Exploring Numeracy in Word Embeddings

Aakanksha Naik*, **Abhilasha Ravichander***,
Carolyn Rose, **Eduard Hovy**
Language Technologies Institute, Carnegie Mellon University
{anaik, aravicha, cprose, ehovy}@cs.cmu.edu

Abstract

Word embeddings are now pervasive across NLP subfields as the de-facto method of forming text representataions. In this work, we show that existing embedding models are inadequate at constructing representations that capture salient aspects of mathematical meaning for numbers, which is important for language understanding. Numbers are ubiquitous and frequently appear in text. Inspired by cognitive studies on how humans perceive numbers, we develop an analysis framework to test how well word embeddings capture two essential properties of numbers: *magnitude* (e.g. 3<4) and *numeration* (e.g. 3=three). Our experiments reveal that most models capture an approximate notion of magnitude, but are inadequate at capturing numeration. We hope that our observations provide a starting point for the development of methods which better capture numeracy in NLP systems.

1 Introduction

Word embeddings operationalize the distributional hypothesis, where a word is characterized by "the company it keeps" (Harris, 1954; Firth, 1957), and have been shown to capture semantic regularities in vector space (Mikolov et al., 2013c). They have been a driving force in NLP in recent years, and enjoy widespread use in a variety of semantic tasks (Rumelhart et al.; Mikolov et al., 2013a,b; Collobert and Weston, 2008; Glorot et al., 2011; Turney and Pantel, 2010; Turney, 2013).

However, to what extent do these word representations capture numeric properties? Numbers often need to be dealt with precisely, and understanding the meaning of text also requires a careful understanding of the quantities involved. They have been identified to play an important role in textual entailment, a benchmark natural language understanding task. Marneffe et al. (2008) extract pairs of contradictions that occur naturally on Wikipedia and Google News, and find that as many as 29% of contradictions arise due to numeric discrepancies. They also identify that on many Recognizing Textual Entailment (RTE) datasets, 8.8% of contradictory pairs feature numeric contradictions. Naik et al. (2018) find that model inability to do numerical reasoning causes 4% of errors made by state-of-the-art models in Natural Language Inference. Spithourakis and Riedel (2018) emphasize the importance of numeracy in language modeling. Yet, numbers are often forgotten and even masked in NLP applications (Mitchell and Lapata, 2009).

In several domains such as economics, finance and scientific articles numbers can play a crucial role in text. Take for example a recent news headline,

> Met Office: Global Warming could exceed 1.5 C within five years

Ideally, the text representation we use should be able to capture that global warming can exceed 1.5 C, not 100 C. **Magnitude** is an essential aspect of a number's *meaning*[1] (Dehaene et al., 1998; Whalen et al., 1999; Cantlon and Brannon, 2006; Gross, 2011; Cutini and Bonato, 2012; Agrillo et al., 2012; Feigenson et al., 2004) . Systems should also be able to draw valid inferences irrespective of whether the text uses "five" or "5". This requires an understanding of symbolic representations used to record numbers in text. Such representation systems are called *numeration systems*, and individual symbols within the system

*The first two authors contributed equally to this work.

[1]Prior work has shown that humans, as well as several species of animals share analogue systems that represent "quantities" or "magnitudes" associated with numbers(Dehaene et al., 1998)

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3374–3380
Florence, Italy, July 28 - August 2, 2019.

are called *numerals*[2]. Systems must handle **numeration**, i.e. associations between distinct symbols used for the same number under different systems (3=three).

In this work, we examine the extent to which word embeddings are capable of representing numeracy attributes, asking the question - *if pretrained word embeddings are utilized for representing text across NLP tasks, what can they represent about numbers?* Our framework formulates triples of numbers to probe word embeddings on their ability to represent magnitude, and their robustness to differences in numeration. We hope this analysis highlights limitations of current pretrained word embeddings at capturing numeracy, and will motivate future research to develop more careful treatments of quantities in text.

2 Analysis Framework

We construct an analysis framework to evaluate embeddings on their ability to capture *magnitude* and *numeration*. Numbers follow a well-defined ordering, under a mathematical system, which holds independent of textual context (e.g.: $0 < 1 < 2$...). This ordering is established by magnitude (Izard and Dehaene, 2008; Russell, 2009) and is consistent across numeration systems. Therefore, an embedding representation that captures magnitude and numeration precisely should maintain this ordering across numeration systems in the embedding space. We evaluate this ability by constructing contrastive tests (Zhu et al., 2018).

A contrastive test for a property p is defined as a triple (x, x_+, x_-) such that x is closer to x_+ than x_- under p. If embeddings capture p, x will be closer to x_+ than x_- in the embedding space, indicating that the embedding method passes the test. We propose three categories of tests, which differ in the choice of x_-[3]:

1. **OVA (One-vs-All)**: Define $x_- = \{y | y \in X - x, y \neq x_+\}$. A model must identify x to be closer to x_+ than all x_-.
2. **SC(Strict Contrastive)**: Choose x_- to be the second-closest to x after x_+ under p.
3. **BC (Broad Contrastive)**: Choose x_- to be the furthest from x under property p.

[2]Several cultures have developed numeration systems (Zhang and Norman, 1995). In this work, we restrict our scope to Arabic and English numeration systems (e.g. Arabic-2, English: two).

[3]x_+ is chosen to be the token closest to x under p.

Model	#English	#Arabic
GloVe-6B-*D	120 (0.03%)	19409 (4.85%)
GloVe-42B-300D	239 (0.01%)	108839 (5.68%)
GloVe-840B-300D	532 (0.02%)	109353 (4.98%)
FastText-Wiki	374 (0.04%)	25549 (2.56%)
FastText-CC	592 (0.03%)	59386 (2.97%)
SkipGram-BoW	114 (0.06%)	2401 (1.31%)
SkipGram-Dep	111 (0.06%)	2416 (1.39%)
GloVe-Num	1117 (0.02%)	318109 (4.4%)
GloVe-All	973 (0.01%)	189598 (2.8%)
FastText-Num	1117 (0.02%)	317627 (4.4%)
FastText-All	973 (0.01%)	189366 (2.8%)
Word2Vec-Num	486 (0.02%)	67908 (2.7%)
Word2Vec-All	434 (0.01%)	37164 (1.2%)

Table 1: Proportion of English and Arabic numerals containing representations in different models. Though embeddings are retrained on the same corpus, preprocessing choices (eg:lowercasing, filtering low frequency words etc.) result in different vocabularies

OVA requires that x_+ must be the closest vector to x in the embedding space. High performance on this test would indicate that the property is captured almost precisely. SC relaxes strictness by only requiring x_+ to be closer than the second-closest token under property p. Finally BC is the least strict of the three. Models can succeed on BC if they manage to capture even an approximate notion of p. We use this framework to construct three categories of contrastive tests for both magnitude and numeration. Example tests for magnitude are shown below[4]:

1. **OVA-MAG**: $(3, 4, x)$, such that $x = \{y | y \in X - \{3\}, y \neq 4\}$
2. **SC-MAG:** $(3, 4, 5)$
3. **BC-MAG:** $(3, 4, 1000000)$

Similarly for numeration,

1. **OVA-NUM:** $(3, three, x)$, such that $x = \{y | y \in Y, y \neq three\}$
2. **SC-NUM:** $(3, three, four)$
3. **BC-NUM:** $(3, three, billion)$

3 Representations

We evaluate the following embedding methods:

Skipgram (Mikolov et al., 2013a): Feedforward network trained to predict words within a fixed window surrounding the current word, with hidden weights used as embeddings. We evaluate with window sizes in $\{2, 5\}$, dependency

[4]Note that we consider 2 and 4 equidistant from 3, so examples like (3,2,4) are removed.

Model	OVA-MAG	SC-MAG	BC-MAG
Random	0.04	49.82	49.34
GloVe-6B-50D	7.70	55.62	82.48
GloVe-6B-100D	10.27	57.83	82.83
GloVe-6B-200D	15.88	62.21	83.94
GloVe-6B-300D	**18.41**	**62.92**	83.98
GloVe-42B-300D	5.18	55.58	91.86
GloVe-840B-300D	11.06	55.40	88.54
FastText-Wiki	13.94	59.96	96.15
FastText-CC	7.83	53.89	85.40
SkipGram-2	7.12	55.49	95.84
SkipGram-5	8.85	55.40	**96.42**
SkipGram-Dep	3.32	51.99	94.60

Table 2: Performance (% accuracy) of various embedding models on magnitude tests. We also report the performance of a random embedding baseline.

parse-based context (Levy and Goldberg, 2014)

GloVe (Pennington et al., 2014): Embeddings generated by training log-bilinear models to predict global word co-occurrence statistics. We evaluate variants with #tokens in {6B, 42B, 840B}; dimensionality in {50, 100, 200, 300}

FastText (Bojanowski et al., 2017): Extended Skipgram model representing words as character n-grams to incorporate sub-word information. We evaluate Wikipedia and Common Crawl variants.

3.1 Retrained Word Vectors

We retrain all models on GigaWord[5] and English Wikipedia[6], under the setting: window size=5; dimensionality=100. To evaluate whether having more occurrences of numerals in the training data correlates with better representations, we train two variants for each model: one on sentences containing numbers (56M in total; 1.5B tokens) (**Num**), and another on 56M sentences (1.5B tokens) subsampled without constraints (**All**).

4 Experiments

How many numerals have representations? Table 1 shows the proportion of English[7] and Arabic numerals in each. Overall, numerals make up less than 5% vocabulary in all models. Despite this, all variants contain representations for sufficient numerals to allow us to apply our framework.

For off-the-shelf variants, we construct 2260 OVA-MAG, SC-MAG and BC-MAG tests. For numeration, we construct separate tests for each model, as there are few common numerals. Further statistics about number of tests for each model are reported in table 3. For retrained embeddings, we construct 31860 OVA-MAG, SC-MAG and BC-MAG tests, 130 OVA-NUM and SC-NUM tests, and 136 BC-NUM tests[8].

4.1 Evaluating Off-The-Shelf Embeddings

Tables 2 and 3 present the performance of off-the-shelf embeddings on magnitude and numeration tests respectively. We use cosine similarity[9] as the distance metric. High performance on BC-MAG indicates that all models capture an approximate notion of magnitude, distinguishing between very large and very small numbers. We speculate this might be because numbers from different magnitude classes often appear in different contexts (See §5.1). As tests become stricter, model performance drops massively. Models perform nearly 10x worse on OVA-MAG as compared to BC-MAG. This suggests model are unable to capture magnitude *precisely*. Across models, SkipGram variants and FastText-Wiki perform best on BC-MAG. However, GloVe outperforms all others on OVA-MAG and SC-MAG. On numeration tests, models fare much worse. With the exception of GloVe models on BC-NUM, no model significantly outperforms a random baseline.

4.2 Evaluating Retrained Embeddings

Table 4 presents the performance of retrained embeddings and a random embedding baseline on magnitude and numeration tests. There is no significant difference in performance between **Num** and **All** variants, suggesting that seeing more numerals during training does not necessarily result in better representations. Results follow similar trends as off-the-shelf embeddings. All models capture an approximate notion of magnitude (high performance on BC-MAG), but do not capture numeration. Across models, FastText variants fare

[5]We use the fourth edition: https://catalog.ldc.upenn.edu/LDC2009T13.

[6]We use the May 1, 2019 dump from https://dumps.wikimedia.org/backup-index.html.

[7]To detect English numerals, we use word2number: https://pypi.org/project/word2number/.

[8]Since all embeddings are trained on the same corpus and share the same vocabulary, there are enough common English numerals to construct a single set of numeration tests.

[9]We experiment with Euclidean distance, and observe similar results (Appendix A and B).

Model	OVA-NUM			SC-NUM			BC-NUM		
	#Tests	Rand	Emb	#Tests	Rand	Emb	#Tests	Rand	Emb
GloVe-6B-50D	117	0.00	0.85	117	49.57	52.99	117	50.43	**79.49**
GloVe-6B-100D	117	0.00	0.85	117	52.99	47.86	117	57.26	**81.20**
GloVe-6B-200D	117	1.71	0.85	117	48.72	**57.26**	117	42.74	**78.63**
GloVe-6B-300D	117	1.71	0.00	117	50.43	**58.97**	117	54.70	**88.89**
GloVe-42B-300D	226	0.44	0.44	226	52.21	51.33	226	53.98	10.18
GloVe-840B-300D	515	0.19	0.19	515	49.90	50.68	515	49.71	**81.94**
FastText-Wiki	360	0.28	0.28	360	50.00	49.72	360	56.67	41.67
FastText-CC	572	0.00	**0.52**	572	46.85	51.22	572	41.26	44.76
SkipGram-2	112	0.00	0.00	112	51.79	48.21	112	49.11	49.11
SkipGram-5	112	0.00	0.89	112	52.68	51.79	112	50.89	14.29
SkipGram-Dep	109	0.92	**1.83**	109	53.21	48.62	109	52.29	31.19

Table 3: Performance (% accuracy) of various embedding models on numeration tests. Since we construct a separate set of tests per model, we report the performance of a random embedding model for each set (Rand). Bolded numbers highlight cases where performance is higher than both random embedding and random choice. Note that random choice performance for OVA-NUM is $\frac{1}{\#Tests}$.

Model	Magnitude			Numeration		
	OVA-MAG	SC-MAG	BC-MAG	OVA-NUM	SC-NUM	BC-NUM
random	0.00	49.62	49.71	**2.31**	47.69	53.68
GloVe-Num	0.01	49.47	72.76	0.00	50.00	19.85
GloVe-All	0.01	49.08	74.02	0.00	46.15	19.85
FastText-Num	**0.09**	51.05	96.69	1.54	**54.62**	58.09
FastText-All	**0.09**	**51.16**	**97.90**	0.00	46.92	**61.03**
Word2Vec-Num	0.02	50.12	93.55	0.77	44.62	33.82
Word2Vec-All	0.02	49.37	94.20	0.00	**54.62**	34.56

Table 4: Performance (% accuracy) of various (retrained) embedding models on magnitude and numeration tests.

best.

5 Discussion

5.1 Performance on Magnitude Tests

Tables 2 and 4 show that most models do not capture magnitude *precisely* (low performance on OVA-MAG; SC-MAG), but seem to learn an approximate notion of magnitude (high performance on BC-MAG) [10]. To examine the difference in contexts that separates numbers of vastly varying magnitudes, we sample 1 million sentences containing numbers from English Wikipedia and GigaWord and compute pointwise mutual information (PMI), defined as

$$\text{PMI}\ (number, class) = \log \frac{p(number,class)}{p(number,\cdot)p(\cdot,class)}$$

between the contexts of primitive numbers (numbers 1-10) and large numbers (>500, >1000, >3000, >10000, >100000) as shown in Table 5. We consider the word immediately following the number as context, since it appears in the context of the number across embedding methods, regardless of sliding window size. We apply add-100 smoothing to identify contexts with maximum discriminatory power.

We observe in table 5 that terms separating primitives from larger numbers fall into categories such as days in a month, which are less than 31, or percentages which are <= 100. In comparison, contexts of larger numbers include terms like *'election'*, *'census'* and *'world'*. As we move beyond numbers that are likely to be dates (>3000), we observe terms such as 'ZIP' occurring with ZIP codes in text, 'block' occurring in contexts such as 'house in 9600 block of Washington Boulevard, 'Refugees' which appears in contexts such as 'relocate about 125,000 refugees away from the border'. We observe that different contexts characterize classes of numbers, and speculate that this may allow embeddings to distinguish between numbers that appear consistently in vastly different contexts

[10] Cognitive studies show that human babies initially start recognizing numbers by approximation and their ability to identify numbers precisely improves over their lifespan (Halberda et al., 2012). (Moyer and Landauer, 1967) were the first to observe that humans took longer to distinguish between closer numbers (eg: 8 and 9) than numbers which were further away in distance (eg: 2 and 9). This finding has since been replicated several times (Dehaene, 2011). In our framework, models find it harder to distinguish between closer numbers (SC-MAG) than distant numbers (BC-MAG)- however the differences here likely arise from different contexts in which numbers of vastly varying magnitudes are used.

	Primitives	λ = 500	Primitives	λ = 1000	Primitives	λ = 3000	Primitives	λ = 10000	Primitives	λ = 100000
Wiki	%	Summer	%	Summer	%	BC	%	Exchange	%	Elected
	July	Census	million	Census	million	RPM	million	HD	million	Ontario
	January	Film	July	Film	July	BCE	July	Departs	May	Owner
	April	World	January	World	January	Inhabitants	January	Delhi	July	Spinneys
	September	Election	April	Election	May	Hollywood	May	Raxaul	January	Thana
GW	percent	index	percent	World	percent	DOWN	percent	novos	percent	novos
	p.m	GMT	million	GMT	million	Composite	million	ZIP	million	Tel
	a.m	World	billion	Olympic	billion	block	billion	University	billion	NDI
	trillion	Olympic	p.m	Olympics	p.m	LAS	points	UP	points	Refugees
	billion	Olympics	years	season	years	UP	p.m	Old	p.m	Eritrean

Table 5: Top 5 nouns by $PMI(word, class)$ for primives and large numbers (numbers $> \lambda$), in 1 million sentences drawn from Wikipedia (wiki) and GigaWord (GW) respectively.

leading to good performance on BC-MAG.

5.2 Recovering magnitude information from nearest neighbours

Model performance on SC-MAG and BC-MAG indicates whether ordering relationships between a number,its closest, second-closest, and furthest numbers are maintained. However, infinite numbers exist, making it infeasible to construct contrastive tests to check ordering relationships between all triples. To mitigate this, we experiment with a paradigm that performs regression with a number's nearest neighbors to predict its magnitude. If magnitude can be recovered from the structure of the embedding space, this provides evidence that magnitude ordering relations are maintained to some extent. For this experiment, we divide the set of 2260 numbers common across off-the-shelf variants[11] into training (80%) and test (20%) sets and run a kNN (k-nearest neighbor) regressor model to predict magnitude. R2 scores for are shown in table 6. Most models show reasonably high R2 scores, indicating that some ordering relationships must be maintained, helping embeddings capture approximate notions of magnitude. While this property of current embedding models is interesting, their failure to capture precise magnitude is an important issue. Word embeddings are used for semantic tasks such as natural language inference or reading comprehension, wherein models might need to reason more precisely about numbers.

Model	R2 Score
GloVe-6B-50D	0.53
GloVe-6B-100D	0.75
GloVe-6B-200D	0.67
GloVe-6B-300D	0.62
GloVe-42B-300D	0.44
GloVe-840B-300D	**0.83**
FastText-Wiki	0.71
FastText-CC	0.56
SkipGram-2	0.67
SkipGram-5	0.76
SkipGram-Dep	0.41
GloVe-Num	0.12
GloVe-All	0.30
FastText-Num	0.73
FastText-All	0.47
Word2Vec-Num	0.68
Word2Vec-All	0.65

Table 6: Results of kNN Regression Test for Magnitude

6 Conclusion

Current NLP systems rely heavily on word embeddings. In this work we demonstrate that three popular embedding models are inadequate at dealing precisely with numbers, in two aspects: magnitude and numeration. We hope this work will promote a more careful treatment of language, and serve a cautionary purpose against using word embeddings in downstream tasks without recognizing their limitations. This work also raises important questions about other categories of word-like tokens that need to be treated like special cases. We hope the community will carefully consider that distributed word representations cannot be relied upon in all scenarios.

7 Acknowledgements

This work has partially been supported by the National Science Foundation under Grant No. CNS 13-30596. The authors would like to thank Thomas Manzini, Shruti Rijhwani and Siddharth Dalmia for helpful discussions and reviews while drafting this paper.

[11] We do this to compare results across all models. Retrained variants contain embeddings for all 2260 numbers.

References

Christian Agrillo, Laura Piffer, Angelo Bisazza, and Brian Butterworth. 2012. Evidence for two numerical systems that are similar in humans and guppies. *PloS one*, 7(2):e31923.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jessica F Cantlon and Elizabeth M Brannon. 2006. Shared system for ordering small and large numbers in monkeys and humans. *Psychological science*, 17(5):401–406.

Ronan Collobert and Jason Weston. 2008. 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. ACM.

Simone Cutini and Mario Bonato. 2012. Subitizing and visual short-term memory in human and non-human species: a common shared system? *Number without language: comparative psychology and the evolution of numerical cognition*, 129.

Stanislas Dehaene. 2011. *The number sense: How the mind creates mathematics*. OUP USA.

Stanislas Dehaene, Ghislaine Dehaene-Lambertz, and Laurent Cohen. 1998. Abstract representations of numbers in the animal and human brain. *Trends in neurosciences*, 21(8):355–361.

Lisa Feigenson, Stanislas Dehaene, and Elizabeth Spelke. 2004. Core systems of number. *Trends in cognitive sciences*, 8(7):307–314.

J. Firth. 1957. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford. Reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520.

Hans J Gross. 2011. To bee or not to bee, this is the question the inborn numerical competence of humans and honeybees: The inborn numerical competence of humans and honeybees. *Communicative & integrative biology*, 4(5):594–597.

Justin Halberda, Ryan Ly, Jeremy B Wilmer, Daniel Q Naiman, and Laura Germine. 2012. Number sense across the lifespan as revealed by a massive internet-based sample. *Proceedings of the National Academy of Sciences*, 109(28):11116–11120.

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

Véronique Izard and Stanislas Dehaene. 2008. Calibrating the mental number line. *Cognition*, 106(3):1221–1247.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Marie-Catherine Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding contradictions in text. *Proceedings of ACL-08: HLT*, pages 1039–1047.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. 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, Atlanta, Georgia. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2009. Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 430–439. Association for Computational Linguistics.

Robert S Moyer and Thomas K Landauer. 1967. Time required for judgements of numerical inequality. *Nature*, 215(5109):1519.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Bertrand Russell. 2009. *Principles of mathematics*. Routledge.

Georgios P Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. *arXiv preprint arXiv:1805.08154*.

Peter D Turney. 2013. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics*, 1:353–366.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

John Whalen, Charles R Gallistel, and Rochel Gelman. 1999. Nonverbal counting in humans: The psychophysics of number representation. *Psychological Science*, 10(2):130–137.

Jiajie Zhang and Donald A Norman. 1995. A representational analysis of numeration systems. *Cognition*, 57(3):271–295.

Xunjie Zhu, Tingfeng Li, and Gerard de Melo. 2018. Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 632–637, Melbourne, Australia. Association for Computational Linguistics.

A Magnitude Tests with Euclidean Distance

Table 7 describes the performance of word embedding models on magnitude tests with Euclidean distance.

Model	OVA-MAG	SC-MAG	BC-MAG
Random	0.04	49.82	49.34
GloVe-6B-50D	7.7	54.87	79.78
GloVe-6B-100D	10.27	57.12	78.5
GloVe-6B-200D	15.88	58.72	80.09
GloVe-6B-300D	18.41	60.44	79.82
GloVe-42B-300D	5.18	55.27	55.09
GloVe-840B-300D	11.06	55.49	98.23
SkipGram-2	8.85	55.35	96.37
SkipGram-5	7.12	55.44	95.8
SkipGram-Dep	3.32	51.95	94.56
FastText-CC	7.83	54.07	91.28
FastText-Wiki	13.94	59.34	98.19

Table 7: Performance (% accuracy) of embedding models on magnitude tests with Euclidean distance

B Numeration Tests with Euclidean Distance

Tables 8 and 9 describe the performance of word embedding models on numeration tests with Euclidean distance.

Model	SC-NUM		
	#Tests	Rand	Emb
GloVe-6B-50D	117	49.57	52.14
GloVe-6B-100D	117	52.99	51.28
GloVe-6B-200D	117	48.72	52.65
GloVe-6B-300D	117	50.43	56.89
GloVe-42B-300D	226	52.21	52.65
GloVe-840B-300D	515	49.90	56.89
FastText-Wiki	360	50.00	49.72
FastText-CC	572	46.85	49.72
SkipGram-2	112	51.79	48.21
SkipGram-5	112	52.68	51.79
SkipGram-Dep	109	53.21	48.62

Table 8: Performance (% accuracy) of embedding models on SC-NUM

Model	BC-NUM		
	#Tests	Rand	Emb
GloVe-6B-50D	117	50.43	**99.15**
GloVe-6B-100D	117	57.26	**100.0**
GloVe-6B-200D	117	42.74	**2.21**
GloVe-6B-300D	117	54.70	**87.57**
GloVe-42B-300D	226	53.98	2.21
GloVe-840B-300D	515	49.71	**87.57**
FastText-Wiki	360	56.67	98.89
FastText-CC	572	41.26	98.89
SkipGram-2	112	49.11	49.11
SkipGram-5	112	50.89	14.29
SkipGram-Dep	109	52.29	31.19

Table 9: Performance (% accuracy) of embedding models on BC-NUM