Computational Brain & Behavior (2021) 4:164–177
https://doi.org/10.1007/s42113-020-00094-8

ORIGINAL PAPER

Scale-Dependent Relationships in Natural Language

Aakash Sarkar[1] · Marc W. Howard[1]

Accepted: 6 October 2020 / Published online: 4 January 2021

Abstract

Language, like other natural sequences, exhibits statistical dependencies at a wide range of scales as discussed by Lin and Tegmark (2017). However, many statistical learning models applied to language impose a sampling scale while extracting statistical structure. For instance, Word2Vec creates vector embeddings by sampling context in a window around each word, the size of which defines a strong scale; relationships over much larger temporal scales would be invisible to the algorithm. This paper examines the family of Word2Vec embeddings generated while systematically manipulating the size of the context window. The primary result is that different linguistic relationships are preferentially encoded at different scales. Different scales emphasize different syntactic and semantic relations between words, as assessed both by analogical reasoning tasks in the Google Analogies test set and human similarity rating datasets WordSim–353 and SimLex–999. Moreover, the neighborhoods of a given word in the embeddings change considerably depending on the scale. These results suggest that sampling at any individual scale can only identify a subset of the meaningful relationships a word might have, and point towards the importance of developing scale-free models of semantic meaning.

Keywords Word2Vec · Embedding · Scale · Context · Analogy · Similarity · Relatedness · NLP · Language · WordSim353 · SimLex999

Introduction

Information in natural sequences often spans across many scales. A mixture of many length scales have been seen to create a power-law decay of long-range correlations in DNA sequences (Li et al. 1994; Peng et al. 1992; Mantegna et al. 1994). Compositions from different composers in Western classical music obey a $1/f^{\alpha}$ power law in both musical pitch and rhythm spectra (Levitin et al. 2012; Roos and Manaris 2007). Such scale-free behavior has been observed in earthquakes (Abe and Suzuki 2005), collective motion of starling flocks (Cavagna et al. 2010), and neural amplitude fluctuations in the human brain (Linkenkaer-Hansen et al. 2001). Samples of natural language also exhibit long-range fractal correlations (Montemurro and Pury 2002). The mutual information (MI) between two symbols, for such sequences, have recently been shown to decay like a power law as well, with the temporal difference between them (Lin and Tegmark 2017) (see Fig. 1).

Analyses on large-text corpora from diverse sources have been shown to have long-range structure beyond the short-range correlations happening at syntactic level between sentences (Ebeling and Neiman 1995; Ebeling and Pöschel 1994). Corpora from different languages have been shown to have a two-scale structure, with the dimension of semantic spaces at short distances being distinctly smaller than at long distances (Doxas et al. 2010). Studies on the statistics of shuffled text corpora seem to confirm this, where a text corpora shuffled even at the sentence level loses large-scale structure (Altmann et al. 2012). There has also been evidence of increased performance of the BEAGLE model on TOEFL synonym scores when entire sentences were used as context windows, and significant variation when the window size was changed (Jones and Mewhort 2007; Sahlgren et al. 2008). However, many prevalent statistical learning models which aim to learn such semantic structure fix a scale when sampling the context around words. We observe one such class of models called Word2Vec, which use a vector embedding to study semantic structure. Word2Vec uses a moving window around each word to gather context, but the size of the window is a fixed parameter. In this paper, we systematically change

✉ Aakash Sarkar
aakash18@bu.edu

[1] Department of Psychological and Brain Sciences, Boston University, Boston, MA 02215, USA

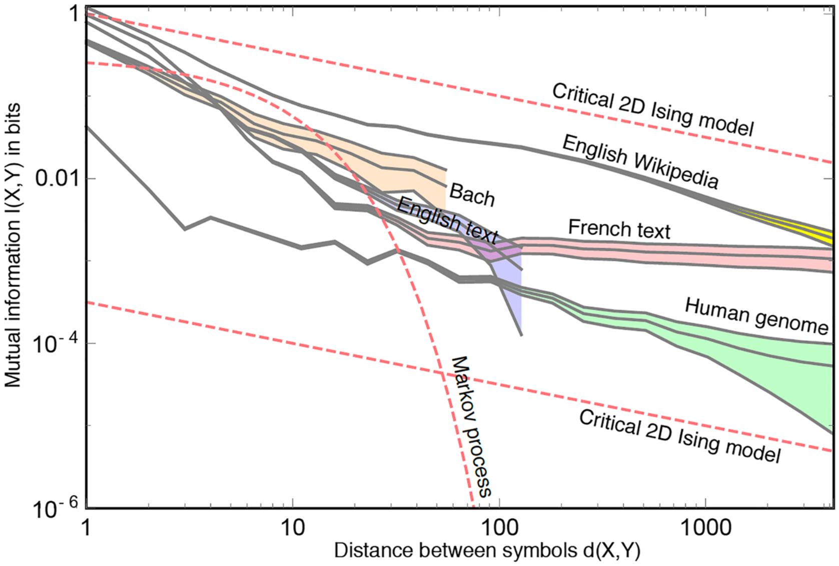

Fig. 1 Language has information at many scales. Mutual information (MI) between a pair of symbols in different natural sequences falls slowly as a function of how far they are spaced (Lin and Tegmark 2017). The MI is a measure of the shared information content between the two symbols, and this seems to decay roughly as a power law for natural language. This is contrasted with the sharp exponential fall seen by a Markov process which has a fixed, predetermined scale. The slow decay of MI suggests that information is contained at a spectrum of different scales, and algorithms sampling natural language at fixed scales might not be sufficient

the size of sampling context used to train Word2Vec, and study the information encoded in the resultant embeddings about the statistical structure of the training text.

Word2Vec and Vector Embeddings

Word2Vec (Mikolov et al. 2013) is a widely used neural network model which learns a vector representation of words, called an embedding, by training on large corpora of text. Word embeddings store a unique vector representation of each word in the vocabulary in a high-dimensional vector space—a good embedding would map semantically similar words onto nearby points onto this vector space. Analyzing the structure of the embedding should also provide insight into the relations between words and how they appear in the source corpus.

Word2Vec is a predictive model which tries to infer a relationship between a central word, referred to as *target*, and its surrounding words, referred to as *context*. It comes in two flavors, which use the same algorithm but act as inverses of each other. The Skip-gram model tries to predict the context words from the target word, and the Continuous Bag-of-Words (CBOW) model tries to predict the target word from the context words around it. In both cases, the training continuously modifies the embedding with each target and context set, so that it would maximize the probability of obtaining one from the other (depending on the flavor). In this article, we focus on the CBOW variant and the structure of the embeddings it generates (Figs. 2 and and 3).

A key aspect of Word2Vec is how the context around each target word is sampled, as this also introduces a definite scale into the algorithm. Word2Vec samples a window of words around the target word w_t, stretching out in both directions (shown in Fig 4). The size of the window is chosen randomly each for each new target word, but there is a maximal size β which is usually defined as a fixed parameter before training commences. It can be shown that the resultant probability of choosing a neighboring word $w_{t\pm k}$ as a context word falls off linearly with the distance k from the target, vanishing completely at β

$$p(w_{t\pm k}) = 1 - \frac{k-1}{\beta}$$

It is interesting to note that both the slope of this probability distribution and the reach of neighboring words accessible to it are governed completely by the choice of parameter β—thus introducing a hard scale in the mechanics of the model.

The vectors in the Word2Vec embeddings have also been seen to have some interesting features—vector arithmetic can often encode mappings of linguistic relations between the corresponding words. For example, vectors which act as directions pointing from the source word (eg. `man`) to the destination word (`woman`) for a particular relation, when then added to a different source word (`king`), could take it very near to the intended destination word (`queen`). This property of the Word2Vec embeddings could be used to test how well the embedding encodes different linguistic relationships, as explored in the next section.

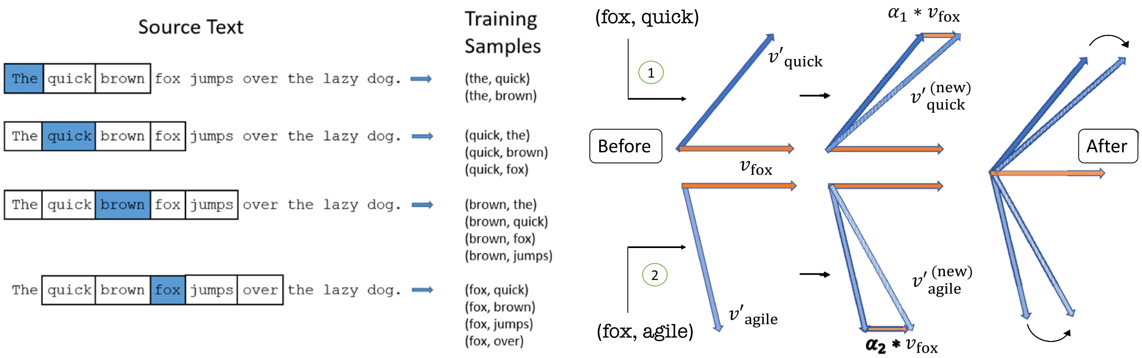

Fig. 2 Word2Vec samples a set window of neighbors around each word, introducing a fixed scale. Left: Word2Vec, a commonly used neural network for analyzing language, categorizes words in a window of fixed maximum size around a target word as its context words, thus introducing a *set* scale. For each word, this generates several `(target, context)` training samples (taken from McCormick 2016). Right: Word2Vec maintains an input and output vector representation for each word in its vocabulary, which are updated at each training sample. For example, when it sees the sample `(fox,quick)` (labeled 1), it brings the output vector for the context word `quick` closer to the input vector for the target word `fox`, and vice versa, which it would again do when it sees the sample `(fox,agile)` (labeled 2). However, by bringing the output vector for `agile` closer to the input vector for `fox`, it has brought the output vectors for `agile` and `quick` closer to each other, which both co-occur in the vicinity of the common word `fox`

Methods

Corpus and Prepossessing

To train Word2Vec, we used the `enwik9` corpus (Mahoney 2006), containing preprocessed text from the first 10^9 bytes of the Wikipedia dump dated March 3, 2006. Wikipedia was chosen to provide a rich representation of words coming from a diverse range of topics. The corpus consists of cleaned-up sentences which only retain text which would be visible to a human reader accessing a Wikipedia web page. Only alphanumeric characters were retained, all numbers were converted to spelled out text, and hyperlinks were processed to contain only the description of the link accessible to the user. After preprocessing, the corpus contained 124 million tokens with a distinct vocabulary of 1.4 million types.

Training Word2Vec

We used the Continuous-Bag-of Words (CBOW) implementation of Word2Vec, written in C, from Mikolov's Word2Vec Github repository (Mikolov 2017). Word2Vec utilizes a shallow three-layer neural network with one hidden layer. It maintains two active vector representations of each word in its vocabulary, called the "input" representation v_i and the "outer" representation v_i', encoded in the weight matrices between the layers. Both of these representations exist in the higher-dimensional vector space of the embedding. The hidden layer shares the same dimensionality, which we denote by N.

The CBOW algorithm tries to guess the target word given the set of context words surrounding that particular word. For each target word, Word2Vec generates (target, context) word pairs for every context word around it and passes each pair onto the neural network for training. Let us assume that, at a given time, the algorithm is given the pair (w_O, w_I). Word2Vec starts with a one-hot representation $\mathbf{x}_{w_I}$, corresponding to the input context word w_I, as its input layer. A one-hot vector has dimension V equaling the size of the vocabulary of the model, and only has a nonzero entry corresponding to the index of the word ($x_k = 1$ only when $k = I$, zero otherwise).

The weight matrix $\mathbf{W}$ (dimension $V \times N$) projects from the input layer onto the hidden layer $\mathbf{h}$. This operation essentially generates the input vector representation $\mathbf{v}_{w_I}$ of the input word

$$\mathbf{h} = \mathbf{W}^T \mathbf{x}_{w_I} := \mathbf{v}^T_{w_I}$$

The hidden layer then projects through another matrix, $\mathbf{W}'$ (dimension $N \times V$), generating a score $\mathbf{u}_k$ for each possible output word w_k

$$\mathbf{u}_k = \mathbf{W}'\mathbf{h} = \mathbf{v}'_{w_k} \cdot \mathbf{v}_{w_I}$$

This effectively computes a dot product of the hidden layer with the output vector for each word w_k in the vocabulary—representing how closely aligned each output vector $\mathbf{v}'_{w_k}$ is to the input vector $\mathbf{v}_{w_I}$. A softmax transformation finally converts this score into a posterior probability distribution. This becomes the corresponding entry $\mathbf{y}_k$ in the output layer of the network

$$\mathbf{y}_k = p(w_k|w_I) := \frac{\exp(\mathbf{v}'_k \cdot \mathbf{v}_I)}{\sum_{m=1}^{V} \exp(\mathbf{v}'_m \cdot \mathbf{v}_I)}$$

This is Word2Vec's best guess about the chances of the word w_k being the target word given that the word w_I appeared in its context window. Given that the actual answer was already known to be w_O for the target word, the error can be computed and the matrices $\mathbf{W}$ and $\mathbf{W}'$ (which

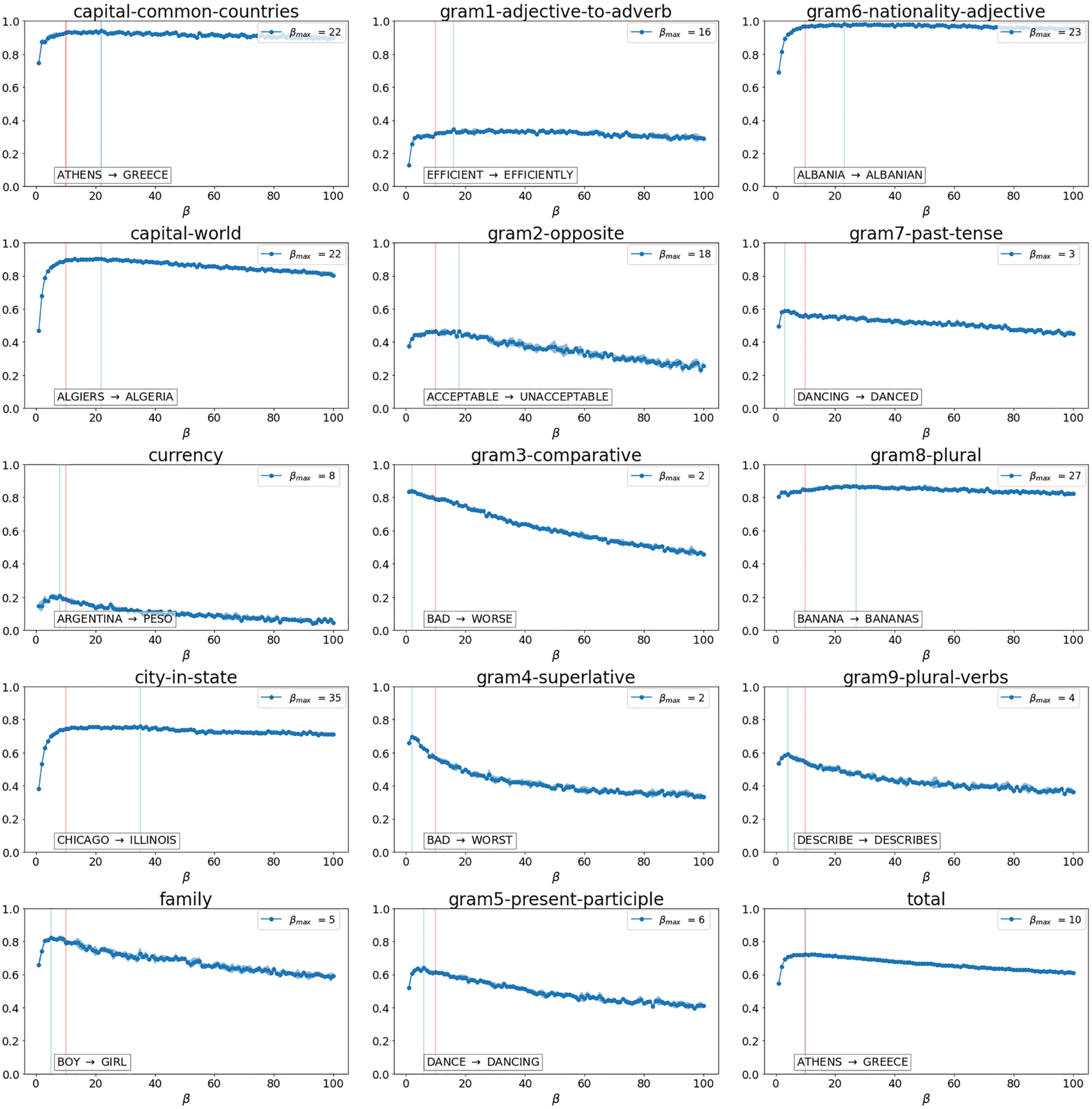

Fig. 3 Different linguistic relations are encoded best at different sampling scales. These graphs show Word2Vec's performance on the analogical reasoning tasks by Mikolov et al. (2013), for different linguistic relationships, as a function of β, the scale of context it is sampling. Analogies in each category test for two word pairs linked by that relation—for instance, a sample analogy in "capital-world" would ask, "if `France` → `Paris`, does India → `Delhi`?". The embedding is correct if by adding the direction vector for the first pair, *vec*(`Paris`) - *vec*(`France`), to the first word of the second pair, vec(`India`), we get a closest match to *vec*(`Delhi`). The y-axis represents the fraction of correctly answered analogies for each linguistic relation. Different relationships show qualitatively different behavior as the sampling scale is changed. Note that the sampling scale corresponding to maximal performance (shown as β_{max} in the upper-right corner) differs across panels, sometimes dramatically (marked with the blue vertical line, while the position of the "best" scale taken across all tests is also marked in red)

generates the input and output representations respectively) can be updated using backpropagation. This ensures that the input vector for the context word ($\mathbf{v}_{w_I}$) and the output vector for the actual target word ($\mathbf{v}'_{w_O}$) move closer to each other, while all the output vectors not associated with the actual target word are moved further away from $\mathbf{v}_{w_I}$. At the end of

Table 1 Chosen values for different parameters used to implement the Continuous-Bag-of Words training in Word2Vec

Description of parameter	Value chosen
Dimensionality of embedding	200
Negative sampling loss (n)	25
Subsampling frequency threshold	10^{-4}
Simultaneous threads running	16
Number of training iterations	30

the training, the space of input vectors **v** becomes the word embedding.

Generating Embeddings for Different Sampling Scales

A range of embeddings were generated by systematically changing the sampling scale from $\beta = 1, 2, 3 \ldots 100$—averaging statistics over 10 instances at each scale to increase consistency. The embeddings were analyzed by using the `gensim` package (Řehůřek and Sojka 2010) in Python.

The number of training iterations was increased to 30 to improve consistency of similarity measurements across embeddings for each sampling scale. The parameters controlling for negative sampling and subsampling frequencies were left unchanged from the default values listed in the repository (refer to Table 1).

The results shown in this article are from embeddings trained with negative sampling. The analysis was also repeated without the use of negative sampling to alleviate concerns of dependency of the negative sampling process on the sampling loss parameter (Johns et al. 2019). Hierarchical sampling (Morin and Bengio 2005), another speedup method used in Word2Vec, which expedites softmax computation with a hierarchical layer that has the words as leaves, was used instead. The trends analyzed were seen to be robust to both speedup methods.

Encoding of Linguistic Relationships at Different Scales: Google Analogies Dataset

To observe how Word2Vec encodes different linguistic relationships, the analogical reasoning tasks in the Google Analogies Dataset (Mikolov et al. 2013) were used. We kept track of whether vector arithmetic can recognize linguistic maps between two words, for instance, `boy` and `girl`, and connect a different word through the same map, like `son` to `daughter`. For this 4-tuple {`boy,girl,son,daughter`}, this was achieved by generating the direction vector going from `boy` to `girl`, and checking if adding this vector to the vector for `son` yields `daughter` as the closest match. A list of such 4-tuples, analyzing maps from a total of 14 different syntactic and semantic relations on the 30,000 most frequent words found in the corpus, was used to compute the fraction of correct choices for each linguistic relation. The performance across different relations, as well as the combined performance, was used to gauge the variability of performance across different sampling scales.

Semantic Similarity vs Relatedness at Different Scales: WordSim – 353 and SimLex – 999

To observe how different metrics of word similarity were captured on an aggregate level in the embeddings, a comparison was made between how the embeddings encoded semantic similarity and relatedness. Two words can often be related, like `coffee` and `cup`, but not

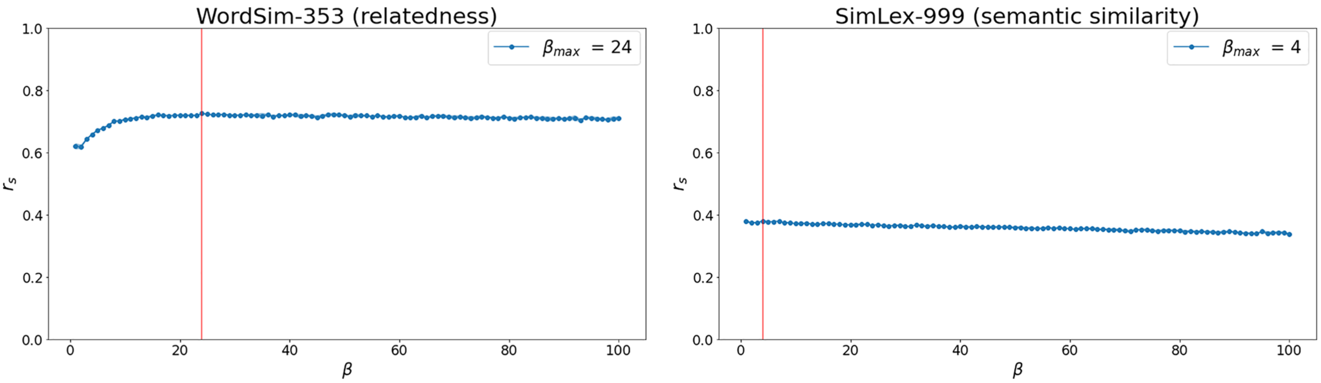

Fig. 4 Semantic similarity and relatedness require different sampling scales. The correlation scores of Word2Vec embeddings with human similarity datasets WordSim–353, quantifying relatedness (Left) and SimLex–999 quantifying semantic similarity (Right), as a function of scale. The y-axis represents the mean Spearman's coefficient (r_s) for each dataset at that sampling scale (β). The scale of highest correlation for each dataset is marked in red, and labeled in the upper-right corner. Note that correlation scores with SimLex, which measures semantic similarity independent of association, decline consistently as the scale is increased, while the correlation with WordSim, which measures more general association or relatedness between word pairs, benefits greatly from larger sampling scales

semantically similar, like cup and mug. Semantically similar words can be interchanged within a sentence and still remain meaningful, while interchanging related words could produce "sentences that often cannot be taken literally" (Lund 1995).

The embeddings were benchmarked with two different human similarity rating datasets to capture this distinction. WordSim–353 (Finkelstein et al. 2001), a benchmark which can measure word relatedness and association (Gabrilovich and Markovitch 2007), consists of 353 word pairs with human participants rating the word pairs on a scale from 0 (totally unrelated) to 10 (very related). SimLex–999 (Hill et al. 2015), on the contrary, was created to explicitly quantify semantic similarity independent of relatedness or association. It consists of 999 word pairs, generated with guidelines to prioritize synonymy in contrast to association.

To compare the similarity datasets with the cosine similarity of the embeddings, their Spearman correlation is computed as a function of the sampling scale. At each sampling scale, the mean correlation and inter-quartile range is calculated by analyzing 10 generated embeddings at that scale β, and repeated for the entire spectrum of sampling scales used.

Capturing Word Neighborhoods at Different Scales

The scale dependence of the embeddings was next examined at a more local level by studying the neighborhood surrounding different word vectors. To look at a diverse set of words, we used the 100 words most frequently used in English, from an analysis on the Oxford corpus (Oxford English Corpus 2011). The top ten words most similar to the central word are chosen in the embeddings trained at sample scales 1, 10, and 100, respectively, with the search constrained to the 10000 most frequent words in the vocabulary. The most similar words were ranked by cosine similarity to the vector for the central word, and was captured using the similarity function in gensim. These words were then combined to get the set of neighbors for each word, and the cosine similarity of these with the central words was examined as a function of scale.

Each curve $sim(w, w_i)$ corresponds to the cosine similarity of neighbor w_i with the central word w, as a function of the sampling scale β. An analysis of these similarity curves can help visualize the changing neighborhood of each central word. The similarity curves of different neighbors can change differently, and this can point towards the inter-relationships between them. For instance, the similarity scores of all neighbors can shift simultaneously with the sampling scale. These monotonic shifts can be contrasted with more immediate changes between neighbors, where the ordinal relationship between pairs, or groups, of neighbors change. Changes like the latter could be indicative of a change in the local semantic space.

Neighbor Statistics of Different Words at Different Scales

The last section looked at the effect of sampling scale on the neighborhoods of word vectors. In this section, this effect is analyzed systematically for a larger set of neighbors for different central words. Each neighbor w_i is characterized by the scale where its vector comes closest to the word vector of the central word w. Cosine similarity is used as the measure of distance between the two vectors, which is computed using the similarity function in gensim. Therefore, for each neighbor w_i, we had a corresponding scale β_i at which $\texttt{similarity}(w, w_i)$ is maximized.

The set of neighbors is chosen in similar fashion to the last section, but in a more exhaustive way, by combining $N = 100$ most similar words to the central word at each scale. The analysis was also repeated for $N = 5, 10, 20, 50$ to see if the distribution of neighbor similarity scores shows robust trends.

For each central word w, there is thus a distribution of sampling scales corresponding to the peak similarity scores between each neighbor and the central word. A histogram of this distribution yields the number of neighbors which reached a peak similarity score at any given sampling scale. Therefore, for each central word, a characteristic curve can be generated as a function of scale, quantifying the distribution of neighbors which would attain the closest similarity to the central word at that particular sampling scale.

Results

We can now examine the effect of the size of sampling context on the structure of semantic space learned by Word2Vec. First, we present the results of assessing the embeddings using analogical reasoning tasks from the Google Analogies test set and examine how scale affects the performance of different linguistic relationships. We then see how well the cosine similarity values of the embeddings are aligned with human similarity benchmarks WordSim–353 and SimLex–999 as the sampling scale is varied. We then move from assessing the embeddings on a global level to looking at the individual neighborhoods of word vectors, and assess if the structure of the local semantic space itself is changing, or if the changes are purely systematic. Each neighbor is characterized by a sampling scale where it achieves maximum similarity with the central word. We then look at the distribution of sampling scales corresponding to peak similarity for neighborhoods of different words, and if there is a central scale around which they are clustered.

Different Relationships at Different Scales

Mikolov et al. (2013) had showed that vector arithmetic in Word2Vec could encode linguistic relationships—adding a direction vector going from *vec*(`France`) - *vec*(`Paris`), to *vec*(`Germany`) can take us very close to *vec*(`Berlin`). To explore whether the efficiency of such encoding was influenced by the sampling scale, we computed the accuracy of the embeddings in answering a set of such 4-vector analogical reasoning tasks for 14 different linguistic relations, as a function of the sampling scale of the embedding (see Fig. 3)

Figure 3 suggests that the different tests have different sensitivity to scale. There seem to be a number of relations (e.g., "gram4-superlative," "currency," "family") for which peak accuracy is reached at fairly low scales, decaying rapidly after. These are contrasted with some other relations (e.g., "gram1-adjective-to-adverb," "gram6-nationality-adjective," "city-in-state") which reach peak accuracy slowly and at increasingly higher scales. There is quite a bit of variability that is seen in the scale for where the best accuracy scores are reached—ranging from $\beta = 2$ for "gram4-superlative" to $\beta = 35$ for "city-in-state," with quite a few clustered towards the higher end of the spectrum.

If the relationships being measured were all best encoded at a single scale, it would be easy to describe the accuracy scores as a function of scale with a common function. However, accuracy for some measures decreases monotonically while for others accuracy reaches a peak at an intermediate scale. Moreover, the scale at which the different measures peak appears to be different across the measures.

Similarity and Relatedness Best Expressed at Different Scales

The distinction between word similarity and relatedness is reminiscent of the dichotomy between syntagmatic vs paradigmatic associations (de Saussure 1916; Rapp 2002). Paradigmatic associations hinge on word interchangeability in similar context, and can be used to detect semantic similarity (Kliegr and Zamazal 2018). Syntagmatic associations, on the other hand, look at words which co-occur together in sentences (Sahlgren 2006), capturing a broader sense of word association or relatedness.

In Fig. 4, we look at the Spearman correlation scores of similarity values of the word embeddings correlated with the human similarity datasets WordSim–353 (relatedness) and SimLex–999 (similarity), as a function of sampling scale. The scale dependence of these two measures was qualitatively different. The correlation with WordSim starts out at its lowest value, and climbs around 16% as the scale is increased to peak to $r_s = 0.725 \pm 0.001$ at $\beta = 24$ (with the inter-quartile variability between the scores of 10 embeddings at that scale). The correlation seems to stay stable at higher scales with only slight drops in value. In contrast, the correlation scores for SimLex seem to decline almost monotonically (by around 11% from its peak to the lowest, although it is difficult to see with this choice of axes), with a peak of around $r_s = 0.379 \pm 0.002$ at $\beta = 4$.

Sampling a larger context has complementary effects for the correlation scores of the two datasets. WordSim, which measures relatedness between word pairs, tends to benefit greatly from having larger window sizes, while SimLex, which aims to measure semantic similarity independent of association, seems to best align with the embeddings at the smallest scales. This runs counter to the expectation of a single scale being able to capture both these metrics effectively.

This suggests that relatedness, like syntagmatic associations, might need larger sampling scales to effectively capture the gamut of co-occurrences of word pairs in sentences, while semantic similarity, like paradigmatic associations, is a more restrictive measure which might be less effective at larger scales as other related words in the sentence could also get associated with the target word.

Different Neighborhoods at Different Scales

We now look at the neighbors of certain words and how the ordering of neighbors changes as the size of the context sampled was varied, which is shown in Fig 5. The neighbors shown in the graphs are picked by combining the top ten most similar words to each central word at scales $\beta = 1, 10, 100$, to show changing neighborhoods at different scales. The central words come from the 100 most frequent words in the Oxford corpus, of which the first four nouns, verbs, and adjectives are shown in the figure (for neighborhoods of the rest see the Supporting Information).

There seems to be both qualitative variability and quantitative variability among the similarity curves. Neighbors of a central word achieve maximum similarity with the central word at very different sampling scales. There is often clustering of neighbors when they appear in similar contexts. There is a heterogeneity of shapes observed in the similarity curves which would be difficult to explain if we assumed that all the curves have similar scale dependence. It would be difficult to capture these intricate trends of behavior by sampling the text at any single, fixed scale.

If the representation was not sensitive to different information at different scales, we would expect all of the curves—for all seed words—to exhibit the same form of scale dependence. Visually, this would manifest as curves

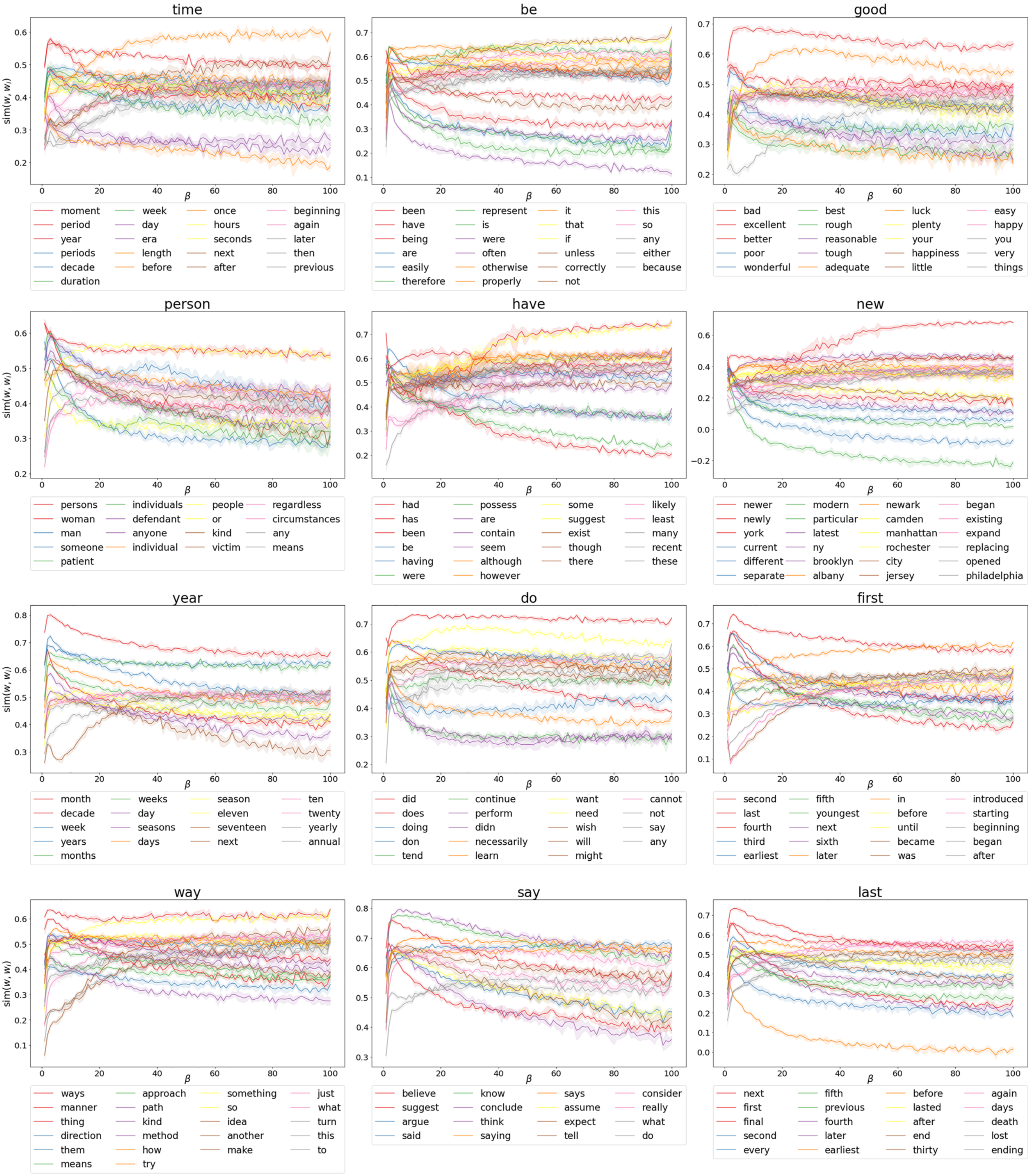

Fig. 5 Relations between words change as a function of scale. The plots show the variation of cosine similarity of neighbors around different central words with the sampling scale (β) of the embedding, for the four most frequent nouns (left), verbs (middle), and adjectives (right) in the Oxford corpus. The neighbors are chosen by pooling together the words most similar to the central word, at scales $\beta = 1, 10, 100$. The similarity curves for different words are seen to cross over at certain scales, which changes the rank ordering of neighbors itself—implying that the shape of the semantic space depends on the scale that we choose

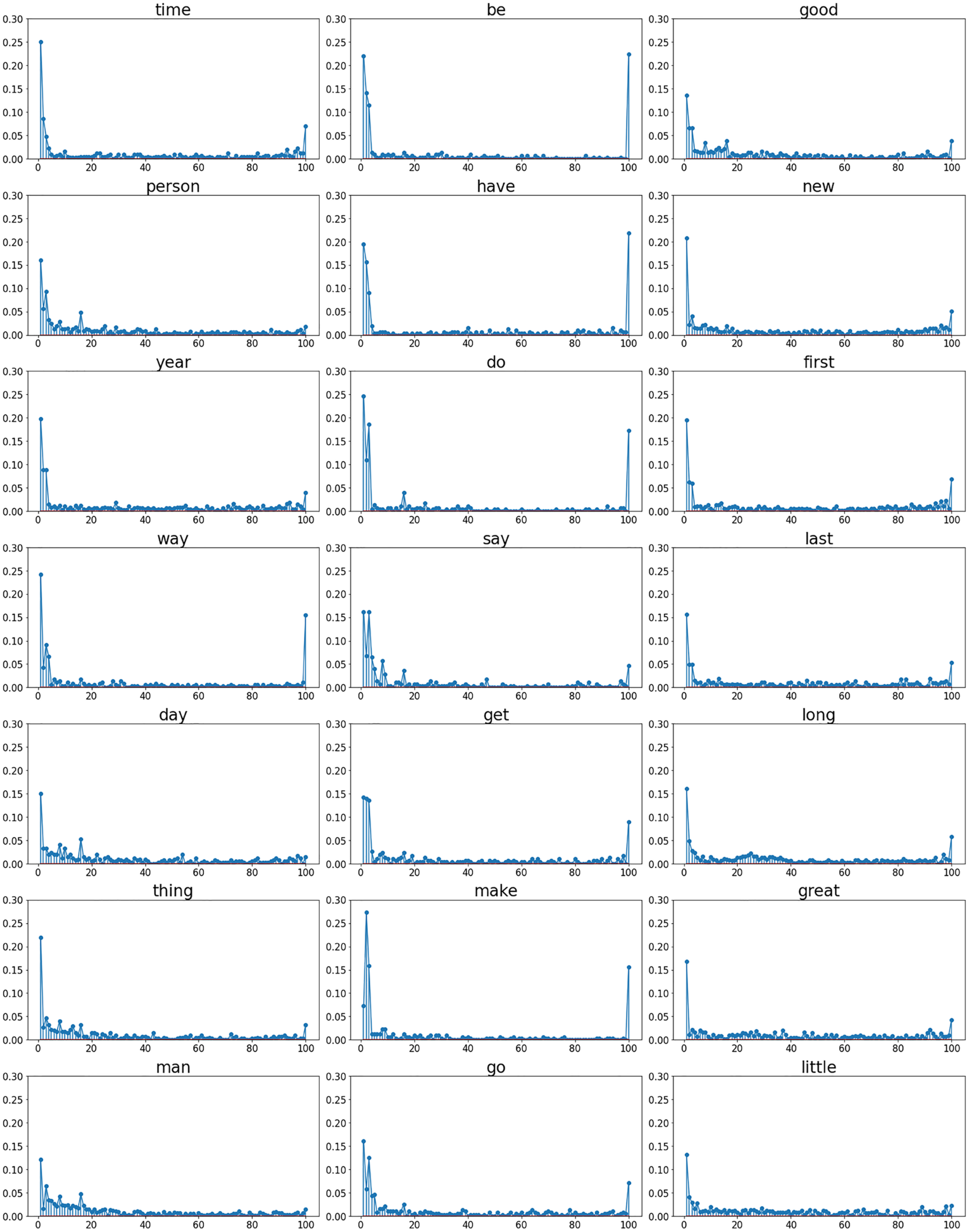

Fig. 6 Neighbors of words show peak similarity at a wide range of scales. The graphs show the normalized fraction of neighbors, of each central word, which attain maximal similarity with the central word at a particular sampling scale (β). The histograms are shown for the seven most frequent nouns (left), verbs (middle), and adjectives (right) in the Oxford corpus. The distributions are not centered around any one scale—the number of neighbors falls off slowly as the scale is varied, with a sizeable fraction of neighbors peaking even at high scales

with similar qualitative shapes. However, the semantic space around a word itself also seems to change as the sampling scale is varied. In each graph, there are multiple instances where the similarity curves intersect, and in many cases different words intersect at different scales. This means that the neighborhood around the seed word changes meaningfully depending on the scale the text is sampled.

Peak Similarity for Neighbors Distributed at Many Different Scales

In the last section, we found that being a neighbor of a central word is a dynamic concept—a word that might be in the top 5 surrounding words at a smaller scale might move to a much more distant position at a larger sampling scale. Each neighbor thus has a range of scales where it is at its closest to the central word. Here, a more systematic analyses of the neighbors are shown, by taking 100 most similar words to the central word at each sampling scale from 1 to 100, and combined to yield the set of neighbors. Choosing a smaller set of similar words did not introduce any qualitative changes in the distributions.

In Fig. 6, neighbors are characterized by the scale at which it came closest to the central word w - each neighbor w_i has a corresponding sampling scale β_i which corresponded to a maxima in similarity (w, w_i). We therefore have a distribution of such sampling scales for each central word, some of which are shown as histograms in Fig 6 (graphs for the rest appear in the Supporting Information). The graphs show the fraction of all the neighbors of each central word which reach a peak similarity score at any given sampling scale.

If language did not carry information about meaning at a range of scales, we might expect the results from this analysis to look quite different. If information about meaning was preferentially carried by a single scale for all words, we would expect these graphs to cluster around this scale, with random fluctuations. In contrast, the results show a heterogeneity in the dependence on scales. Some words peak sharply around a scale of one (e.g., TIME) whereas others show a longer tail (e.g., MAN). Other words do not decrease monotonically but show a second peak at higher scales (e.g., LONG). The Supporting Information shows many more examples.

Discussion

We have shown that the size of context while training Word2Vec can substantially change the properties of the resultant embedding. It is seen that to capture the semantic structure of different linguistic relationships, context has to be captured at a wide spectrum of scales. Because different forms of information are carried at different scales, the performance of a language model depends on its sensitivity to scale. One can classify extant language models based on how they treat information at different timescales.

Language Models with a Single, Fixed Scale

Many contemporary language models sample context at fixed scales. For instance, the introduction of self-attention mechanisms in the Transformer architecture (Vaswani et al. 2017) allowed it to look at the relationships between words and model long-term dependencies without the need for recurrent units or convolution. However, the algorithm trains on fixed-length segments of text, and the self-attention looks at the contribution of all words within this fragment to decipher the meaning of each word. This still constrains the architecture to a fixed scale of context. It also introduces the problem of context fragmentation (Dai et al. 2019), as the fragments scoop up a fixed length of symbols without consideration of sentence structure or semantics. Thus, the model remains completely unaware of the context present in the previous segments when it trains on the current segment, limiting its efficiency in looking at the large-scale contexts present in the text. Transformers are used as building blocks in many state-of-the-art language modeling architectures like BERT (Devlin et al. 2018) from Google and GPT from OpenAI (Radford et al. 2018).

The use of a fixed scale is seen also in older distributional models like latent semantic analysis (LSA) and the topic model (Griffiths et al. 2007; Landauer and Dumais 1997), which work with co-occurrence of words inside larger structures of text (documents). In LSA, the size of the document is chosen a priori (the default choice being 300 words), thus setting a fixed scale. The topic model is generative, as it tries to infer the distribution of words in each topic (a probability distribution over words) and distribution of topics in each document which would best account for the semantic structure in the source text. One still has to choose the number of topics beforehand, however, thus enforcing a scale.

An effective scale is also seen in the syntagmatic-paradigmatic model (SP, Dennis 2004; 2005), which tries to extract structure from text by simultaneously keeping track of syntagmatic and paradigmatic associations between words. Syntagmatic associations are formed between words that occur together, like `run` and `fast`, as opposed to paradigmatic associations, which form between words which appear in similar context, like `run` and `walk`. The model keeps track of these by maintaining memory traces which evaluates and stores different kinds of associations between words. However, these connections are computed between words within sentence-sized chunks, which sets a scale.

A fixed scale buffer has also carried over to moving window models like Word2Vec, and other vector embedding models like GloVe (global vectors for word representation, Pennington et al. 2014). Although the GloVe vectors are constructed to marry the best of both these worlds by calculating the co-occurrence matrix of a word around the context window of another word, choosing the size of the context window still sets a scale.

Language Models that Learn Relevant Timescales

Other contemporary language models do not a priori fix a scale, but nonetheless have a set of scales that are learned via training.

In recurrent neural networks (Elman 1990; Lawrence et al. 2000; Mikolov et al. 2010; Yao et al. 2013), the hidden state at a given time is computed as a function of both the input at that step and the hidden state for the preceding time step. This allows the network to learn dependencies technically without a fixed timescale (Alpay et al. 2016). It can be shown that a RNN learns the relevant timescales it needs to maintain by updating the eigenvalues of the weight matrix connecting the hidden states corresponding to sequential time steps. However, focusing on learning dependencies on only some preferred timescales, even if not fixed, could lead a recurrent network to ignore information at other timescales which could be essential in learning the causal structure of the input data. Training RNNs to learn long-term dependencies using standard gradient descent has also been shown to get increasingly difficult as the timescales to be captured become longer (Bengio et al. 1993; Bengio et al. 1994).

Long-short memory (LSTM) networks (Schmidhuber and Hochreiter 1997) were introduced to tackle both the vanishing and exploding gradient problem in RNNs (Hochreiter et al. 2001) and efficiently learn long-range dependencies (Hochreiter and Schmidhuber 1997). They have been successful in learning structure in language modeling (Sundermeyer et al. 2012; Wang and Jiang 2015; Sundermeyer et al. 2015) and has shown strong performance in benchmarks (Graves 2012; Gers et al. 1999; Greff et al. 2016). However, LSTMs can still suffer from exploding gradients (Pascanu et al. 2012; Le and Zuidema 2016; Grosse 2017). LSTMs have also been shown to empirically use 200 context words on average regardless of the hyperparameters chosen, and start to disregard word order significantly after the first 50 tokens (Khandelwal et al. 2018). More recent language modeling architectures like Ulm-Fit (Howard and Ruder 2018) and contextualized word representations like ELMo (Peters et al. 2018) also use LSTM units as their building blocks, implying that they could also suffer from a effective maximal size of context.

Towards Scale-Invariant Language Models

We have seen that the statistical structure of language simultaneously carries different forms of information at different scales. However, many state-of-the-art language models still address timescales as either a fixed buffer storing context, or attempt at learning relevant timescales as it parses through text. There has been recent efforts to combine features from both these classes (Dai et al. 2019), but the entire spectrum of timescales contained in the data are still not treated equivalently.

Language models with fixed scale inherit this idea from short-term memory models from mid-twentieth century psychology. George Miller's influential paper (Miller 1956) argued the result that we can store "seven plus-or-minus two" simultaneous items of information in short-term memory. The idea of short-term memory as a fixed buffer store existing independently and separately from long-term memory was further developed in the dual-store model (Atkinson and Shiffrin 1968). This classical view of short-term memory acting as fixed-capacity buffer in turn led to early computational models like HAL and BEAGLE (Jones and Mewhort 2007; Lund et al. 1996) which featured a moving window which gathered context around a target word, a feature still used in many contemporary language models.

In the intervening decades, ideas in psychology and neuroscience have evolved towards a scale-invariant working memory (Balsam and Gallistel 2009; Chater and Brown 2008; Gibbon 1977). Biological neural networks exhibit a wide range of timescales and carry information about many different scales, including systematic changes at the scale of seconds, minutes, hours, and even days. (Bernacchia et al. 2011; Mau et al. 2018; Rubin et al. 2015; Cai et al. 2016; Bright et al. 2019). Neuronal ensembles have been seen to fire at increasing latencies following a stimulus with a gradually increasing firing spread (Pastalkova et al. 2008; Eichenbaum 2014; Salz et al. 2016). These *time cells* behave like a short-term memory, retaining information not only about the timing but also the identity of the stimulus (Tiganj et al. 2018; Cruzado et al. 2018), on have a spectrum of timescales. It is possible to build cognitive models from scale-invariant time cells that describe behavior, underscoring the usefulness of a scale-invariant representation of temporal history in models of cognition (Howard et al. 2015).

How would one incorporate these insights into a new generation of language models? It seems like a new generation of language models employing scale-free buffers (Shankar and Howard 2013), which can store information from exponentially long timescales at the cost of discounting temporal accuracy, might be able to learn structure simultaneously from different scales of context. Such a model would not have to direct attention only to a fixed subset of scales, either predetermined or learned, but would be able to attend equally to the entire spectrum of observed timescales, extracting useful predictive information about scale-dependent relationships in natural language.

Conclusion

In this work, we have investigated how the scale of sampling context around each word changes the structure of semantic space learned by Word2Vec. It is seen that different relationships can have markedly different performances at different scales and they seem to be best encoded at a large spectrum of sampling scales. Looking at the individual neighborhoods of word vectors, we find that the local semantic space around words seems to change qualitatively and that the ordering of neighbors around a word can be drastically different based on the scale that context is sampled. We also find that a sizeable fraction of neighbors of a central word can come closest to it even in embeddings sampling context at considerably large scales. The statistics of such maximal scales does not seem to be peaked at any central scale but rather seem to follow a slowly decaying distribution as the sampling scales are increased. These results seem to indicate that there is not a preferred scale to study language—there is different information about the structure of the semantic space at different scales, which would be better analyzed by scale-invariant models of statistical learning.

Acknowledgments We gratefully acknowledge discussions with Kim Stachenfeld, Per Sederberg, and Brandon Jacques. This work was supported by IIS 1631460 and Google AIFRA.

References

Abe, S., & Suzuki, N. (2005). Scale-free statistics of time interval between successive earthquakes. *Physica A: Statistical Mechanics and its Applications*, *350*(2-4), 588–596.

Alpay, T., Heinrich, S., Wermter, S. (2016). Learning multiple timescales in recurrent neural networks. In *International conference on artificial neural networks* (pp. 132–139).

Altmann, E.G., Cristadoro, G., Esposti, M.D. (2012). On the origin of long-range correlations in texts. *Proceedings of the National Academy of Sciences*, *109*(29), 11582-7. https://doi.org/10.1073/pnas.1117723109.

Atkinson, R.C., & Shiffrin, R.M. (1968). Human memory: a proposed system and its control processes. In Spence, K.W., & Spence, J.T. (Eds.) The psychology of learning and motivation, (Vol. 2 pp. 89-105). New York: Academic Press.

Balsam, P.D., & Gallistel, C.R. (2009). Temporal maps and informativeness in associative learning. *Trends in Neuroscience*, *32*(2), 73–78.

Bengio, Y., Frasconi, P., Simard, P. (1993). The problem of learning long-term dependencies in recurrent networks. In *IEEE international conference on neural networks* (pp. 1183–1188).

Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Bernacchia, A., Seo, H., Lee, D., Wang, X.J. (2011). A reservoir of time constants for memory traces in cortical neurons. *Nature Neuroscience*, *14*(3), 366–72.

Bright, I.M., Meister, M.L., Cruzado, N.A., Tiganj, Z., Howard, M.W., Buffalo, E.A. (2019). A temporal record of the past with a spectrum of time constants in the monkey entorhinal cortex. Submitted.

Cai, D.J., Aharoni, D., Shuman, T., Shobe, J., Biane, J., Song, W., Silva, A. (2016). A shared neural ensemble links distinct contextual memories encoded close in time. *Nature*, *534*(7605), 115–118.

Cavagna, A., Cimarelli, A., Giardina, I., Parisi, G., Santagati, R., Stefanini, F., Viale, M. (2010). Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences*, *107*(26), 11865–11870.

Chater, N., & Brown, G.D.A. (2008). From universal laws of cognition to specific cognitive models. *Cognitive Science*, *32*(1), 36–67. https://doi.org/10.1080/03640210701801941.

Cruzado, N.A., Tiganj, Z., Brincat, S.L., Miller, E.K., Howard, M.W. (2018). Compressed temporal representation during visual paired associate task in monkey prefrontal cortex and hippocampus. In *Program no. 243.03 2018 neuroscience meeting planner. San Diego, Society for Neuroscience*.

Dai, Z., Yang, Z., Yang, Y., Cohen, W.W., Carbonell, J., Le, Q.V., Salakhutdinov, R. (2019). Transformer-xl: attentive language models beyond a fixed-length context. arXiv:1901.02860.

Dennis, S. (2004). An unsupervised method for the extraction of propositional information from text. In *Proceedings of the National Academy of Science, USA 101 Suppl*, (Vol. 1 pp. 5206–13).

Dennis, S. (2005). A memory-based theory of verbal cognition. *Cognitive Science*, *29*, 145–193.

de Saussure, F. (1916). Cours de Linguistique g en erale. Paris: Payot. edited posthumously by C. Bally and A. Riedlinger.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Doxas, I., Dennis, S., Oliver, W.L. (2010). The dimensionality of discourse. *Proceedings of the National Academy of Science, USA*, *107*(11), 4866–71.

Ebeling, W., & Neiman, A. (1995). Long-range correlations between letters and sentences in texts. *Physica A: Statistical Mechanics and its Applications*, *215*(3), 233–241.

Ebeling, W., & Pöschel, T. (1994). Entropy and long-range correlations in literary english. *EPL (Europhysics Letters)*, *26*(4), 241.

Eichenbaum, H. (2014). Time cells in the hippocampus: a new dimension for mapping memories. *Nature Reviews Neuroscience*, *15*(11), 732-44. https://doi.org/10.1038/nrn3827.

Elman, J.L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E. (2001). Placing search in context: the concept revisited. In *Proceedings of the 10th international conference on world wide web* (pp. 406–414).

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis, (Vol. 7 pp. 1606–1611).

Gers, F.A., Schmidhuber, J., Cummins, F. (1999). Learning to forget: continual prediction with LSTM.

Gibbon, J. (1977). Scalar expectancy theory and Weber's law in animal timing. *Psychological Review*, *84*(3), 279–325.

Graves, A. (2012). Long short-term memory. In *Supervised sequence labelling with recurrent neural networks* (pp. 37–45): Springer.

Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J. (2016). LSTM: a search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), 2222–2232.

Griffiths, T.L., Steyvers, M., Tenenbaum, J.B. (2007). Topics in semantic representation. *Psychological Review*, *114*(2), 211-44.

Grosse, R. (2017). Lecture 15: Exploding and vanishing gradients. University of Toronto Computer Science.

Hill, F., Reichart, R., Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, *41*(4), 665–695.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks. IEEE Press.

Hochreiter, S., & Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. In *Advances in neural information processing systems* (pp. 473–479).

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv:1801.06146.

Howard, M.W., Shankar, K.H., Aue, W., Criss, A.H. (2015). A distributed representation of internal time. *Psychological Review*, *122*(1), 24–53.

Johns, B.T., Mewhort, D.J., Jones, M.N. (2019). The role of negative information in distributional semantic learning. *Cognitive Science*, *43*(5), e12730.

Jones, M.N., & Mewhort, D.J.K. (2007). Representing word meaning and order information composite holographic lexicon. *Psychological Review*, *114*, 1–32.

Khandelwal, U., He, H., Qi, P., Jurafsky, D. (2018). Sharp nearby, fuzzy far away: how neural language models use context. arXiv:1805.04623.

Kliegr, T., & Zamazal, O. (2018). Antonyms are similar: towards paradigmatic association approach to rating similarity in simlex-999 and wordsim-353. *Data & Knowledge Engineering*, *115*, 174–193.

Landauer, T.K., & Dumais, S.T. (1997). Solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240.

Lawrence, S., Giles, C.L., Fong, S. (2000). Natural language grammatical inference with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering*, *12*(1), 126–140.

Le, P., & Zuidema, W. (2016). Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. arXiv:1603.00423.

Levitin, D.J., Chordia, P., Menon, V. (2012). Musical rhythm spectra from Bach to Joplin obey a 1/f power law. *Proceedings of the National Academy of Sciences*, *109*(10), 3716–3720.

Li, W., Marr, T.G., Kaneko, K. (1994). Understanding long-range correlations in dna sequences. *Physica D: Nonlinear Phenomena*, *75*(1-3), 392–416.

Lin, H.W., & Tegmark, M. (2017). Critical behavior in physics and probabilistic formal languages. *Entropy*, *19*(7), 299. arXiv:1606.06737.

Linkenkaer-Hansen, K., Nikouline, V.V., Palva, J.M., Ilmoniemi, R.J. (2001). Long-range temporal correlations and scaling behavior in human brain oscillations. *Journal of Neuroscience*, *21*(4), 1370–1377.

Lund, K. (1995). Semantic and associative priming in high-dimensional semantic space. In *Proc. of the 17th Annual conferences of the cognitive science society*.

Lund, K., Burgess, C., Atchley, R.A. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments & Computers*, *28*(2), 203–208.

Mahoney, M. (2006). About the test data. https://cs.fit.edu/mmahoney/compression/textdata.html. Online; accessed 12-Dec-2019.

Mantegna, R.N., Buldyrev, S.V., Goldberger, A.L., Havlin, S., Peng, C.-K., Simons, M., Stanley, H.E. (1994). Linguistic features of noncoding DNA sequences. *Physical Review Letters*, *73*(23), 3169.

Mau, W., Sullivan, D.W., Kinsky, N.R., Hasselmo, M.E., Howard, M.W., Eichenbaum, H. (2018). The same hippocampal CA1 population simultaneously codes temporal information over multiple timescales. *Current Biology*, *28*, 1499–1508.

McCormick, C. (2016). Word2Vec tutorial - the skip-gram model. http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model. Online; accessed 12-Dec-2019.

Mikolov, T. (2017). GitHub - tmikolov/word2vec: automatically exported from code.google.com/p/word2vec. https://github.com/tmikolov/word2vec. Online; accessed 12-Dec-2019.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., Khudanpur, S. (2010). Recurrent neural network based language model Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miller, G.A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, *63*, 81–97.

Montemurro, M.A., & Pury, P.A. (2002). Long-range fractal correlations in literary corpora. *Fractals*, *10*(04), 451–461.

Morin, F., & Bengio, Y. (2005). Hierarchical probabilistic neural network language model. In *Aistats Aistats*, (Vol. 5 pp. 246–252).

Oxford English Corpus (2011). Facts about the language. Facts about the language. http://oxforddictionaries.com/words/the-oec-facts-about-the-language. Online;accessed from archived version on 26-May-2020.

Pascanu, R., Mikolov, T., Bengio, Y. (2012). Understanding the exploding gradient problem. arXiv:1211.5063,2.

Pastalkova, E., Itskov, V., Amarasingham, A., Buzsaki, G. (2008). Internally generated cell assembly sequences in the rat hippocampus. *Science*, *321*(5894), 1322-7.

Peng, C.-K., Buldyrev, S.V., Goldberger, A.L., Havlin, S., Sciortino, F., Simons, M., Stanley, H. (1992). Long-range correlations in nucleotide sequences. *Nature*, *356*(6365), 168.

Pennington, J., Socher, R., Manning, C. (2014). GloVe: global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv:1802.05365.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstanding paper.pdf.

Rapp, R. (2002). The computation of word associations: comparing syntagmatic and paradigmatic approaches. In *Proceedings of the 19th international conference on Computational linguistics*, (Vol. 1 pp. 1–7).

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 45–50 Valletta, Malta ELRA*. *http://is.muni.cz/publication/884893/en*.

Roos, P., & Manaris, B. (2007). A music information retrieval approach based on power laws. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, (Vol. 2 pp. 27–31).

Rubin, A., Geva, N., Sheintuch, L., Ziv, Y. (2015). Hippocampal ensemble dynamics timestamp events in long-term memory. *eLife*, *4*, e12247.

Sahlgren, M. (2006). The Word-Space Model. Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces (Unpublished doctoral dissertation).

Sahlgren, M., Holst, A., Kanerva, P. (2008). Permutations as a means to encode order in word space.

Salz, D.M., Tiganj, Z., Khasnabish, S., Kohley, A., Sheehan, D., Howard, M.W., Eichenbaum, H. (2016). Time cells in hippocampal area CA3. *Journal of Neuroscience*, *36*, 7476–7484.

Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. *Neural Comput*, *9*(8), 1735–1780.

Shankar, K.H., & Howard, M.W. (2013). Optimally fuzzy temporal memory. *Journal of Machine Learning Research*, *14*, 3753–3780.

Sundermeyer, M., Ney, H., Schlüter, R. (2015). From feedforward to recurrent LSTM neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(3), 517–529.

Sundermeyer, M., Schlüter, R., Ney, H. (2012). LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Tiganj, Z., Cromer, J.A., Roy, J.E., Miller, E.K., Howard, M.W. (2018). Compressed timeline of recent experience in monkey lPFC. *Journal of Cognitive Neuroscience*, *30*, 935–950.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Polosukhin, I. (2017). Attention is all you need Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wang, S., & Jiang, J. (2015). Learning natural language inference with LSTM. arXiv:1512.08849.

Yao, K., Zweig, G., Hwang, M.Y., Shi, Y., Yu, D. (2013). Recurrent neural networks for language understanding. In *Interspeech* (pp. 2524–2528).