

# Exploring Internal Numeracy in Language Models: A Case Study on ALBERT

Ulme Wennberg, Gustav Eje Henter

Division of Speech, Music and Hearing, KTH Royal Institute of Technology  
Lindstedtsvägen 24, SE-100 44 Stockholm, Sweden  
ulme@kth.se, ghe@kth.se

## Abstract

It has been found that Transformer-based language models have the ability to perform basic quantitative reasoning. In this paper, we propose a method for studying how these models internally represent numerical data, and use our proposal to analyze the ALBERT family of language models. Specifically, we extract the learned embeddings these models use to represent tokens that correspond to numbers and ordinals, and subject these embeddings to Principal Component Analysis (PCA). PCA results reveal that ALBERT models of different sizes, trained and initialized separately, consistently learn to use the axes of greatest variation to represent the approximate ordering of various numerical concepts. Numerals and their textual counterparts are represented in separate clusters, but increase along the same direction in 2D space. Our findings illustrate that language models, trained purely to model text, can intuit basic mathematical concepts, opening avenues for NLP applications that intersect with quantitative reasoning.

**Keywords:** Language models; Transformer-based models; Numerical data representation; Word embeddings; PCA; Numerals in NLP

## 1. Introduction

The Transformer architecture introduced by Vaswani et al. (2017) has led to major advances in computational linguistics. Transformer-based models of language like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), and ELECTRA (Clark et al., 2020) have excelled in a range of tasks, from machine translation (Lample et al., 2018; Gu et al., 2018) to question answering (Yamada et al., 2020) and beyond. Studies have also evaluated these models' numerical reasoning. For example, Saxton et al. (2019) found a Transformer-based language model to perform at an E-grade level on a British math exam for 16-year-olds. Kalyan et al. (2021) subsequently developed benchmarks for mathematical commonsense reasoning to track the progress of models in this respect, highlighting the growing interest in this research area.

Despite demonstrable performance on numerical tasks, the origin of these abilities in text-trained models, and the root of their numeracy and quantitative understanding, remain obscure. In this paper, we aim to illuminate this aspect by analyzing the learned embeddings these models use to represent lexical tokens internally. We study how the models have learned to embed numerals and their written representations, unearthing evidence that various embedding vectors capture the essence of numerical concepts. Instead of studying whether the embeddings of different numbers are distributed close together (like one might cluster the embeddings of synonyms; cf. Mikolov et al. (2013)), we thus consider how their representa-

tions *differ*, and how the axes of greatest variation among these concepts relate to the intrinsic ordering and numeric value of the different tokens.

Our paper makes two contributions:

1. We propose a novel way to study internal numerical cognition in language models.
2. We use our proposed method to investigate how ALBERT encodes numerical and ordinal information, and how this varies across different versions of ALBERT of various sizes, independently trained.

Using our proposed method, we find:

- Trained ALBERT models consistently use primary principal component axes to denote ordering and spacing of numbers, ordinals, and magnitude orders.
- The representations are closer together for higher values, suggesting a logarithmic representation of numbers.
- Numerals and their textual counterparts are represented in separate clusters, but increase along the same direction in 2D PCA space.

## 2. Background

In this section, we give a background on numeracy in language models and highlight previous works investigating their internal token representations.

Numeracy is critical for complex reasoning in NLP. Investigations into models' numerical reasoning abilities (Wallace et al., 2019; Jin et al., 2021; Thawani et al., 2021; Duan et al., 2021; Sakamoto and Aizawa, 2021) have shown promising enhancements in model numeracy. Further studies (Kim et al., 2021; Lin et al., 2020; Shah et al., 2023) have

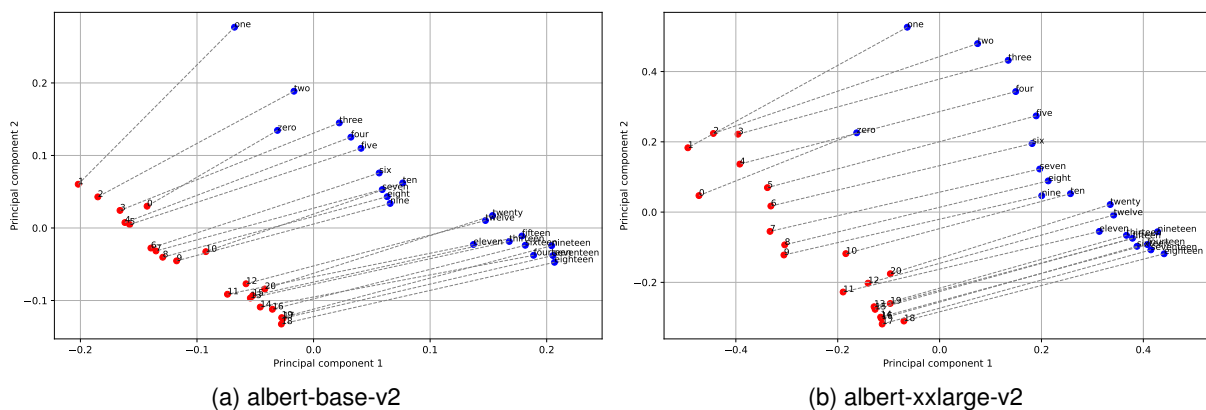


Figure 1: Visualization of the two first principal components of word embeddings for numbers zero through twenty and their textual counterparts in two ALBERT models.

explored models’ abilities to extrapolate and their numerical commonsense knowledge. In conjunction with these explorations, recent methodologies (Sundararaman et al., 2022; Saeed and Papotti, 2022; Jiang et al., 2020; Liang et al., 2022) introduce a variety of approaches to improve numerical representation and processing, showing ongoing efforts to refine numeracy in language models.

Internally, Transformers use self-attention to capture dependencies between all pairs of input vectors. The models use a variety of mechanisms to represent positional information for sequential input data. One common approach, used already in Vaswani et al. (2017), is to encode each token  $w_i$  as a vector that is the sum of a content-embedding vector (that depends only on the token  $w_i$ ) and a position-encoding vector (that depends only on  $i$ , the position in the input sequence). The position-dependence mechanisms either explicitly represent the inherent ordering of input symbols, or, when they do not, have been shown to learn positional representations that capture both this ordering and the translation equivariance of text sequences (Wennberg and Henter, 2021).

Other research (e.g., Mikolov et al., 2013; Vylomova et al., 2016; Durrani et al., 2022) has sought to shed light on information processing in neural language models by analyzing their learned embeddings of different words, concepts, and lexical tokens. A consistent finding is that synonyms cluster together in latent space, meaning that linguistic similarity is reflected internally in the learned model. In this work, we apply a similar analysis, but to concepts that are numerical rather than linguistic. The key difference between our present study and prior work on language-model numeracy is that we look directly at the internal embeddings that Transformer-based language models have learned for numerical concepts, and investigate to what extent *differences* between these embeddings are reflective of differences in numerical value between the mathematical concepts they represent.

### 3. Experiments

We now describe the method and results of our study of the word embeddings inside eight different Transformer-based language models, namely the ALBERT family (Lan et al., 2020). We choose to study ALBERT because it is available in four different model sizes (starting at “base” and going up to “xxlarge”), each with checkpoints at two different points during training (“v1” vs. “v2”, with v2 having been trained for longer), allowing for a comparison of embeddings in different models and their evolution. In our analysis, we specifically examine numerical ranges from zero to twenty and one to one hundred, not only to cover a broad spectrum of basic and multi-digit numerals but also because these numbers are consistently tokenized as single tokens by the ALBERT models (Lan et al., 2020). This choice aligns with our objective to study unambiguous, uncontextualized numerical representations within the model. Many submissions to the GLUE (Wang et al., 2019) and SuperGLUE (Wang et al., 2020) leaderboards are descendants of the ALBERT architecture.

#### 3.1. Analysis Methodology

All the analyses in this paper follow the same underlying recipe:

First, we extract uncontextualized embeddings for selected tokens (single-token words only). These word embeddings are prior to position embedding addition or self-attention layer processing.

We then conduct PCA on these embeddings to identify principal variation axes. This is a linear dimensionality reduction technique, meaning that linear structures like a number line are preserved.

Lastly, we plot embeddings along principal component axes to assess if they capture mathematical concept ordering and if distances reflect mathematical relationships, like proximity of similar numbers.

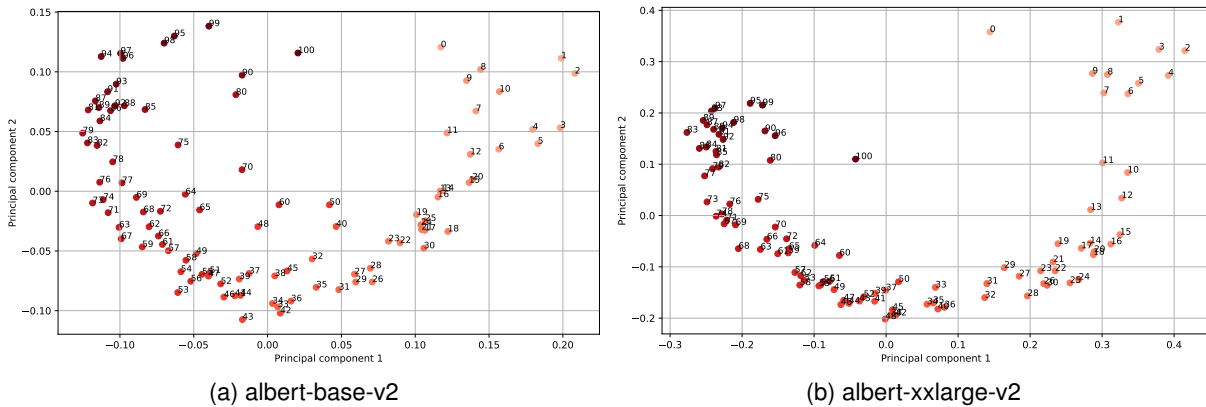


Figure 2: The first and second PCA components for all numbers 1 to 100 in two different ALBERT models.

### 3.2. Numerical vs. Lexical Embedding

We first compare the learned embeddings of the numbers zero through twenty, juxtaposed with their word representations (e.g., “7” versus “seven”), across different ALBERT models. The results of applying our PCA-based analysis to the resulting 42 different representations is visualized in Figure 1 for the smallest and largest ALBERT models (the other models yield very similar-looking plots).

A number of observations are immediately apparent in the figure:

1. Numbers and words representing occupy two distinct, elongated clusters.
2. Within each cluster, there is a direction along which numerical values generally increase. In other words, the values are mostly in order, and we (approximately) recover a number line in the PCA space for each cluster (numbers vs. number words).
3. The direction along which the values increase is the same for both numbers and number words. It would thus easily be possible, particularly for the bigger model, to project the embeddings onto a single axis in PCA space that approximates the number line.
4. When values exceed ten, numbers begin to bunch up more.

The fact that the two different kinds of embeddings can be projected onto something like the number line strongly suggests a learned ability to link numerical symbols to their word forms, and to their approximate value and ordering.

We can also make some minor observations about individual numbers, such as the positions of the numbers and words for zero being idiosyncratic, and (more curiously) that numbers and words for twenty also consistently are out of place.

### 3.3. Numbers 1 Through 100

Next, we performed the same analysis on integers 0 to 100 (excluding word forms) and charted their

2D PCA distribution for the same two ALBERT models. The findings, displayed in Figure 2, mirror those from other models. We observe that:

1. As numbers increase, they approximately trace out a horseshoe shape in 2D space.
2. Larger numbers gradually compress closer together, especially for the larger model.
3. Rounded numbers (i.e., those ending in zero) lie closer to middle of the space. This is more visible for the smaller model, but true for both. 25, 75, and numbers with many powers of two are also closer to the middle. 100, with two zeroes, sticks out particularly much.

The most important conclusion is that the ability to use embeddings to order numbers by size persists into larger numbers, though the spacing gets more compressed as the numerical values increase.

### 3.4. Representing Orders of Magnitude

Having looked at numbers up to 100, we also studied the embeddings of words for different orders of magnitude. Specifically, we performed PCA on the embedding representations of the words “hundred,” “thousand,” “million,” “billion,” and “trillion.” Figure 3 shows these words’ positions on the first principal axis across eight ALBERT models. We see that:

1. The words always respect the expected ordering based on their numerical value.
2. The separation between “hundred” and “thousand” is consistently the shortest, typically by some margin. This evokes comparisons to the logarithmic axis at the bottom of the figure.

There is a close call between “hundred” and “thousand” for the xlarge model, but the separation increases with longer training (model v1 vs. v2).

### 3.5. Words for Ordinals

As our last experiment, we visualize the representation of words for ordinals rather than numerals. Specifically, we apply the same PCA-based method to the embeddings of the terms

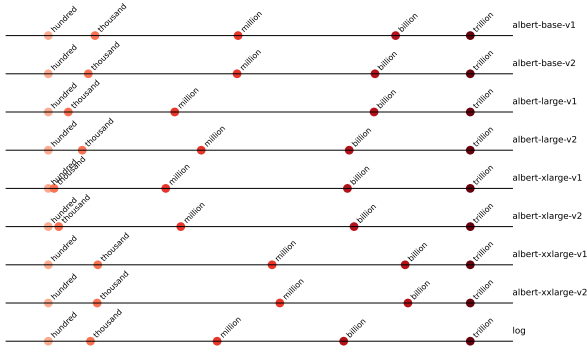


Figure 3: Orders-of-magnitude word embeddings visualized along the first PCA axis across eight ALBERT configurations. Axes have been affinely transformed so that the first and last embeddings line up vertically. The last row shows the concepts arranged on a logarithmic axis for comparison.

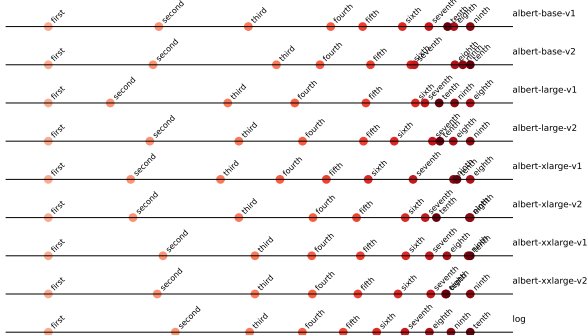


Figure 4: Visualization of ordinal term embeddings along the first PCA axis across eight ALBERT configurations. The axes have been affinely transformed so that the first and last embeddings line up vertically. The last row shows the concepts arranged on a logarithmic axis for comparison.

“first” through “tenth” and visualize the first principal axis for all eight ALBERT models, like in the previous section. The results are shown in Figure 4, from which we make the following observations:

1. Ordinals consistently appear in the correct order along the principal axis of variation up until and including “seventh”.
2. The distance between ordinals gradually decreases as the numbers increase, with the last three ordinals generally being close together and often out of order.

Embeddings do not become obviously better with longer training, especially as they already mostly appear in the correct order for the v1 models.

## 4. Discussion

Reviewing all four analyses, it’s evident that ALBERT models’ internal representation of various numbers and numerical concepts in the embedding

layer directly reflects their numerical value. The representations are closer to a logarithmic than a linear scale. These trends are very consistent across models of different sizes and trained for different amounts of time.

While the fact that Transformer-based language models can support simple mathematical reasoning has indicated some level of numeracy within the model, we can now open the black box and see that numerical knowledge evident within the basic vector representations inside the model. It is not at all obvious that reasonable representations of numerical concepts would arise in these models, given that they are pre-trained exclusively on text to optimize standard language-modeling objectives, with no direct mathematical training.

The observation that larger numbers cluster closer, hinting at logarithmic scaling, and the unique behavior of round numbers in Figure 2, may stem from their occurrence frequency in data, aligning with Benford’s law (Benford, 1938). This law suggests that smaller leading digits are more common in real-world numerical data, yielding a near uniform distribution of digits on a logarithmic scale.

A notable limitation of our study is its focus on single-token numbers, which excludes decimals and larger numerical values from our analysis.

## 5. Conclusion

We have introduced a novel approach to analyzing the quality of numerical representations in language models. This offers insights into model numeracy, which matters for developing improved numerical-understanding capabilities for Transformer-based language models.

We use our method to investigate how ALBERT, an important Transformer-based language model architecture, represents different numerical and ordinal inputs. Our results demonstrate a clear concept of numerical ordering within the vector representations inside the model. Representations of larger numbers fall closer together, suggestive of models using logarithmic axis representations internally. The findings are very robust, in that they appear essentially unchanged across eight different models that differ in size and training duration.

Going beyond numerical order, future work should seek to quantify to what extent learned internal structures reflect interval and ratio scales, as well as to what extent factors like model architecture and term frequency in the corpus contribute to (or otherwise influence) these structures. Another goal is to extend the analysis to multi-token numbers and mathematical operators, and connect with emerging understanding of how models then perform stepwise mathematical processing in latent space (Lee et al., 2019; Valentino et al., 2023).



## 6. Acknowledgments

We thank the anonymous reviewers for constructive feedback. This research was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

## 7. Bibliographical References

- Frank Benford. 1938. The law of anomalous numbers. *Proc. APS*, pages 551–572.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [Pre-training transformers as energy-based cloze models](#). In *Proc. EMNLP*, pages 285–294.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proc. NAACL-HLT*, pages 4171–4186.
- Hanyu Duan, Yi Yang, and K. Tam. 2021. Learning numeracy: A simple yet effective number embedding approach using knowledge graph. In *Proc. EMNLP*.
- Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Firoj Alam. 2022. [On the transformation of latent space in fine-tuned NLP models](#). In *Proc. EMNLP*, pages 1495–1516.
- Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor O. K. Li. 2018. [Meta-learning for low-resource neural machine translation](#). In *Proc. EMNLP*, pages 3622–3631.
- Chengyue Jiang, Zhonglin Nian, et al. 2020. Learning numeral embedding. *arXiv preprint arXiv:2001.00003*.
- Zhihua Jin, Xin Jiang, et al. 2021. NumGPT: Improving numeracy ability of generative pre-trained models. *arXiv preprint arXiv:2101.03137*.
- Ashwin Kalyan, Abhinav Kumar, Arjun Chandrasekaran, Ashish Sabharwal, and Peter Clark. 2021. [How much coffee was consumed during EMNLP 2019? Fermi problems: A new reasoning challenge for AI](#).
- Jeonghwan Kim, Giwon Hong, Kyung min Kim, Junmo Kang, and Sung-Hyon Myaeng. 2021. Have you seen that number? investigating extrapolation in question answering models. In *Proc. EMNLP*.
- Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018. [Phrase-based & neural unsupervised machine translation](#). In *Proc. EMNLP*, pages 5039–5049.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. [ALBERT: A lite BERT for self-supervised learning of language representations](#). In *Proc. ICLR*.
- Dennis Lee, Christian Szegedy, Markus N. Rabe, Sarah M. Loos, and Kshitij Bansal. 2019. [Mathematical reasoning in latent space](#).
- Zhenwen Liang, Jipeng Zhang, et al. 2022. MWP-BERT: Numeracy-augmented pre-training for math word problem solving. In *Proc. NAACL*.
- Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. In *Proc. EMNLP*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#).
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Distributed representations of words and phrases and their compositionality](#). *Proc NIPS*, 26.
- Mohammed Saeed and Paolo Papotti. 2022. You are my type! type embeddings for pre-trained language models. In *Proc. EMNLP*.
- Taku Sakamoto and Akiko Aizawa. 2021. Predicting numerals in natural language text using a language model considering the quantitative aspects of numerals. *Proc. DEELIO*.
- David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *Proc. ICLR*.
- Raj Sanjay Shah, Vijay Marupudi, Reba Koenen, Khushi Bhardwaj, and S. Varma. 2023. Numeric magnitude comparison effects in large language models. In *Proc. ACL*.
- Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Liyan Xu, and L. Carin. 2022. Improving downstream task performance by treating numbers as entities. In *Proc. ICLR*.
- Avijit Thawani, Jay Pujara, and Fredrik Ilievski. 2021. Numeracy enhances the literacy of language models. In *Proc. EMNLP*.

- Marco Valentino, Jordan Meadows, Lan Zhang, and André Freitas. 2023. [Multi-operational mathematical derivations in latent space](#).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Proc. NIPS*, pages 5998–6008.
- Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proc. ACL*, pages 1671–1682.
- Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proc. EMNLP*.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. [Superglue: A stickier benchmark for general-purpose language understanding systems](#).
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. [Glue: A multi-task benchmark and analysis platform for natural language understanding](#).
- Ulme Wennberg and Gustav Eje Henter. 2021. The case for translation-invariant self-attention in transformer-based language models. In *Proc. ACL*.
- Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. [LUKE: Deep contextualized entity representations with entity-aware self-attention](#). In *Proc. EMNLP*, pages 6442–6454.