

Sequential Modeling in Vector Space

Benyou Wang¹, Emanuele Di Buccio^{1,2} and Massimo Melucci¹

¹Department of Information Engineering, University of Padova, Padova, Italy

²Department of Statistical Sciences, University of Padova, Padova, Italy

Abstract

In Information Retrieval and Natural Language Processing, representation of discrete objects, e.g., words, usually relies on embedding in vector space; this representation typically ignores sequential information. One instance of such sequential information is temporal evolution. For example, when discrete objects are words, their meaning may smoothly change over time. For this reason, previous works proposed dynamic word embeddings to model this sequential information in word representation explicitly. This paper introduces a representation that relies on sinusoidal functions to capture the sequential order of discrete objects in vector space.

Keywords

sequential modeling, vector space, dynamic word embedding, sinusoidal functions

1. Introduction

Vector space methods have been used in IR for many decades [1]. Recently, the increasing availability of computing resources makes it feasible to embed various types of discrete objects (e.g., words) as dense vectors. For example, word embedding learns a map from a word (denoted as a specific integer index in a pre-defined vocabulary with arbitrary index order) to a D -dimension vector:

$$f : \mathbb{N} \rightarrow \mathbb{R}^D \quad (1)$$

However, such a embedding cannot deal with the spatially or temporally sequential information of objects. One spatial scenario is to encode word order in bag-of-words neural networks like Transformer [2, 3]. Regarding the temporal scenario, word meaning may change over time [4]. For instance, the word `gay` shifted from the meaning `cheerful` in the 1900s to the meaning `frivolous` in the 1950s and finally to the meaning `homosexuality` since the 1990s [5].

In this work, we will focus on the temporally sequential aspect: temporal evolution. This work adopts sinusoidal functions to encode sequential evolution of word meaning change in vector space. The advantages over existing methods might be: 1) it is more *efficient* since the proposed method do not need to maintain a copy of word representation for each timestamp as required by previous works [6]; 2) it can be more *effective* to model semantic evolution since functions can deal with long-term but gradual meaning changes thanks to the continuity of

IIR 2021 – 11th Italian Information Retrieval Workshop, September 13–15, 2021, Bari, Italy

✉ wang@dei.unipd.it (B. Wang)

ORCID 0000-0002-1501-9914 (B. Wang)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Binary coding: orders in 16 numbers (0 – 15) are encoded as four-digit binary numbers 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, . . . Observe that the last digit in red is a periodical sequence of $[0, 1, \dots]$ with a period of 2, the second last digit in blue is a periodical sequence of $[0, 0, 1, 1, \dots]$ with a period of 4, and so on¹.

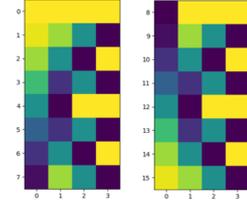


Figure 1: Sinusoidal coding for 0 – 15

functions. In Section 3 we will show how the proposed method could approximate any word meaning evolution.

2. Problem Definition

A object-agnostic order (e.g., position and time) embedding [7, 2] is defined as:

$$f : \mathbb{R} \rightarrow \mathbb{R}^D \quad (2)$$

One may consider binary coding for order embedding. However, it is not differentiable and thus unfriendly to neural networks. To this end, one may consider designing continuous coding with the same periodical property. Fig. 1 shows an alternative sinusoidal encoding [2, 8] with periods of 2, 4, 8, 16. Such continuity will facilitate back-propagation if such embedding is used in neural networks.

Object-aware dynamic evolution. Sequential encoding becomes more challenging when such sequential evolution is not shared among objects; for example, an individual word may change meaning over time, but other words may not share the same trend in meaning change. Therefore, such dynamic evolution processes are object-aware. Formally, evolution of an object with index i can be formalized as a mapping from object (indexed in \mathbb{N}) and time ($t \in \mathbb{R}$) to a D -dimensional vector:

$$f : \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{R}^D \quad (3)$$

3. Methodology: Dynamic Object Embedding

To smoothly model object-aware dynamic evolution, we represent each object as a continuous function: a specific object embedding at time t is represented as the values of the function when the variable equals t . More formally, our approach aims to learn a mapping that maps each object w_i to functions over time/order:

$$f : \mathbb{N} \rightarrow (g : \mathbb{R} \rightarrow \mathbb{R}^D) \quad (4)$$

where f maps a object, e.g., w_i with index i , to a function g , which is a function over a variable $t \in \mathbb{R}$. Note that the output of g is a D -dimensional vector, $g(t) \in \mathbb{R}^D$. Let us denote $f(i)$ as g_i . A object w_i at time t is represented as a D -dimensional vector $\mathbf{U}_{i,t} = f(i)(t) = g_i(t)$.

¹The example is from https://kazemnejad.com/blog/transformer_architecture_positional_encoding

Examples of g are linear functions $g(t) = \mathbf{b} + \mathbf{k}t$ with parameters $\mathbf{b}, \mathbf{k} \in \mathbb{R}^D$ or a sinusoidal functions $g(t) = \mathbf{b} + \mathbf{v} \sin(\omega t + \theta)$ with parameters $\mathbf{b}, \mathbf{v}, \omega, \theta \in \mathbb{R}^D$.

A typical way for word vectors is factoring positive point-wise mutual information (PPMI) matrices [9]. Note that in a temporal scenario, PPMI matrices also changes over time. Assume that the PPMI between a word pair (i, j) at time t is $y_{i,j}(t)$; our goal is to approximate $y_{i,j}(t)$ by a dot product between dynamic word embedding of i , denoted as $f(i)(t) \in \mathbb{R}^D$, and a static compass [10] of j , denoted as $h(j) = \mathbf{v}_j \in \mathbb{R}^D$:

$$y_{i,j}(t) \approx f(i)(t)h(j)^T \quad (5)$$

Sinusoidal Parameterization. By formalizing $f(i, t)$ as sinusoidal functions, i.e., a mixture of cosine and sine functions plus a bias term:

$$f(i)(t) \stackrel{\text{def}}{=} [b_{i,1} + r_{i,1} \sin(\omega_1 t); b_{i,2} + r_{i,2} \cos(\omega_1 t); \dots; b_{i,D-1} + r_{i,D-1} \sin(\omega_{D/2} t); b_{i,D} + r_{i,D} \cos(\omega_{D/2} t);] \quad (6)$$

Eq. 5 will result in:

$$f(i)(t)h(j)^T = \underbrace{\sum_{k=1}^D b_{i,k} v_{j,k}}_{\Delta} + \sum_{k=1}^{\frac{D}{2}} \underbrace{r_{i,2k-1} v_{j,2k-1}}_{\alpha_{i,j,k}} \sin(\omega_k t) + \underbrace{r_{i,2k} v_{j,2k}}_{\beta_{i,j,k}} \cos(\omega_k t) \quad (7)$$

Therefore, $y_{i,j}(t)$ is a weighted sum of sinusoidal functions plus a constant term Δ , i.e., $y_{i,j}(t) = \Delta + \sum_{k=1}^{\frac{D}{2}} \alpha_{i,j,k} \sin(\omega_k t) + \beta_{i,j,k} \cos(\omega_k t)$ $\{\alpha_{i,j,k}\}_{k=1}^{\frac{D}{2}}$ and $\{\beta_{i,j,k}\}_{k=1}^{\frac{D}{2}}$ are the coefficients and $\{\omega_k\}_{k=1}^{\frac{D}{2}}$ are the corresponding frequencies. [11] states that linear combinations of sine and cosine functions could approximate all continuous functions in $\mathcal{C}(I)$. Thus, Eq. 7 could approximate any $y_{i,j}(t) \in \mathcal{C}(I)$, and therefore capture any word meaning evolution. Static object vectors, e.g., [12], can be considered as a special case of constant functions: $g_i = \mathbf{b}_i$, or a specific case of sinusoidal function when $\mathbf{r}_i = \mathbf{0}$ or ω_i is small enough. The additional parameters ω_i and \mathbf{r}_i are expected to capture the dynamic aspect of word meaning evolution. Intuitively, long periods reflect some long-range evolution, although in practice, such sinusoidal functions would not necessarily be periodical with an extremely long period in a limited timespan [13].

4. Ongoing and Future Work

This paper proposes a sinusoidal parameterization to capture the sequential aspects of objects embedded in vector space. We focused on modeling change in word meaning over time; the considered parameterization is promising since, in principle, it could approximate any word meaning evolution. We are currently focusing on the evaluation of the proposed approach to investigate both its effectiveness and efficiency. Experiments will consider diverse tasks, e.g., temporal analogy [6] and semantic change detection [14]. Future work will consider other discrete objects, e.g., user profiles. Moreover, further theoretical and empirical investigations are needed to deal with the optimization issues when sinusoidal activation functions are used, i.e., infinity local minima [13].

Acknowledgments

The work is supported by the Quantum Access and Retrieval Theory (QUARTZ) project, which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 721321.

References

- [1] G. Salton, A. Wong, C.-S. Yang, A vector space model for automatic indexing, *Communications of the ACM* 18 (1975) 613–620.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, *NIPS 2017* (2017).
- [3] B. Wang, D. Zhao, C. Lioma, Q. Li, P. Zhang, J. G. Simonsen, Encoding word order in complex embeddings, in: *ICLR, 2020*.
- [4] B. Wang, E. Di Buccio, M. Melucci, Representing words in vector space and beyond, in: *Quantum-Like Models for Information Retrieval and Decision-Making*, Springer, 2019, pp. 83–113.
- [5] W. L. Hamilton, J. Leskovec, D. Jurafsky, Diachronic word embeddings reveal statistical laws of semantic change, in: *ACL, 2016*, pp. 1489–1501.
- [6] Z. Yao, Y. Sun, W. Ding, N. Rao, H. Xiong, Dynamic word embeddings for evolving semantic discovery, in: *WSDM, 2018*, pp. 673–681.
- [7] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupard, M. Brubaker, Time2vec: Learning a vector representation of time, *arXiv preprint arXiv:1907.05321* (2019).
- [8] B. Wang, L. Shang, C. Lioma, X. Jiang, H. Yang, Q. Liu, J. G. Simonsen, On position embeddings in bert, in: *ICLR, 2021*.
- [9] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, *NIPS 27 (2014)* 2177–2185.
- [10] V. Di Carlo, F. Bianchi, M. Palmonari, Training temporal word embeddings with a compass, in: *AAAI, volume 33, 2019*, pp. 6326–6334.
- [11] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of control, signals and systems* 2 (1989) 303–314.
- [12] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781* (2013).
- [13] G. Parascandolo, H. Huttunen, T. Virtanen, Taming the waves: sine as activation function in deep neural networks, *Openreview preprint* (2016).
- [14] P. Shoemark, F. F. Liza, D. Nguyen, S. Hale, B. McGillivray, Room to glo: A systematic comparison of semantic change detection approaches with word embeddings, in: *EMNLP-IJCNLP, 2019*, pp. 66–76.