UNIVERSITY OF CALIFORNIA
SANTA CRUZ

ISLANDS AND BRIDGES OF LANGUAGE: BIO-INSPIRED STRUCTURAL ANALYSIS OF LANGUAGE EMBEDDING DATA

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTATIONAL MEDIA

by

Hongwei (Henry) Zhou

March 2022

The Thesis of Hongwei (Henry) Zhou
is approved:

Associate Professor Angus G. Forbes (Advisor)

Assistant Professor Adam M. Smith

Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Islands and Bridges of Language:

Bio-inspired Structural Analysis of Language Embedding Data

by

Hongwei (Henry) Zhou

In this thesis, I propose a method of applying an agent-based model named Monte Carlo Physarum Machine (MCPM) to language embedding data. This method has been previously applied in astronomy for inferring the quasi-fractal structure of the cosmic web. In this thesis, I show that this model can provide a distinct scope to understand, analyze and extract information from language embedding data. I assess the novelty of the algorithm first by identifying the characteristics of the revealed structure through visualization, and generate word similarity metrics in comparison with other status quo similarity metrics. In addition, I propose a visualization tool to further help explore the language embedding space in 3D. As a result, I argue that both the MCPM method and the visualization tool can assist examining the structure of language embedding in the reduced 3D space.

To my parents and friends,

and those who didn't find me annoying.

Acknowledgments

I want to acknowledge and thank Oskar Elek for being such an important contributor to this project. This project was not possible without his technical knowledge and encouragement.

I also want to express my gratitude to Montana Fowler for being one of my best friends in grad school to this day. Her continuing support of my pursuit has been irreplaceable in my graduate school journey.

In addition, I'd like to extend my appreciation to all of my cohorts in graduate school, as well as faculties who supported me throughout this work.

Chapter 1

Introduction

Word embedding is a family of algorithms that transforms words into a set of numbers that supposedly embed their semantic and syntactic content, where semantic content correlates to meaning of the word, while syntactic content correlates to their structural roles [34]. The algorithm imagines that word tokens' content can be interpreted as points in a continuous space, and their distribution can be mathematically produced through input data, by looking at the usage of each word. The key assumption is that words used in similar context tend to have similar semantic and syntactic content [26]. The word tokens with similar content are thus positioned closely in the continuous space. The points with similar values can be interpreted as being used in similar context. This process can be carried out in either frequency-based or prediction-based methodologies [34]. The output of word embedding algorithms has been widely useful for many downstream natural language processing tasks such as part of speech tagging [14], named entity recognition [50] and machine translation [15].

As mentioned, word embeddings are points distributed in a continuous space with arbitrary dimensions. Each point refers to a word token. This naturally raises questions about the meaningfulness of many geometric properties in the continuous space [41, 57]. Specifically, many studies of the geometric arrangement of word embedding focus on two main mathematical units for analysis: offset vectors and clusters [27]. Offset vectors allow the linear translation from one point/word to another point/word, while clusters focus on the spatial proximity between points [37, 31, 8]. Since the word embedding algorithm itself presupposes that words positioned near each other are similar semantically and syntactically, clusters become an important part of word embedding examination [9]. Offset vectors, on the other hands, were discovered to embed human-interpretable meaning such as gender switching (*man* - *woman* $\approx$ *king* - *queen*). Offset vectors are also termed *word analogies*, and have become a wide-spread mathematical unit to discover structures in word embedding [27]. Both mathematical units rely on linear relations (translation and Euclidean distances) to understand the relationship between words.

In this thesis, I present the preliminary work of applying Monte Carlo Physarum Machine, MCPM for short, as a way to visualize, interpret and make sense of word embedding. MCPM is an agent-based model inspired by the self-organizing characteristics of slime mold, initially studied by Jeff Jones [32]. It was then modified by Burchett et al. with an additional Monte Carlo decision-making process, and was shown to be empirically accurate in predicting the pattern of cosmic web of the universe [10, 62], where it has successfully recovered the theoretically predicted filamentary patterns over sparse

galaxy data. MCPM can be understood as bio-inspired modeling of optimal transport networks. The mathematics of optimal transport [69, 52] is based on the principle of least effort, which applies to phenomena ranging from particle and light transport to the behavior of living beings. As such, MCPM can be understood as an alternative way to discover structures in word embedding. Specifically, the optimal transport network is not constrained to linear relations like the aforementioned two mathematical units, offset vectors and clusters, are. For this purpose, I seek to investigate the potential application of MCPM or slime mold in the context of language embedding [22].

My method stands in contrast with the on-going investigations of language embedding data in natural language processing communities. First of all, MCPM highlights non-linear and more global structural relations comparing to offset vectors and clusters, which only consider local pair-wise relations or local neighborhood, this distinction will be further clarified in Section 2. Second of all, I do not conduct my research with the motivation to find a specific linguistic phenomenon, or to apply MCPM to improve performance of a specific downstream NLP tasks such as machine translation. Rather, I am interested in a more structure-focused and exploratory approach: I deploy my method, which has shown to discover a very specific structure, to language embeddings, and see what linguistic properties this method reveals.

In order to apply MCPM and visualization methods, a dimensionality reduction technique called UMAP is used to reduce the original word embedding data to 3D. This raises two concerns: 1) whether the structural pattern extracted in the reduced-dimensionality space also exists in the original space, and 2) since UMAP is stochastic,

how can we understand what is invariant across multiple outputs. I address the first concern by comparing the results from the reduced dimensions to the status quo measurement in the original dimensions, and the second concern by generating two datasets under the exact same condition, and try to discover structures that appear in both of them using a visualization tool.

Thus, the contribution of this thesis is divided into two parts: information retrieval with MCPM and a visualization tool for language embedding. For the information retrieval aspect, I examine the behavior of MCPM probing through visualization results. The behavior shows that MCPM does not only reveal structures based on proximity, but also the connectivity of the dataset. I then interpret MCPM probing as a way to retrieve word similarity lists in language embedding. MCPM probe method reveals that both connectivity and euclidean distance embed salient structural information in the reduced 3D dataset. For the visualization tool aspect, I demonstrate the potential the tool affords by identifying salient structures in the reduced 3D language embedding dataset. During this process, the observation using the visualization tool also supports the hypothesis that data connectivity embeds salient information in the reduced 3D dataset. The earlier version of this work was published in IEEE VIS4DH 2020 workshop [72]. It was a preliminary result on the information retrieval aspect of this thesis.

Chapter 2

Literature Review

To reiterate, this thesis focuses on a specific structure discovery algorithm, namely optimal transport networks and Monte Carlo Physarum Machine, and investigates its potential for contribution in existing natural language processing data, specifically language embedding in this thesis. My approach rests on a foundation spanning two fields: information retrieval and information visualization.

For this purpose, I break down this section into three sections. Section 2.1 first introduces the general background of word embedding and its influence on the natural language processing community today. Section 2.2 focuses on the information retrieval aspect of language embedding, where I highlight the distinction between extrinsic and intrinsic evaluation, as well as the desire for more complex evaluation. Section 2.3 focuses on the relevant visualization tools used to evaluate word embedding, and mentions particular tools from which I take direct inspiration for my own visualization tool.

2.1 Word Embeddings

Much work extracting meaning from text has relied on relational structures that can be represented (and visualized) as graphs. Phrase Nets [66], 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 graph visualization, such as documents [51], stories [64], or even ideas [46] with suitable relational axioms applied to them. At a more granular level, syntactic relations in linguistics are often represented as graph diagrams [49], as are the ontological relationships between words [24]. 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.

In recent years, word embeddings, such as Word2Vec, GloVe, and ELMo, have gained 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. 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 [40, 12]. Because they explicitly consider the token's context [18, 54], it has been shown that embeddings contain information that can be processed to extract a range of useful properties: clustering by token usage [57, 71] as well as different kinds of syntactic information [57, 35, 28]. Thus, there is the promise

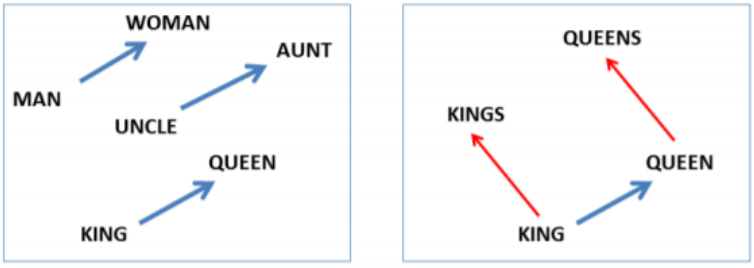

Figure 2.1: **Linguistic regularities shown through vector offsets, credit to Mikolov et al. [41].**

that this kind of method could provide high-dimensional representations that encode a large manner of relations implicitly without having to hand-code them in advance.

Consequently, encoding language using vectors provides affordance to analyze linguistic regularities within language embedding data. Most notably, Mikolov et al. discover that an offsetted vector can be used to describe syntactic and semantic properties in the embedding [41]. One such example is shown in Figure 2.1. The left subfigure shows similarity among vectors that describe binary gender relation (masculine-feminine) between two words. The right figure demonstrates how two vectors, one describing singular-plural relation and the other binary gender relation, can be used to navigate within the word embedding space from "Kings" to "Queens." Fundamentally, this specific phenomenon discover that analogies (word pairs) embed regularities in natural language.

To enable similar discoveries through word analogy, tools aimed to help exploring word relations are proposed for literary experts and natural language processing researchers [37, 27]. These focus on exploring linear relationships between word embed-

dings, identifying concepts and experimenting with attribute vectors. Notably, Heimerl et al. conduct a survey on the usage of word embeddings and identify key tasks and applications. As a result, Heimerl et al. find that, in addition to word pairs as basic units of analysis, grouping words based on spatial proximity can be another way to analyze word embedding [27]. My proposal of the MCPM probe method follows this line of inquiry of finding structural regularity within language embedding. The main innovation is that MCPM, comparing to both word pairs and spatial proximity, goes beyond the limit of euclidean distances and linear transformations. It is sensitive to the connectivity within the data itself, which is not necessarily linear.

Most notably, Vaswani et al. propose a novel model named transformers to generate word embeddings. The main innovation of the transformer model is to replace the traditional sequence modeling, which linearly processes through the text, to attention mechanism, which creates word embeddings by looking at context of the entire sentence [68]. In short, transformer models have the advantage of global information tracking instead of local processing algorithms. Transformer models gained much attention for its state-of-the-art performance in many domains. Surprisingly, when used in self-supervised pretraining methods, transformers have shown superior performance than many directly supervised models in downstream natural language processing tasks such as question answering, machine translation, reading comprehension, and summarization [54]. Similar discoveries were also made in image representation [11]. Because transformers can also be used as generative models, many studies and applications have used it for creative tasks. For example, AI Dungeon and AI Dungeon 2 utilize BERT

to create a choose-your-own-adventure games that respond to arbitrary player text prompts and generate unexpected and oftentimes hilarious responses [1]. Jukebox also utilizes transformer architecture to generate many song samples based on lyrics, music genre and artist style [19]. Most recently, DALL·E utilizes GPT 3, the most advanced transformer model, to develop a decoder-only architecture to allow text-to-image generation [55]. The resulting images are found to be surprisingly coherent even when the input text combines unrelated concepts. Word embedding generated by transformers is examined lightly in my thesis, particularly in section 4.1. The majority of the thesis is devoted to studying the applicability of MCPM to global (context-independent) word embedding methods such as Word2Vec, rather than contextualized word embedding methods such as transformers.

2.2 Evaluation of Word Embeddings

As word embedding algorithms are adopted and utilized, the notion of quality naturally develops to guide researchers to improve and evaluate word embedding algorithms. The distinction between *intrinsic* and *extrinsic* evaluation methods is used to differentiate the two families of quality assessment [59, 6].

Extrinsic evaluation methods binds the quality of the word embedding to its utility to a specific downstream language processing task, such as name entity recognition, sentiment analysis and semantic role labeling [59]. By this definition, any natural language processing task that can integrate with word embedding is an extrinsic evalu-

ation method. This naturally raises the problem on the narrowness of each evaluation method and the incommensurability among different extrinsic evaluation scores, as different language tasks highlight different features in the word embeddings. In their survey of language embedding evaluation methods, Barakov argues that no global evaluation score exists through extrinsic evaluation due to "the lack of performance correlation on different downstream tasks" [6].

In contrast, intrinsic evaluation method binds the notion of quality to its closeness to human cognition: embeddings are measured against human judgements. Most notably, datasets such as SimLex-999 [29] are created by human subjects, in the format as a list of words considered semantically proximate to a given word. However, Gladkova et al. express concern on the vagueness of semantic proximity as it conflates different linguistic features such as semantic similarity, relatedness, morphological relations (plural, tense) and collocations [25]. Barakov also complicates the datasets as different datasets are generated from different cognitive processes. As a result, they separate intrinsic evaluation datasets to conscious and subconscious evaluations [6]. In addition to testing against a dataset, another category of intrinsic evaluation uses human as direct judgement of word embeddings. For example, the subjects compare two word embeddings to see which similarity list is more intuitive. The consistency of word embeddings can also be evaluated by how fast a human subject can pick out a randomly inserted word in a similarity list [25].

The two mathematical units have dominated the interpretive work in word embeddings: analogies through word pairs (offset vectors) and grouping based on prox-

imity (clusters). They have become essential to describe the underlying structure of word embedding. So much so that dimensionality reduction methods have come under scrutiny in terms of its ability to preserve word analogies. Liu et al. propose modification to existing dimensionality reduction method in order to highlight the two features in visualization [36]. The local neighborhood is emphasized as a major comparative determinant of embedding quality in Embedding Comparator [9]. Heimerl et al., based on their identification of these two basic units for embedding analysis, develop visualization techniques to better compare and measure qualities based on local neighborhood and concept axis (a vector from one neighborhood to the other) [27]. We also see this in the intrinsic evaluation listed above, where the most prominent evaluation is around the concept of word similarity, a measurement defined based on spatial proximity. Many of the information extraction also focus on human-identifiable linguistic traits: a vector that codes the transformation from singular to plural, or a cluster that designates animal names. This leads to Gladkova et al. to call for a more sophisticated way to interpret word embeddings: "...the most perfect word embedding is unlikely to have exactly the same 'concepts' as us...by focusing on the structures that we expect the word embeddings to have, we might be missing the structures that they actually have." [25].

Going beyond simple mathematical units, Hewitt et al. manage to identify that syntactic tree information can be extracted by linearly transforming contextualized embeddings [28]. Coenen et al. also show evidence that BERT encodes semantic and syntactic information in its sub-spaces representation [12]. I see these two studies as

more sophisticated ways to probe embedding structure as they do not assume the primitives in the space have any interpretable meaning. Rather, they can claim such meaning is embedded somehow that needs to be extracted through linear transformation.

I see my thesis as being similar to works that try to go beyond understanding the structure of language embedding with either offset vectors or proximity-based clustering. MCPM probing method strictly tries to discover structure based on not only distance, but also connectivity of the datasets. Since there is no standard way to evaluate its extrinsic utility, I will not be making any claim about its applicability to downstream tasks. However, I do extract word tokens and argue about the quality of word similarity ranking in later chapters. Therefore, from the perspective of word embedding evaluation, this thesis proposes an intrinsic evaluation method based on MCPM probing, which goes beyond linear relations as basic structural units.

2.3 Visualization of Word Embeddings

As word vectors are points in continuous space, visualization is a natural extension to comprehend word embedding. But the task is complicated due to the their high dimensionality. While parallel coordinates are suitable for high dimensional data [20, 13], 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 [38, 5], optionally with additional semantic annotations [12]. In that process, two different distortions

Figure 2.2: **BERT embeddings of “die” are clustered into different senses, credit to Coenen et al. [12].**

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 [8] and identification of salient dimensions [30] does seem to alleviate some of these issues.

There are many tools designed to explore the reduced word embedding data. Most of which take the form of scatter plots. For contextualized embeddings such as BERT, Coenen et al. visualize the distribution of embeddings of a single word and discover that the clusters and spans of the clusters have human-interpretable meaning to them, meaning that they can identify the general semantic seperation between clusters, as well as from one end to the other end of the cluster as demonstrated in Figure 2.2 [12]. Another notable embedding visualization to directly visualize points in space is the Embedding Projector developed by Tensorflow, it allows the user to visualized word

embeddings in interactive 3D space, and the user can choose different dimensionality reduction techniques to see the difference in the final result [2].

Many language embedding tools attempt to innovate on tasks relevant to embedding exploration. For example, Heimerl et al. identify tasks to extract learned information in language embedding, such as inspecting local neighborhoods (around one single point) or analyze vector relations (the relationship between two points), and develop several different views to facilitate these tasks [27]. To address the specific concern with the stochastic nature of language embedding, as well as various embedding algorithms that can generate vastly different outputs, Boggust et al. propose Embedding Comparator [9]. The tool allows multiple scatter plot views, with some views localizing around a single point for neighborhood inspection, as well as some views taking a more global view with other information such as similarity distribution. The design allows the researchers to gain a better understanding of the difference between two embedding data. The visualization tool in this thesis allows a simple 3D navigation, which is shown to be sufficient for my tasks without the need for multiple views.

Some studies see the potential unreliability in existing dimensionality reduction schemes for studying language embedding and intend to develop new methods in order to preserve the relevant structures in the embedding data. Liu et al. propose a dimensionality reduction method that designed specifically to preserve local neighborhood clusters as well as word analogy vectors [36]. Molino et al. from Uber AI lab propose a simple tool named Parallax to allow users to interactively choose the axes of projection through algebraic formulae [42].

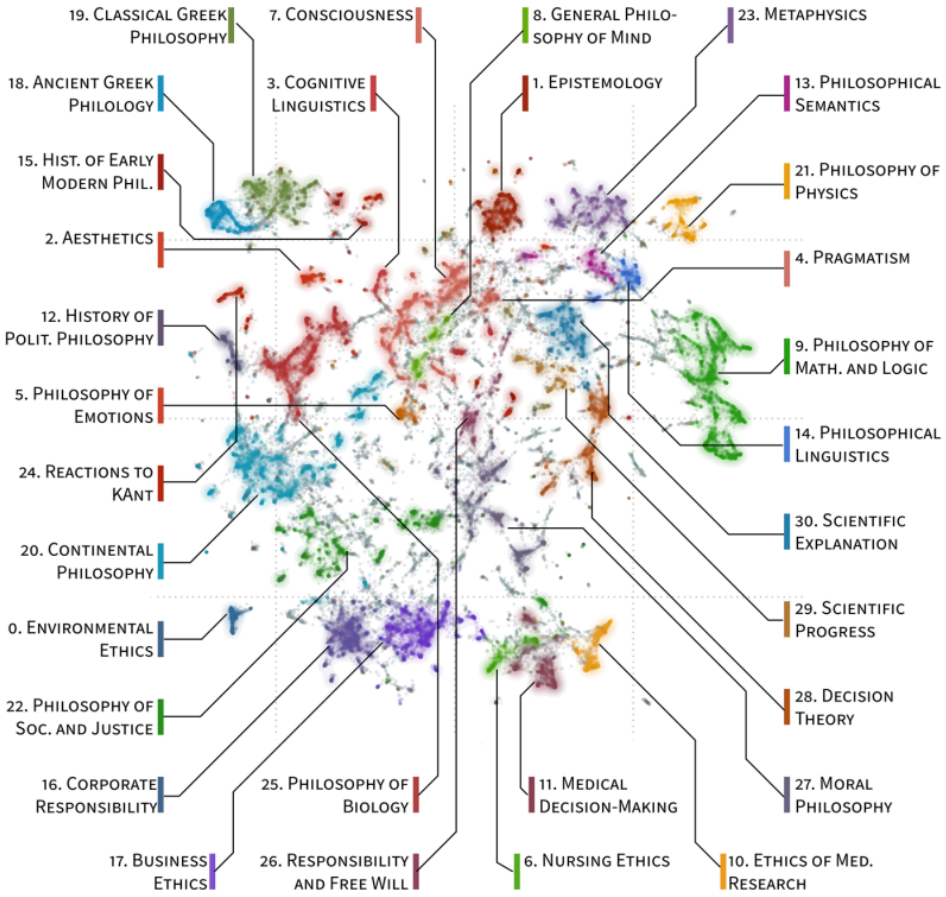

Figure 2.3: **Structure of philosophy visualization by Noichl [45]. UMAP is used to reduce the scatter plot to two-dimensional representation (distribution of points at the center), which allows the author to identify distinct clusters of specialization within philosophy (the text around the central scatter plot).**

For many embedding visualizations, scatter plots with UMAP are fairly effective to conduct studies and learning insights from the data. It has been used to process through philosophical text to understand the association between different traditions of philosophy [45], as shown in Figure 2.3, as well as the general network of academic papers through citations [8]. The visualization tool proposed in this thesis takes direct inspiration from Embedding Projector. It allows users to examine word embeddings by allowing the users to navigate the 3D scatter plot, with additional features to expose not only MCPM probe results as well as relevant linguistic information such as part of speech. For concerns regarding the reliability of preserving structures in dimensionality reduction, I dedicate section 4.3 to examine structures persistent between two datasets produced under the same hyperparameter for dimensionality reduction.

Chapter 3

Method

3.1 Dataset

The original word embeddings used in this thesis are high-dimensional: same as the base model BERT in [18], our BERT embeddings are 768-dimensional. Our Gensim Continuous Skipgram [39] embeddings are 300-dimensional. To make visualization and analysis feasible, we rely on UMAP (neighborhood size of 15) to project the data to 3D space. The dimensionality reduction is necessary, due to the high memory requirements of our simulation of MCPM.

It is important to consider to what extent the discovered structure is inherent to the embeddings. It has been shown that non-linear methods such as t-SNE and UMAP distort pairwise relationships between embeddings, while PCA can introduce false positive parallel pairs in its result [37]. I chose UMAP because it has been shown to strike a balance between preserving global and local structures. This is in contrast

to PCA and t-SNE, which are known to destroy both global and local structures.

To assess the applicability of MCPM and the visualization tool in language embedding data in general, I first select two datasets generated by different language models. The first dataset is generated by Gensim Continuous Skipgram—a variation of Word2Vec—fed with English Wikipedia Dump of February 2017, with approximately 300k word tokens. 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. In the majority of section 4, I focus on this dataset unless otherwise specified. Since the UMAP algorithm is somewhat stochastic, I generated two different result with the same dataset and the same parameter. I'll refer to these two datasets as "W2V-300k-1" and "W2V-300k-2".

The second dataset is generated by BERT fed with Wikipedia text corpus. Different from Word2Vec, BERT generates multiple datasets. Each generated dataset is particular to a single word, and defines the context relative to that word – typically resulting in 1000s of tokens. Each token in the space is a single instance of a particular word being used in a sentence. Different from W2V-300k, the part of speech information is not included in the data, which is a built-in design of most BERT implementations. I generated the BERT embedding dataset with the word *back*, which I will call "BERT-back".

3.2 MCPM Probing and Similarity

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. To put it simply, the trace is stored as a density field in a 3D grid, where the value of each cell represents the thoroughput of the MCPM agents during simulation. This representation of trace is advantageous for my further analysis, serving as a guidance mechanism for exploring the connections between different embedding tokens or, generally, distinct regions in the embedding.

Having extracted the trace field representing the transport network over the embedded tokens, I deploy an agent-based algorithm inspired by MCPM, but significantly simplified. I will refer to the agents of this process as MCPM probe agents. The main difference is that MCPM probe agents traverse the already detected trace field without modifying it. In addition, their geometric behavior is more basic in comparison to MCPM agents.

Each step of the MCPM probe agents consists of two phases: *sensing* and *steering*. The process is illustrated in Figure 3.1. In the sensing step, an agent samples values p_0 and p_1 from the trace. The sample distance $sense\_distance$ is determined

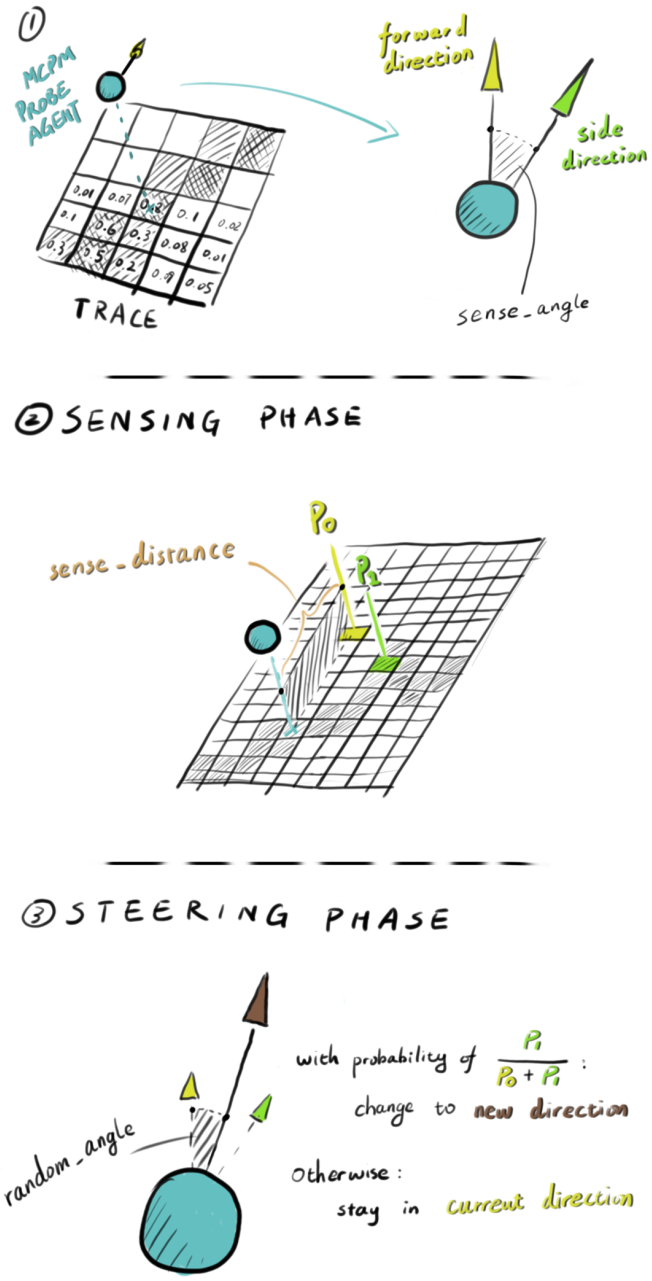

Figure 3.1: **Illustration of the probe agent behavior. The values p_0 and p_1 are sampled from the trace field. Both *sense_distance* and *sense_angle* are determined prior to simulation. More detailed description of MCPM is provided in [10].**

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. If the agent turns, its new movement direction is then changed by $0 < random\_angle < sense\_angle$ towards the sensing direction, with *random_angle* sampled uniformly in the given interval.

Each MCPM 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. I 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.

This probing process is designed from the perspective of one particular data point, or the anchor point. More specifically, the probing agents are spawned on one single anchor point with randomized directions. They explore the trace and mark down discovered points around the anchor. In addition to discovering the point, each data point excluding the anchor point holds a counter, which is incremented every time the agent discovers the point. The result is similar to a sparse graph, where the edge can be denoted as [*anchor*, *data_point*, *count*]. The *anchor* and *data_point* are the two nodes at the end of each edge, while the *count* is the number of times agents manage to discover the data point. The result is a sparse graph because the edge only connects the anchor point to other data points without consideration of connection among non-anchor points. Figure 3.2 demonstrates this sparse graph representation. The counter can be used to calculate the likelihood of traversing from the anchor point to the data

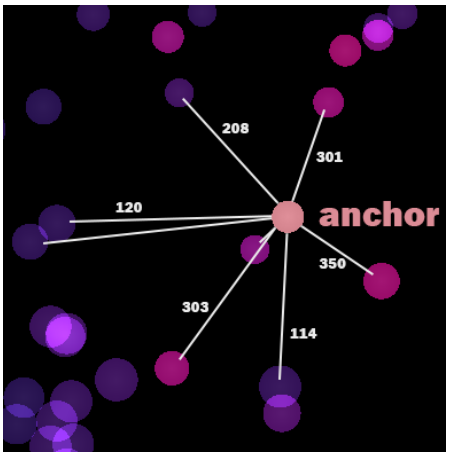

Figure 3.2: **A simplified illustration of MCPM probing results. Edges are built from a chosen anchor point to other data points. The number on the edge denotes how many times MCPM probe agents discover the data point.**

point. Hypothetically, the closer and the more connected a data point is to the anchor point, the higher its counter is.

To apply our method to applications in Natural Language Processing (NLP), I interpret the MCPM probing results as similarity metrics for word embeddings, which I term MCPM similarity. Word similarity is a way to measure similarity of tokens geometrically. One standard practice of measuring word similarity in NLP is Cosine Similarity ($d_{cos}(v1, v2) = v1 \cdot v2$). 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. We also compare our similarity metric to Euclidean similarity ($d_{euclid}(v1, v2) = ||v1 - v2||$) to assess how much spatial distance matters in similarity metrics. For MCPM similarity, the *data_point* thus becomes a candidate that is similar to the *anchor*. The *count* becomes a metric for how

similar the *data_point* is to the *anchor*; the higher the count, the more similar the pair is.

3.3 Visualization Tool

To enable easier examination of the word embedding and the MCPM output, I develop a visualization tool in Three.js [3]. The tool is designed to help freely explore the dimensionally reduced 3D word embedding space, identify significant structures in the word embedding visible through the dimensionality reduction progress or the alternate similarities suggested by MCPM. The visualization tool is broken down to three aspects: 3D word embedding examination, MCPM-based examination and linguistic-based examination.

3.3.1 3D Word Embedding Examination

When the application first launches, the user is presented with the interface as shown in Figure 3.3. The most prominent view is the 3D scatter plot view, where the exploration of word embedding data takes place, with a panel on the top right to provide various options that change the main scatter plot. The 3D scatter plot, mouse navigation and mouse pointer content examination constitute the main interactions given to visually explore the word embedding dataset.

The main scatter plot view is a 3D scatter plot. The user mainly uses the mouse input to navigate the space. The camera view is always pointing towards a focal point. The mouse wheel controls the distance to that focal point, allowing the user to

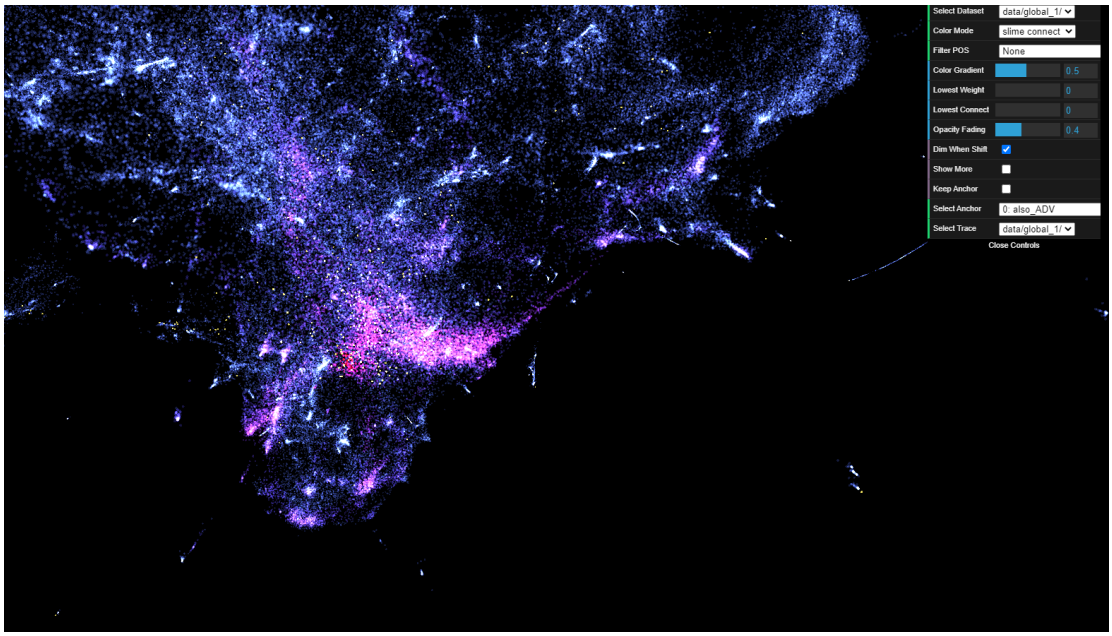

Figure 3.3: **An overview of the slime mold visualization tool. Each dot represents a word embedding. Their coloring, from red to blue, indicates their connected-ness to the focused anchor point. Each yellow point is a possible anchor point to examine.**

zoom in and out between a view on the larger structure and on the local neighborhood. Left mouse button controls the rotation of the camera around the focal point. Right mouse button controls the position of the focal point.

Each point in the scatter plot view represents a single word in the word embedding. To view its content, the user simply hovers the mouse pointer over the point. The content of the point appears in a tooltip next to the point as illustrated in the left figure in Figure 3.4. By default, the point closest to the camera is selected. The user can turn on the *Show More Tokens*, which will show content of all points intersected by the ray cast from the mouse pointer, which is shown in the middle figure of 3.4.

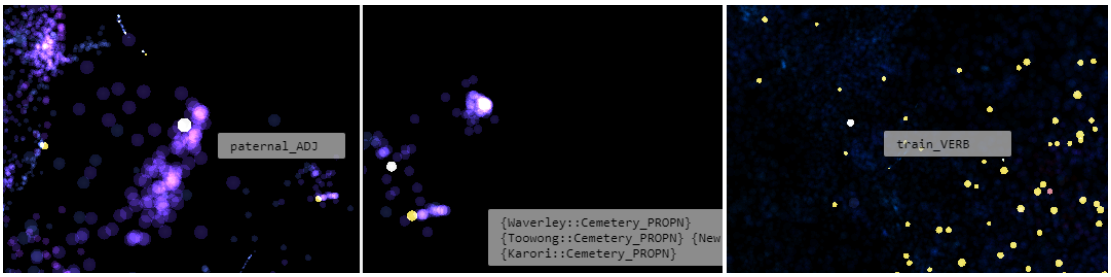

Figure 3.4: **Left: The tool tip displays the word of the point that is hovered over, and is closest to the camera. Center: When *Show More Tokens* is checked, the tool tip displays multiple tokens on the mouse pointer. Right: When left shift is pressed, all non-anchor points are dimmed, and only anchor points (yellow) and focused anchor point (pink) are selectable.**

3.3.2 MCPM-based Examination

Since MCPM simulation is rather demanding of computational resource, I choose to pre-generate the MCPM data for examination. Since there are too many points in the word embedding, I decide to only include a number of points as anchor points - points that we generate MCPM probing results for. The selection criteria is: 1) The top 200 most frequently used words in English, 2) 200 randomly selected words with multiple part-of-speech and 3) 100 randomly selected words based on their position in the data in order to evenly cover the space. Number 2 is done to approximate a list of polysemous words, or words with multiple meanings. Number 3 is done by randomly sample a point in the 3D space and find a word that is closest to it. The MCPM result is mainly presented through the anchor point examination in the visualization tool. To examine the MCPM result, only one anchor point can be focused on at a time. Each anchor point is color-coded as yellow as shown in Figures 3.3 and 3.4.

To switch anchor points, the user simply needs to double left click on the

selected anchor point. The tool will load the MCPM result and re-color all the points. Some areas can have high density of points, which makes selecting anchor points difficult. By pressing *left shift*, the user can switch between anchor points while non-anchor points are dimmed and cannot be selected, as shown in right of Figure 3.4. In addition to switching anchor points in the main scatter plot view, the user can also use the *select anchor* tab in the top-right panel to quickly navigate between anchor points based on their IDs and word string. The list is sorted by IDs in ascending order to allow quick selection.

Each point is color coded, from red to blue, to indicate its connected-ness to the focused anchor point. We can see how it looks in Figure 3.3. There's a small area of red points, which are the most salient points discovered by the slime mold agents. They typically are the closest points to the focused anchor points. The most un-salient points are coded in blue, some are undiscovered by the agents during simulation. The intermediate points are coded in the pink to purple spectrum. These are the most interesting points as they reveal how the MCPM agents explore the surrounding area.

Since there are many points in high density, it can be hard to visually examine the result because of cluttering. To allow better examination, the user can control the *Lowest Connect* value in the top-right panel. Since each point stores a value, a counter indicating how many times MCPM agents discover the point. If *Lowest Connect* value is zero, all words are displayed in the scatter plot view. If the value is non-zero, all words with counter values under the set value are invisible and un-selectable. This is demonstrated in Figure 3.5, where *Lowest Connect* changes from high to low value,

showing a pattern of how MCPM agents travel for the focused anchor point *tour_VERB*.

3.3.3 Linguistic-based Examination

In addition to the MCPM result, the word embedding also includes the part-of-speech tag for each word. The visualization tool implements the functionality to color code word embeddings based on their part-of-speech tag. Each part-of-speech is assigned a color to visually identify their distribution pattern. This is closely examined in the next chapter, section 4.3. Specifically, Figure 4.4 provides a visual overview of W2V-300k-1 and W2V-300k-2.

The embedding examination, MCPM examination and tag-of-speech examination constitute the three main functionalities of the visualization tool. I find that MCPM and tag-of-speech look at the dataset in different granularity. MCPM focus more on the local neighborhood, where the interesting results are typically highlighted on the couple hundred to thousands of points surrounding a given anchor point, while part-of-speech examination concerns with the makeup of the entire dataset.

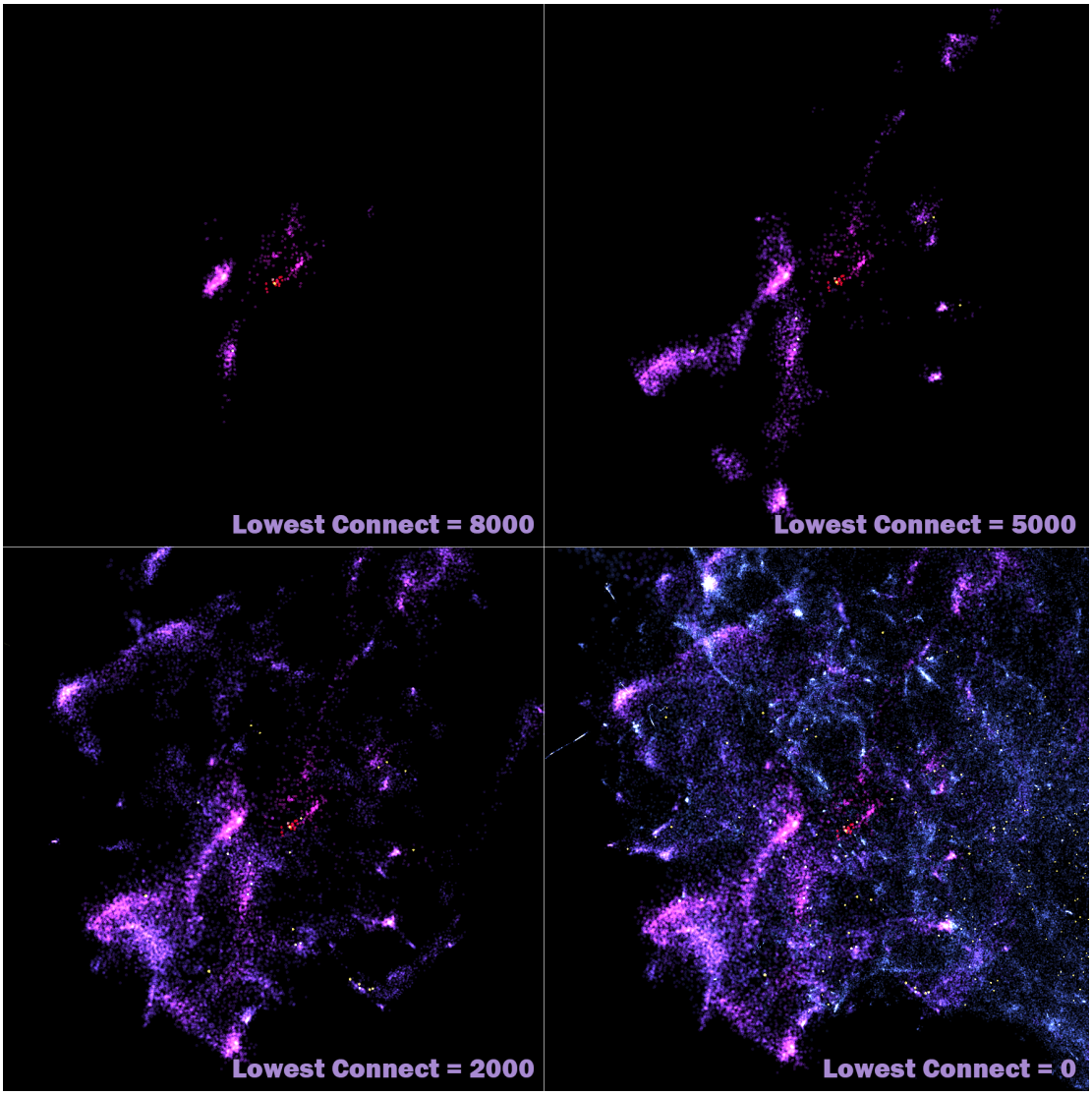

Figure 3.5: **Visualization results with different *Lowest Connect* values for the anchor word *tour_VERB*, the yellow point in the center of each figure, surrounded by red points. A number is generated and stored in each point, which is the counter designating the connected-ness to the anchor word *tour_VERB*. During slime mold simulation, the slime agent increments the counter when sufficiently close to it. *Lowest Connect* filters out points whose counters are below the given *Lowest Connect* value. When the *Lowest Connect* is zero, all points in the data are displayed.**

Chapter 4

Result and Discussion

4.1 Trace-guided Exploration

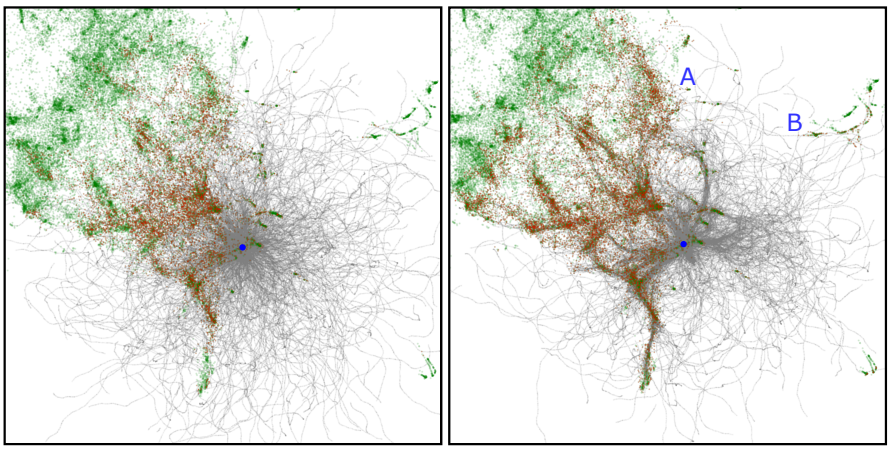

Figure 4.1: **MCPM agent exploration results for token *class_NOUN* in W2V-300k-1 comparing unguided (left) and trace-guided traversal (right).**

In this section, I mainly try to explore the behavior of MCPM probe through visualization. I apply MCPM probing to two language embedding datasets: W2V-300k-1 and BERT-Back. The main observation is that MCPM probe agents are not only

sensitive to distance during its exploration, but also connectivity within the dataset.

In W2V-300k-1, MCPM agent exploration results for token *class_NOUN* are shown in Figure 4.1. The left shows the exploration path for unguided agents, while the right shows the path while agents are guided by trace. 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.

One can see 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 4.1-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 being closer within the paradigm of optimal transport.

In BERT-back (BERT dataset generated for token *back*), the contextual embeddings visualized in Figure 4.2 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,

Table 4.1: **Three samples from each of the three major clusters detected in BERT-back. See Figure 4.2 for the corresponding visualization.**

Top cluster (spatial relation)
Partition walls constructed from fibre cement *backer* 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 *back-mounted* cylinders.
Bottom-left cluster (back in time)
Mono Lake is believed to have formed at least 760,000 years ago, dating *back* to the Long Valley eruption.
Other settlements were Toro, in the extreme south, 1827, and Noble, in the north portion, dating *back* to the 1830s.
Bottom-right cluster (direction of communication)
Decisions must be unanimous: any divided decision sends the question *back* to the House at large.
He ends by saying that, if he does not hear *back* from Romani, he will not write to him again.

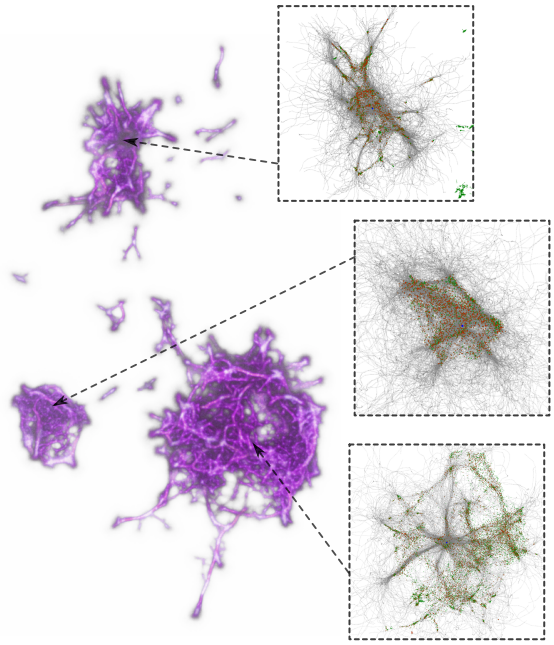

Figure 4.2: **Visualization of intra-cluster exploration for BERT-back, starting in locations inside each respective cluster. We observe distinct topologies within each cluster, corresponding to the different contexts of word *back* captured by the embedding (see Table 4.1).**

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 Table 4.1.

The samples found within each cluster demonstrate clear differences in the word usage patterns (see Table 4.1). 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* 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.

Both visualization results in Figure 4.1 and 4.2 demonstrate clear distinction between MCPM probing and purely distance-based probing. While, MCPM agents are still sensitive to euclidean distances, they are also sensitive to the connectivity of the dataset by following throughput of the trace. The result of this additional consideration is that the clustering and pattern finding is a lot more flexible comparing to other distance-based clustering methods such as K-Means.

4.2 Word Similarity

By interpreting MCPM probing results as finding similar word tokens in language embedding data, I can compare the MCPM results to other similarity measurements: Cosine similarity and Euclidean similarity. These similarity metrics emphasize different mathematical relations. Cosine similarity measures orientation with respect to origin, while Euclidean similarity measures geodesic distance in a homogeneous space, and MCPM similarity builds on the optimal-transport throughput. For this section, I queried the word *wind_NOUN* in W2V-300k-1 and generated five different similarity

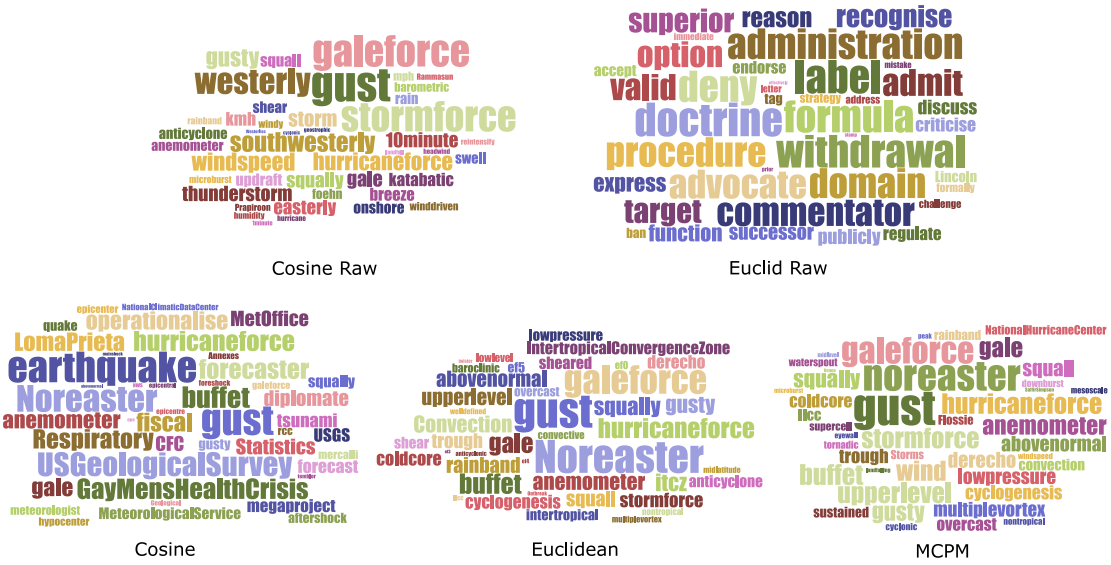

Figure 4.3: **Word clouds of top 50 most similar words for *wind_NOUN* in the W2V-300k according to five similarity metrics. The bigger the word, the more similar it is to *wind_NOUN*.**

rankings for each metrics, as shown in Figure 4.3. Two similarity rankings: *Cosine Raw* and *Euclid Raw* are cosine and euclidean similarities in the original 300 dimensions, while *Cosine*, *Euclidean* and *MCPM* are the three measurements in the reduced three dimensions. The *Cos Raw* similarity is considered as the benchmark: it is often used in downstream tasks such as machine translation and sentimental analysis.

Immediately, one can observe that *Euclid Raw*, or the Euclidean similarity in the original 300 dimension, does not provide useful similarity lists for *wind_NOUN*. Top words such as *formula*, *doctrine* and *withdrawal* are not related to *wind* in any sense. However, *Euclidean*, or the euclidean similarity list in the reduced dimension, does yield more salient results with some common words to the benchmark *Cosine Raw* similarity list. One can conclude that geodesic distance does not embed semantic information in

the original word embedding. However, distance is embedded with information during the process of dimensionality reduction. Based on this observation, *Euclid Raw* will not be included in the rest of this section.

Turning attention to *Cosine* similarity, one can see that, while containing some common words such as *Noreaster* and *gust*, a large portion of the list contains out-of-place words such as *diplomate*, *fiscal* and *statistics*. Considering the observation that distance contains semantic information in the reduced dimension, one can conclude that cosine similarity becomes a less precise measurement as a result. Therefore, *Cosine* similarity is also not included in the rest of the section. I mainly focus on *Cosine Raw*, *Euclidean* and *MCPM* similarity lists for the next comparison.

Euclidean and MCPM rankings have much more agreeable candidates in higher ranks such as *gust*, *hurricane-force* and *anemometer*. There are still disagreement among the similarity lists. For example, *coldcore* appears in both *Euclidean* and *MCPM* but not in *Cosine Raw*, while *lowpressure* appears in *MCPM* but not in the other two lists. But one can see how these words relate to *wind_NOUN*, unlike *Cosine* similarity, where some obviously irrelevant entries appear. At this point, word cloud visualization is inefficient in comparing the three similarity rankings. Since all the words seem relevant to *wind_NOUN*, I want to focus on the difference of rankings between *Euclidean* and *MCPM* similarities, with *Cosine Raw* as the benchmark. Word cloud does not allow enough space for comparing many words and does not make visualizing ranking differences easy.

To further compare *Euclidean* and *MCPM* rankings, I extract the the top 5000

Table 4.2: **Ranking difference Cosine Raw, Euclidean and MCPM. The entries are ordered in descending order of their ranking difference. 20000+ indicates that the word is not found within the most 20000 similar words in Cosine Raw Ranking.**

Word	**Cosine Raw Rank**	**Euclidean Rank**	**MCPM Rank**
cumulonimbus_NOUN	178	454	4944
sedimentology_NOUN	20000+	399	4660
Wash_PROPN	20000+	363	4610
surge_NOUN	666	110	4350
landlocked_ADJ	20000+	474	4700
post-katrina_ADJ	20000+	748	4869
stalagmite_NOUN	17878	956	4953
massif_NOUN	20000+	782	4747

words in both similarity lists. I compare their difference by sorting them in descending order of their ranking differences, as shown in Table 4.2. In other words, the top word *cumulonimbus_NOUN* has the greatest difference between *Euclidean* and *MCPM* rankings. The *Cosine Raw* ranking is then used as benchmark to see which ranking is closer to it. I choose the top 5000 words for *Euclidean* and *MCPM* ranking because it strikes a good balance where the ranking difference is significant, but not big enough to include words completely irrelevant to *wind_NOUN*. For *Cosine Raw Rank*, any ranking that is higher than 20000 is marked as *20000+*, as the specific ranking past that number is no longer relevant for comparison.

As one can observe, excluding two words *cumulonimbus_NOUN* and *surge_NOUN*,

most words are much further down the ranking than both *Euclidean* and *MCPM* rankings suggest. However, their *MCPM* rankings are much further down the list than *Euclidean* rankings. In this sense, one can say that *MCPM* ranking agrees with *Cosine Raw* more, as it considers these words to be far less important than *Euclidean* ranking does. I agree with many of the rankings in *Cosine Raw*. Words like *sedimentology_NOUN*, *stalagmite_NOUN* and *massif_NOUN* are more relevant to geology and rock formations than anything directly related to wind and storms. The disparity between *Euclidean* and *MCPM* rankings can be explained by how these rankings are generated. Since *MCPM* ranking is also sensitive to connectivity in addition to distance, one can conclude that not only distance is embedded information in the process of dimensionality reduction, but the *relative distance between words* also embeds similarity information. This is also supported by the observation in the next section. Using the visualization tool, I observe that similar words are often clustered in filaments rather than semi-uniformly distributed clusters, as shown in Figure 4.10. Nevertheless, the conclusion is preliminary. It needs to be verified by a broader, quantitative study in the future.

It is also important to realize that the definition of similarity as 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 *statistics*, 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* [29]. It remains an open question whether

there is a way to extract the distinction between associated and similar words in word embeddings.

4.3 Global Structure of W2V-300k

In this section, I mainly focus on the contribution of the word embedding visualization tool without considering its capability to examine MCPM fitting results. To recap, I generated two different datasets of W2V-300k with the exact same UMAP hyperparameters, in order to identify any consistent pattern between the two datasets. The only variability within these two datasets is caused by the stochasticity of UMAP dimensionality reduction process. Identifying consistent patterns between the two allows me to argue that certain patterns are intrinsic within the data rather than an artifact created by UMAP. I'm going to mostly examine the structure by visualizing the global structure through parts of speech color filter.

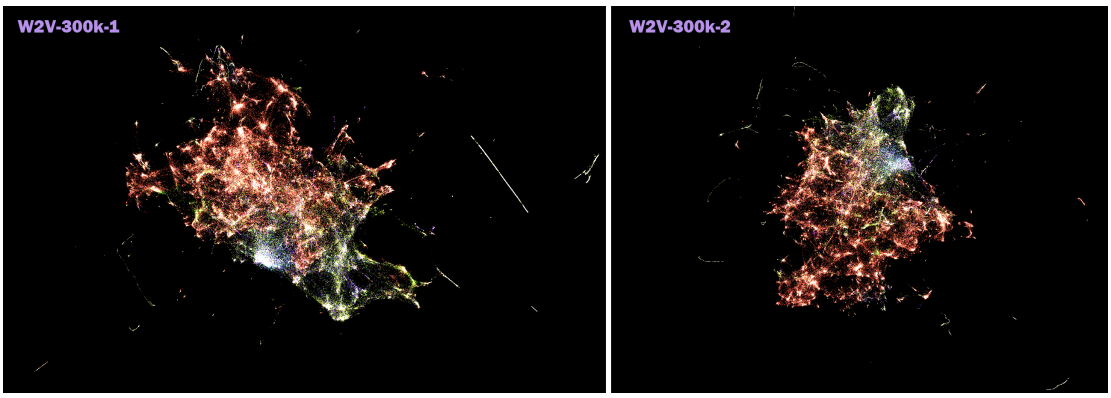

Figure 4.4: **Overview of W2V-300k-1 and W2V-300k-2. Different color represents different part-of-speech tag of words.**

Both W2V-300k-1 and W2V-300k-2 are visualized in Figure 4.4. Different

part-of-speech tags are indicated by their colors. We can already discern some features from the this view: 1) Majority of the points are distributed in the central cluster. 2) There are scattered filaments outside the main cluster. 3) The main cluster is split into two sections based on color, one is mainly occupied by orange points (Proper Nouns), the other mainly by green points (Nouns). In the rest of the section, I will discuss the global structure shared across the two datasets in more detail. First, I break down the entire dataset into three different zones based on the prominence of part-of-speech tags: *Mixed* (no prominence of part-of-speech), *Proper Nouns* and *Numbers*. Second, I further break down the *Mixed* zone - one with mixed part-of-speech tags, to look at the difference in distribution across different part-of-speech tags. Third, informed by the spatial division of zones, I look at the whether there's any distinguishing feature in the actual content of words: specifically, cultural, scientific and general.

The first consistent global pattern is visualized in Figure 4.5. I find that the datasets can be separated into three distinct zones: the first zone (*Mixed* in the figure) contains mixed parts of speech - mostly nouns, verbs, adjective and adverbs, the second zone (*Proper Noun*) contains mostly proper nouns, and the last zone (*Numbers*) contains mostly numbers. When visualized separately from the exact same view position, we can see that the three zones occupy distinctly different area of the dataset, with *Mixed* zone and *Proper Noun* zone being complement to each other forming the large cluster in the middle, while *Numbers* zone consists of loose filaments outside the main cluster.

Shifting attention to the *Mixed* zone - the zone containing a mixture of noun, verb, adjective and adverb, I visualize the part-of-speech separately as shown in Figure

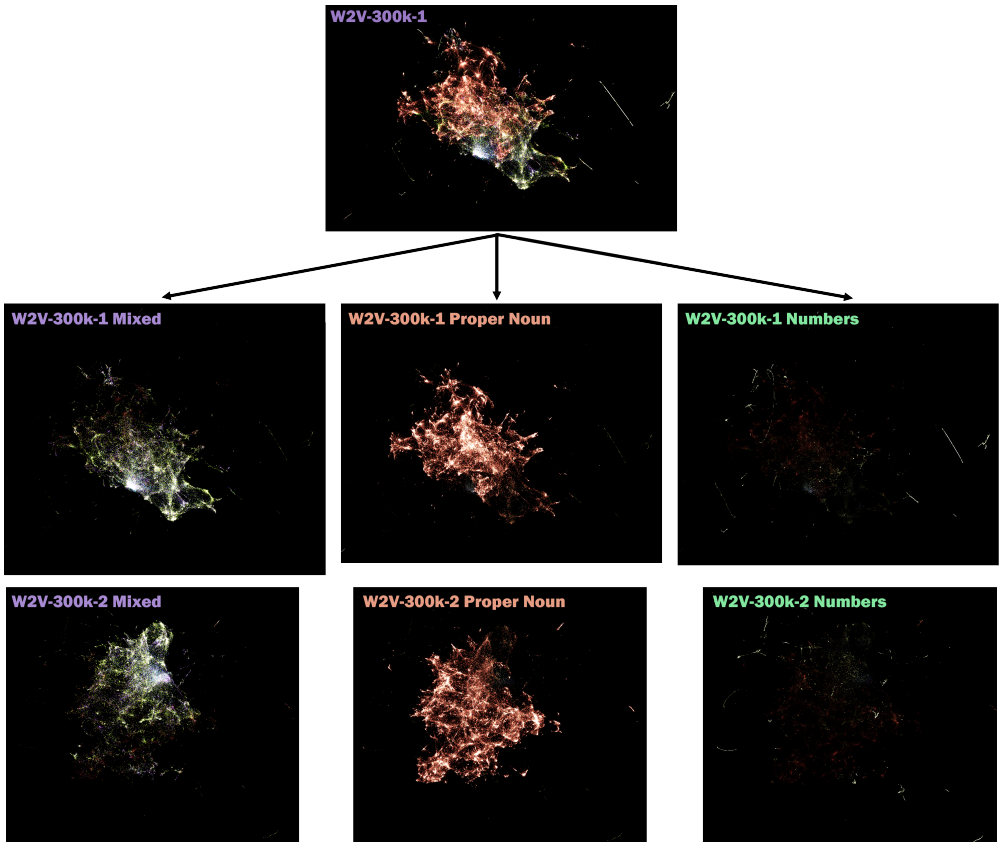

Figure 4.5: **Visualization of three zones of W2V-300k-1 and W2V-300k-2 based on prominence of part-of-speech tags. Mixed mode includes noun, verb, adjective and adverb. We can see that three different modes cover distinctly different areas of the datasets, with Numbers consistently cover the outer loose filaments. Only original visualization of W2V-300k-1 is included.**

4.6. Interestingly, the noun-adj and verb-adv form two separate pairs with similar distributions within them. For both datasets, the nouns are quite spread out, with many points bleeding into *Proper Noun* zone. While there are a lot less adjectives than nouns, we can see that they are distributed quite similarly in terms of spread. Similar visual correlation can be seen between verbs and adverbs, only there are a lot less data points. The verb-adv pair is more concentrated in a particular area. We can make sense of this correlation as noun-adj pair and verb-adv pair are intrinsically defined to be used

together.

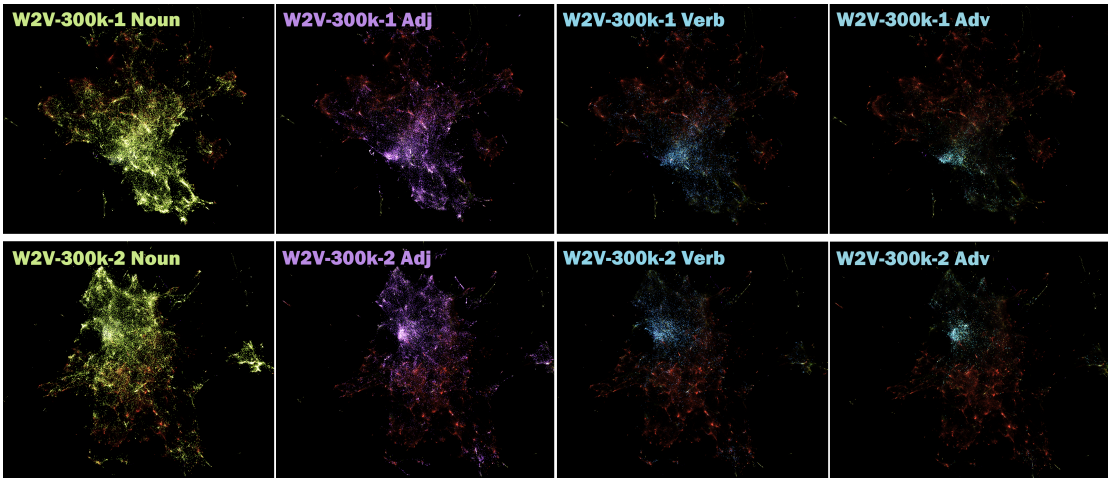

Figure 4.6: **Noun, verb, adjective, adverb visualization of global embeddings. One can see that Noun-Adj pair, as well as Verb-Adv pair, have similar spread and center of gravity. This might point to the relatedness of the part-of-speech pairs. Namely, adjectives and nouns are used together, similar to adverbs and verbs.**

Additionally, we can see that all four part-of-speech tags have a high density in a concentrated area, which can be spotted by the brighter area in adjectives, verbs and adverbs visualizations. There's also an extension of nouns next to the concentrated area that is not covered by verbs and adverbs (bottom right in W2V-300K-1, top mid-left in W2V-300-2). This is shown in Figure 4.7. We're going to explore those areas in the next paragraph.

To further examine the nature of the concentrated area as well as the extension of nouns next to it, I visualize the embedding by only displaying nouns and verbs in Figure 4.8. The purpose of displaying verbs next to nouns is to show where the concentrated area is, as seen by the white concentration in the figure. I mark down three distinct areas in both datasets. I find that, in *Concentration* area, where there

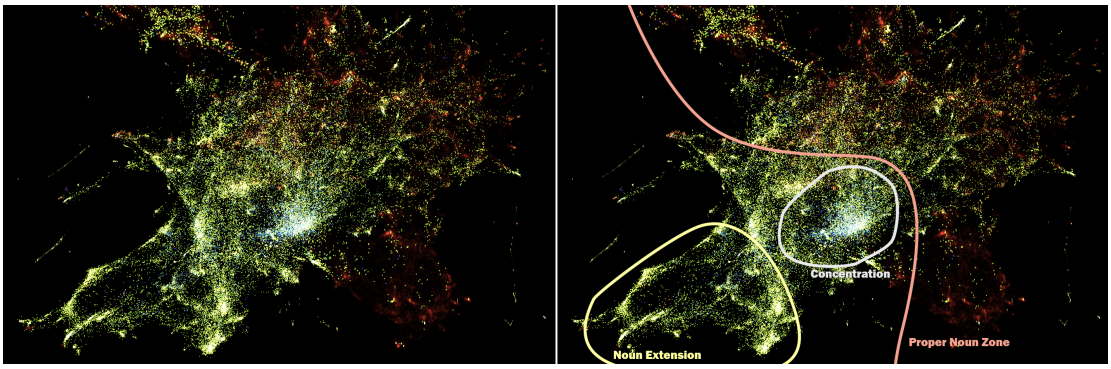

Figure 4.7: **Illustration of the extended noun area, compared to concentrated area.**

are high density and evenly spread out words with a mixture of nouns, adjectives and adverbs, the words are more commonly used than in *Noun Extension* area, where the nouns are more distinctively specialized, many times pertaining to scientific uses. I mark down 16 words in each dataset as a path traveling from the *Concentration* area into the *Noun Extension* area to demonstrate this change.

Closely examining the *Proper Noun* zone, where proper nouns dominate the data points, we can immediately see the connection within each small local section. Some specific examples are shown in Figure 4.9. For example, a section includes proper nouns that directly relate to geographical names in China, or proper nouns that include English names.

When examining *Num* area, we find the word content to be less predictable, as shown in Figure 4.10. Some loose filaments represent specific location names as opposed to numbers as shown on the left side of the figure. We also find commonly used word such as *regional_ADJ*, which should belong to the *Mixed* zone. This suggests that some loose filaments might be an artifact created by UMAP. Despite this, large portion of the

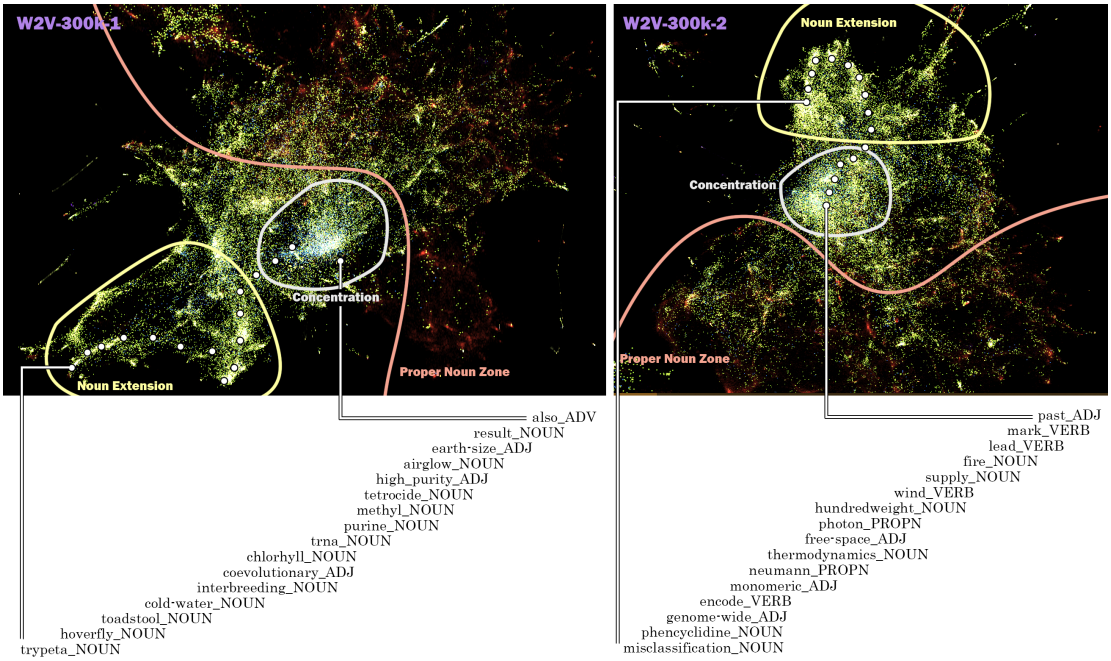

Figure 4.8: **View of two datasets marked with *Proper Noun*, *Concentration*, *Noun Extension* zones. Each dataset has 16 words marked down from the concentration area to the noun extension area. The 16 words show a shift from common words to more specialized, scientific words.**

loose elements still concerns with numbers. The right side of the figure shows that the word's part-of-speech tag does not need to be NUM in order to be grouped together, this makes us hypothesize that the number are represented in the loose filament area because of its context rather than the word content itself. In the middle we identified both *1962-1963* and *1960-1964*. Looking these words up on Google, we find that most Wikipedia entries with these words do no use them in the context of a sentence, such as *Uganda (1962-1963)*, *List of avant-garde films of the 1960s: 1960–1964* or *Kerala MLAs 1960–1964*.

In this section, I have shown some interesting patterns in the global structure that appear in both datasets. This serves three purposes:

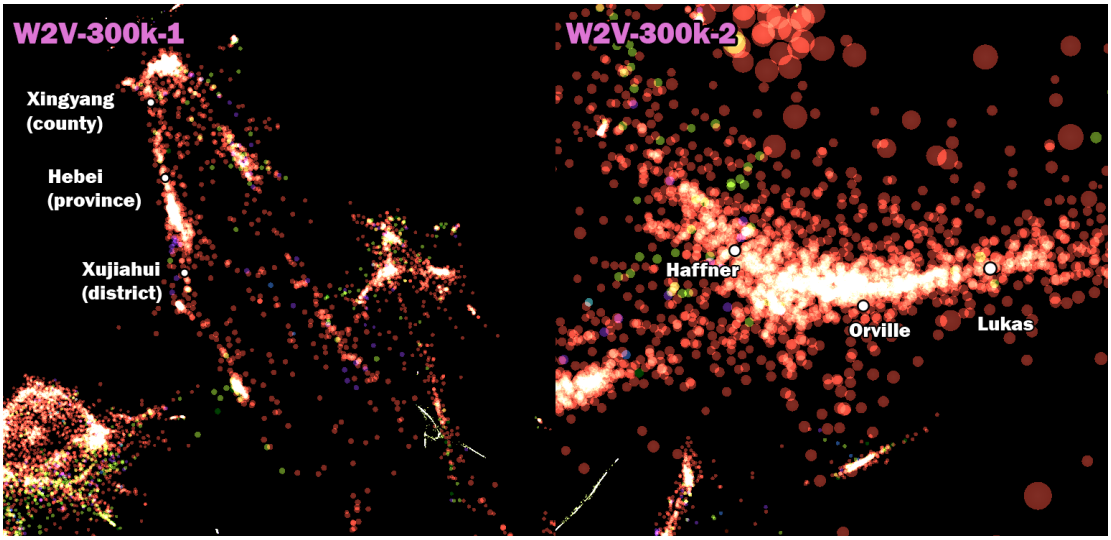

Figure 4.9: **Examples of Proper Noun sections. The left shows a filament of geographical Chinese proper nouns. The right shows a filament of English names.**

- By finding and showing consistent visual patterns across two datasets, I demonstrate that the embedding visualization tool, along with the speech-of-tags visualization, provides researchers and interested users opportunities to discover patterns and to generate hypotheses in the word embedding data.

- It strengthens the observations in section 4.2. The word similarity results provided by MCPM and Euclidean measurements in the reduced dimension have a degree of consistency that I argue is present in the original dimension, rather than artifacts created by dimensionality reduction.

- Some observations in this section also support the observation in section 4.2. Specifically, I observe that, for word similarity in reduced three-dimensional embeddings, semantic information is not only embedded in euclidean distances, but also connectivity of the points. The most clear demonstration of this can be seen

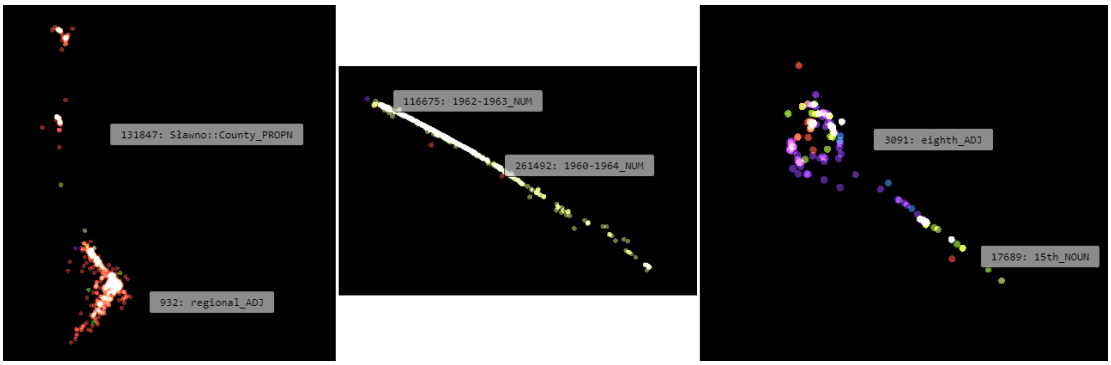

Figure 4.10: **Examples of Number loose filaments. The left shows a filament consists of not number words. The center and the right sub-figures concern with numbers, but the right figure consists of a mixture of part-of-speech tags.**

in both Figure 4.9 and 4.10, where similar words are clustered in filament-like structures.

Chapter 5

Conclusion and Future Work

In this thesis, I propose a novel method that makes use of the MCPM algorithm to discover the structure of language embedding data. The main innovation of this method is to introduce non-linear structural discovery into embedding evaluation, which has been dominated by offset vectors and proximity-based clusters. The contribution is broken into two aspects: information retrieval and a data visualization tool.

For the information retrieval aspect, I first visually examine the structural probing through visualising travel path of the agent. The trace-guided probe is shown to be superior to purely unguided probe (random walk). While unguided probe is similar to the nearest neighbor search, the trace-guided probe shows that the probe agents tend to follow salient structures in the data regardless of its distance from the emission point. The second observation is that MCPM acts as a robust clustering method that allows random sampling within each cluster. Purely distance-based methods, on the other hand, might fluctuate on their reliablity based on distance between clusters and where

the sampling point is located. I posit that these two properties are mainly due to MCPM's sensitivity to both spatial distance as well as connectivity of the data.

Then, I interpret the MCPM probe as finding word similarity in word embedding. By comparing Cosine and Euclidean-based measurement in both original dimension and reduced dimension, I find that euclidean distance embeds relevant semantic information only in the reduced dimension. By comparing MCPM ranking and Euclidean ranking in the reduced dimension to the benchmark (the Cosine ranking in the original dimension) I find that MCPM ranking agrees with Cosine ranking more than Euclidean. I suggest that this shows that UMAP utilizes both euclidean distance as well as connectivity between data to conduct dimensionality reduction, both of which MCPM is sensitive to.

For the visualization tool aspect, I demonstrate the potential of the tool by identifying consistent structures between W2V-300k-1 and W2V-300k-2, both generated under the same condition. I managed to identify different sections of the dataset and how each section houses different types of words: cultural-related proper nouns, common words and specialized words. This exercise also helps strengthening the observation in the information retrieval part, because it shows that there is consistent structure in the W2V-300k data that is preserved through the dimensionality reduction process. It also supports the observation that UMAP utilizes both distance and connectivity to produce the data in the reduced dimension. Overall, this thesis shows that MCPM is a notable information retrieval method for certain 3D scatter plots where euclidean distance and connectivity between data points are known to contribute to their structure.

There are many directions one can take for future projects. The first direction is fully exploring the potential of MCPM in language embedding evaluation. One can compare MCPM evaluation to SIMLEX-99 to see if its similarity method correlate with human intuition. The troubling aspect, however, is that currently MCPM results are still applied in the reduced dimension. The main result of this thesis shows that MCPM is useful in reduced dimension because of how UMAP structures the data. For this reason, MCPM can perhaps never compete with the Cosine similarity result in the original dimension.

This opens up two lines of research questions, one concerned with MCPM, the other with language embedding algorithms. On the side of MCPM, the next step is to extend MCPM to arbitrary dimensions. Because of curse of dimensionality, it is perhaps impossible to preserve the exact simulation steps in higher dimensions. Therefore, more computer science work needs to be done in order to make this goal possible. But as I've shown in section 4.2, euclidean distance does not embed explicit meaning in the original dimension for Word2Vec. So applying high-dimensional MCPM to the same algorithm is not a fruitful task. This brings us to the other line of research question, which is on the side of language embedding algorithm. Further research has to dive deeper into the language embedding algorithms themselves. This thesis starts out with rather little motivation from the inner working of specific algorithms. The main motivation is that the result data is in a format (scatter plot) that allows MCPM exploration. It is not enough to simply seek correlation and to blindly follow empirical processes. A truly rigorous work needs to make informed arguments about the mechanisms of

data production below the surface that connects the nature of MCPM to the nature of language embedding algorithm.

Bibliography

[1] https://play.aidungeon.io/main/home.

[2] http://projector.tensorflow.org/.

[3] https://threejs.org/.

[4] https://github.com/CreativeCodingLab/Polyphorm.

[5] Ehsan Amid and Manfred K. Warmuth. Trimap: Large-scale dimensionality reduction using triplets, 2019.

[6] Amir Bakarov. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*, 2018.

[7] Matthew Berger. Visually analyzing contextualized embeddings. *arXiv preprint arXiv:2009.02554*, 2020.

[8] Matthew Berger, Katherine McDonough, and Lee M Seversky. cite2vec: Citation-driven document exploration via word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):691–700, 2016.

[9] Angie Boggust, Brandon Carter, and Arvind Satyanarayan. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples. *arXiv preprint arXiv:1912.04853*, 2019.

[10] Joseph N Burchett, Oskar Elek, Nicolas Tejos, J Xavier Prochaska, Todd M Tripp, Rongmon Bordoloi, and Angus G Forbes. Revealing the dark threads of the cosmic web. *The Astrophysical Journal Letters*, 891(2):L35, 2020.

[11] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[12] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of BERT. *arXiv preprint arXiv:1906.02715*, 2019.

[13] 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*, pages 91–98, 2009.

[14] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[15] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer,

and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[16] Xiangfeng Dai and Robert Prout. Unlocking super bowl insights: Weighted word embeddings for twitter sentiment classification. 2016.

[17] Xiangfeng Dai and Robert 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*, pages 1–6, 2016.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[19] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[20] 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*, pages 231–240, 2011.

[21] Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In *Proceedings of*

coling 2016, the 26th international conference on computational linguistics: Technical papers, pages 3519–3530, 2016.

[22] Oskar Elek, Joseph N Burchett, J Xavier Prochaska, and Angus G Forbes. Monte Carlo Physarum Machine: An agent-based model for reconstructing complex 3d transport networks. In *Artificial Life Conference Proceedings*, pages 263–265. MIT Press, 2020.

[23] Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings, 2019.

[24] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[25] Anna Gladkova and Aleksandr Drozd. Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 36–42, 2016.

[26] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[27] Florian Heimerl and Michael Gleicher. Interactive analysis of word vector embeddings. *Computer Graphics Forum*, 37(3):253–265, 2018.

[28] John Hewitt and Christopher 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)*, pages 4129–4138, 2019.

[29] Felix Hill, Roi Reichart, and Anna Korhonen. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[30] 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.

[31] Xiaonan Ji, Han-Wei Shen, Alan Ritter, Raghu Machiraju, and Po-Yin 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.

[32] Jeff Jones. Characteristics of pattern formation and evolution in approximations of physarum transport networks. *Artificial life*, 16(2):127–153, 2010.

[33] K. Kucher and A. Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *IEEE Pacific Visualization Symposium*, pages 117–121, 2015.

[34] Yang Li and Tao Yang. Word embedding for understanding natural language: a survey. In *Guide to big data applications*, pages 83–104. Springer, 2018.

[35] Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside bert's linguistic knowledge. *arXiv preprint arXiv:1906.01698*, 2019.

[36] 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.

[37] Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. Latent space cartography: Visual analysis of vector space embeddings. *Computer Graphics Forum*, 38(3):67–78, 2019.

[38] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

[39] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[40] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[41] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[42] Piero Molino, Yang Wang, and Jiawei Zhang. Parallax: Visualizing and understanding the semantics of embedding spaces via algebraic formulae. *arXiv preprint arXiv:1905.12099*, 2019.

[43] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.

[44] Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, 2004.

[45] Maximilian Noichl. Modeling the structure of recent philosophy. *Synthese*, pages 1–12, 2019.

[46] Deniz Cem Önduygu, Hüseyin Kuşçu, and Eser Aygün. History of philosophy: Summarized & visualized. https://www.denizcemonduygu.com/philo/.

[47] 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.

[48] Deokgun Park, Seungyeon Kim, Jurim Lee, Jaegul Choo, Nicholas Diakopoulos, and Niklas Elmqvist. Conceptvector: text visual analytics via interactive lexicon building using word embedding. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):361–370, 2017.

[49] Barbara H. Partee, Alice ter Meulen, and Robert E. Wall. *Mathematical Methods in Linguistics. Corrected first edition.* Kluwer Academic Publishers, 1990.

[50] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global

vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[51] Ignacio Perez-Messina, Claudio Gutierrez, and Eduardo Graells-Garrido. Organic visualization of document evolution. In *International Conference on Intelligent User Interfaces*, page 497–501, 2018.

[52] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[53] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 3rd edition, 2016.

[54] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[55] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

[56] Christof Rezk-Salama. *Volume rendering techniques for general purpose graphics hardware*. PhD thesis, Universität Erlangen-Nürnberg, 2001.

[57] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how BERT works. *arXiv preprint arXiv:2002.12327*, 2020.

[58] R. M. Rohrer, D. S. Ebert, and J. L. Sibert. The shape of shakespeare: visualizing text using implicit surfaces. In *Proceedings IEEE Symposium on Information Visualization (Cat. No.98TB100258)*, pages 121–129, 1998.

[59] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 298–307, 2015.

[60] Rita Sevastjanova, Aikaterini-Lida Kalouli, Christin Beck, Hanna Schäfer, and Mennatallah El-Assady. Explaining contextualization in language models using visual analytics.

[61] Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. Towards generating long and coherent text with multi-level latent variable models, 2019.

[62] Sunil Simha, Joseph N Burchett, J Xavier Prochaska, Jay S Chittidi, Oskar Elek, Nicolas Tejos, Regina Jorgenson, Keith W Bannister, Shivani Bhandari, Cherie K Day, et al. Disentangling the cosmic web towards FRB 190608. *arXiv preprint arXiv:2005.13157*, 2020.

[63] Manjeet Singh. Word embedding, Jul 2020.

[64] I. Subašic and B. Berendt. Web mining for understanding stories through graph visualisation. In *IEEE International Conference on Data Mining*, pages 570–579, 2008.

[65] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations, 2019.

[66] 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.

[67] 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.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[69] Cédric Villani. *Optimal Transport: Old and new*. Springer, 2009.

[70] M. Wattenberg. Arc diagrams: visualizing structure in strings. In *IEEE Symposium on Information Visualization*, pages 110–116, 2002.

[71] Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint arXiv:1909.10430*, 2019.

[72] Hongwei Henry Zhou, Oskar Elek, Pranav Anand, and Angus G Forbes. Bio-inspired structure identification in language embeddings. In *2020 IEEE 5th Workshop on Visualization for the Digital Humanities (VIS4DH)*, pages 7–13. IEEE, 2020.

[73] George Kingsley Zipf. *Human behavior and the principle of least effort*. Addison-Wesley, 1949.