

As stated by Firth [1] and further popularized in the computational linguistics community by Church and Hanks [2], "You shall know a word by the company it keeps". Departing from this principle, which can be traced further back to analytic philosophy, there have been substantial efforts to infer semantic and syntactic meaning from words through their effective usage in text [3]. Although the same principle has been applied in different and seemingly distinct domains, such as bibliometrics [4] and bioinformatics [5], generalizing the notion of characterizing objects through their contexts into a broader fundamental principle for similarity discovery is so far largely unexplored.

Extending Firth's line of thought we argue that, with respect to observed data, the effective semantics of any object are given 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. We then 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. [6], 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[1] – which is the case e.g. for gene co-expression [7], semantic [8], word co-occurrence [9] and social networks [10], as well as for many other graphs of interest [6] – we can also prune the correlation graph substantially prior to 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

[1]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.

objects, while the approach presented here is both highly 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, concept 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. As an illustrative example, it may for instance not be appropriate to analyze a large text corpus in terms of its individual characters, when the data can be described in terms of words or on more abstract levels still. We will here demonstrate that CCM can be used for this purpose. Specifically, we will show that *concepts* – coarse-grained abstractions of objects – are constituted by groups of inter-similar objects that play analogous roles in data, and that we can discover these by clustering the objects in the similarity graph.

A. 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. III, followed by a description of the proposed method in Sec. IV. In Sec. V we demonstrate the versatility of the method by applying it in three distinct domains, with proof-of-concepts in computational linguistics, music and molecular biology. Sec. VI treats the scaling properties of the method, where we show that it is scalable both in theory and practice. The paper is concluded in Sec. VII with a summary of our findings and a discussion on possible future directions.

II. 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. The method that is closest in spirit to ours is SimRank [11], 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" [11]. 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 [12], 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 graphs with tens of millions of edges, doing only a single pass over the data. Ravasz et al. propose a related approach for finding similar vertices using so called topological overlap measures [5], which they apply on metabolic networks. Zhang et al. [13] generalized this approach for use on weighted gene co-expression networks. As in our case, these methods relate vertices by assigning higher similarity scores to 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 [14]. 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 [15], in which concepts are discovered through clustering. A large number of methods to find semantic similarities have been developed – see [3] for a recent review – from the seminal work by Church and Hanks [2], and Brown et al. [16], to more recent approaches, e.g. based on vector representations, such as GloVe [17], and neural networks, such as word2vec [18]. Several of these methods could be used to produce the equivalent of the similarity graph in which we perform clustering to find concepts. These methods, however, are limited to natural language processing while our approach is domain-agnostic. 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 [19], 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 [20].

Another related area is ontology learning [21], 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 [4] – 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 [22] as well as web pages [23]. 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.

III. Background

A. 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, probabilities, 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 friendship.

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 its friendships.

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 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, $\mathcal{R} = (C, R)$, 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-friendship example, the correlation network is simply a social network.

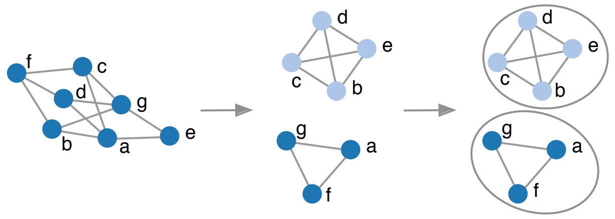

Fig. 1. A correlation graph is transformed to a similarity graph in which clustering is performed.

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 social network example, a concept is a group of people that have similar circles of acquaintances.

B. 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 figure is 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 [24], [25], that, in turn, is analogous to the commonly used Jaccard coefficient [26] through a monotonic transformation.

IV. Methods

A. Similarity calculations

In order to efficiently and scalably transform a correlation graph into a similarity graph, we utilize two observations. Firstly, 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

$$L_1(i,j) = \sum_{k \in n_i} |\rho_{i,k}| - \sum_{k \in n_i \cap n_j} |\rho_{i,k}|$$
$$+ \sum_{k \in n_j} |\rho_{j,k}| - \sum_{k \in n_i \cap n_j} |\rho_{j,k}| + \sum_{k \in n_i \cap n_j} |\rho_{i,k} - \rho_{j,k}|$$
$$= |\rho_i|_1 + |\rho_j|_1 + \sum_{k \in n_i \cap n_j} (|\rho_{i,k} - \rho_{j,k}| - |\rho_{i,k}| - |\rho_{j,k}|). \quad (6)$$

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, give 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. 6, and adding this to the edge weight sums of i and j.

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 regard 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,

$$E_D(i,j) \leq \theta_D \quad (7)$$

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

$$E_1(i,j) = |L_1(i,j) - \tilde{L}_1(i,j)| \leq \theta_1. \quad (8)$$

If we would like to remove terms by approximating by zero while keeping the total approximation error E_D as small as possible, we should remove the smallest 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 correlations of object i are approximated by zero, and $|\check{\rho}_i|_1$ the norm of discarded correlations:

$$|\check{\rho}_i|_1 = \sum_{\rho_{i,k} < \tau_i} |\rho_{i,k}|. \quad (9)$$

The upper bound of the error is then given by

$$E_1(i,j) \leq |\check{\rho}_i|_1 + |\check{\rho}_j|_1, \quad (10)$$

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. 6 while guaranteeing an error $E_1(i,j) \leq \theta_1$ is then a matter of sorting correlations $\rho_{i,k}$ and, starting with the smallest one, removing all relations until the cumulative sum exceeds half the distance error threshold, $\theta_1/2$.

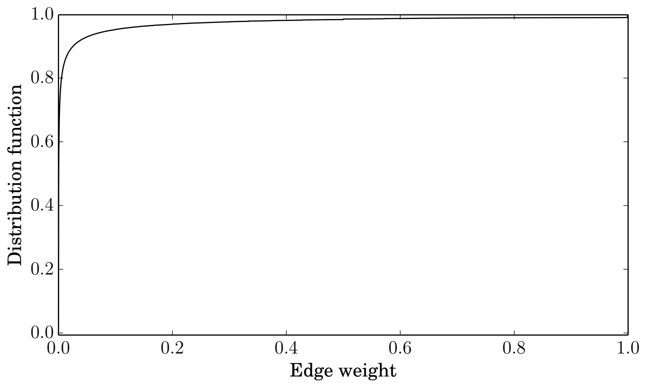

Fig. 2. The cumulative distribution function of edge weights in the Billion word correlation graph described in Sec. V-A shows that a large fraction of edges with low weights can be pruned. For example, approximately 90% of the edges are discarded when considering edges with weights ≥ 0.01.

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. Thus, we may effectively prune a large fraction of the links while keeping the cumulative discarded weight (and error) comparatively low, further reducing computational complexity.

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 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. 10.

B. Clustering

After transforming a correlation graph to a similarity graph, the latter typically exhibits tightly grouped objects that are similar according to measure $\sigma_{i,j}$. We can therefore identify concepts by clustering the vertices, which is also known as community detection. There is a large number of available algorithms with varying suitability with regard to accuracy and scalability [27]. However, it is beyond the scope of this paper to evaluate the performance of different clustering algorithms in this context. Instead we use a simple and transparent clustering method. The approach resembles standard distributed algorithms for identifying connected components in graphs and works as follows: We begin by initializing each vertex i to form its own cluster, indexed by $c_i = i$. Then, for each vertex i, we set its cluster index to be the smallest cluster index of i:s neighbours j for which $\sigma_{i,j} \geq \sigma_{min}$, where σ_{min} is a threshold value. This is repeated until no more cluster indices are changed. In this way, cluster memberships are propagated

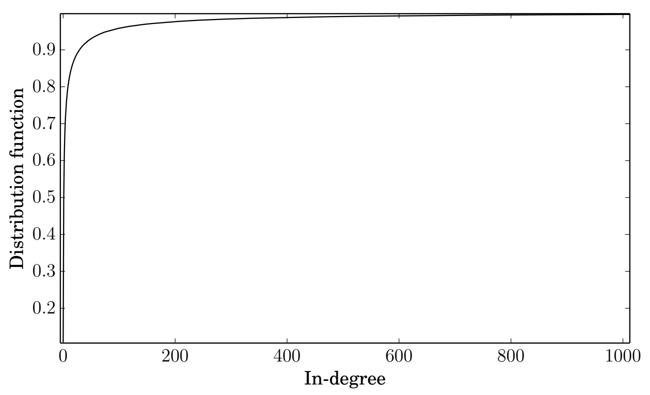

Fig. 3. The cumulative distribution function of in-degrees for the graph referred to in Fig. 2 illustrates that it is possible to apply an in-degree threshold while affecting comparably few vertices. Only a small percentage of the vertices are affected, for instance, when capping the in-degree at 500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

within components that are separated by edges with weights $\sigma_{i,j} \leq \sigma_{min}$. The interpretation of – and rationale for – this approach is that clusters in the graph are groups of vertices that are interlinked with a certain degree of similarity, as specified by σ_{min}, and where the clusters, in turn, are interlinked with weaker similarity relations.

C. Implementation

The calculations of the approximations and error bounds of the norms of the differences $|\rho_i - \rho_j|_1$, 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) 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.
2) For each vertex i, calculate the norms $|\rho_i|_1$ (the weight sum prior to pruning), $|\check{\rho}_i|_1$ (the weight sum of discarded edges) and $|\hat{\rho}_i|_1$ (the weight sum of kept edges), where $|\check{\rho}_i|_1$ is simply acquired by subtracting $|\hat{\rho}_i|_1$ from $|\rho_i|_1$.
3) Calculate the sum term in Eq. 6, denoted $\Lambda_{i,j}$, for each pair of vertices that share a 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 (i, j) in the previous step, calculate the approximate relative L_1-norm, $\tilde{l}_{i,j}$, as $\tilde{l}_{i,j} = (\Lambda_{i,j} + \psi_{i,j})/\psi_{i,j}$ and the upper error bound, $\epsilon_{i,j}$, as $\epsilon_{i,j} = (|\check{\rho}_i|_1 + |\check{\rho}_j|_1)/\psi_{i,j}$, where the normalizing factor $\psi_{i,j} = |\rho_i|_1 + |\rho_j|_1$ is the maximum possible difference between i and j.

After completing step 4 it is straightforward to calculate the approximate similarity $\tilde{\sigma}_{i,j} = 1 - \tilde{l}_{i,j}$ according to Eq. 1. Note that $\tilde{l}_{i,j}$ is a conservative approximation of the true relative L_1-norm, $l_{i,j}$, since $\tilde{l}_{i,j} - \epsilon_{i,j} \leq l_{i,j} \leq \tilde{l}_{i,j}$. For this reason, the acquired similarity approximation will be the "worst case scenario" in the sense that it is always larger than the true relative L_1-norm.

The method is implemented in the Scala programming language and uses the in-memory data processing framework Apache Spark [28], which enables us to employ the method at scale in terms of computing hardware. To facilitate reproducibility, the implementation will be made available with an open source license in an online repository.[2] 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 [29], is also possible.

V. Experiments

In order to demonstrate the broad applicability of our approach, we will now showcase it for three distinct types of objects: words, artists and codons. Here we prioritize breadth over depth, and more in-depth evaluations of the method's performance with respect to specific applications will be topics of future publications.

A. 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 therefore co-occur if they are adjacent. There exist many different word association measures, see [30] for a large number of examples, such as pointwise mutual information [2] and normalized versions thereof [31]. Here we simply measure the association between i and j as the relative frequency of j occurring in i:s context, 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. There likely exists more appropriate measures, such as the aforementioned pointwise mutual information, with regard to specific applications. However, for the purpose of demonstrating our approach, we believe the conditional probability measure suffices.

The method is applied on two datasets: the Billion word [32] and the Google Books n-gram [33], [34] corpora. The former consists of nearly one billion tokens and originates from crawled online news texts. From these 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

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

```
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))) => ((i,j),abs(rik-rjk)-abs(rik)-abs(rjk)))
4:                .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) All terms in the sum in Eq. 6 are calculated and 4) summed per two-hop neighbour pair.

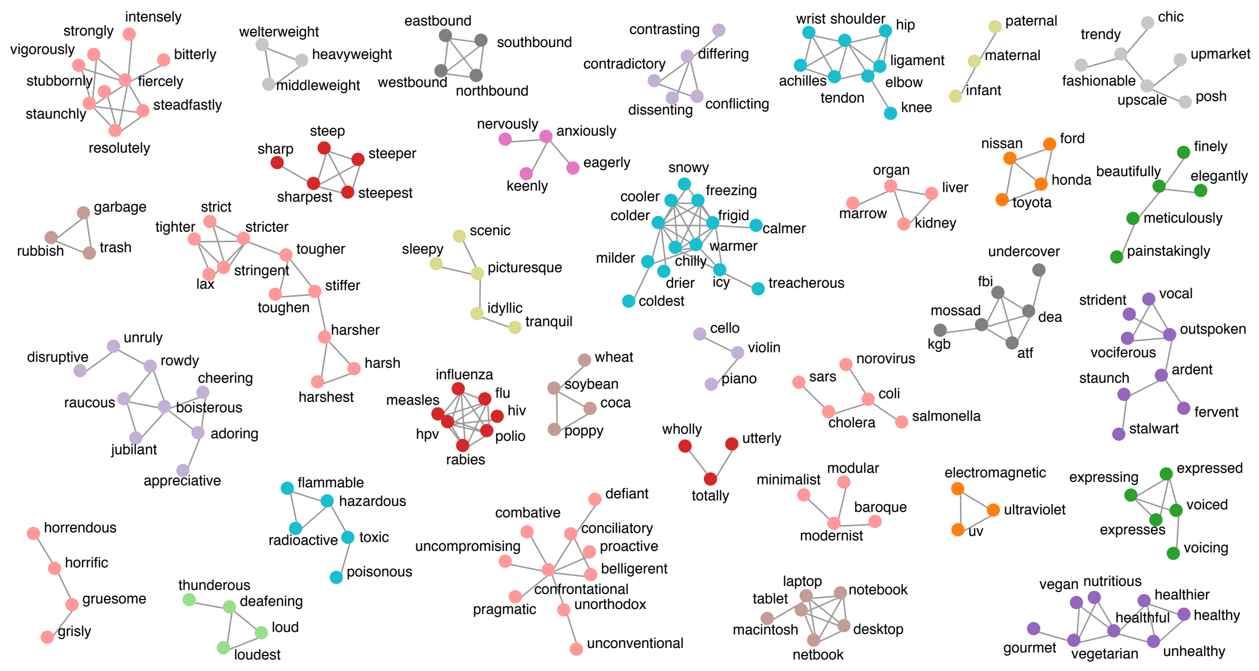

Fig. 5. Examples of concepts in a word similarity graph based on the Billion word corpus are constituted by clusters of similar words. For sake of clarity, edges with weights $\sigma_{i,j} \geq 0.15$ are shown.

groups of words that reflect both syntactic and semantic concepts. Examples of such concepts are shown in Fig. 5, where we see that the clusters 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 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.

The Google Books n-gram dataset, which consists of 361 billion tokens for the English language version of the dataset, is used both to evaluate the scalability of the method, which will be discussed in Sec. VI, and to quantify the quality of resulting similarity relations. An n-gram can be defined as a contiguous sequence of n words in a text. To further challenge the method, we apply it on correlation graphs with respect to co-occurrence windows of size 5. This results in a denser correlation graph, since a word has more neighbors due to the larger co-occurrence window size. Nevertheless, the key properties that we describe in Sec. IV-A1 still apply and we can prune away a large number of edges with low weights.

A common approach to quantitatively evaluate the performance of word association methods is to use benchmarks with word pairs that have been manually graded with respect to degree of association. 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). However, to give an indication of the method's performance, we measure the Spearman rank correlation coefficient between benchmark similarities and $\sigma_{i,j}$ for word pairs (i, j) that *do* exist in the similarity graph. For this purpose we use the standard WS-353 test collection [35], which consists of 353 word pairs that have been graded by human annotators. We build a similarity graph from co-occurrence windows of size 5, 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 this graph, which is built in less than 10 minutes (cf. Fig. 9), 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 [36], [37]. These figures represent the correlation with respect to the average annotator score. Note, however, that there is low inter-annotator agreement in WS-353, where the mean performance of individual annotators, with respect to the mean score of the remaining annotators, is in fact also 0.76 [38].

B. 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.[3] This dataset, provided by Celma [39], consists of approximately 19 million track plays of 992 users. 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, 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. 6. 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 [40].

C. 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 [41]. 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 [41], 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 substitution matrix provided by Schneider et al. [42], 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.

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

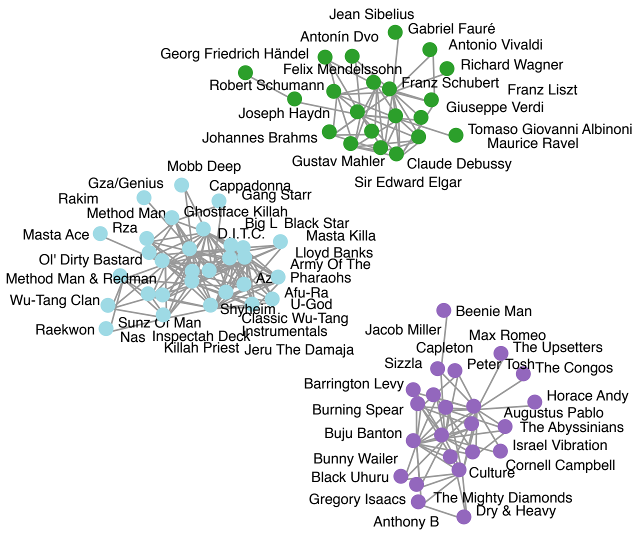

Fig. 6. Examples of clusters in an artist similarity graph correspond to three distinct music genres. Edges with weights $\sigma_{i,j} \geq 0.5$ are shown.

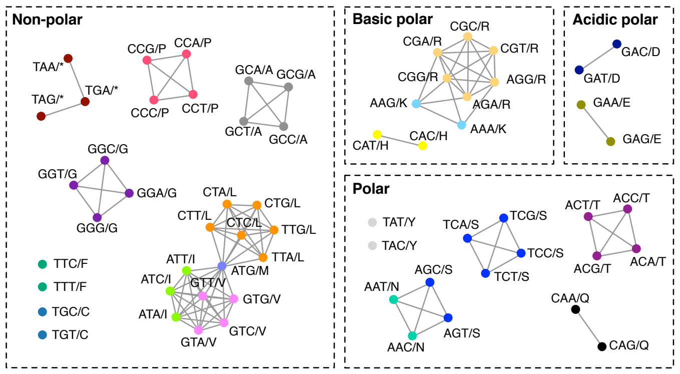

Fig. 7. Codon similarity graph where vertices are labeled with c/a for codon c coding to amino acid a. Edges with weights $\sigma_{i,j} \geq 0.45$ are shown. Vertices are color coded with respect to amino acids and grouped by properties. Note that when the edge weight threshold is lowered, clusters containing several amino acids are split by amino acid. The rare and low mutable amino acid tryptophan is omitted.

As seen in the resulting codon similarity graph in Fig. 7, codons that translate to the same amino acid according to the standard genetic code [43] 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. 7, 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. [44]. In comparison, Wu and Brutlag derived amino acid substitution groups by group-wise (as opposed to pairwise) statistical analysis of protein databases [45]. The groups shown in Fig. 7 ({I, L, M, V}, {K, R} and {N, S}) all agree with their

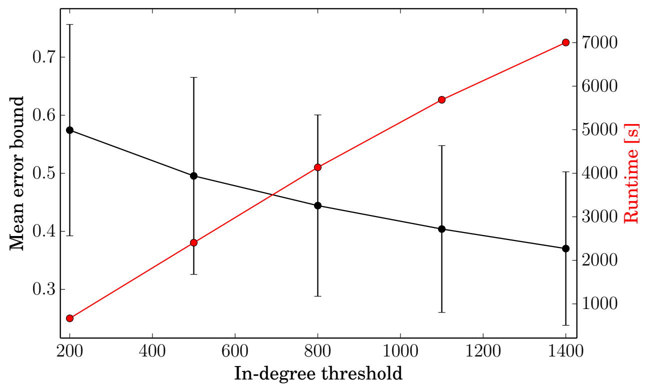

Fig. 8. Runtime, mean error bound and standard deviation of error bound (shown as error bars) for different in-degree thresholds, and $\rho_{i,j} \geq 10^{-5}$. Built from bigrams in the Billion word corpus using a commodity laptop.

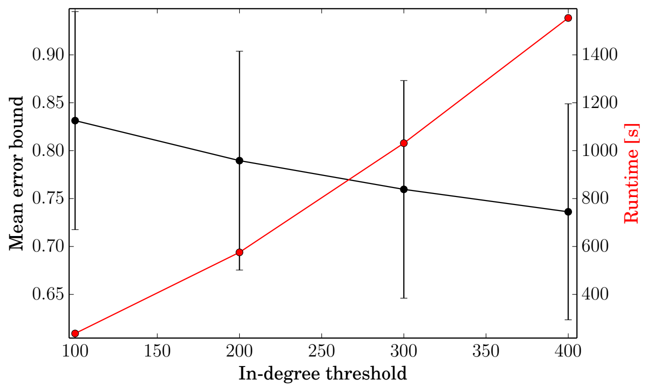

Fig. 9. Runtime, mean error bound and standard deviation of error bound (shown as error bars) for different in-degree thresholds, and $\rho_{i,j} \geq 10^{-3}$. Built from Google Books 5-grams using an Amazon EC2 cluster (see text for details).

findings. In summary, the codon similarity graph captures both concepts and higher-order concepts: from codons to amino acids, via the genetic code, to higher-order amino acids that constitute known substitution groups.

VI. Scalability

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 [46], [47].

The most computationally demanding component of the algorithm is building the two-hop graph through a self-join operation (the third step in Sec. IV-C). Since a self-join is a conjunctive query [46] in relational algebra terms, we can reason about its computational cost. Specifically for a distributed environment, Koutris et al. [48] 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. [28] 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.

To demonstrate that our approach is applicable at scale in practice, we apply it on one of the largest, to our knowledge, text corpora currently available, the Google Books n-gram dataset [33], [34], 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.[4] As described in Sec. V-A, 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.[5] 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, and together with the pruning described in Section IV-A1 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. 8, for building similarity graphs using the Billion word corpus as described in Sec. V-A. Analogous results are achieved in the Google 5-gram case, here with runtimes on the order of minutes, as seen in Fig. 9. The experiments were replicated three times, and the runtimes are reported in Table I. Fig. 8 and Fig. 9 also illustrate the trade-off between accuracy, controlled via the in-degree threshold, 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. 10. This reflects the situation exemplified in Fig. 3, namely that comparably few vertices are affected by the in-degree threshold.

[4]https://aws.amazon.com/datasets/8172056142375670

[5]http://aws.amazon.com/ec2/

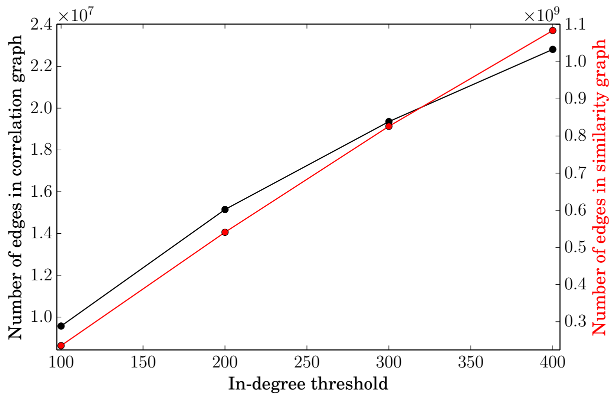

Fig. 10. Number of edges in the correlation- and similarity graph, respectively, for different in-degree thresholds. Same configuration as in Fig. 9.

TABLE I. 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

VII. Conclusions

This paper proposes a conceptually simple method for discovering similarities and concepts through transforming a correlation graph to a similarity graph on which clustering is performed. As the method does not rely on any intermediate representation or dimensionality reduction, 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.

Due to the generality of the approach there is a vast number of possible directions to take. For instance, CCM 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.

Acknowledgment

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

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

[2] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[3] S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain, "Semantic Similarity from Natural Language and Ontology Analysis," *Synthesis Lectures on Human Language Technologies*, vol. 8, no. 1, pp. 1–254, 2015.

[4] M. Kessler, "Bibliographic coupling between scientific papers," *American Documentation 14*, pp. 10–25, 1963.

[5] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[6] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, no. 1, pp. 47–97, Jan. 2002.

[7] I. K. Jordan, L. Mariño Ramírez, Y. I. Wolf, and E. V. Koonin, "Conservation and Coevolution in the Scale-Free Human Gene Coexpression Network," *Molecular Biology and Evolution*, vol. 21, no. 11, pp. 2058–2070, 2004.

[8] M. Steyvers and J. B. Tenenbaum, "The large-scale structure of semantic networks: statistical analyses and a model of semantic growth." *Cognitive science*, vol. 29, no. 1, pp. 41–78, 2005.

[9] R. F. Cancho and R. V. Solé, "The small world of human language," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 268, no. 1482, pp. 2261–2265, 2001.

[10] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 29–42.

[11] G. Jeh and J. Widom, "SimRank: A Measure of Structural-context Similarity," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 538–543.

[12] W. Yu, W. Zhang, X. Lin, Q. Zhang, and J. Le, "A space and time efficient algorithm for SimRank computation," *World Wide Web*, vol. 15, no. 3, pp. 327–353, 2012.

[13] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," *Statistical applications in genetics and molecular biology*, vol. 4, p. Article17, 2005.

[14] Z. Harris, "Distributional structure," *Papers in structural and transformational Linguistics*, 1970.

[15] M. Sahlgren, "The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces." Ph.D. dissertation, Stockholm University, 2006.

[16] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based N-gram Models of Natural Language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[17] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014, pp. 1532–1543.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

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

[20] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[21] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the future," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 20:1–20:36, 2012.

[22] H. Small, "Co-citation in the scientific literature: A new measure of the relationship between two documents," *Journal of the American Society for Information Science*, vol. 24, no. 4, pp. 265–269, 1973.

[23] R. Larson, "Bibliometrics of the World Wide Web: An exploratory analysis of the intellectual structure of cyberspace," *Ann. Meeting of the American Soc. Info. Sci.*, 1996.

[24] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[25] T. Sørensen, "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.*, vol. 5, pp. 1–34, 1948.

[26] P. Jaccard, "The Distribution of the Flora in the Alpine Zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[27] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[28] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauly, M. J. Franklin, S. Shenker, and I. Stoica, "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, 2012, pp. 15–28.

[29] A. Alexandrov, R. Bergmann, S. Ewen, J.-C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M. J. Sax, S. Schelter, M. Höger, K. Tzoumas, and D. Warneke, "The Stratosphere platform for big data analytics," *The VLDB Journal*, pp. 163–181, 2014.

[30] P. Pecina, "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, 2008, pp. 54–57.

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

[32] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, and P. Koehn, "One billion word benchmark for measuring progress in statistical language modeling." *CoRR*, vol. abs/1312.3005, 2013.

[33] J. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, T. G. B. Team, J. P. Pickett, D. Holberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden, "Quantitative analysis of culture using millions of digitized books," *Science*, 2010.

[34] Y. Lin, J. Michel, E. L. Aiden, J. Orwant, W. Brockman, and S. Petrov, "Syntactic Annotations for the Google Books Ngram Corpus," in *Proceedings of the ACL 2012 System Demonstrations*, ser. ACL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 169–174.

[35] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," in *Proceedings of the 10th International Conference on World Wide Web*, ser. WWW '01. New York, NY, USA: ACM, 2001, pp. 406–414.

[36] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, "Large-scale learning of word relatedness with constraints," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2012, pp. 1406–1414.

[37] W. Yih and V. Qazvinian, "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*, ser. NAACL HLT '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 616–620.

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

[39] Ò. Celma, *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

[40] Ò. Celma and P. Cano, "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, 2008, p. 5.

[41] M. Anisimova and C. Kosiol, "Investigating protein-coding sequence evolution with probabilistic codon substitution models." *Molecular Biology and Evolution*, vol. 26, no. 2, pp. 255–271, 2009.

[42] A. Schneider, G. Cannarozzi, and G. Gonnet, "Empirical codon substitution matrix," *BMC Bioinformatics*, vol. 6, no. 134, 2005.

[43] M. Nirenberg, P. Leder, M. Bernfield, R. Brimacombe, J. Trupin, F. Rottman, and C. O'Neal, "RNA Codewords and Protein Synthesis, VII. On the General Nature of the RNA Code," *Proceedings of the National Academy of Science*, vol. 53, pp. 1161–1168, May 1965.

[44] M. O. Dayhoff and R. M. Schwartz, "Chapter 22: A model of evolutionary change in proteins," in *Atlas of Protein Sequence and Structure*, 1978.

[45] T. D. Wu and D. L. Brutlag, "Discovering empirically conserved amino acid substitution groups in databases of protein families," in *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology, St. Louis, MO, USA, June 12-15 1996*, D. J. States, P. Agarwal, T. Gaasterland, L. Hunter, and R. Smith, Eds. AAAI, 1996, pp. 230–240.

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

[47] D. Bitton, H. Boral, D. J. DeWitt, and W. K. Wilkinson, "Parallel algorithms for the execution of relational database operations," *ACM Transactions in Database Systems*, vol. 8, no. 3, pp. 324–353, Sep. 1983.

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