

Bio-inspired Structure Identification in Language Embeddings

Hongwei (Henry) Zhou* Oskar Elek† Pranav Anand‡ Angus G. Forbes§
University of California, Santa Cruz

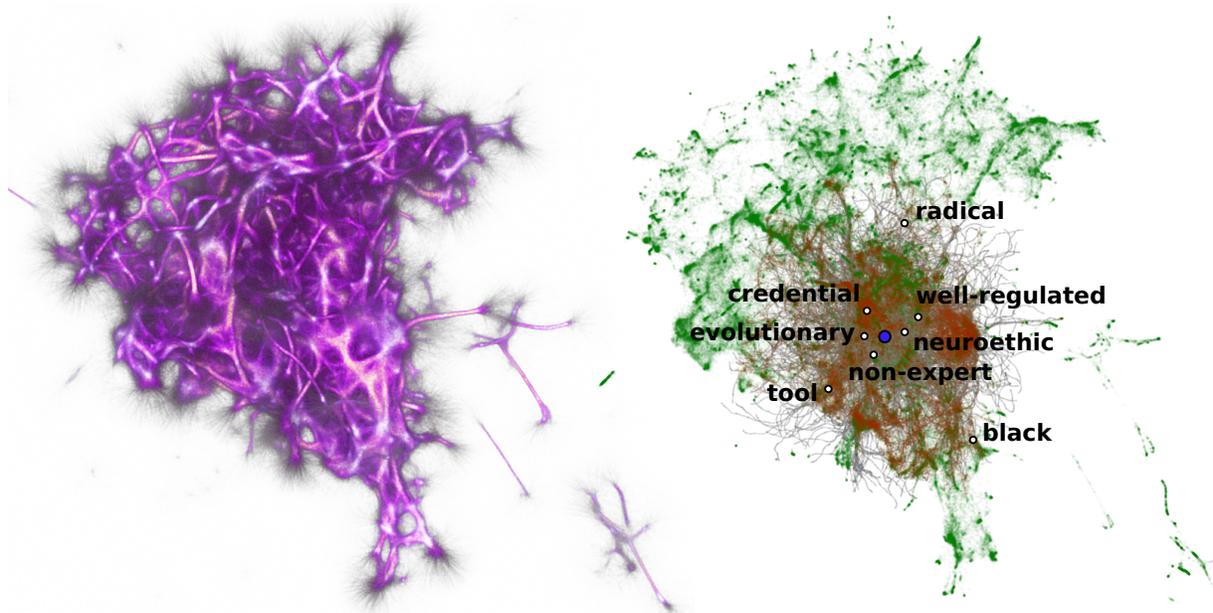


Figure 1: Left: 3D volume color-map visualization of the geometric structures recovered from Gensim Continuous Skipgram embedding of Wikipedia 2017 (300k tokens). Right: structure-guided exploration of the intermediate neighborhood of the noun *research*, highlighting the discovered tokens (red) from among all tokens (green). Tokens are discovered by structure-guided agents that stochastically explore the embedding space.

ABSTRACT

Word embeddings are a popular way to improve downstream performances in contemporary language modeling. However, the underlying geometric structure of the embedding space is not well understood. We present a series of explorations using bio-inspired methodology to traverse and visualize word embeddings, demonstrating evidence of discernible structure. Moreover, our model also produces word similarity rankings that are plausible yet very different from common similarity metrics, mainly cosine similarity and Euclidean distance. We show that our bio-inspired model can be used to investigate how different word embedding techniques result in different semantic outputs, which can emphasize or obscure particular interpretations in textual data.

Index Terms: [Human-centered computing]: Visualization—Visualization techniques; [Computing methodologies]: Artificial intelligence—Natural language processing;

1 MAPPING LANGUAGE

Much work extracting meaning from text has relied on relational structures that can be represented (and visualized) as graphs. Phrase Nets [34], for instance, uses nodes to represent words (‘tokens’) and

edges for the user-defined relations between them. Depending on the interpretation of the working data, higher-level entities can be mapped to this structure, such as documents [26], stories [33], or even ideas [23] – with suitable relational axioms applied to them. At a more granular level, syntactic relations in linguistics are often represented as graph diagrams [25], as are the ontological relationships between words [11]. While such relational structures have proven incredibly valuable, they are difficult to automatically generate from text, a problem since there are often countless relations one might wish to extract from a text. Moreover, from a computational perspective, graphs can be very costly: even in the simple example of the Bigram model, the resulting relationship graph will have $O(N^2)$ edges in a dataset with N tokens. For more complex relations, this cost can grow still further, rendering graphs difficult to handle for datasets with 10^4 – 10^5 or more tokens.

In recent years, word embeddings – such as Word2Vec, GloVe, ELMo, and BERT – have gained real remarkable traction as representations of word-level information. Their key computational idea is to transform topological information contained in a relational graph to geometric information encoded in a D -dimensional vector (‘embedding’) space by using a deep learning model. This brings the representational cost down from $O(N^2)$ to $O(ND)$, where $D \ll N$, typically in the 100s. On top of the efficiency increase, embeddings have a number of interesting algebraic properties: most importantly, the contextual similarity of the embedded tokens is transformed into geometric proximity in the embedding [5, 21]. Because they explicitly consider the token’s context [8, 29], it has been shown that embeddings contain information that can be processed to extract a range useful properties: clustering by token usage [31, 37] as well as different kinds of syntactic information [13, 18, 31]. Thus, there is the promise that this kind of method could provide high-dimensional

*e-mail: hzhou55@ucsc.edu

†e-mail: oelek@ucsc.edu, website: <https://cgg.mff.cuni.cz/~oskar/>

‡e-mail: panand@ucsc.edu, website: <https://people.ucsc.edu/~panand/>

§e-mail: angus@ucsc.edu, website: <https://creativecoding.soe.ucsc.edu/>

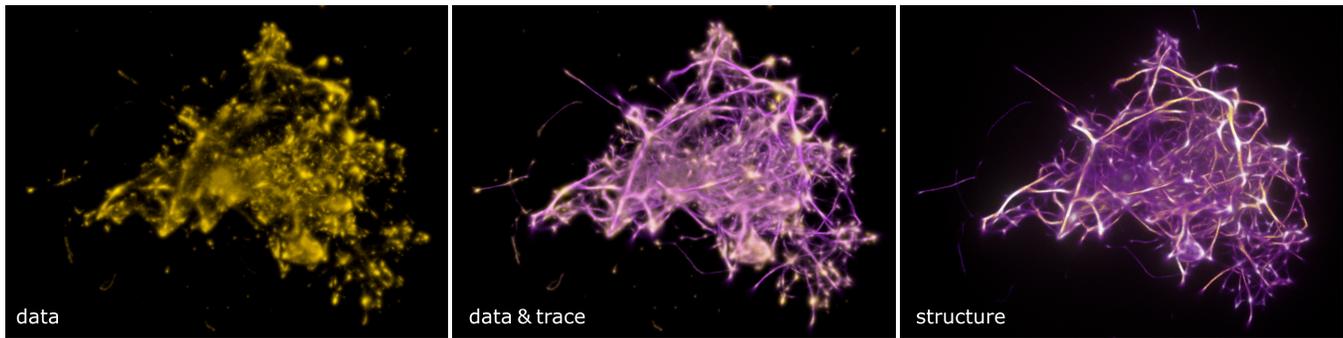


Figure 2: 3D volume visualization of global embedding for W2V-300k. Left: The token data are represented as a density field, rendered in yellow. Middle: geometry discovered in these data by MCPM is represented by another density field – trace – overlaid over the data in purple. Right: the structure of the trace field is rendered as an emissive ‘heatmap’, with the highest density values also emitting more light.

representations that encode a large manner of relations implicitly without having to hand-code them in advance.

In spite of the progress in language processing tasks, understanding the information contained in embeddings is still challenging due to their high dimensionality. While parallel coordinates are suitable for high dimensional data [6, 9] they do not capture the spatial relationships critical in embeddings. Therefore, the standard way to visualize embeddings is currently to project the token data to 2D or 3D using PCA, UMAP and other dimensionality reduction techniques [1], optionally with additional semantic annotations [5]. In that process, two different distortions happen to the data: distortion of high-level structure, and induction of relations that are not part of the original embedding. The inclusion of explicit referencing information between the tokens [3] and identification of salient dimensions [15] does seem to leverage some of these issues. In addition, interactive visualization tools have been proposed for literary experts and natural language processing researchers [12, 20]. These focus on exploring linear relationship between word embeddings, identifying concepts and experimenting with attribute vectors.

Yet our understanding of the embeddings’ structural properties remains far from complete [17]. For instance, recent work on contextual embeddings has found that the embedded tokens have a highly anisotropic distribution, which impacts the standard cosine distance used to measure their closeness [37]. At least in the case of non-contextual (‘global’) embeddings, normalizing the distribution to be more isotropic and centered about the origin improves the performance of downstream tasks that build on this way of measuring distances [22]. These and other results show that there are gaps in our understanding of the various embedding data.

2 LANGUAGE AS ORGANIC STRUCTURE

We propose a visualization and analysis framework for language embeddings based on bio-inspired **optimal transport networks**. The mathematics of optimal transport [27, 36] is based on the principle of least effort, which applies to phenomena ranging from particle and light transport to the behavior of living beings. Ubiquitous in nature, we posit that this principle applies to language alike – an idea already explored by Zipf [38] and others since.

We make the following key assumption: that the relational structure of language is reasonably preserved in (or transferred into) the embedded representations. Building on that, we look into the possibility of discovering the geometric structures in both local and global embeddings (Section 3). Our contributions towards this include:

- Recovering geometric structures in these data by using a combination of dimensionality reduction techniques (mainly UMAP) and a pattern-finding algorithm based on optimal transport in biological systems dubbed MCPM (Figure 1-left, Section 4).

- Design of a custom random-walk exploration technique to examine the recovered structures through the lens of a few standard language processing tasks (Figure 1-right, Section 5).
- Demonstration of the utility of these techniques for both gaining insight into the embedding data and improving the performance of standard language processing tasks (Section 6).

Ultimately, this work is a first step towards developing a framework to enable human-readable exploration of machine-generated language data. Section 7 discusses the future steps we plan to undertake to make this happen.

3 DATASETS

Language embedding data can be generated through a variety of embedding algorithms applied to specific text corpora. To take the examples of BERT [35] and Word2Vec [21], both propose novel neural network architectures to vectorize English words. The networks are both trained to guess masked words given its surrounding word tokens. The assumption behind these systems is that in the embedded space each word’s surroundings can capture its context.

We select these two language embedding models – BERT (Figure 3) and Word2Vec (Figure 2) – and generate the following datasets that are the basis for our further analysis.

Local (contextual) embeddings . Similar to Coenen et al., this dataset uses base BERT for embedding generation and Wikipedia as language corpus [5]. Each generated dataset is particular to a single word, and defines the context relative to that word – typically resulting in 1000s of tokens. Each data point then refers to a sentence in which the word occurred within the corpus. We can interpret this as a volume of semantic space where the meaning of a token can fluctuate based on its actual usage in the text. We will be referring to these as “BERT-X”, with “X” being the represented word.

Global embeddings. This dataset uses Gensim Continuous Skipgram – a variation of Word2Vec – to process the English Wikipedia Dump of February 2017, resulting in approximately 300k tokens. In contrast to contextual embeddings, each data point refers to a single token instead of the sentence in which a token is used. A token includes two pieces of information: the word and its part of speech. For example, *wind_NOUN* and *wind_VERB* are considered separate tokens and occupy different positions. This is different from BERT where the part of speech is not made explicit. We will be referring to this dataset as “W2V-300k”.

Dimensionality reduction. The original word embeddings are high-dimensional: BERT embeddings are 768-dimensional while those by Gensim Continuous Skipgram are 300-dimensional. To make visualization and analysis feasible, we rely on UMAP (with neighborhood size of 15) to project the data to 3D space. The

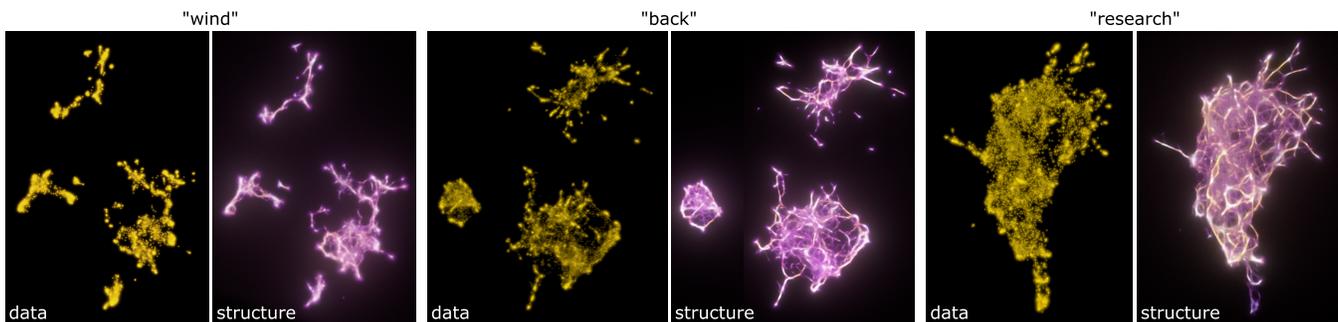


Figure 3: 3D volume visualization of local embeddings for words *wind*, *back* and *research*. The modalities are identical with Figure 2 (with the joint visualization of data and trace skipped). For each token, we observe different numbers of clusters and their shape.

dimensionality reduction is necessary, due to the high memory requirements of our notion of transport network: this is based on a continuous representation by a **density field**, both in the visualization (Section 4) and exploration (Section 5) stages.

Naturally, this implies the loss of some information, particularly sacrificing the global structure of the data to preserve local neighborhood relations (however, less so than other dimensionality reduction methods such as PCA or TriMap [2]). It is also likely that additional geometric structures are induced in this process. Yet in Section 5 we show that even these distortions do not render the dataset unusable when the detected structures are used for navigating the embedding. Sections 6 and 7 discuss this in further detail.

4 STRUCTURE DETECTION AND VISUALIZATION

Having the embedding data projected to 3D as a point cloud (Section 3), the next step is to find a transport network that spans it. For this purpose we use the Monte Carlo Physarum machine (MCPM), a **pattern-finding algorithm** inspired by the growth and foraging behavior of Physarum polycephalum ‘slime mold’ [10]. This method has been previously applied in astronomy for inferring the quasi-fractal structure of the cosmic web [4, 32], where it has successfully recovered the theoretically predicted filamentary patterns over sparse galaxy data.

MCPM is a hybrid method, in which a swarm (10^6 – 10^7) of discrete agents explores a domain represented by a continuous 3D lattice. This lattice stores the spatial footprint of the input data, which then acts as an attractor for the agents. As a result, the agents interconnect the input data in a single continuous transport network. This emergent network is represented by another lattice referred to as **trace**, effectively storing the scalar spatio-temporal density of the model’s agents. This representation of the transport network is advantageous for our further analysis (Section 5), serving as a guidance mechanism for exploring the connections between different embedding tokens or, generally, distinct regions in the embedding.

Visualization. To visualize the overall 3D network structure, we rely on a combination of direct volume rendering [30] and physically-based volumetric path tracing [28]. The main tasks the visualization addresses at this stage are:

- understanding the overall structure of each embedding,
- identifying the number of distinct and significant clusters in the analyzed embedding, as well as their shape, and
- recognizing whether different embeddings contain similar structural patterns and on what scales they are present.

In the remainder of this section, we focus on a qualitative analysis of the global and local embedding datasets introduced in Section 3, through the lens of the above tasks. Section 5 then focuses on a finer, token-level exploration of these data.

Global embeddings. Even though the token data in W2V-300k appear disorganized on the first glance (Figure 2), MCPM reveals

rich network-like geometry. This structure is chaotic, even fractal, with filaments folded into themselves – probably as a result of compressing the high-dimensional embedding data into 3D. The structures exist at the level of entire clusters, rather than interconnecting individual words, although this might be a limitation of the underlying lattice resolution.

Even though some outlying tokens are present, the vast majority of these data forms a single densely interconnected network. This might be a reflection of how interlinked the source Wikipedia corpus is. We observe two recurrent features: **filaments** and **knots**. Some token clusters are distributed along the filaments, while others lie in the knots where multiple filaments intersect. We also notice different strengths (densities) of filaments, usually in proportion to the number of tokens contained within them.

Local embeddings. Since each local embedding corresponds to a specific word, we have generated the BERT embeddings for several interesting words and then chose three representative ones: *wind* (a homonym), *back* (a polysemous word), and *research*. The geometric structures that MCPM infers here (Figure 3) are similar to W2V-300k, but exist on a smaller scale: the filaments interconnect words or groups of words, rather than entire clusters.

MCPM also functions as an implicit **clustering mechanism**. UMAP is designed to preserve components during the dimensionality reduction, but clustering them is not trivial due to their irregular shapes. MCPM manages to identify and interconnect the clusters (as a function of the model’s characteristic feature scale). Some clusters are interconnected densely similar to W2V-300k, others are sparse and branchy, perhaps indicative of the usage patterns of these words. The number of connected components also matches intuition: words like *wind* and *back* have multiple discrete contexts, while *research* has only one but very broad context. General terms like *class* and *suit* yield more than 10 components. Further discussion on clustering is provided in Section 6.

5 EXPLORING THE EMBEDDING

Having extracted the trace field representing the **transport network** over the embedded tokens, this section covers the mechanisms of navigating this structure and the resulting word similarity measure we propose based on it. For this purpose we adopt the following metaphor: we interpret the transport network as an electrically conductive medium. Filaments with high density offer high throughput, while low-density areas (lacunae) have high resistance. The shortest path in such a structure is one that minimizes travelling distance and maximizes throughput.

Navigating the trace. We deploy an agent-based algorithm inspired by MCPM (see Section 4), but significantly simplified. We will refer to the agents of this process as **probe agents**. The main difference is that probe agents traverse the already detected trace field without modifying it. Second, their geometric behavior is more basic in comparison to MCPM agents.

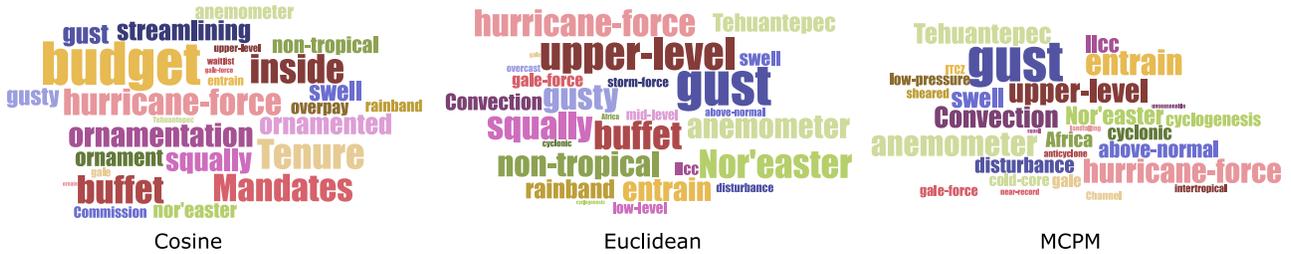


Figure 4: Word clouds of top 30 most similar words for *wind_NOUN* in the W2V-300k according to three similarity metrics. The bigger the word, the more similar it is to *wind_NOUN*.

Table 1: Ranking difference list between MCPM, Euclidean and cosine rankings. The entries are ordered by difference of MCPM and Euclidean rankings.

| Word | MCPM Rank | Euclid Rank | Cosine Rank |
|---------------|-----------|-------------|-------------|
| unseasonable | 28 | 271 | 908 |
| near-record | 26 | 262 | 796 |
| anticyclone | 25 | 65 | 181 |
| intertropical | 24 | 44 | 125 |
| squall | 29 | 49 | 138 |

Each step of the probe agents consists of two phases: **sensing** and **steering**. In the sensing step, an agent samples values p_0 and p_1 from the trace (Figure 5-left). The sample distance *sense_distance* is determined prior to the simulation. The value p_0 lies along the agent’s current movement direction, while p_1 is sampled from a cone determined by a constant *sense_angle*. Then in the steering step, the agent makes a decision whether to turn or not based on the probability proportional to p_0 and p_1 (Figure 5-right). If the agent turns, its new movement direction is then changed by $0 < \text{random_angle} < \text{sense_angle}$ towards the sensing direction, with *random_angle* sampled uniformly in the given interval.

Each probe agent’s behavior is a **random walk process**. Due to the probabilistic steering step, the trace guides the agents so that they effectively follow the transport network structure. Our typical random walk search uses 900 probe agents, each performing 500 steps. We consider a token ‘discovered’ if any of the agents passes around it within a small distance, usually between 1/400 and 1/200 of the domain size.

Trace-guided exploration. Figure 6 demonstrates the impact of the trace guiding, in comparison to unguided, purely random search. With trace guiding, most agents follow a few distinct paths to discover the surrounding token clusters. Without guiding, the random-walk process ends up being equivalent to the nearest neighbor search: the likelihood of a token being discovered decreases as a square of distance from the origin, as the agents become more spread-out. The two marked regions A and B in Figure 6-right, illustrate this contrast: from the random walk density we see that region A is more thoroughly explored than B in spite of both having a similar Euclidean distance from the source. This translates to A

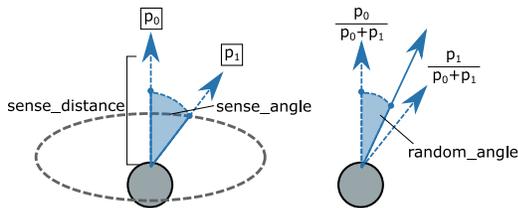


Figure 5: Illustration of the probe agent behavior: sensing phase (left) and steering phase (right). The values p_0 and p_1 are sampled from the trace field.

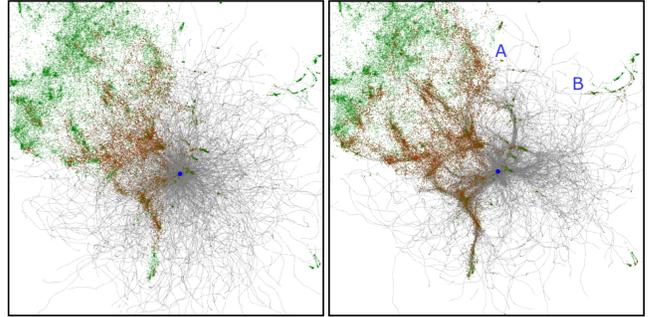


Figure 6: MCPM agent exploration results for token *class_NOUN* in W2V-300k comparing unguided (left) and trace-guided traversal (right). The agents are spawned at the same starting position (blue dot) and their trajectories are marked in grey. They are set out to discover the green data points (tokens), which are marked in red when discovered. To avoid cluttering we only draw a subset of the token data (within a narrow slice centered around the starting point), but still draw all the agent trajectories to emphasize the patterns of their movement.

being closer within the paradigm of optimal transport.

Similarity ranking. Using word embeddings, we can evaluate the relations between words geometrically. For word similarity, the two widely used similarity metrics [3, 7, 16, 24] are the standard Euclidean distance $d_{Euclid}(v_1, v_2) = \|v_1 - v_2\|$ and cosine similarity $d_{cos}(v_1, v_2) = v_1 \cdot v_2$. Cosine similarity assumes that two words represent directions on an N-dimensional hypersphere: the closer the directions, the more similar the words. The implication of this metric is that the spatial distance between two data points matters less than their direction from the origin.

To provide a **structure-aware measure**, we define our similarity metric by how reachable one data point is from another. In contrast with cosine and Euclidean similarity, MCPM similarity is defined by connectedness rather than their pure distance in space. In other words, the closeness of two data points is measured by whether other data points lay down a clear path between them. To this end we deploy agents from a chosen source point (Figure 6). If a point is found sufficiently close to an agent at any simulation step, a counter for that point is incremented. The resulting similarity to the source data point is proportional to the value of this counter at the end of the search (normalized over all discovered points).

To explore this new similarity measure, we choose *wind_NOUN* in W2V-300k to generate the similarity rankings based on the three different metrics. We discuss how the metrics fare in section 6, but some distinctions can be gleaned from the word clouds of the top 30 discovered tokens shown in Figure 4 and the difference list between MCPM ranking and other two rankings shown in Table 1. The entries are ordered by descending difference between the MCPM ranking and Euclidean ranking.

Table 2: Three samples from each of the three major clusters detected in BERT-back. See Figure 7 for the corresponding visualization.

| Top cluster |
|--|
| Partition walls constructed from fibre cement <i>backer</i> board are popular as bases for tiling in kitchens or in wet areas like bathrooms. |
| At one time a firm called Submarine Products sold a sport air scuba set with three manifolded <i>back-mounted</i> cylinders. |
| The harnesses of many diving rebreathers made by Siebe Gorman included a large <i>back-sheet</i> of reinforced rubber. |
| Bottom-left cluster |
| Mono Lake is believed to have formed at least 760,000 years ago, dating <i>back</i> to the Long Valley eruption. |
| Other settlements were Toro, in the extreme south, 1827, and Noble, in the north portion, dating <i>back</i> to the 1830s. |
| Early history: The area comprising the city of Bell has a Native American history dating <i>back</i> thousands of years. |
| Bottom-right cluster |
| Decisions must be unanimous: any divided decision sends the question <i>back</i> to the House at large. |
| He ends by saying that, if he does not hear <i>back</i> from Romani, he will not write to him again. |
| Cartoons often would be rejected or sent <i>back</i> to artists with requested amendments, while others would be accepted and captions written for them. |

6 DISCUSSION

Clusters in contextual embeddings. The contextual embeddings visualized in Figure 3 show a clear separation of clusters. MCPM acts as a robust clustering method here, in spite of their highly irregular shape. We identify these clusters visually as components (sub-networks) interconnected by MCPM. Specifically, two tokens belong to the same cluster if one can be reached from the other by following the MCPM trace network. To explore the contents of these clusters, we sample several sample locations inside the embedding BERT-back, and then visualize the resulting searches in Figure 7.

The samples found within each cluster demonstrate clear differences in the word usage patterns (see Table 2). The irregular top cluster usages of *back* as an indication of spatial relation. Both bottom-left and bottom-right clusters demonstrate *back* as verbal particles used in phrasal verbs. The smaller cluster in the bottom-left shows usages of *back* as a movement in time. Finally, the large bottom-right cluster indicates directionality of communication.

The separation of clusters as seen for polysemous words like *back* and *wind* (see Figure 3) indicates clear boundaries of these volumes and hints on the number of distinct contexts in which these words occur. MCPM similarity is useful here to not only identify the clusters, but to allow for their efficient exploration starting at arbitrary seed points within the clusters.

Knot words versus filament words. With the structural information carried by the MCPM trace, we observe two distinct types of data points: **knot words** and **filament words**. As their names suggest, knot words are words positioned inside clusters and their emitted agents travel with no clear directionality, while filament words are positioned on the paths connecting clusters. For instance, the position of *research_NOUN*, visualized in Figure 8, stands in contrast with that of *class_NOUN*. We observe a clear difference in the distribution of agents' travel directions.

So far, the identification of these concepts has been based on the visual analysis (Section 4). To enable automatic extraction of these properties, we propose to measure the directional statistics of agents spawned by a given query token. Per intuition provided by

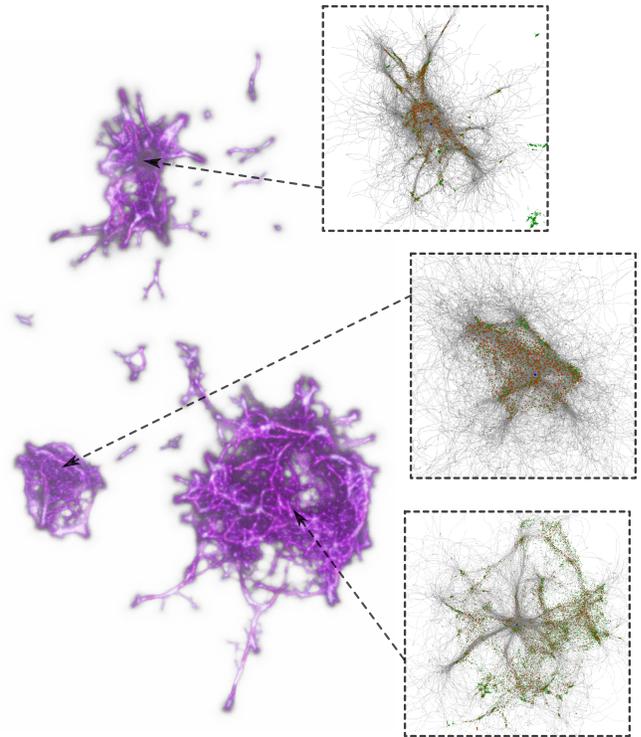


Figure 7: Visualization of intra-cluster exploration for BERT-back, starting in locations inside each respective cluster. We observe distinct topologies within each cluster, corresponding to the different contexts of word *back* captured by the embedding (see Table 2).

the visual analysis, the directional distribution of probe agents from a filament word should be bi-modal, while knot words should yield more complex multi-modal distributions. Based on these criteria, we plan to study these properties in bulk, and determine their origins and semantic significance.

Similarity ranking. The three compared rankings emphasize different mathematical relations. Cosine similarity measures orientation with respect to origin, Euclidean similarity measures geodesic distance in a homogeneous space, and MCPM similarity builds on the optimal-transport throughput. Our aim is to explore how these properties can be used for a more intuitive way to understand machine-generated language representations. Based on the similarity rankings, different properties seem to highlight some word tokens that others do not. Curiously, the cosine similarity shows many items not found in other rankings. For the query *wind*, words such as *budget*, *ornamentation* and *overpay* seem rather out-of-place but are still placed highly in the ranks. Euclidean and MCPM rankings have much more agreeable candidates in higher ranks such as *gust*, *hurricane-force* and *anemometer*. Many of them are specialized terms in climatology. This implies that geodesic distance is an important factor when considering word closeness in W2V-300k.

We also observe a divergence between Euclidean ranking and MCPM ranking in Table 1. MCPM similarity manages to capture words quite far away from the source point. Interestingly, the two words *unseasonable* and *near-record* are still consistent with the general climatological theme of the similarity ranking. Words like *anticyclone* and *intertropical* are very similar to many of the top candidates such as *cyclogenesis* and *non-tropical*. This finding seems to suggest that spatial distance is also an imperfect measurement of similarity, and the measurement of similarity should also consider the throughput between words. This of course needs to be verified by a broader, quantitative study in the future.

It is also important to realize that the definition of similarity as

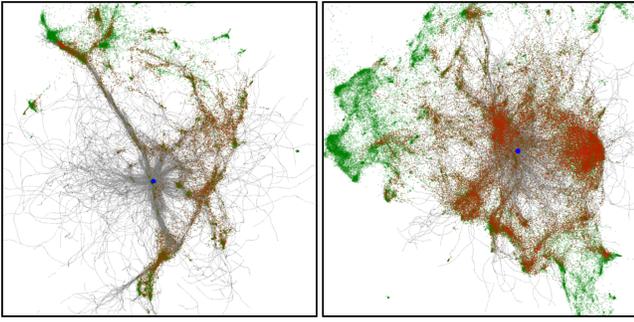


Figure 8: MCPM agent exploration results for *class_NOUN* (left) and *research_NOUN* (right). The comparison illustrates the difference between filament words (left) and knot words (right), which can be differentiated by the clearly articulated paths of the probe agents, as seen in the left figure.

such is rather vague semantically. Computational linguistics distinguishes between the concept of association and similarity. While one would agree that *tropical* is more similar to *wind* than *overpay*, we can only claim they are more similar because it's easier to associate the word *tropical* to *wind*. At the same time, the word *gust* and *squall* can be said to be associated with, but also similar to *wind* [14]. It remains an open question whether there is a way to extract the distinction between associated and similar words in word embeddings.

Since the Monte Carlo exploration is a stochastic process by definition, it is important to address the stability of our similarity ranking. On the word embedding level, studies have shown embedding algorithms to be non-deterministic even with the same training data and configuration [12]. On the trace generation level, and the subsequent trace-guided exploration by probe agents, we solely rely on converged aggregate solutions. In the MCPM similarity ranking, we observe that the results are fairly stable as each ranking only shifts by 1 to 2 places between different results. From this we conclude that the probabilistic nature of our framework does not render the results unreliable, especially considering that even human-produced rankings tend to have significant inter-subject variation.

Origin of the structures. Ultimately, an important question to consider is to what extent are the discovered transport networks and resulting structures inherent to the embeddings. It has been shown that non-linear methods such as t-SNE and UMAP distort pairwise relationship between embeddings, while PCA can introduce false positive parallel pairs in its result [19]. The important consideration is therefore the choice of the projection method: we chose UMAP because of its recognized ability to preserve both global and local structures, as well as the number of components in the source data. This is in contrast to other methods like t-SNE and PCA, which are known to destroy all global and local structure, respectively. Our experiments with different configurations of UMAP also showed that regardless of the resulting projection shape, the structures were present and reliably recovered by MCPM. Finally, the fact that MCPM similarity yields reasonable results even in 3D is very encouraging, and motivates us to look for solutions that operate in the native, high-dimensional embedding space.

7 CONCLUSION

In this exploratory paper, we have investigated how a bio-inspired random walk model pioneered for structure-finding in astronomy [4, 10] can help us identify and visualize interesting geometric configurations in word embeddings. We show that this method can reveal a range of potential structural factors in the embedding space, including the number of components for a word, as well as its neigh-

borhood structure. Our approach combines a holistic view of an embedding dataset as an optimal transport network, while enabling us to pay attention to structures on the word and cluster level.

We believe these results are a promising start to a new line of research, and a great deal of work lies ahead of us to understand if these structures correspond to meaningful semantic distinctions. Building a robust interpretation for the information encoded in the embeddings would also improve natural language processing tasks like word similarity and disambiguation.

We are currently in the process of redesigning the MCPM simulation to run in the original high-dimensional spaces instead of a reduced one, to overcome the question which of the detected structures are valid and which ones are induced by the dimensionality reduction. This challenging extension is also attractive for other application domains which make use of embeddings, for instance recommender engines, music processing, game state space exploration, and generative systems.

We also plan to more deeply probe the filament/knot distinction. We intend to develop a concrete mathematical characterization of these two concepts – something that is needed to automatically extract them and explore to what extent is this information semantically meaningful and, in the long run, to understand where it comes from. This understanding could be instrumental for structural comparison of different text corpora and even entire languages.

REFERENCES

- [1] <http://projector.tensorflow.org/>.
- [2] E. Amid and M. K. Warmuth. Trimap: Large-scale dimensionality reduction using triplets, 2019.
- [3] M. Berger, K. McDonough, and L. M. Seversky. cite2vec: Citation-driven document exploration via word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):691–700, 2016.
- [4] J. N. Burchett, O. Elek, N. Tejos, J. X. Prochaska, T. M. Tripp, R. Bordoloi, and A. G. Forbes. Revealing the dark threads of the cosmic web. *The Astrophysical Journal Letters*, 891(2):L35, 2020.
- [5] A. Coenen, E. Reif, A. Yuan, B. Kim, A. Pearce, F. Viégas, and M. Wattenberg. Visualizing and measuring the geometry of BERT. *arXiv preprint arXiv:1906.02715*, 2019.
- [6] C. Collins, F. B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE Symposium on Visual Analytics Science and Technology*, pp. 91–98, 2009.
- [7] X. Dai and R. Prout. Unlocking super bowl insights: Weighted word embeddings for twitter sentiment classification. In *Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016*, pp. 1–6, 2016.
- [8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [9] W. Dou, X. Wang, R. Chang, and W. Ribarsky. Paralleltopics: A probabilistic approach to exploring document collections. In *IEEE Conference on Visual Analytics Science and Technology*, pp. 231–240, 2011.
- [10] O. Elek, J. N. Burchett, J. X. Prochaska, and A. G. Forbes. Monte Carlo Physarum Machine: An agent-based model for reconstructing complex 3d transport networks. In *Artificial Life Conference Proceedings*, pp. 263–265. MIT Press, 2020.
- [11] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
- [12] F. Heimerl and M. Gleicher. Interactive analysis of word vector embeddings. *Computer Graphics Forum*, 37(3):253–265, 2018.
- [13] J. Hewitt and C. D. Manning. A structural probe for finding syntax in word representations. 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)*, pp. 4129–4138, 2019.
- [14] F. Hill, R. Reichart, and A. Korhonen. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

- [15] X. Ji, H. Shen, A. Ritter, R. Machiraju, and P. Yen. Visual exploration of neural document embedding in information retrieval: Semantics and feature selection. *IEEE Transactions on Visualization and Computer Graphics*, 25(6):2181–2192, 2019.
- [16] X. Ji, H.-W. Shen, A. Ritter, R. Machiraju, and P.-Y. Yen. Visual exploration of neural document embedding in information retrieval: semantics and feature selection. *IEEE Transactions on Visualization and Computer Graphics*, 25(6):2181–2192, 2019.
- [17] K. Kucher and A. Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *IEEE Pacific Visualization Symposium*, pp. 117–121, 2015.
- [18] Y. Lin, Y. C. Tan, and R. Frank. Open sesame: Getting inside bert’s linguistic knowledge. *arXiv preprint arXiv:1906.01698*, 2019.
- [19] S. Liu, P. Bremer, J. J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, and V. Pascucci. Visual exploration of semantic relationships in neural word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):553–562, 2018.
- [20] Y. Liu, E. Jun, Q. Li, and J. Heer. Latent space cartography: Visual analysis of vector space embeddings. *Computer Graphics Forum*, 38(3):67–78, 2019.
- [21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.
- [22] J. Mu, S. Bhat, and P. Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.
- [23] D. C. Öndüğü, H. Kuşçu, and E. Aygün. History of philosophy: Summarized & visualized. <https://www.denizcemondüğü.com/philolo/>.
- [24] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist. Conceptvector: Text visual analytics via interactive lexicon building using word embedding. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):361–370, 2018.
- [25] B. H. Partee, A. ter Meulen, and R. E. Wall. *Mathematical Methods in Linguistics. Corrected first edition*. Kluwer Academic Publishers, 1990.
- [26] I. Perez-Messina, C. Gutierrez, and E. Graells-Garrido. Organic visualization of document evolution. In *International Conference on Intelligent User Interfaces*, p. 497–501, 2018.
- [27] G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- [28] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 3rd ed., 2016.
- [29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.
- [30] C. Rezk-Salama. *Volume rendering techniques for general purpose graphics hardware*. PhD thesis, Universität Erlangen-Nürnberg, 2001.
- [31] A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in bertology: What we know about how BERT works. *arXiv preprint arXiv:2002.12327*, 2020.
- [32] S. Simha, J. N. Burchett, J. X. Prochaska, J. S. Chittidi, O. Elek, N. Tejos, R. Jorgenson, K. W. Bannister, S. Bhandari, C. K. Day, et al. Disentangling the cosmic web towards FRB 190608. *arXiv preprint arXiv:2005.13157*, 2020.
- [33] I. Subašić and B. Berendt. Web mining for understanding stories through graph visualisation. In *IEEE International Conference on Data Mining*, pp. 570–579, 2008.
- [34] F. van Ham, M. Wattenberg, and F. B. Viegas. Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1169–1176, 2009.
- [35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
- [36] C. Villani. *Optimal Transport: Old and new*. Springer, 2009.
- [37] G. Wiedemann, S. Remus, A. Chawla, and C. Biemann. Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint arXiv:1909.10430*, 2019.
- [38] G. K. Zipf. *Human behavior and the principle of least effort*. Addison-Wesley, 1949.