

LGDE: Local Graph-based Dictionary Expansion

Juni Schindler^{1*}, Sneha Jha², Xixuan Zhang³, Kilian Buehling³, Annett Heft^{3,4}, and Mauricio Barahona^{1*}

¹Department of Mathematics, Imperial College London, United Kingdom
juni.schindler19@imperial.ac.uk, m.barahona@imperial.ac.uk

²Department of Surgery & Cancer, Imperial College London, United Kingdom
sneha.jha20@imperial.ac.uk

³Weizenbaum Institute Berlin, Germany; Institute for Media and Communication Studies, Free University of Berlin, Germany
x.zhang@fu-berlin.de, kilian.buehling@fu-berlin.de

⁴Institute for Research on Far Right Extremism, University of Tübingen, Germany
annett.heft@uni-tuebingen.de

We present Local Graph-based Dictionary Expansion (LGDE), a method for data-driven discovery of the semantic neighborhood of words using tools from manifold learning and network science. At the heart of LGDE lies the creation of a word similarity graph from the geometry of word embeddings followed by local community detection based on graph diffusion. The diffusion in the local graph manifold allows the exploration of the complex nonlinear geometry of word embeddings to capture word similarities based on paths of semantic association, over and above direct pairwise similarities. Exploiting such semantic neighborhoods enables the expansion of dictionaries of pre-selected keywords, an important step for tasks in information retrieval, such as database queries and online data collection. We validate LGDE on two user-generated English-language corpora and show that LGDE enriches the list of keywords with improved performance relative to methods based on direct word similarities or co-occurrences. We further demonstrate our method through a real-world use case from communication science, where LGDE is evaluated quantitatively on the expansion of a conspiracy-related dictionary from online data collected and analyzed by domain experts. Our empirical results and expert user assessment indicate that LGDE expands the seed dictionary with more useful keywords due to the manifold-learning-based similarity network.

* Corresponding authors.

Action Editor: Saif Mohammad. Submission received: 17 July 2024; revised version received: 12 February 2025; accepted for publication: 3 April 2025.

<https://doi.org/10.1162/COLLa.00562>

1. Introduction

Dictionary expansion aims to expand a set of pre-selected keywords by adding related terms that can enhance original queries in information retrieval tasks. Designing a dictionary without in-depth knowledge of the vocabulary in the domain of interest is prone to inaccurate, non-specific, or incomplete results. Therefore, expert-generated seed dictionaries are typically expanded with domain-specific keywords for diverse applications, such as patent searches (Lee et al. 2014), queries of bibliometric databases (Yin et al. 2020) and online forums (Gharibshah et al. 2022), **query expansion** for more effective Web searches (Roy et al. 2016; Kuzi, Shtok, and Kurland 2016), or collecting topic-specific content from social media platforms (Bruns, Harrington, and Hurcombe 2020; Zeng and Schäfer 2021; Klinger et al. 2022; van Atteveldt, van Kuppevelt, and Welbers 2022). Dictionary expansion is particularly relevant, and challenging, in domains with evolving semantics where word choice and language style are highly specialized and diverge from general usage or are constantly in flux to ensure exclusive, community-internal communication, to adjust to ongoing events or the emergence of topics and cultural changes, or to avoid legal prosecution (Heft et al. 2023). Approaches based on large static corpora such as WordNet (Fellbaum 1998) are unable to capture semantic change, which usually requires large diachronic corpora collected over long periods of time and are not ideal for highly dynamic linguistic environments such as social media.

When retrieving information around a certain topic, the challenge becomes to find a “good” dictionary that leads to a corpus containing most documents associated with the topic (high recall) and with few irrelevant documents (high precision). New approaches for data-driven dictionary expansion have leveraged word embedding models to find semantically similar words to pre-selected keywords (Roy et al. 2016; Amsler 2020; Gharibshah et al. 2022; van Atteveldt, van Kuppevelt, and Welbers 2022; Stoll, Wilms, and Ziegele 2023).

In this work, we build on the idea of data-driven dictionaries, but rather than focusing only on words that are most directly similar to pre-selected keywords, we introduce **Local Graph-based Dictionary Expansion (LGDE)**, a method that incorporates tools from manifold learning and network science to capture the complex geometry of the space of word embeddings. Previous studies (Jakubowski, Gasic, and Zibrowius 2020) have suggested that the space of word embeddings is close to a “pinched” manifold within a high-dimensional ambient space. Our methodology contains two main ingredients: the representation of the complex geometry of word vector embeddings through a similarity graph, and the detection of local graph communities around seed keywords using random walks. Specifically, we start with domain-specific word vector embeddings and use the **Continuous k -Nearest Neighbors (CkNN)** geometric graph construction (Berry and Sauer 2019) to create a semantic similarity network of words. We then expand the pre-selected set of keywords by adding the words that belong in their corresponding **semantic communities** as determined by **Severability** (Yu et al. 2020), a fast local community detection method that exploits random walks to explore the local nonlinear geometry of the word graph to find local communities around seed keywords, and thus naturally includes weighted graph paths corresponding to multi-step word associations. Importantly, the CkNN graph provides a consistent description in the sense that its unnormalized graph Laplacian, which determines the properties of graph diffusions, converges to the Laplace-Beltrami operator in the limit of large data. This intrinsic consistency linking geometry and diffusion further motivates our use of severability, a diffusion-based method for local community detection. Severability

detects the semantic community of a seed keyword through the local exploration of the word similarity graph via a random walk. In particular, the semantic community of a given seed keyword is a subgraph in which a random walk starting from the keyword mixes quickly, yet from which it has low probability of escaping. In the context of LGDE, this translates into the random walker quickly discovering words of high similarity and remaining within the set of newly discovered words. Importantly, the random walk explores longer paths, and thus allows the uncovering of longer chains of strong word associations. We show below that, thanks to its foundation on local graph diffusions, severability provides a quantitative advantage over other local community detection algorithms for our task.

To evaluate our method, we consider the task of expanding a dictionary of pre-selected keywords from a human-coded hate speech English-language dataset from the social media platforms Reddit (<https://reddit.com>) and Gab (<https://gab.com>) (Qian et al. 2019). When compared with baseline approaches based on direct word similarities or co-occurrences, LGDE leads to improved dictionary expansion, as measured by a higher F_1 score of discovered words significantly more likely to appear in hate speech-related communication. In addition, to showcase the general-purpose applicability of LGDE, we use a text classification setting on the 20 Newsgroups dataset (Mitchell 1997), also showing an improvement over baseline methods. Finally, we apply our method to a real-world scenario, where LGDE is used to aid subject matter experts in analyzing conspiracy-related posts from the message forum 4chan (<https://4chan.org>) on the “Great Replacement” and “New World Order” conspiracy theories. In this case, LGDE shows a quantitative advantage in discovering additional relevant words that would be missed without the graph-based perspective, as assessed by domain expert-based (blind) annotation.

1.1 Problem Definition

Let us consider a pre-selected list of n keywords $W_0 = \{w_1, w_2, \dots, w_n\}$, denoted the **seed dictionary**, which are known to be relevant to a certain topic. These seed terms are typically derived from expert knowledge, literature research, or existing keyword lists (Gharibshah et al. 2022; Heft et al. 2023). Let us further assume that we have access to a domain-specific corpus of l documents $D = \{d_1, d_2, \dots, d_l\}$ related to the topic of interest, and that each keyword in W_0 is contained in at least one document of D .

The dictionary expansion problem can then be formulated as follows: Expand the seed dictionary W_0 by m new words from the domain-specific corpus D to obtain a data-driven **expanded dictionary** $W = \{w_1, \dots, w_n, w_{n+1}, \dots, w_{n+m}\}$ such that the new words make W “more representative” of the topics of interest (compared to W_0) as measured by evaluation metrics such as the F_1 score that balances precision and recall.

1.2 Related Work

Keyword extraction and query expansion are related tasks but not equivalent to dictionary expansion. The former refers to the extraction of representative keywords from text without an initial seed dictionary (Firoozeh et al. 2020), whereas the latter relies on user input on query words or phrases, possibly expanding the current query term or suggesting additional terms (Schütze, Manning, and Raghavan 2008). Both tasks have been studied mostly in the context of information retrieval in search engines and often involve relevance user feedback on retrieved documents (Zheng et al. 2020). Here, on the other hand, we use semantic relationships already captured in the latent

representation space and focus on generating relevant keyword or query terms based on curated seed keywords, without explicit user feedback.

Early statistical approaches for dictionary expansion were based on ranking candidate words using TextRank (Mihalcea and Tarau 2004), or by analyzing word co-occurrences directly (Yin et al. 2020). Promising recent work has also leveraged pre-trained word embeddings to expand a seed dictionary by choosing the most similar words according to the embeddings (Amsler 2020; Gharibshah et al. 2022; van Atteveldt, van Kuppevelt, and Welbers 2022; Stoll, Wilms, and Ziegele 2023). Prior work also suggests that global word embeddings, such as GloVe (Pennington, Socher, and Manning 2014), may underperform in tasks that benefit from local properties (Diaz, Mitra, and Craswell 2016), which is achieved in our work through the use of local graph techniques for manifold learning and diffusion-based community detection. Farmanbar, Van Ommeren, and Zhao (2020) explore domain-specific query terms but focus on important challenges in end-to-end production pipelines using direct cosine similarities. Most similar to our domain-specific setting, Tsai and Wang (2014) and Gharibshah et al. (2022) use similar methods adapted to custom domains but do not utilize the properties of the semantic network, as we do in our work.

1.3 Motivation

A data-driven augmented dictionary can be constructed by adding words from a domain-specific vocabulary $V \subseteq \mathbb{R}^r$ of word embedding vectors that are most similar to the keywords in the seed dictionary $W_0 \subseteq V$ according to the cosine similarity

$$S_{\cos}(\mathbf{u}, \mathbf{v}) := \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2} \quad (1)$$

for word embeddings $\mathbf{u}, \mathbf{v} \in V$, where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. Then, given a threshold $0 \leq \epsilon \leq 1$, the **thresholding-based** expanded dictionary $W(\epsilon)$ is defined as

$$W(\epsilon) := \bigcup_{\mathbf{w} \in W_0} \{\mathbf{v} \in V \mid S_{\cos}(\mathbf{w}, \mathbf{v}) \geq \epsilon\} \quad (2)$$

Similarly, the **kNN-expanded dictionary** $W(k)$ is defined as

$$W(k) := \bigcup_{\mathbf{w} \in W_0} \{\mathbf{v} \in V \mid \mathbf{v} \in N_k(\mathbf{w})\} \quad (3)$$

where $N_k(\mathbf{w})$ denotes the $k \geq 1$ most similar words to \mathbf{w} according to S_{\cos} . Increasing the parameter ϵ (or k) expands the seed dictionary with most similar words based on direct word similarity. To improve the quality of the expanded dictionary, the word embeddings in V are typically fine-tuned on a domain-specific corpus D to better capture contextual semantics for the particular task at hand.

An issue not easily addressed by such ϵ -thresholding or kNN approaches is that direct (cosine) similarities can be uninformative in noisy, high-dimensional latent space representations, such as word embeddings. It is well known that the **relative contrast**

between closest and furthest neighbors of a set of vectors decreases rapidly for the Euclidean distance (and cosine similarity) as the dimension of the space increases (Aggarwal, Hinneburg, and Keim 2001). Yet, on the other hand, the high-dimensionality of word vector embeddings, typically hundreds of dimensions, is crucial to encode the rich semantic and syntactic characteristics of words learned through training in the large datasets used in textual analysis. This means that there is an implicit tension when using high-dimensional word vector embeddings: Although high dimensional embeddings are desirable to enhance semantic content, high dimensionality also makes embedding vectors less distinguishable, thus leading to relatively unspecific word associations when using direct word similarities.

To help circumvent this conundrum, LGDE uses tools from manifold learning and network science to account for chains of word associations and to better capture the local nonlinear geometry of seed keywords; see Figure 1 for an illustration. In particular, LGDE constructs a graph (a geometric CkNN graph) to represent consistently the geometry of the local manifold of the word embeddings as a means to capture a local representation of the strongest word similarities. This graph is then explored via diffusion to obtain consistent groups of words accessible from the seed dictionary using local community detection. This construction allows us to consider chains of word associations as paths on the graph. Indeed, a seed keyword $w \in W_0$ could be similar to a word $u \in V$ ($S_{\cos}(w, u) > \epsilon$) which is in turn similar to another word $v \in V$ ($S_{\cos}(u, v) > \epsilon$), yet we might have low direct similarity between w and v ($S_{\cos}(w, v) < \epsilon$), reminiscent of cosine similarity not fulfilling the standard triangle inequality (Schubert 2021). A thresholding method based only on direct similarities would then exclude the word v from the data-driven dictionary $W(\epsilon)$, although the chain of strong word associations arguably makes v a sensible candidate for dictionary expansion. Similar problems may also occur when adding the k most similar words for each seed keyword (Equation (3)) or when thresholding is applied iteratively, for example, with the IKEA method (Gharibshah et al. 2022).

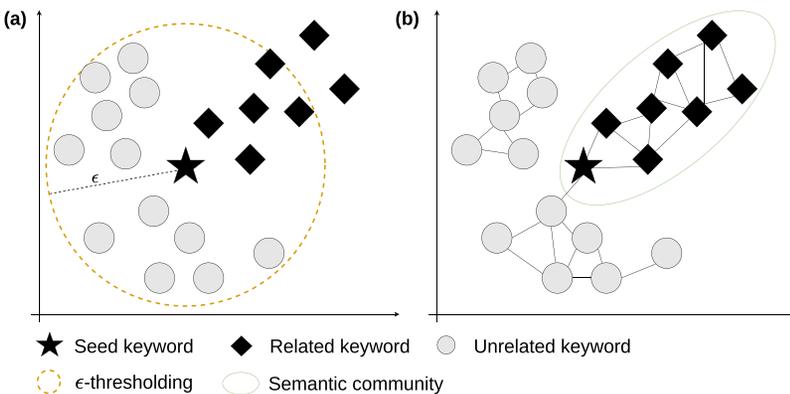


Figure 1

(a) A simple thresholding of the cosine similarity with a fixed radius $0 \leq \epsilon \leq 1$ cannot account for inhomogeneities in the word embedding space. (b) The LGDE method, based on tools from manifold learning and network science, represents consistently the nonlinear local geometry of word embeddings around a seed keyword through a geometric CkNN graph and derives its semantic community through diffusion-based local community detection that captures chains of strong word associations.

2. Methodology

LGDE consists of two steps. In the first step, given a set of word embedding vectors for seed and candidate keywords, we compute a similarity graph that captures the local semantic similarities of the corpus. In the second step, we use local community detection based on graph diffusion to obtain semantic communities of each seed keyword as candidates for dictionary expansion. We detail these steps in the following subsections. We also describe a method to compute domain-specific word representations and a validation strategy for our dictionary expansion problem.

2.1 Semantic Network Construction

To explore the semantic space of our vocabulary $V \subseteq \mathbb{R}^r$ with size $N = |V|$, we construct an undirected, weighted **semantic similarity graph** $G = (V, E)$, where the nodes correspond to the words in V . The weighted edges E are computed from the $N \times N$ matrix of **normalized cosine distances**:

$$\tau := \|1 - S_{\cos}\|_{\max} \tag{4}$$

where S_{\cos} is the cosine similarity (Equation (1)) and $\|\cdot\|_{\max}$ is the element-wise normalization with the max-norm so that all elements of τ are normalized between $[0, 1]$. We also define the matrix of **normalized cosine similarities** (Altuncu et al. 2019):

$$S := 1 - \tau \tag{5}$$

We would like to obtain a semantic network with edges weighted by the similarity S but S is a dense matrix that contains many small values corresponding to negligible word similarities. To produce a robust semantic network that only contains the most relevant chains of semantic similarities, we first obtain the undirected and unweighted CkNN graph from the distance matrix τ (Berry and Sauer 2019). The $N \times N$ adjacency matrix $B^{(k)}$ of the CkNN graph is given by:

$$B_{\mathbf{u},\mathbf{v}}^{(k)} := \begin{cases} 1 & \text{if } \tau(\mathbf{u}, \mathbf{v}) < \delta \sqrt{\tau(\mathbf{u}, \mathbf{u}_k)\tau(\mathbf{v}, \mathbf{v}_k)}, \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\mathbf{u}_k, \mathbf{v}_k \in V$ denote the k -th nearest neighbors of $\mathbf{u}, \mathbf{v} \in V$, respectively, and $\delta > 0$ controls the graph density. In contrast with a **k -Nearest Neighbors** (kNN) graph, which does not account for inhomogeneities in the data as it connects a node to all of its k nearest neighbors, the CkNN construction corrects for different densities and has been shown to approximate consistently the geometry of complex manifolds embedded in a Euclidean space (Berry and Sauer 2019, Theorem 7). Note that τ is equivalent to using Euclidean distances of normalized word vectors when $\delta = 1$. In that case, and assuming $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$, we have:

$$\tau(\mathbf{u}, \mathbf{v}) < \sqrt{\tau(\mathbf{u}, \mathbf{u}_k)\tau(\mathbf{v}, \mathbf{v}_k)} \iff \|\mathbf{u} - \mathbf{v}\|_2 < \sqrt{\|\mathbf{u} - \mathbf{u}_k\|_2 \|\mathbf{v} - \mathbf{v}_k\|_2}$$

which follows directly from:

$$\|\mathbf{u} - \mathbf{v}\|_2^2 = \|\mathbf{u}\|_2^2 + \|\mathbf{v}\|_2^2 - 2\langle \mathbf{u}, \mathbf{v} \rangle = 2(1 - S_{\cos}(\mathbf{u}, \mathbf{v})) = a \tau_{\mathbf{u},\mathbf{v}}$$

where $a := 2 \max_{\mathbf{u}, \mathbf{v} \in V} (1 - S_{\cos}(\mathbf{u}, \mathbf{v}))$ is a constant. Moreover, empirical studies have shown that CkNN with $\delta = 1$ and adequate choice of k outperforms other graph constructions for downstream tasks such as data clustering (Liu and Barahona 2020) and classification (Qian et al. 2021).

Finally, we can define the weighted semantic similarity network $G^{(k)} = (V, E^{(k)})$ with adjacency matrix

$$A^{(k)} := S \odot B^{(k)} \tag{7}$$

where \odot denotes the element-wise (Hadamard) product. Therefore, the edge weights of the semantic network $G^{(k)}$ are given by the normalized semantic similarity S , and its backbone is the sparse CkNN graph $B^{(k)}$ that preserves the topological and local geometric features of the semantic space V .

2.2 Semantic Community Detection

The graph $G^{(k)}$ encodes the semantic information contained in our domain-specific corpus D at a word-level representation with the inter-word weighted edges $E^{(k)}$ capturing relevant semantic similarities between words. Hence, paths in this graph can be interpreted as chains of word associations. Moreover, the keywords in the seed dictionary W_0 are nodes in the graph and we can study their context and related words by computing their local semantic community structure. In network science, **community** refers to a group of nodes in the graph that are tightly connected within the group and more loosely connected with the other communities. **Community detection** is then an unsupervised task to find such communities in graph data without specifying the number or type of communities. The result is a partitioning of the graph into mutually exclusive subgraphs (hard partitions) or into potentially overlapping subgraphs (soft partitions). See Fortunato (2010) for a review and Schaub et al. (2017) and Schindler and Fuller (2023) for a discussion of the notion of community. Here we use the **severability** method, a diffusion-based framework for fast local community detection (Yu et al. 2020), and determine the **semantic community** $C(\mathbf{w})$ for each seed keyword $\mathbf{w} \in W_0$. Briefly, severability is a community detection algorithm that detects local graph communities by exploiting a discrete-time random walk on the graph of words, which is governed by the transition probability matrix P , where $P_{\mathbf{u}, \mathbf{v}}$ is the probability of the random walk jumping from word $\mathbf{u} \in V$ to $\mathbf{v} \in V$:

$$0 \leq P_{\mathbf{u}, \mathbf{v}} := \frac{A_{\mathbf{u}, \mathbf{v}}^{(k)}}{\sum_{\mathbf{v}' \in V} A_{\mathbf{u}, \mathbf{v}'}^{(k)}} \leq 1 \tag{8}$$

The semantic community $C(\mathbf{w})$ of $\mathbf{w} \in W_0$ is the subset $C \subseteq V$ (with $\mathbf{w} \in C$) that maximizes the severability quality function $\sigma(C, t)$ for the time scale t :

$$0 \leq \sigma(C, t) := \frac{\rho(C, t) + \mu(C, t)}{2} \leq 1 \tag{9}$$

The severability $\sigma(C, t)$ consists of two terms: the **mixing term** $\mu(C, t)$, which measures the mixing of the random walker within C over time t , and the **retention term** $\rho(C, t)$, which quantifies the probability of the random walker not escaping C by time t (Yu et al. 2020). A semantic community $C(\mathbf{w})$ of seed keyword \mathbf{w} has high severability when

a random walker starting from \mathbf{w} , and observed over time horizon t , quickly explores all words within the community, due to the existence of paths of high semantic similarity, and remains within that community over a long time, that is, it exits the community with low probability over time t .

Specifically, let Q denote the (substochastic) submatrix of P corresponding to the nodes in a connected subset $C \subseteq V$ and let $\mathbf{q}_i^{(t)}$ denote the i -th row of the matrix Q^t . Then Yu et al. (2020) define the retention term as:

$$0 \leq \rho(C, t) := \frac{1}{|C|} (\mathbf{1}^T Q^t \mathbf{1}) \leq 1 \tag{10}$$

where $\mathbf{1}$ is the $|C|$ -dimensional vector of ones, and the mixing term as:

$$0 \leq \mu(C, t) := 1 - \frac{1}{|C|} \sum_{i=1}^{|C|} \left\| \bar{\mathbf{q}} - \frac{\mathbf{q}_i^{(t)}}{\mathbf{q}_i^{(t)} \mathbf{1}} \right\|_{TV} \leq 1 \tag{11}$$

where $\|\cdot\|_{TV}$ is the total variation distance that measures the convergence towards $\bar{\mathbf{q}}$, the quasi-stationary distribution of Q . As t increases, we need a larger-sized community $C \subseteq V$ to trap the random walk and achieve high retention $\rho(C, t)$, but simultaneously, increasing the size of C makes mixing more difficult and leads to a reduced $\mu(C, t)$. We then optimize the severability quality function to obtain local semantic communities around the seed keywords based on the derived semantic similarity word graph. To indicate the dependency of the semantic community of word \mathbf{w} on the temporal scale t and the CkNN parameter k we write $C^{(k,t)}(\mathbf{w})$. Note that severability captures chains of word associations through paths of length larger than two in the random walk and also allows for overlapping semantic communities (potentially capturing polysemy, if present). Code for optimizing the severability quality function and performing local community detection is available at <https://github.com/barahona-research-group/severability>.

The result of the LGDE method is then an extended dictionary $W(k, t)$, which is defined as the union of (overlapping) local semantic communities:

$$W(k, t) := \bigcup_{\mathbf{w} \in W_0} C^{(k,t)}(\mathbf{w}) \tag{12}$$

By construction $W_0 \subseteq W(k, t) \subseteq V$, and the size of $W(k, t)$ generally grows with increasing k and t such that in the set-theoretic limit, we have $\lim_{k,t \rightarrow \infty} W(k, t) = V$, that is, the extended dictionary would become the whole vocabulary. As remarked above, our extended dictionary can include words that are connected to a seed keyword $\mathbf{w} \in W_0$ via a chain of strong word associations, rather than only by strong direct similarity.

The computational complexity of LGDE depends on the vocabulary size N , the number of neighbors k in the CkNN construction, the time scale t of severability, and the size n of the seed dictionary W_0 . The complexity of the CkNN construction is $O(N^2k)$, as it is dominated by the $\tau(\mathbf{u}, \mathbf{v})$ rescaling in Equation (6), which requires finding the k -th nearest neighbor (out of N words) for each of the N words. The complexity of severability is linear in n with $O(nb \log_2 t)$, where b is a constant that depends on the maximum local community size allowed in the optimization algorithm (Yu et al. 2020). Hence, the overall complexity for a single run of LGDE is $O(N^2k + nb \log_2 t)$.

In subsequent runs, for example, when adding a new word to the seed dictionary, no additional graph construction is required and only severability has to be applied with complexity $O(b \log_2 t)$ for a single keyword. The favorable computational complexity of LGDE thus follows from the use of fast local community detection.

2.3 Domain-specific Word Representations

To construct the input space V , we use word embeddings generated from general-purpose corpora as vector inputs. In our setting of community detection on a word similarity graph, we wish to allow a word to belong to multiple, overlapping communities, to be able to capture multiple word meanings if they exist. Hence, we use static word embeddings such as word2vec (Mikolov et al. 2013) or GloVe (Pennington, Socher, and Manning 2014) instead of contextualized embeddings such as BERT (Devlin et al. 2019) or ELMo (Peters et al. 2018). It may be worth noting that BERT-based (Devlin et al. 2019) models rely on subword tokenization and, typically, word embeddings are extracted by summing or averaging the subword token embeddings, a heuristic that often degrades the word-level semantic graph representation (Altuncu 2020). This follows from the fact that BERT-style models are designed for supervised end-to-end fine-tuning, and not for extracting intermediate embedding layers. For instance, a Semeval task on diachronic semantics (Schlechtweg et al. 2020) showed that static or type-based embeddings outperformed contextualized embeddings. Furthermore, studies on representations from BERT-like models (Vulić et al. 2020) have shown various issues with representations produced by averaging hidden layers. These are highly task- and domain-dependent and there are no techniques to select a single layer that is reliably better (Bommasani, Davis, and Cardie 2020). Work on hidden layer selection for good word-level representations may be an interesting future direction that could complement our work here but is beyond the focus of the present study.

For our experiments, we use GloVe as base word embeddings, generated from general-purpose corpora like Wikipedia 2014 and Gigaword 5 (Pennington, Socher, and Manning 2014). The base GloVe embeddings are available in different dimensions $r = 50, 100, 300$. It is well known that word embeddings are dependent on the corpus they are trained on, given that terms often adopt new or additional meanings across domains or over time (e.g., “Apple” as a concept in technology or business domains that came into existence only in the 1970s). We use Mittens (Dingwall and Potts 2018) to tune the GloVe representations to better represent the semantic relationships in our use-case domains. However, note that base embeddings could be used as-is with our method. In particular, for a set of N words in our domain-specific corpus D , the word embeddings \mathbf{v}_i , $i = 1, \dots, N$, are computed from a retrofitting model that optimizes the Mittens cost function

$$J_{\text{Mittens}} = J_{\text{GloVe}} + \mu \sum_{i \in U} \|\mathbf{v}_i - \mathbf{u}_i\|_2 \quad (13)$$

where J_{GloVe} is the standard GloVe cost function (Pennington, Socher, and Manning 2014) based on the word co-occurrences in D , and U is the index set of words for which pre-trained vector representations \mathbf{u}_i are available. The hyperparameter $\mu \geq 0$ determines the extent of fine-tuning: A large μ favors remaining closer to the base embeddings \mathbf{u}_i , whereas a small μ favors greater adaptation of the word representations \mathbf{v}_i , such that setting $\mu = 0$ means that the word vectors are computed from scratch on the new

corpus. Although a value of $\mu = 0.1$ was the default in Dingwall and Potts (2018), we find that a larger value of μ improves the quality of embeddings for a small domain-specific corpus D , as is the case in our study. By training the Mittens model, we compute fine-tuned r -dimensional word vectors $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\} \subset \mathbb{R}^r$, for $r = 50, 100, 300$, and we assume $W_0 \subsetneq V$. Following Schnabel et al. (2015), we normalize the word vectors to unit length, as the relative angle is sufficient to capture semantic similarity.

2.4 Evaluation of Expanded Dictionaries

Consider a domain-specific corpus of l documents $D = \{d_1, \dots, d_l\}$ with ground-truth (human-coded) labels \mathbf{y} such that $\mathbf{y}_i = 1$ if document d_i has a certain topic of interest (true) and $\mathbf{y}_i = 0$ otherwise (false). To assess the quality of a data-driven dictionary W we can evaluate the performance of a simple document classifier $f_W : D \rightarrow \{0, 1\}$ associated with W , where we define $f_W(d) = 1$ if there exists a keyword in W that appears in the document $d \in D$ and $f_W(d) = 0$ otherwise. To evaluate the trade-off between precision and recall of the dictionary W on the potentially unbalanced benchmark data (D, \mathbf{y}) , we compute the macro F_1 score of its associated classifier f_W and denote this number by $F_1[W]$. Similarly, $P[W]$ denotes the macro precision and $R[W]$ the macro recall. This evaluation strategy can also be used to optimize the hyperparameters ϵ , in the case of the threshold dictionary $W(\epsilon)$ in Equation (2), or k, t , in the case of the LGDE dictionary $W(k, t)$ in Equation (12), on a train split of the benchmark data.

In addition, it is also possible to evaluate the contribution of a single keyword $\mathbf{w} \in W$ to the performance of the dictionary W . Let us consider the probability $\mathbb{P}[\mathbf{w} \in d_i \mid \mathbf{y}_i = 1]$ that the word \mathbf{w} appears in a true document and $\mathbb{P}[\mathbf{w} \in d_i \mid \mathbf{y}_i = 0]$ that it appears in a false document. Then we define the likelihood ratio (LR) (van der Helm and Hische 1979) of word \mathbf{w} as

$$\text{LR}(\mathbf{w}) := \frac{\mathbb{P}[\mathbf{w} \in d_i \mid \mathbf{y}_i = 1]}{\mathbb{P}[\mathbf{w} \in d_i \mid \mathbf{y}_i = 0]} \quad (14)$$

which is larger than 1 if \mathbf{w} is more likely to appear in true documents than in false ones. Words with larger LR can thus be considered more representative of true documents in the text or corpus under study. The median LR for all words $\mathbf{w} \in W$ denoted by $\text{LR}[W]$ can be used to summarize the contributions of keywords to the performance of the dictionary.

3. Experiments

We evaluate the performance of LGDE on two benchmark datasets and compare it to four baseline methods: (i) thresholding (Equation (2)); (ii) kNN (Equation (3)); (iii) IKEA, where the seed dictionary is expanded iteratively adding words with average similarity to seed and previously discovered keywords larger than a threshold until convergence (Gharibshah et al. 2022); and (iv) TextRank (Mihalcea and Tarau 2004), a popular keyword extraction method that retrieves the k top-ranked words according to the PageRank centrality in a word co-occurrence graph with co-occurrence window sized w .

3.1 Application to a Benchmark Hate Speech Dataset

In the first experiment, we apply dictionary expansion to a seed dictionary of the most frequent hate speech-related words in a dataset collected from online platforms.

3.1.1 Data. We use a benchmark dataset collected by Qian et al. (2019) from Reddit and Gab consisting of 56,099 manually annotated hate speech posts, of which 19,873 (35.4%) are categorized as “hate speech” according to Facebook’s definition (Qian et al. 2019, p. 4758). The data is available at: <https://github.com/jing-qian/A-Benchmark-Dataset-for-Learning-to-Intervene-in-Online-Hate-Speech>. We split the data into train data (75%) and test data (25%) using stratification. Following Qian et al. (2019), we choose our seed dictionary W_0 as the five most frequent keywords in the benchmark dataset: “ni**er”, “fa**ot”, “ret**d”, “ret***ed”, and “c**t” (unmasked keywords in Table 13). The task is to expand this seed dictionary with additional hate speech-related keywords.

3.1.2 Experimental Setup. As outlined in Section 2.3, we compute domain-specific word embeddings $V \subseteq \mathbb{R}^r$, for all words appearing in at least 15 posts but in no more than 80% of the posts. This selection criterion filters out stopwords, infrequent misspellings, and rare words, and leads to a set with $N := |V| = 7,093$ words. We found that choosing a relatively large $\mu = 1.0$ in Equation (13) improves word embeddings (i.e., embeddings for out-of-vocabulary words are learned while retaining the quality of pre-existing word embeddings) for various embedding dimensions $r = 50, 100, 300$. From the semantic space V , we compute the $N \times N$ matrix of normalized cosine similarities S (Equation (5)). Finally, for each dimension r , we compare the expansion of the seed dictionary W_0 into the thresholding- (Equation (2)), kNN- (Equation (3)), IKEA- and TextRank-based expanded dictionaries against the LGDE dictionary $W(k, t)$ (Equation (12)). For this dictionary expansion task, TextRank is only applied to documents in the training corpus with at least one seed keyword and we consider context windows $2 \leq w \leq 10$.

As shown in Figure 2, LGDE consistently outperforms the other baseline methods across different sizes of the expanded dictionary. To facilitate proper comparisons, and to control for the size of the different dictionary expansion methods, we require that expanded dictionaries include at least 30 keywords but no more than 50 (with the exception of IKEA where the maximum is set to 60 due to the large gap in attained dictionary sizes). We then optimize the hyperparameters of LGDE and the other baseline methods on the train split of the benchmark data (see Table 1 for optimal hyperparameters and Figure 6 in the Appendix for additional information).

3.1.3 Results. Table 2 shows that the optimal LGDE dictionaries $W(k, t)$ outperform the optimal dictionaries obtained with four baseline methods for all dimensions r of the word embeddings, as measured by increased F_1 scores. The quality of the LGDE dictionary improves with increasing embedding dimension and the overall best dictionary is achieved with LGDE at dimension $r = 300$. On the other hand, the dictionaries based on thresholding, kNN, IKEA, and TextRank do not improve upon the performance of the seed dictionary W_0 , even when increasing the embedding dimension because of the inclusion of many non-relevant words.

A qualitative assessment of the discovered words (Table 3) shows that LGDE discovers keywords relevant to hate speech documents, whereas thresholding often produces expected offensive, but largely standard, words without bringing in new semantic context. Examples of relevant keywords only found by LGDE include: “be**er”, a racist

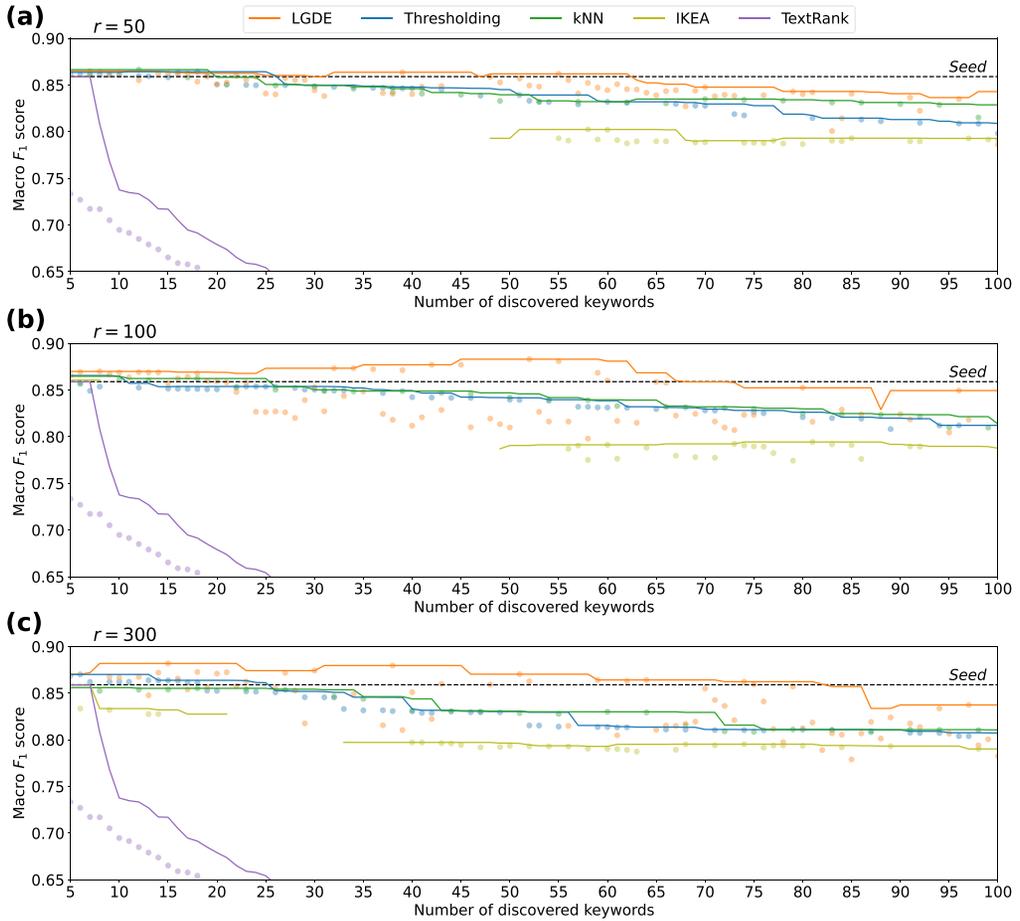


Figure 2 Hate speech benchmark dataset. Train macro F_1 scores for LGDE, thresholding, kNN, IKEA, and TextRank dictionaries computed from word embedding dimensions (a) $r = 50$, (b) $r = 100$, and (c) $r = 300$. Solid lines are F_1 scores max-pooled with window size 15 for each method, whereas the dotted line is the train F_1 score of the seed dictionary for comparison. LGDE scores are consistently higher across dictionary sizes.

Table 1 Optimal hyperparameters for LGDE and four baselines in hate speech and 20 Newsgroups experiments. In the 20 Newsgroups experiment, no hyperparameters were found for IKEA that could produce a dictionary within our size constraints (100–150 discovered words).

	LGDE	Thresholding	kNN	IKEA	TextRank
Hate speech, $r = 50$	$k = 13, t = 2$	$\epsilon = 0.774$	$k = 7$	$\epsilon = 0.701$	$w = 2, k = 35$
Hate speech, $r = 100$	$k = 12, t = 5$	$\epsilon = 0.681$	$k = 8$	$\epsilon = 0.610$	$w = 2, k = 35$
Hate speech, $r = 300$	$k = 19, t = 2$	$\epsilon = 0.552$	$k = 8$	$\epsilon = 0.507$	$w = 2, k = 35$
20 Newsgroups, $r = 50$	$k = 3, t = 8$	$\epsilon = 0.891$	$k = 7$	–	$w = 2, k = 106$
20 Newsgroups, $r = 100$	$k = 5, t = 8$	$\epsilon = 0.807$	$k = 7$	–	$w = 2, k = 106$
20 Newsgroups, $r = 300$	$k = 6, t = 6$	$\epsilon = 0.673$	$k = 8$	–	$w = 2, k = 106$

Table 2

Hate speech benchmark dataset. Test macro F_1 scores of the seed dictionary and expanded dictionaries obtained with LGDE and four baseline methods (thresholding, kNN, IKEA, TextRank) for different embedding dimensions r .

r	Seed dictionary	LGDE	Thresholding	kNN	IKEA	TextRank
50	0.856	0.861	0.852	0.850	0.800	0.607
100	0.856	0.874	0.850	0.848	0.787	0.607
300	0.856	0.875	0.846	0.846	0.794	0.607

Table 3

Hate speech benchmark dataset. LGDE- and thresholding-based discovered dictionaries for different embedding dimensions r . Words in bold have a large likelihood ratio with $\text{LR}[w] \geq 2$. See Table 13 in the Appendix for unmasked keywords.

r	LGDE only	Intersection	Thresholding only
50	alright, awww , baiting, be**er , braindead , brainless , btw, commie , commies , c**ts , f*g , fa**ts , f**ker , gg, goyim , h*g , hater, liar, limey, mo*on, pedo , simp, sk**k , s**ts , snowflake, tl, tr**ny , wetback , woah	as**le, aww , crybaby, ni**a , ni**ers , prob , sc**ag, slur, spouting , weasel	autistic , b**ch , bu****it, cared, competent, convicted, f**k , insane, intellectually , liking , mentally , morally, offenders, olds, paki , puke, ret**ds , sane, ugh, yikes
100	as**les, awww , badass , be**er , betas, b**bs , btw, chi**an , commies , c**ts , douche , fa**ts , f**kers , f**ks , goyim , ho*os, limey, maggot , motherf**ker , motherf**kers , pedo , pilled , prob , ra**ead , shitty, simp, sk**k , spangle, stfu , th*t, tw*t , ummm, walkaway, wi**er , woah	b**ch , braindead , commie , f*g , f**kin , ni**a , ret**ds , shite, turd , wetback	as**le, autistic , cared, competent, crybaby, engineers, f**k , f**king , hillbilly, honest, incompetent, intellectually , mentally , morally, murderers , ni**ers , offenders, slur, weasel, w**re
300	be**er , braindead , brainwashing, btw, commies , c**ts , douche , fa**ts , foxnews, f**ker , f**kers , goalposts, goyim , ho*os, misandry, motherf**ker , motherf**kers , pedo , potus, ra**ead , sc**ag, shite, sk**k , stfu , sucks, th*t, tr**ny , tw*t , ummm, wetback , wi**er , woah	aww , awww , ni**a , ret**dation , ret**ds , simp	ahhh, autistic , bi**h , cared, competent, convicted, cruel, derogatory, disabled, execute , executed, execution, f*g , f**k , f**king , incompetent, mental, mentally , mildly, murderers , ni**ers , offenders, slur, ugh, word

slur for Hispanic men, “h*g”, a misogynist slur for older women, and “tr**ny”, a transphobic slur for a transgender people (see Table 13 for unmasked keywords). Such terms, including neologisms and online slang, are part of an informal jargon and potentially difficult to anticipate without being part of the relevant online community. To discover these terms using thresholding—chosen for comparison here as the best-performing baseline method—requires choosing very small values of ϵ at the price of adding irrelevant terms to $W(\epsilon)$, so that the overall F_1 performance is reduced. For example, discovering the term “tr**ny” at $r = 50$ requires $\epsilon \leq 0.719$ such that $|W(\epsilon = 0.719)| = 192$ with $F_1 = 0.767$, as compared to the optimal LGDE score $F_1 = 0.861$, and at $r = 300$ it

Table 4

Hate speech benchmark dataset. Pairwise comparisons of median likelihood ratio LR (Equation (14)) of words discovered only by LGDE *vs.* median LR of words discovered only by each of the baseline methods (thresholding, kNN, IKEA, TextRank) for different embedding dimensions, r . The asterisks indicate that the median LR of LGDE-only words is significantly higher than baseline methods-only words (ns means $p \geq 0.05$; * means $p < 0.05$; ** means $p < 0.01$; *** means $p < 0.001$; **** means $p < 0.0001$; Mann-Whitney U test).

r	LGDE	Thresholding	LGDE	kNN	LGDE	IKEA	LGDE	TextRank
50	2.25 (*)	1.45	2.23 (ns)	1.91	2.28 (*)	1.41	2.28 (****)	1.00
100	2.59 (*)	1.53	2.59 (**)	1.46	2.73 (***)	1.44	2.79 (****)	1.03
300	2.66 (*)	1.25	2.73 (ns)	1.55	2.97 (**)	1.44	2.97 (****)	0.98

requires $\epsilon \leq 0.485$ such that $|W(\epsilon = 0.485)| = 263$ with $F_1 = 0.741$, as compared to the optimal LGDE score $F_1 = 0.875$ (Table 2).

To further evaluate these differences, Table 4 shows that the median LR for words discovered only by LGDE is significantly higher than the median LR for words discovered only by thresholding, the best-performing baseline method: $LR[W(k, t) \setminus W(\epsilon)] > LR[W(\epsilon) \setminus W(k, t)]$. This result is statistically significant for all dimensions ($p < 0.05$, Mann-Whitney U test) indicating that LGDE discovers words more representative of hate speech-related content. Similarly, we find that the median LR for words discovered only by LGDE is higher (and almost always significantly so) than the median LR for words discovered by each of the other baseline methods kNN, IKEA, and TextRank (Table 4).

3.2 Application to a Benchmark 20 Newsgroups Dataset

To demonstrate the broader applicability of LGDE as a dictionary expansion method, we extend our evaluation to the 20 Newsgroups dataset in a second experiment.

3.2.1 Data. We use the popular 20 Newsgroups dataset (Mitchell 1997), comprising 18,846 posts from 20 newsgroups on different topics split into train (60%) and test data (40%). Following Keerthi and DeCoste (2005), we create a size-balanced two-class variant of the dataset with the positive class consisting of the 10 newsgroups with names from `sci.*`, `comp.*`, or `misc.forsale` and the negative class consisting of the other 10 groups with names from `alt.atheism`, `rec`, `soc.religion.christian`, and `talk.politics`. As our seed dictionary W_0 we choose the 16 words that appear in the group names of the groups in the positive class: “computer”, “cryptography”, “electronics”, “graphics”, “hardware”, “ibm”, “mac”, “medicine”, “microsoft”, “os”, “pc”, “sale”, “science”, “space”, “sys”, “windows”.

3.2.2 Experimental Setup. We compute domain-specific GloVe word embeddings $V \subseteq \mathbb{R}^r$, for the words appearing in at least 15 posts and filter out stopwords to obtain a set of $N := |V| = 10,751$ words. Due to the more standard semantic relationships in the 20 Newsgroups dataset, we chose $\mu = 5.0$ in Equation (13) for various embedding dimensions $r = 50, 100, 300$. We then repeat the previous experimental setup for dictionary expansion as described in Section 3.1.2. Again, we find that LGDE consistently outperforms the other baseline methods across different sizes of the expanded dictionary (Figure 3). Reflecting the larger seed dictionary size in this application, we require that

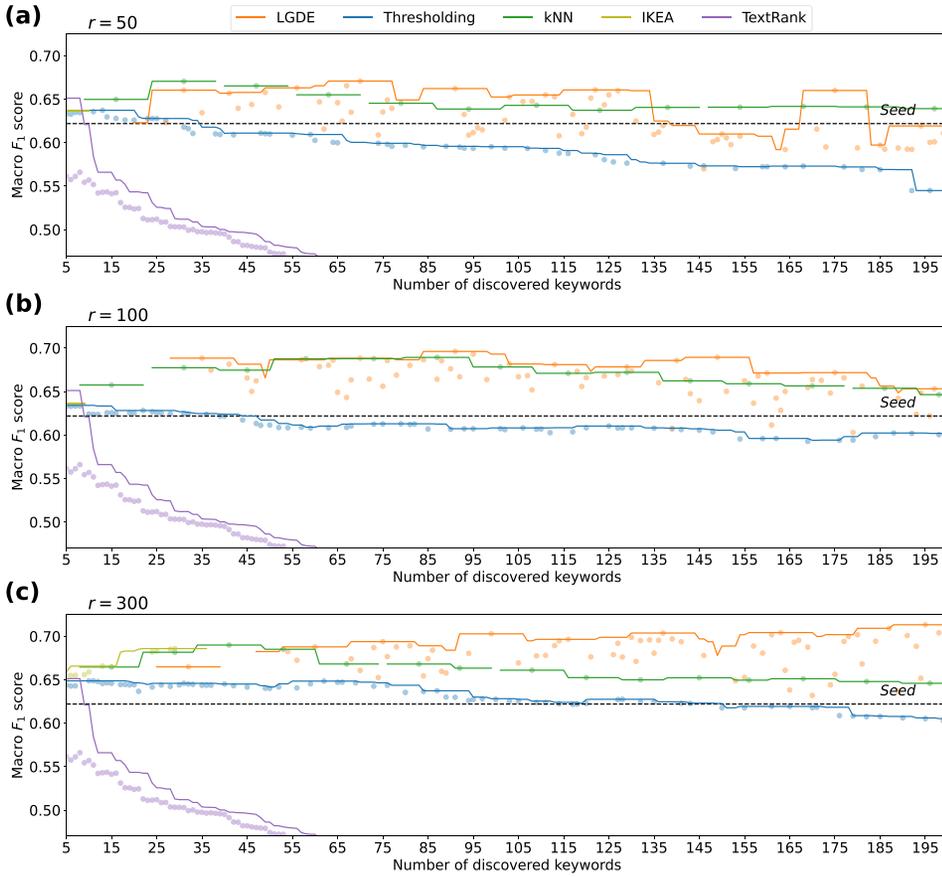


Figure 3 20 Newsgroups benchmark dataset. Train macro F_1 scores for LGDE, thresholding, kNN, and TextRank dictionaries computed from word embedding dimensions (a) $r = 50$, (b) $r = 100$, and (c) $r = 300$. Solid lines are F_1 scores max-pooled with window size 15 for each method, and the dotted line is the train F_1 score of the seed dictionary for comparison. LGDE scores are consistently higher across dictionary sizes. Note that the IKEA method does not produce dictionaries with sizes between 100 and 150 and is thus excluded from this comparison.

expanded dictionaries include at least 100 keywords but no more than 150. Optimal hyperparameters for the different methods can be found in Table 1, and Figure 7 in the Appendix contains additional information. Note that the IKEA method only produced dictionaries of size smaller than 30 or larger than 800 (see Figure 3) which were not deemed reasonable for the use case. Hence, we exclude IKEA from further comparison in this example.

3.2.3 Results. Table 5 shows that the optimal LGDE dictionaries $W(k, t)$ outperform the other baseline methods across all dimensions r as measured by increased F_1 scores. Again, the performance of LGDE improves with increased dimension and the overall best performance is achieved with LGDE at dimension $r = 300$. Again we find that the thresholding-based dictionaries do not improve upon the performance of the bare seed dictionary W_0 . The kNN-based dictionaries can improve upon W_0 but still lead to lower performance than LGDE.

Table 5

20 Newsgroups benchmark dataset. Test macro F_1 scores of the seed dictionary and expanded dictionaries obtained with LGDE and baseline methods (thresholding, kNN, TextRank) for different embedding dimensions r .

r	Seed dictionary	LGDE	Thresholding	kNN	TextRank
50	0.621	0.653	0.580	0.636	0.437
100	0.621	0.692	0.587	0.674	0.437
300	0.621	0.694	0.618	0.649	0.437

Table 6

20 Newsgroups benchmark dataset. Pairwise comparisons of median likelihood ratio LR of words discovered only by LGDE *vs.* median LR of words discovered only by baseline methods (thresholding, kNN, TextRank) for different embedding dimensions, r . Statistical significance according to Mann-Whitney U test is indicated by asterisks, as in Table 4.

r	LGDE	Thresholding	LGDE	kNN	LGDE	TextRank
50	2.15 (**)	0.92	1.61	1.99 (*)	2.02 (ns)	1.17
100	4.98 (****)	1.47	3.98 (ns)	2.57	4.60 (****)	1.17
300	11.95 (****)	1.33	9.20 (**)	2.53	10.12 (****)	1.15

We also find that words discovered only by LGDE have a significantly higher LR than words discovered only by thresholding across all dimensions (Table 6). Relative to TextRank, LGDE leads to a significantly higher LR at higher dimensions $r = 100, 300$, whereas relative to kNN, LGDE only leads to a significantly higher LR at dimension $r = 300$ ($p < 0.01$), where the advantages of LGDE are more apparent. Overall, these results demonstrate that LGDE is an effective method of discovering relevant words for expansion above baseline methods, particularly at higher embedding dimensions that capture more semantic information from text.

Finally, in Figure 4 we examine the relationship between the quality of the seed keyword and the quality and size of its semantic community. We find a positive Pearson correlation (ρ) between the LR of the seed keyword and the mean LR of the words in their semantic community, which becomes stronger as the embedding dimension grows—it increases from 0.330 for $r = 50$ to 0.960 for $r = 300$. On the other hand, as r grows, there is also an increasingly negative Pearson correlation between the LR of the seed keyword and the size of their semantic community— ρ becomes negative with value -0.114 for $r = 100$ and decreases further to -0.499 for $r = 300$. This means that high quality seed keywords tend to retrieve a small number of high-performing discovered words, whereas lower quality keywords generate larger, more generic semantic communities with lower-performing discovered keywords. Note that the seed dictionary in our 20 Newsgroups experiment used solely words found in the class labels; our analysis thus highlights the importance of carefully selecting the seed dictionary.

4. Comparison of Local Community Detection Methods

In this section, we examine the use of severability as a local community detection method in the LGDE pipeline. As described in the Introduction, severability complements consistently the CkNN construction of semantic networks as they are both

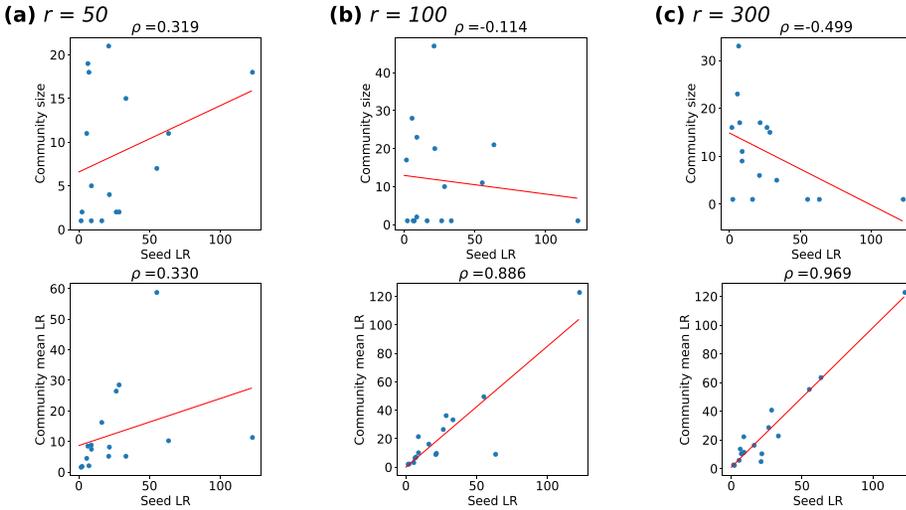


Figure 4 20 Newsgroups benchmark dataset. Scatter plot and Pearson correlation (ρ) between the likelihood ratio LR of the 16 seed keywords and the size (top) or mean LR (bottom) of their LGDE semantic community for different embedding dimensions: (a) $r = 50$, (b) $r = 100$, and (c) $r = 300$. Red solid lines are least-squares fits. As r increases, the correlation between the LR of the seed keyword and the size of its community becomes increasingly negative, whereas the correlation between the LR of the seed keyword and the mean LR of its community becomes increasingly positive. Hence, for larger embedding dimensions, LGDE finds smaller but more specific semantic communities for high-quality seed keywords.

underpinned by graph diffusions, which we use as our basis for dictionary expansion. Whereas most community detection methods are global in nature, we have compared to popular local community detection methods (Baltsou, Christopoulos, and Tsihlas 2022): **local modularity** R (Clauset 2005), **local modularity** M (Luo, Wang, and Promislow 2006), and **SIWO** (Zafarmand et al. 2023). However, these methods face several limitations in the context of dictionary expansion. Firstly, local modularity R and M are defined for unweighted graphs and thus fail to capture the weighted semantic similarity in our word network. Secondly, the fact that the optimization of local modularity R and M is not deterministic (i.e., seed dependent) introduces additional, undesired variability in the dictionary expansion task. Thirdly, these algorithms exploit combinatorial features (density of edges for local modularity R and M or triangles for SIWO) to identify a local community, thus neglecting paths and the ensuing chains of strong word associations. Finally, these methods are parameter-free and return fixed-size local communities that do not accommodate naturally the different sizes of local communities. Our experiments confirm that severability outperforms the three other local community detection methods for our dictionary expansion task in both the benchmark hate speech dataset and the 20 Newsgroups dataset, as shown in the consistently higher F_1 scores in Table 7.

5. Application to Conspiracy-related Content on 4chan

As an application to a real-world use case, we apply LGDE to the problem of collecting conspiracy-related 4chan posts. All content on 4chan is ephemeral and, together with

Table 7

Test macro F_1 scores of the seed dictionary compared to expanded dictionaries obtained with LGDE (CkNN+severability) and CkNN with three standard local community detection methods (SIWO, local modularity R , local modularity M) for different embedding dimensions r : (a) hate speech benchmark dataset; (b) 20 Newsgroups benchmark dataset. Because the optimization of local modularity R and M is not deterministic, we repeat the computations with 50 different seeds and present the average score and 95% confidence intervals for dictionaries within the admissible size range.

(a) Hate speech benchmark dataset					
r	Seed dictionary	LGDE (CkNN + severability)	CkNN + SIWO	CkNN + Mod R	CkNN + Mod M
50	0.856	0.861	0.859	0.8272 (0.8270–0.8275)	0.835 (0.833–0.836)
100	0.856	0.874	0.827	0.8505 (0.8493–0.8516)	0.8597 (0.8594–0.8600)
300	0.856	0.875	0.828	0.852 (0.851–0.853)	0.862 (0.861–0.863)
(b) 20 Newsgroups benchmark dataset					
r	Seed dictionary	LGDE (CkNN + severability)	CkNN + SIWO	CkNN + Mod R	CkNN + Mod M
50	0.621	0.653	0.616	0.606 (0.604–0.608)	0.634 (0.632–0.637)
100	0.621	0.692	0.648	0.6505 (0.6502–0.6508)	0.659 (0.658–0.660)
300	0.621	0.694	0.664	0.669 (0.666–0.673)	0.663 (0.661–0.666)

complete user anonymity and the absence of content moderation, a highly vernacular user culture has developed, partly characterized by racist and misogynous content (De Zeeuw and Tuters 2020). We follow the definition of conspiracy-related content in Heft et al. (2023, p. 3) as “the full public communication *on* and *about* (alleged) conspiracy theories in various communication venues, including their narrations, counter-narrations, and debunking, as well as neutral observation forms”. Detecting such content can be challenging as participants use slang and insider humor to explicitly distinguish themselves from a perceived out-group (De Zeeuw et al. 2020). Therefore, terms used in public debate or the scientific literature might deviate from the vocabulary used by 4chan users. Furthermore, the vocabulary and the narratives change over time as new events lead to adaptations of existing narratives or new theories emerge and are then included in the existing canon (Garry et al. 2021; Heft et al. 2023). Therefore, relying solely on pre-existing domain knowledge in such an open-ended, evolving domain often leads to high-precision but low-recall retrieval methods. In particular, using only literature-based dictionaries to retrieve conspiracy-related posts from 4chan could miss neologisms and creative composition of existing concepts, leading to incomplete collections of longitudinal datasets of relevant user posts. Improving the recall is thus crucial for data collection in communication science and requires improved methods of dictionary expansion. Below, we show that starting from an expert-selected seed dictionary, LGDE discovers new conspiracy-related words that would be missed without a local graph-based perspective.

5.1 Data

We assemble an *initial* seed dictionary \tilde{W}_0 with 215 keywords representative of two conspiracy theories (“Great Replacement” and “New World Order”) based on the RPC-Lex dictionary (Puschmann et al. 2022) and other literature (full list in Table 11 in Appendix). Using the fourPlebsAPI (Buehling 2022), we collect all English language posts published in 22 sample weeks (2 weeks in each year from 2011 to 2021; see Table 12 in Appendix) on 4chan’s political discussion board /po1/ (<https://boards.4chan.org/po1/>) containing at least one of the words in \tilde{W}_0 . This leads to a corpus D with 102,058 unique documents. Because many conspiracy-related keywords are multi-word phrases (e.g., “great replacement”), we pre-process the input to include hyphenated terms and noun phrases. In particular, we first detect multi-word phrases in the corpus that are (1) seed keywords, (2) hyphenated terms, or (3) noun phrases with spaCy (Honnibal et al. 2020) and then make the phrases uniform by lemmatizing their tokens that are nouns, verbs, adjectives, and adverbs. We then use Stanza (Qi et al. 2020) for further pre-processing of the data.

We prepared two human-coded benchmark datasets as follows. As a training set, a random sample of 500 documents from D was given a ‘yes/no’ label of being conspiracy-related according to the majority vote of three independent human coders (trained student assistants). We found that 65 out of 500 documents (13.0%) were classified as conspiracy-related. We independently collected a corpus of 47,272 4chan posts from the sample weeks. To reduce the imbalance of the dataset, we clustered document-level embedding representations of these samples using the k -medoids method (Park and Jun 2009). This lets us identify 225 documents as those medoids which are most similar to, but distinct from, the posts with coded ground-truth. These 225 documents serve as our test set, of which 69 are conspiracy-related (30.7%) according to the majority vote of the three independent human coders. To avoid extended exposure of human coders to potentially abusive 4chan content, we kept the size of the manual annotation datasets small.

5.2 Experimental Setup

The vocabulary V is formed by the 5,000 most frequent words (excluding stop words) in the domain-specific corpus D and it includes our seed keywords. We compute fine-tuned word embeddings $V \subseteq \mathbb{R}^{100}$ from D , starting from pre-trained 100-dimensional GloVe base embeddings using Mittens in Equation (13) with the default $\mu = 0.1$, as our corpus D is reasonably large (Dingwall and Potts 2018). Our seed dictionary $W_0 = \tilde{W}_0 \cap V$ is composed of the 109 keywords that appear in the corpus.

Next, we compare the dictionary expansions obtained using thresholding and LGDE with a maximum of 150 discovered keywords. We perform hyperparameter tuning to obtain the optimal dictionaries by evaluating the performance of the discovered words $W(\epsilon) \setminus W_0$ and $W(k, t) \setminus W_0$, since our human-coded train data was collected using the seed dictionary W_0 . We find that, in this case, $k = 7$ and $t = 1$ leads to the best dictionary for LGDE and $\epsilon = 0.730$ for thresholding.

5.2.1 Protocol for Expert-based Annotation. Three domain experts, proficient in identifying and collecting data related to conspiracy theories, were presented with the complete set (union) of words discovered by the thresholding method and LGDE, in a random order. Each expert independently carried out blind annotation of whether discovered terms are suitable to be used as a keyword to search for conspiracy-related content on

Table 8

Test macro scores (precision P , recall R , F_1 score) for the seed, threshold-expanded, and LGDE dictionaries in the conspiracy-related dataset.

	P	R	F_1
Seed dictionary	0.769	0.559	0.529
Thresholding	0.595	0.609	0.558
LGDE	0.700	0.621	0.629

4chan, as defined above (Heft et al. 2023, p. 3). The annotation guideline required a keyword to be labeled as 1 if: it is related to a narration of a specific kind of conspiracy theory; or it talks about possibly conspicuous actors and actions; or it is a form of “outside” observation and reporting on actors spreading conspiracy theories or actions perceived by others as conspiracies (or 0 otherwise). The final labels are assigned based on majority voting. We discuss the results in the next section.

5.3 Results

We first evaluate the performance of the thresholding and LGDE expanded dictionaries using the human-coded test data described in Section 5.1. Table 8 shows that the LGDE dictionary has the highest macro F_1 -Score of 0.629, and LGDE achieves both higher macro precision and recall than the thresholding-based dictionary. As expected, the expert-selected seed dictionary has the highest precision, P , but LGDE improves substantially the recall, R , and hence the F_1 score, whereas the thresholding-based expansion is not as effective.

The words discovered through both dictionary expansions were evaluated and annotated according to the majority vote of three independent domain experts following the protocol described in Section 5.2.1. We find that LGDE discovers significantly more conspiracy-related keywords ($p < 0.01$, Fisher’s exact test): 30.2% of the words discovered by LGDE are found to be conspiracy-related in contrast to 18.9% of the words discovered by thresholding (Table 9).

A qualitative assessment of the discovered terms shows that this quantitative improvement coincides with the semantic content of the discovered words. Many of the terms discovered via thresholding are formal words relating to parts of the population, political philosophies, individuals, or entities, whereas the words discovered via LGDE are more closely associated with 4chan users’ platform-specific rhetoric. For example, LGDE discovers anti-Semitic jargon that might seem unremarkable in other contexts, such as “golem”, an anti-Semitic concept that refers to people “brainwashed” by Jewish media, or “good goy”, an anti-Semitic term related to Jewish domination. Other key vocabulary related to conspiracy narratives include “globoho*o”, an anti-Semitic and homophobic portmanteau of the words “global” and “homosexual”, “predictive programming”¹ or “MKUltra”.² Our corpus contains noun phrases as terms, and we observe that LGDE captures more multi-word phrases useful for the identification of

1 “Predictive programming” refers to the idea that governments use popular media like books or films to control people.

2 “MKUltra” refers to human experiments conducted by the CIA in the past including psychological torture.

Table 9

Discovered words in the 4chan dataset. Words in bold are classified as conspiracy-related by majority vote of three independent domain experts (30.2% of the words discovered by LGDE vs. 18.9% of the words discovered by thresholding). LGDE discovered significantly more conspiracy-related words than thresholding ($p < 0.01$, Fisher’s exact test). See Table 13 in the Appendix for unmasked keywords.

LGDE only	:^ , adolf_hitler, afd, ain’t, alt_right, amalek, american_government, architect, bannon, bioweapon , bogdanoffs , bombing, btfo, cabal , cattle, civic_nationalism, cold_war, communist_party, cont, ctr, cull, death_camp, disguis , donald_trump, durr, eas, end_game , end_result, end_time , engineering, entire_population , etc, executive_order, expulsion, extermination_camp, false_narrative , false_prophet , general_public, george_washington, gib, globalist_jews , globalist_puppet , globoho*o , golem , good_goy , good_job, good_people, great_ally, hahaha, hahahaha, happ, ho*o, hurr, imo, international_criminal_court, invasion, israeli_puppet , jeff_bezos , jesus_christ, jewish_plan , jewish_role , jewish_state , jp_morgan , juden , kamala_harris, lebensraum, libt**d, main_reason, mass_extinction , master , memeflag, mental_illness, mkultra , mutt, muzzy, new_age, new_zealand, nonce, nsg , occupy, oligarch, orthodox_church, ottoman_empire, pharisee, pogrom, predictive_programming , prev, ptg, pu**ie, racemixing , real_power , removal, rule_class , saa, schlo*o, scientist, sperg, stooge, supreme_court, tldr, tpp , vast_majority, video_game, welfare_state, white_supremacy, year_old, zog_puppet
Intersection	9/11, admit, annihilation , assimilation, atrocity, banker, be**er, billionaire, cia , commie, commy, conquer, degeneracy, depopulation , deportation, destroy, eliminate, engineer, extinct, financier , freemasonry , heroine, invade, k**e, luciferian , masonic , mass_murder , massacre, neocon, ohhhh, open_border , pedo, persecution, relocate , rid, rockefeller , skill , slaughter, species, terror, wipe, wwiii
Thresholding only	2., 3., act, annihilate, armenian, assassinate , assassination , attack, attempt, auschwitz, banish, bank, basically, Biden, capitalism, capitalist, colonize, commit, communism, communist, conservatism, conspiracy , consumerism, crime, c**k, cult, cure, deport, displace , diversity, elite , embrace, empire, enslave , eu, existence, expansion, exploit, fa***t, fascism, feminism, fuck, fund, globalization, guy, happen, hate, hillary, holocaust , homosexuality, human, imperialism, internationalism, isolate, jesuit , jew, jews, justify, k**e_puppet , kill, leftist, liberal, liberalism, liberate, like, literally, marxist, mass_genocide , member, money, mossad , murder, nationalism, nazi, need, neoliberalism, ni**er, oppress, president, promote, protect, push, racism, rape, religion, rWanda, save, shit, shitskin, socialism, spread, subjugate , terrorist, tolerance, torture, tribunal, trump, try, uk, vatican, violence, war, weed, win, zionism , zionist

conspiracy-related posts, such as “Jewish plan” or “Israeli puppet”, instead of the less useful unigrams “Jewish”, “plan”, or “Israeli”.

6. Discussion

In this work, we have proposed LGDE, a graph-based, local-diffusion framework for the discovery of keywords that are semantically similar to a pre-defined seed dictionary. Using tools from manifold learning and network science, LGDE captures the complex nonlinear geometry of word embeddings and finds not only most similar words but also chains of word associations. Our results on two benchmarks (a standard 20 Newsgroups dataset and a hate speech corpus from online posts) show that LGDE outperforms baseline methods (thresholding, kNN, iterative thresholding [IKEA], TextRank) with improved macro F_1 scores and larger likelihood ratio for words discovered, indicating that LGDE expands the dictionary with more representative terms. Keyword expansion is especially important in highly dynamic text corpora (such as the online hate speech example), which typically use an informal lexicon or jargon that is challenging to anticipate without in-depth knowledge of online communities.

Table 10 presents a comparative overview of the principles underpinning the different methods examined here. TextRank stands in contrast to the other methods in

Table 10

Overview of dictionary expansion methods considered here. LGDE is the only graph-based method applied to word embeddings that uses diffusion-based local community detection.

	Word embeddings	Geometric cut-off	Graph	Local community	Diffusion-based
TextRank			★		★
Thresholding	★	★			
IKEA	★	★			
kNN	★		★		
CkNN + Mod <i>R</i>	★		★	★	
CkNN + Mod <i>M</i>	★		★	★	
CkNN + SIWO	★		★	★	
LGDE (CkNN + severability)	★		★	★	★

that it does not rely on similarities from word embeddings but rather selects relevant words by ranking them according to their PageRank, a random-walk centrality in the graph of word co-occurrences. In our experiments, TextRank disproportionately included less relevant terms and resulted in reduced F_1 and LR scores, thus confirming the importance of going beyond mere word co-occurrences, and instead considering the rich semantic representation provided by high-dimensional word embeddings.

The rest of the methods considered here use word embeddings and exploit the geometry of the embedding space in different ways. In particular, both thresholding and IKEA enlarge the dictionary with neighborhoods around seed keywords defined via a geometric cut-off—in one swoop in the case of thresholding, and iteratively in the case of IKEA. Neither thresholding or IKEA improve upon the F_1 scores of the seed dictionaries in either the hate speech or 20 Newsgroups data, and are both outperformed by LGDE both in terms of F_1 scores and likelihood ratios in all cases. In our experiments, the iterative IKEA method failed to effectively manage the trade-off between discovering new terms and maintaining the dictionary quality, and produced dictionaries that were either too small or excessively large with low overall performance and limited applicability for real-world use cases. These results from thresholding and IKEA reveal the limitations of using Euclidean neighborhoods in a high-dimensionality and highly non-linear embedding space.

One way to capture the complexity of the embedding space is through the use of graphs to describe its local geometry, as is done in geometric and manifold learning. All the remaining methods in Table 10 use local geometric graphs to capture the neighborhoods of seed keywords. The simplest of all is the kNN graph constructed from the k closest neighbors, which shows improved F_1 scores relative to the seed dictionary in the 20 Newsgroups data. However, because kNN dictionary expansion only considers direct word similarities, it fails to take advantage of additional properties of the local graph construction. This leads to lower F_1 and likelihood ratio scores, especially as the dimensionality of the embedding space increases.

The remaining methods not only construct a graph representation of local neighborhoods (via CkNN), but also exploit graph-theoretical properties to obtain local communities around each seed keyword. Three of the local community detection methods (SIWO, local modularity R and M) use combinatorial graph properties (edge or clique high density) to identify local communities as a means to capture neighborhood inhomogeneity in embedding space. In our setting, these criteria translate into semantic

communities with consistent, high word similarities, and these methods often improve on the seed dictionary and can outperform other baseline methods. However, the reliance on density graph measures for local community detection limits the effectiveness of CkNN graph-based manifold learning for dictionary expansion.

LGDE, which uses a CkNN graph construction combined with severability to identify local communities based on graph diffusions, consistently outperforms the other methods in our experiments. Using random walks to explore graph neighborhoods in an adaptive manner allows for the identification of local communities that include strong long paths that capture semantic similarity via chains of word associations. LGDE can thus take advantage of the semantic content in high-dimensional word embeddings, while avoiding the lack of contrast that ensues from high dimensionality. This is seen in the higher F_1 -scores and likelihood ratios for LGDE as the dimensionality of the word embeddings increases from $r = 50$ to $r = 300$, in contrast to the reduction of performance observed for the thresholding-, kNN-, and IKEA-based approaches.

Although LGDE performs well in dictionary expansion for both standard text from the 20 Newsgroups and online hate speech text with highly subjective and evolving language usage, LGDE may be particularly suitable in cases like the latter, which require differentiation between complex topics or dealing with specialized jargon or colloquial vocabulary. In particular, we find that in the real-world use case of a corpus of conspiracy-related 4chan posts, LGDE discovers platform-specific words, noun phrases, and jargon terms that represent the neologisms and informal register specific to the 4chan corpus. This makes LGDE especially informative when researchers cannot assume a priori a comprehensive knowledge of the lexical variety of the object of study.

Our LGDE method has two hyperparameters, k and t , which control the sparsity of the graph and the time horizon of the random walk exploration, respectively. These hyperparameters were tuned on a train split of the data (see Table 1, and Figures 6 and 7 in the Appendix). Our numerics indicate that LGDE achieves higher performance in two broad regimes: a sparser graph with longer time diffusion (small k and large t) or, conversely, a denser graph with shorter time diffusion (large k and small t). These results show that LGDE can accommodate differences in the word embeddings for different corpora (via the sparsity of the similarity graph) that are then exploited differently via the diffusion. In particular, these higher performing regimes reflect the expected balance in a graph diffusion, whereby if we had instead a long-time diffusion (high t) over a dense graph (high k) it would lead to large, unspecific semantic communities around the seed words.

As a final illustration of LGDE, Figure 5 shows the semantic similarity graph for the 4chan dataset with the conspiracy-related seed keywords (red) and discovered words (orange) clustered closely together on the left side of the layout. Semantic communities obtained by LGDE often overlap and can contain multiple seed keywords, for example, the semantic community of “deep state” in Figure 5 (b). The semantic communities can also contain chains of strong word associations. For example, the seed keyword “helter_skelter”³ is linked to the word “entire_population”, a term used within Great Replacement narratives, via a chain through the term “good_goy” with similarities of 0.61 and 0.62, respectively, whereas the direct similarity is only 0.50; see Figure 5 (c). If we were to set $\epsilon = 0.50$ in a thresholding approach to discover “entire_population” from the seed “helter_skelter”, the expanded dictionary would contain 4,891 words (almost the full vocabulary) with a poor test $F_1 = 0.246$. In summary, the improved performance

3 “Helter_skelter” refers to an apocalyptic vision of race conflict.

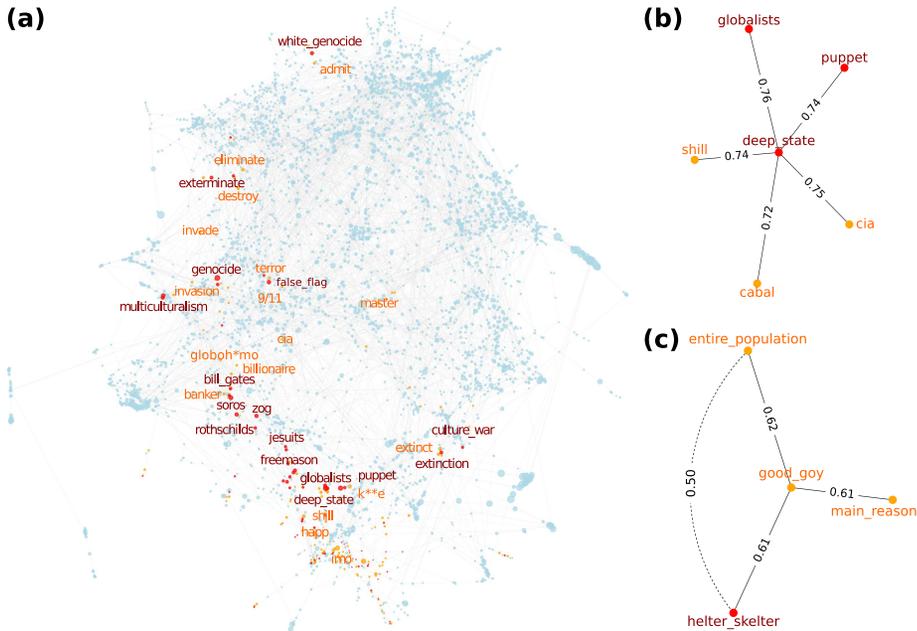


Figure 5
 (a) Largest connected component of the semantic similarity graph obtained within LGDE for the 4chan dataset, where red nodes indicate seed keywords and orange nodes correspond to discovered words. (b) The semantic community of “globalists” contains two other seed keywords and three discovered words, all conspiracy-related according to domain expert evaluation, see Table 9. The edge weights represent word similarities. (c) The semantic community of “helter_skelter” contains the two conspiracy-related discovered words “good_goy” and “entire_population”, where the latter is connected to the seed keyword via a chain of high similarities whereas the direct similarity (0.50) to the seed keyword (dashed line) is lower. If such a low similarity is used in the thresholding-based approach, the expanded dictionary $W(\epsilon)$ would include 4,891 words (out of 5,000 in the whole vocabulary) with a low performance of $F_1 = 0.246$ on the test set.

of LGDE stems from the combination of a graph-based description of the local geometry of the word-embedding space together with the diffusion-based detection of semantic communities in the local neighborhoods of seed keywords.

7. Limitations and Future Work

We list some limitations in the current work and areas of open research on which we would like to expand in the future. While our experiments here use only English language data, our method is general and the application could be useful for similar data in other languages. Also, although qualitative assessment is invaluable for the validation of our results, the process of manual annotation can be slow and costly and, in particular, labeling hate speech- or conspiracy-related content can pose severe mental health risks to human annotators. Hence, in future work, we would like to use LGDE as part of mixed methods approaches (Puschmann et al. 2022). This is especially important for domains like hate speech or conspiracy-related communication where systems based on large language models (Jagerman et al. 2023; Wang, Yang, and Wei 2023; Lei et al. 2024) could be curtailed due to strict moderation filters. Other areas of further

work include other specialized domains where the adaptation, creation, and evolution of terminology is a central feature, for example, in the scientific literature in relation to the emergence of new sub-disciplines or research areas. Furthermore, it will be important to carry out a detailed error analysis of LGDE to better understand the limitations of the method. To facilitate the applicability of LGDE, we also plan to carry out further numerics to characterize the heuristics for the selection of the hyperparameters k and t , as discussed above. Finally, it would be interesting to evaluate the applicability of LGDE to specialized word disambiguation tasks, since we have observed preliminary evidence of polysemy being captured through overlapping semantic communities (Yu et al. 2020).

Data and Code Availability

The hate speech data is publicly available at <https://github.com/jing-qian/A-Benchmark-Dataset-for-Learning-to-Intervene-in-Online-Hate-Speech> and the 4chan data may be made available upon evaluation of reasonable requests. Code for optimizing the severability quality function and performing overlapping community detection is available at <https://github.com/barahona-research-group/severability>. An implementation of LGDE and code to reproduce the results and figures in this study is available at <https://github.com/barahona-research-group/LGDE>.

Funding

Juni Schindler acknowledges support from the EPSRC (PhD studentship through the Department of Mathematics at Imperial College London) and the Weizenbaum Institute (Research Fellowship). Annett Heft, Kilian Buehling, and Xixuan Zhang acknowledge support by grants from the German Federal Ministry of Education and Research (grant numbers 13N16049 [in the context of the call for proposals Civil Security – Societies in Transition] and 16DII135 [in the context of the Weizenbaum Institute]). Mauricio Barahona acknowledges support from EPSRC grant EP/N014529/1 supporting the EPSRC Centre for Mathematics of Precision Healthcare.

8. Appendix

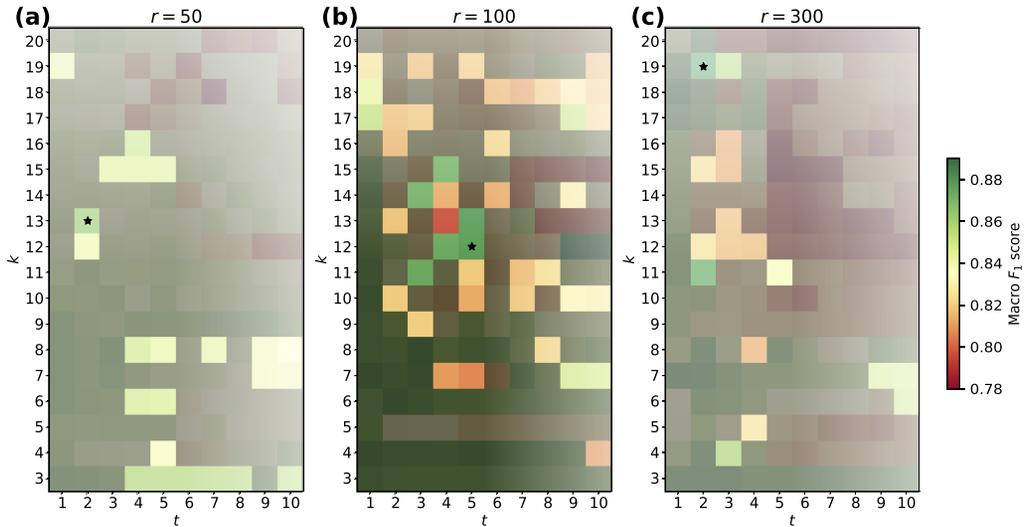


Figure 6
 Hyperparameter tuning of (k, t) for LGDE dictionary $W(k, t)$ in the hate speech dataset is performed using the train macro F_1 score of expanded dictionaries (optimal values indicated by stars). Note that the hyperparameter combinations where the number of discovered words is less than 30 or more than 50 are not considered (shaded fields).

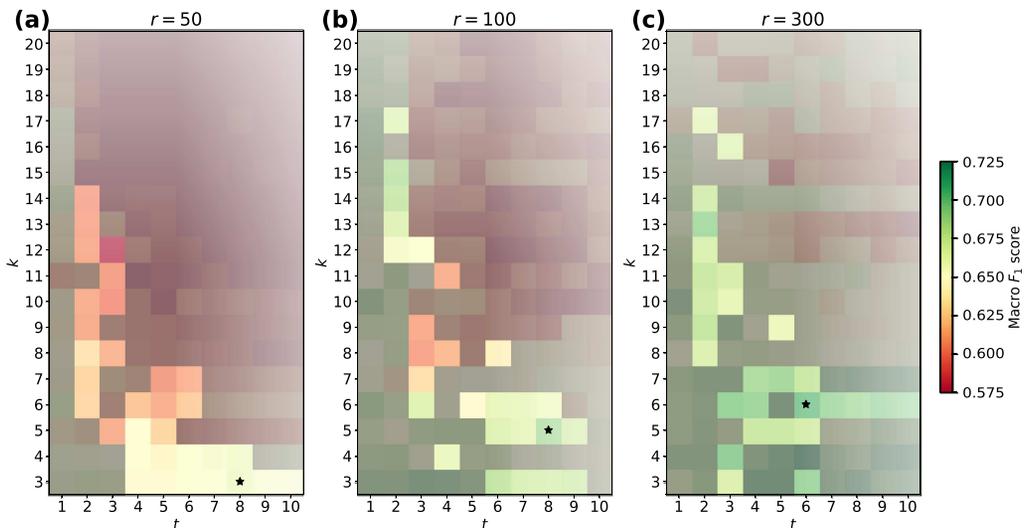


Figure 7
 Hyperparameter tuning of (k, t) for LGDE dictionary $W(k, t)$ in the 20 Newsgroups dataset is performed using the train macro F_1 score of expanded dictionaries (optimal values indicated by stars). Note that the hyperparameter combinations where the number of discovered words is less than 100 or more than 150 are not considered (shaded fields).

Table 11

Expert-selected seed dictionary with 215 English and German conspiracy-related keywords based on a German language dictionary by Puschmann et al. (2022) and additional literature. While the seed dictionary retains German-language keywords, only English-language 4chan posts were collected in our experiment.

14_words, 14_worte, 14words, abolition_of_the_german_people, abolition_of_the_occident, abolition_of_white_people, abschaffung_des_abendlandes, abschaffung_des_deutschen_volkes, abschaffung_des_deutschen_volks, auslöschen, ausmerzen, ausrotten, ausrottung, bevölkerungsaustausch, bevölkerungsreduzierung, bilderberg, bilderberg-, bilderberg_group, bilderberg-club, bilderberg-conference, bilderberg-conferences, bilderberger, bilderberger-conference, bilderberger-konferenz, bilderbergern, bilderbergers, bilderberg-group, bilderberg-gruppe, bilderberg-konferenz, bilderberg-konferenzen, bill_gates, birth_jihad, birth-jihad, black_helicopters, bürgerkriegsflüchtlinge, cemetery_in_prague, civil_war_refugees, coudenhove-kalergi, culture_war, deckmantel, deckmäntelchen, deep_state, der_große_austausch, drahtzieher, drahtziehern, edmond_de_rothschild, elders_of_zion, eradicate, eurabia, eurabien, exterminate, extermination, extinction, fadenzieher, false_flag, false_flag_angriff, false_flag_angriffe, false_flag_attack, false_flag_attacks, false_flag-angriff, false_flag-angriffe, financial_elite, financial_elites, financial_lobby, financial_mafia, finanzelite, finanzeliten, finanzlobby, finanzmafia, flüchtlingsinvasion, foreingnation, freemason, freimaurer, friedhof_in_prag, geburten-dschihad, geburten-jihad, geheimbund, geheimbünde, geheimgesellschaft, geheimpolitik, genocide, genozid, george_soros, germoney, global_replacism, globalism, globalismus, globalist, globalisten, globalists, great_exchange, great_replacement, große_austausch, großem_austausch, großen_austausch, großer_austausch, helter_skelter, illuminaten, illuminati, islamisierung, islamization, jesuit_order, jesuiten, jesuitenorden, jesuitism, jesuitismus, jesuits, jewish_republic, judenrepublik, kalergi_plan, kalergi-plan, killuminati, kulturkampf, marionette, marionetten, marionettenregierung, marionettentheater, massenaustausch_der_bevölkerung, mastermind, menschenlenkung, menschheit_reduzieren, migration_als_waffe, migration_as_a_weapon, migration_weapon, migrationswaffe, multiculturalism, multiculturalist, multikulturalismus, multikulturalist, neue_weltordnung, neweweltordnung, new_world_order, nwo, one_world_government, owg, pegida, people_management, people's_death, population_exchange, population_reduction, population_replacement, protocols_of_the_elders_of_zion, protocols_of_zion, protokolle_der_weisen_von_zion, protokolle_zions, puppet, puppet_government, puppet_master, puppet_theater, puppets, race_genocide, racial_genocide, rassenmischung, reduce_humankind, refugee_invasion, relocation, repopulation, repopulation_machinery, repopulation_policy, rothschild, rothschilds, satanic_world_elite, satanic_world_elites, satanische_weltelite, satanische_welteliten, secret_policy, secret_societies, secret_society, sir_john_retcliffe, soros, string_puller, string_pullers, strippenzieher, strippenzieherin, strippenziehern, the_octopus, tiefe_staat, tiefen_staat, tiefer_staat, umsiedlung, umvolkung, umvolkungsmaschinerie, umvolkungspolitik, usrael, verausländerung, volksaustausch, volkstod, war_on_whites, weise_von_zion, weisen_von_zion, weißer_genozid, weltherrschaft, weltjudentum, weltordnung, weltregierung, weltverschwörung, white_extinction, white_genocide, white_replacement, world_conspiracy, world_control, world_domination, world_government, world_judaism, world_order, zionist_congress, zionist_lodges, zionist_occupied_government, zionist_protocols, zionistisch_besetzte_regierung, zionistische_logen, zionistische_protokolle, zionistischer_kongress, zog, zog-

Table 12

Twenty-two sample weeks (2 weeks in each year from 2011 to 2021) used to collect 4chan data.

	Start date	End date
Week 1	28.02.2011	06.03.2011
Week 2	15.08.2011	21.08.2011
Week 3	12.03.2012	18.03.2012
Week 4	15.10.2012	21.10.2012
Week 5	07.01.2013	13.01.2013
Week 6	04.11.2013	10.11.2013
Week 7	24.02.2014	02.03.2014
Week 8	04.08.2014	10.08.2014
Week 9	19.01.2015	25.01.2015
Week 10	02.11.2015	08.11.2015
Week 11	30.05.2016	05.06.2016
Week 12	12.09.2016	18.09.2016
Week 13	27.03.2017	02.04.2017
Week 14	21.08.2017	27.08.2017
Week 15	02.04.2018	08.04.2018
Week 16	03.12.2018	09.12.2018
Week 17	22.04.2019	28.04.2019
Week 18	16.12.2019	22.12.2019
Week 19	03.02.2020	09.02.2020
Week 20	26.10.2020	01.11.2020
Week 21	31.05.2021	06.06.2021
Week 22	11.10.2021	17.10.2021

Table 13

Unmasked offensive keywords.

A-H		K-Z	
as**le	asshole	k**e	kike
as**les	assholes	libt**d	libtard
be**er	beaner	mo*on	moron
b**ch	bitch	motherf**ker	motherfucker
b**bs	boobs	motherf**kers	motherfuckers
bu****it	bullshit	ni**a	nigga
chi***an	chinaman	ni**er	nigger
c**k	cuck	ni**ers	niggers
c**t	cunt	pu**ie	pussie
c**ts	cunts	ra**ead	raghead
f*g	fag	ret**d	retard
fa***t	faggot	ret**dation	retardation
fa**ts	faggots	ret***ed	retarded
f**k	fuck	ret**ds	retards
f**ker	fucker	schlo*o	schlomo
f**kers	fuckers	sc***ag	scumbag
f**kin	fuckin	sk**k	skank
f**king	fucking	s**ts	sluts
f**ks	fucks	th*t	thot
globoho*mo	globohomo	tr**ny	tranny
h*g	hag	tw*t	twat
ho*o	homo	w**re	whore
ho*os	homos	wi**er	wigger

Acknowledgments

This study has benefitted from discussions in the *Advancing Cross-Platform Research in Political Social Media Communication* workshop at the Weizenbaum Institute in 2022 and from the first author's presentation in the Computational Methods Division at the International Communication Association Conference in 2023. We thank our student assistants Joana Becker, Dominik Hokamp, Angelika Juhász, and Katharina Sawade for their invaluable work of labelling benchmark data of conspiracy-related 4chan posts. We also thank Yun William Yu and Angad Khurana for helpful discussions on the implementation of severability, and Asem Alaa for valuable suggestions on the Python packaging of the code for this project. Finally, we are grateful to the anonymous reviewers for their constructive comments that significantly enhanced the manuscript.

References

- Aggarwal, Charu C., Alexander Hinneburg, and Daniel A. Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory—ICDT 2001*, Lecture Notes in Computer Science, pages 420–434, Springer. https://doi.org/10.1007/3-540-44503-X_27
- Altuncu, M. Tarik, Erik Mayer, Sophia N. Yaliraki, and Mauricio Barahona. 2019. From free text to clusters of content in health records: An unsupervised graph partitioning approach. *Applied Network Science*, 4(1):2. <https://doi.org/10.1007/s41109-018-0109-9>, PubMed: 30906850
- Altuncu, M. Tarik. 2020. *A Graph Theoretical Perspective for the Unsupervised Clustering of Free Text Corpora*. Ph.D. thesis, Imperial College London.
- Amsler, Michael. 2020. *Using Lexical-Semantic Concepts for Fine-Grained Classification in the Embedding Space*. Ph.D. thesis, University of Zurich.
- Baltsou, Georgia, Konstantinos Christopoulos, and Konstantinos Tsihclas. 2022. Local community detection: A survey. *IEEE Access*, 10:110701–110726. <https://doi.org/10.1109/ACCESS.2022.3213980>
- Berry, Tyrus and Timothy Sauer. 2019. Consistent manifold representation for topological data analysis. *Foundations of Data Science*, 1(1):1–38. <https://doi.org/10.3934/fods.2019001>
- Bommasani, Rishi, Kelly Davis, and Claire Cardie. 2020. Interpreting pretrained contextualized representations via reductions to static embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781. <https://doi.org/10.18653/v1/2020.acl-main.431>
- Bruns, Axel, Stephen Harrington, and Edward Hurcombe. 2020. 'Corona? 5G? or both?': The dynamics of COVID-19/5G conspiracy theories on Facebook. *Media International Australia*, 177(1):12–29. <https://doi.org/10.1177/1329878X20946113>
- Buehling, Kilian. 2022. `fouRplebsAPI`: R package for accessing 4chan posts via the 4plebs.org API.
- Clauset, Aaron. 2005. Finding local community structure in networks. *Physical Review E*, 72(2):026132. <https://doi.org/10.1103/PhysRevE.72.026132>, PubMed: 16196669
- De Zeeuw, Daniël, Sal Hagen, Stijn Peeters, and Emilija Jokubauskaite. 2020. Tracing normification: A cross-platform analysis of the QAnon conspiracy theory. *First Monday*, 25(11). <https://doi.org/10.5210/fm.v25i11.10643>
- De Zeeuw, Daniël and Marc Tuters. 2020. Teh Internet is serious business. *Cultural Politics*, 16(2):214–232. <https://doi.org/10.1215/17432197-8233406>
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Diaz, Fernando, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*. <https://doi.org/10.18653/v1/P16-1035>
- Dingwall, Nicholas and Christopher Potts. 2018. Mittens: An Extension of GloVe for learning domain-specialized representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 212–217. <https://doi.org/10.18653/v1/N18-2034>
- Farmanbar, Mojtaba, Nikki Van Ommeren, and Boyang Zhao. 2020. Semantic search with domain-specific word-embedding

- and production monitoring in Fintech. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, pages 28–33. <https://doi.org/10.18653/v1/2020.coling-demos.6>
- Fellbaum, Christiane. 1998. WordNet: An electronic lexical database. *MIT Press Google Schola*, 2:678–686. <https://doi.org/10.7551/mitpress/7287.001.0001>
- Firoozeh, Nazanin, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. 2020. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3):259–291. <https://doi.org/10.1017/S1351324919000457>
- Fortunato, Santo. 2010. Community detection in graphs. *Physics Reports*, 486(3–5):75–174. <https://doi.org/10.1016/j.physrep.2009.11.002>
- Garry, Amanda, Samantha Walther, Rukaya Rukaya, and Ayan Mohammed. 2021. QAnon conspiracy theory: Examining its evolution and mechanisms of radicalization. *Journal for Deradicalization*, (26):152–216.
- Gharibshah, Joobin, Jakapun Tachaiya, Arman Irani, Evangelos E. Papalexakis, and Michalis Faloutsos. 2022. IKEA: Unsupervised domain-specific keyword-expansion. In *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 496–503. <https://doi.org/10.1109/ASONAM55673.2022.10068656>
- Heft, Annett, Kilian Buehling, Xixuan Zhang, Juni Schindler, and Miriam Milzner. 2023. Challenges of and approaches to data collection across platforms and time: Conspiracy-related digital traces as examples of political contention. *Journal of Information Technology & Politics*, 21(3):323–339. <https://doi.org/10.1080/19331681.2023.2250779>
- Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in Python. <https://spacy.io/>
- Jagerman, Rolf, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv:2305.03653*.
- Jakubowski, Alexander, Milica Gasic, and Marcus Zibrowius. 2020. Topology of word embeddings: Singularities reflect polysemy. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 103–113.
- Keerthi, S. Sathya and Dennis DeCoste. 2005. A modified finite Newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 6(12):341–361.
- Klinger, Ulrike, W. Lance Bennett, Curd Benjamin Knüpfer, Franziska Martini, and Xixuan Zhang. 2022. From the fringes into mainstream politics: Intermediary networks and movement-party coordination of a global anti-immigration campaign in Germany. *Information, Communication & Society*, 26(9):1890–1907. <https://doi.org/10.1080/1369118X.2022.2050415>
- Kuzi, Saar, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM '16*, pages 1929–1932. <https://doi.org/10.1145/2983323.2983876>
- Lee, Yongho, So Young Kim, Inseok Song, Yongtae Park, and Juneseuk Shin. 2014. Technology opportunity identification customized to the technological capability of SMEs through two-stage patent analysis. *Scientometrics*, 100(1):227–244. <https://doi.org/10.1007/s11192-013-1216-0>
- Lei, Yibin, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–401.
- Liu, Zijing and Mauricio Barahona. 2020. Graph-based data clustering via multiscale community detection. *Applied Network Science*, 5(1):3. <https://doi.org/10.1007/s41109-019-0248-7>
- Luo, Feng, James Z. Wang, and Eric Promislow. 2006. Exploring local community structures in large networks. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pages 233–239. <https://doi.org/10.1109/WI.2006.72>
- Mihalcea, Rada and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411. <https://doi.org/10.3115/1220575.1220627>

- Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. https://doi.org/10.1007/978-1-4757-6744-5_4
- Mitchell, Tom. 1997. Twenty newsgroups. <https://doi.org/10.1016/j.eswa.2008.01.039>
- Park, Hae-Sang and Chi-Hyuck Jun. 2009. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341. <https://doi.org/10.1016/j.eswa.2008.01.039>
- Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. <https://doi.org/10.18653/v1/N18-1202>
- Puschmann, Cornelius, Hevin Karakurt, Carolin Amlinger, Nicola Gess, and Oliver Nachtwey. 2022. RPC-Lex: A dictionary to measure German right-wing populist conspiracy discourse online. *Convergence*, 28(4):1144–1171. <https://doi.org/10.1177/13548565221109440>, PubMed: 36186680
- Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108. <https://doi.org/10.18653/v1/2020.acl-demos.14>
- Qian, Jing, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4755–4764. <https://doi.org/10.18653/v1/D19-1482>
- Qian, Yifan, Paul Expert, Pietro Panzarasa, and Mauricio Barahona. 2021. Geometric graphs from data to aid classification tasks with Graph Convolutional Networks. *Patterns*, 2(4):100237. <https://doi.org/10.1016/j.patter.2021.100237>, PubMed: 33982027
- Roy, Dwaipayan, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. *arXiv:1606.07608*.
- Schaub, Michael T., Jean-Charles Delvenne, Martin Rosvall, and Renaud Lambiotte. 2017. The many facets of community detection in complex networks. *Applied Network Science*, 2(1):1–13.
- Schindler, Juni and Matthew Fuller. 2023. Community as a vague operator: Epistemological questions for a critical heuristics of community detection algorithms. *Computational Culture*, (9).
- Schlechtweg, Dominik, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised lexical semantic change detection. *arXiv preprint arXiv:2007.11464*. <https://doi.org/10.18653/v1/2020.semeval-1.1>
- Schnabel, Tobias, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307. <https://doi.org/10.18653/v1/D15-1036>
- Schubert, Erich. 2021. A triangle inequality for cosine similarity. In *Similarity Search and Applications, Lecture Notes in Computer Science*, pages 32–44, Springer International Publishing. https://doi.org/10.1007/978-3-030-89657-7_3
- Schütze, Hinrich, Christopher D. Manning, and Prabhakar Raghavan. 2008. *Introduction to Information Retrieval*, volume 39. Cambridge University Press Cambridge.
- Stoll, Anke, Lena Wilms, and Marc Ziegele. 2023. Developing an incivility dictionary for German online discussions – A semi-automated approach combining human and artificial knowledge. *Communication Methods and Measures*, 17(12):131–149. <https://doi.org/10.1080/19312458.2023.2166028>
- Tsai, Ming-Feng and Chuan-Ju Wang. 2014. Financial keyword expansion via continuous word vector representations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1453–1458. <https://doi.org/10.3115/v1/D14-1152>

- van Atteveldt, Wouter, Dafne van Kuppevelt, and Kasper Welbers. 2022. CAVA: An open source R toolkit for dictionary Coherence, Adaptation, Validation, and Analysis. <https://github.com/vanatteveldt/CAVA>
- van der Helm, H. J. and E. A. Hische. 1979. Application of Bayes's theorem to results of quantitative clinical chemical determinations. *Clinical Chemistry*, 25(6):985–988. <https://doi.org/10.1093/clinchem/25.6.985>, PubMed: 445835
- Vulić, Ivan, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240. <https://doi.org/10.18653/v1/2020.emnlp-main.586>
- Wang, Liang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *arXiv:2303.07678*. <https://doi.org/10.18653/v1/2023.emnlp-main.585>, PubMed: 37358388
- Yin, Xicheng, Hongwei Wang, Pei Yin, Hengmin Zhu, and Zhenyu Zhang. 2020. A co-occurrence based approach of automatic keyword expansion using mass diffusion. *Scientometrics*, 124(3):1885–1905. <https://doi.org/10.1007/s11192-020-03601-7>
- Yu, Yun William, Jean-Charles Delvenne, Sophia N. Yaliraki, and Mauricio Barahona. 2020. Severability of mesoscale components and local time scales in dynamical networks. *arXiv:2006.02972*.
- Zafarmand, Mohammadmahdi, Yashar Talebirad, Eric Austin, Christine Largeton, and Osmar R. Zaiane. 2023. Fast local community discovery relying on the strength of links. *Social Network Analysis and Mining*, 13(1):112. <https://doi.org/10.1007/s13278-023-01115-7>
- Zeng, Jing and Mike S. Schäfer. 2021. Conceptualizing “dark platforms”. Covid-19-related conspiracy theories on 8kun and Gab. *Digital Journalism*, 9(9):1321–1343. <https://doi.org/10.1080/21670811.2021.1938165>
- Zheng, Zhi, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: Contextualized query expansion for document re-ranking. *arXiv preprint arXiv:2009.07258*.