Sparser is better: one step closer to word embedding interpretability

Simon Guillot, Thibault Prouteau, Nicolas Dugué

Sparser is better: one step closer to word embedding interpretability

Simon Guillot[1,2] **Thibault Prouteau**[1] **Nicolas Dugué**[1]
Le Mans Université, LIUM[1]
INaLCO, ERTIM[2]
firstname.lastname@univ-lemans.fr

Abstract

Sparse word embeddings models (SPINE, SINr) are designed to embed words in interpretable dimensions. An interpretable dimension is such that a human can interpret the semantic (or syntactic) relations between words active for a dimension. These models are useful for critical downstream tasks in natural language processing (*e.g.* medical or legal NLP), and digital humanities applications. This work extends interpretability at the vector level with a more manageable number of activated dimensions following recommendations from psycholinguistics. Subsequently, one of the key criteria to an interpretable model is sparsity: in order to be interpretable, not every word should be represented by all the features of the model, especially if humans have to interpret these features and their relations. This raises one question: to which extent is sparsity sustainable with regard to performance? We thus introduce a sparsification procedure to evaluate its impact on two interpretable methods (SPINE and SINr) to tend towards sustainable vector interpretability. We also introduce stability as a new criterion to interpretability. Our stability evaluations show little albeit non-zero variation for SPINE and SINr embeddings. We then show that increasing sparsity does not necessarily interfere with performance. These results are encouraging and pave the way towards intrinsically interpretable word vectors.

1 Introduction

Word embeddings models (Mikolov et al., 2013; Pennington et al., 2014; Devlin et al., 2018) allowed tremendous evolution in natural language processing. However, they embed the lexicon in dense representation spaces with opaque dimensions. It is possible to obtain an understanding of these models via probing (Rogers et al., 2021) and embedding matrix analysis (Shin et al., 2018). However such methods are subject to criticism with regard to the interpretation that can actually be drawn from them (Hewitt and Liang, 2019; Ravichander et al., 2021; Elazar et al., 2021). This *a posteriori* approach to understanding models' decisions corresponds to the explainability paradigm in machine learning.

On the other hand, interpretability (Rudin, 2019) is defined for word embedding models as the possibility to find semantic (or syntactic) consistency in the dimensions of the embedding space (Murphy et al., 2012; Faruqui et al., 2015; Subramanian et al., 2018; Prouteau et al., 2022). Models such as SPINE (Subramanian et al., 2018) and SINr (Prouteau et al., 2021) meet this requirement: Table 1 illustrates the interpretability of the dimensions resulting from such methods. These inherently interpretable approaches to represent the lexicon are deemed preferable for high-stakes downstream use such as medical or legal NLP (Rudin, 2019). Interpretability also eases connection between word embeddings and linguistic models of the lexicon, since consistent semantic dimensions can be grasped as semantic features, which are used in a variety of theoretical models (Jackendoff, 1983; Pottier, 1963; Rastier, 2009).

As far as we know, only the interpretability of dimensions is considered in the literature and human evaluations such as the *Word Intrusion Detection* (Murphy et al., 2012) are targeted specifically towards this aspect. In this paper, we introduce **vector-level interpretability** and define it as the capacity for a speaker to make sense of the set of activated dimensions in a word vector. It is possible only if the set of dimensions to describe the word is limited. The size of this set is bounded by two different kinds of psychological experiments: semantic features production (Garrard et al., 2001; McRae et al., 2005) and features retention (Miller, 1956; Peterson and Peterson, 1959). This body of literature comes to an agreement at roughly ten fea-

tures. We consider in this paper that this number of features is a desirable horizon for vector-level interpretability. Following this objective and to further reduce the amount of information provided to the speaker, we also consider binary word vectors as in Faruqui et al. (2015). Moreover, this binary approach is consistent with componential analysis (Goodenough, 1956; Katz and Fodor, 1963).

Considering these criteria, and to tend towards more interpretability, our work offers the following contributions :

- Refine interpretability by introducing additional criteria: stability and increased sparseness for vector-level interpretability.
- Evaluate the effects of increased word vector sparseness and binarity on performance.
- Illustrate the effects of increasing vector sparseness on the embedding space.

To this end, we introduce Section 2 the criteria for interpretability and their different settings in the literature. Section 3 introduces the models considered for our experiments. In Section 4, we detail the experimental setup adopted to evaluate the impact of sparsity as well as binarity on performance and vector-level interpretability. In Section 5, we demonstrate that the trade-off between sparsity and interpretability is not as strong as one would think. Finally, Section 6 illustrates the impact of sparsity on word vectors and discusses its benefits.

2 Related work

Interpretability : criteria and models. The seminal article of (Murphy et al., 2012) paves the way towards psycholinguistically plausible distributional representations. The authors fix the following set of constraints on the representation space: sparseness, positivity and performance. Sparseness is justified by the difficulty to cover a vast vocabulary comprised of many different topics with a small set of features. Thus, a large number of dimensions is needed, but only some of those are activated for the description of each word. Positivity is motivated by the fact that storing null or negative features for each item of the lexicon is not cognitively efficient (Palmer, 1977; Lee and Seung, 1999). The performance criterion is needed since it is possible to produce interpretable representations of the lexicon (*e.g* raw co-occurrence matrices) with subpar performances on intrinsic or extrinsic evaluations. This sparse interpretable word model research is carried on with `SPOWV` (Faruqui et al., 2015), `SPINE` (Subramanian et al., 2018) and `SINr` (Prouteau et al., 2021). The first two models transform previously trained dense representations into sparse word embeddings while the latter builds a sparse embedding space from a word co-occurrence matrix. The word intrusion tests (Murphy et al., 2012; Senel et al., 2018; Subramanian et al., 2018; Prouteau et al., 2022) are designed to assess the internal consistency of dimensions in the embedding space. As introduced Section 1, we wish to allow interpretability at the vector level which might benefit from a smaller set of activated components in word vectors.

Stability. Pierrejean (2020) demonstrate the non-determinism of neural models' training which lead to variations in evaluation scores and word neighborhoods. On the front of explicability, new deterministic methods are emerging (Zafar and Khan, 2021). However, Rudin (2019) encourages to prioritise interpretable approaches over explicable approaches, motivating this work.

From these observations and as stated Section 1, we refine the criteria necessary to enable vector-level interpretability by redefining sparsity and adding stability.

Binary embeddings. Prototypicality theory (Rosch, 1975; Rosch et al., 1976) introduced the paradigm of weighted features in psychology and linguistics. However, feature-based analysis preempted this theoretical framework with componential analysis. This approach based on binary features was used by anthropological linguists (Goodenough, 1956), in structuralist work (Pottier, 1963) and in cognitively informed generativist frameworks (Katz and Fodor, 1963). Faruqui et al. (2015) construct binary vectors using sparse coding to sparsify dense word embeddings in more dimensions than the original space—called overcomplete vectors (`SPOWV`). The model is then binarized simply by setting each non-zero value to one. In computer science, another use to binary models is to reduce the memory footprint of word embeddings by replacing floats with bits and also the compute needed to exploit these representations. It is especially critical in low-resource embedded systems—*e.g* mobile phones. Tissier et al. (2019) and Navali et al. (2020) introduce autoencoder approaches to binarize

	Word2Vec	SPINE	SINr
insulin	scalar, tablespoon, vesicular, dystrophy antiserum, falsifiable, experimenter, internat PBS, NC, arginine, IFN	glutathione, pancreas, gastroduodenal, vitamin immunologically, hyperplasia, transgene, nociceptive insulin, sulphasalazine, interferon, cholangitis	hypertriglyceridaemia, mellitus, porcine, insulin aldosterone, aminotransferase, creatinine, glycated ulcerative, sulphasalazine, colitis, sera
mint	scalar, tablespoon, vesicular, dystrophy cube, geranium, Berowne, curiosities polyunsaturated, misfire, margarine, methile	spoonfuls, parsnips, kebabs, preheat onion, basil, yogurt, coriander dial, screams, vibration, spadefoot	tbsp, oregano, diced, dijon Gibson, gigged, charvel, Ibanez minted, minting, hoards, coinages
oxygen	scalar, tablespoon, vesicular, dystrophy herbicides, menstrual, deprave, angiotensin pou, tenascin, cytoplasm, platelet	glutathione, pancreas, gastroduodenal, vitamin lipid, crypt, tris, calcium monoxide, oxides, sulphuric, nitrogen	monoxide, dioxide, nitrous, oxides supplemental, hypoxaemic, electrocardiographic, gastroscopy diastolic, systolic, transfusion, transfusions

Table 1: Words with the highest values on the top three dimensions of "*insulin*", "*mint*" and "*oxygen*" in Word2Vec, SPINE and SINr sparsified to 100 active dimensions per vector according to the protocol described Section 4.

dense representations. Both of these models optimize for non-redundancy among dimensions and conservation of semantic information. Once vectors are binary, classical evaluation tasks such as word similarity or analogy may be redefined with bitwise operations (Sokal and Michener, 1958; Tissier et al., 2019). These models achieve competitive results to the baseline considering their small footprint.

3 Interpretable word embeddings

SPINE and SPOWV achieve close results on intrinsic and downstream evaluations but SPINE scores better in terms of interpretability (Subramanian et al., 2018), we thus do not consider SPOWV in the experiments that follow. Furthermore, SINr performances and interpretability are on a par with SPINE, we thus consider both SPINE and SINr as our reference interpretable models.

SPINE. SPINE, first introduced in Subramanian et al. (2018) derives sparse word embeddings from a previously trained dense model such as Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). Architecturally, it is an autoencoder whose hidden layer is of higher dimension than the dense input—*e.g* sparsifying from 300 dense dimensions to 1000 sparse dimensions. Three losses are implemented to enforce sparsity and interpretability. The *Reconstruction Loss* penalizes the poor reconstruction of the input representation from the output of the hidden layer, the *Average Sparsity Loss* and the *Partial Sparsity Loss* enforce sparse representations by limiting the number of active dimensions and skew vector values towards 0 or 1. SPINE has multiple hyperparameters: the minimum sparseness, the number of epochs and the vector output dimension.

SINr. Introduced in Prouteau et al. (2021), SINr is a graph-based approach to word embeddings. From a co-occurrence matrix extracted on a corpus, SINr builds a weighted word co-occurrence graph—words are represented by nodes and the number of co-occurrences by edges. A community detection algorithm, the *Louvain* method (Blondel et al., 2008), then uncovers dense groups of co-occurring words in the graph. SINr then leverages the distribution of each node over this partition to derive a sparse representation—not all words co-occur with words from each community. The representation is sparse by design, each component of the embedding space is related to a community. Community detection is an unsupervised process admitting a single parameter allowing to potentially control the number of communities detected.

4 Methodology

Models. Alongside the models presented Section 3, Word2Vec is used as a baseline. We use the *Skip-gram with negative sampling* (SGNS) architecture and the parameters described in Levy and Goldberg (2014). Word2Vec embeddings have 300 dimensions with a context window of 5 words. Since SPINE's number of dimensions is adjustable when SINr's is not—it is dependent on the number of communities detected—we base the number of dimensions of SPINE on SINr. Optimal performances for SINr are observed with the hyperparameter controlling the number of communities set to 50 resulting in 4460 dimensions for OANC (Nancy et al., 2011) and 8454 for BNC (Consortium, 2007) —the English corpora we use in our experiments is presented at the end of the next section. SPINE embeddings are trained from the Word2Vec model previously presented. The sparsity parameter of SPINE has little impact on the sparsity of the output. Subsequently, after several rounds of training, the model selected is that which achieves the best performances on the similarity task with a sparseness—95% after 1000 epochs—allowing further sparsification according to our experimental setup described hereinafter.

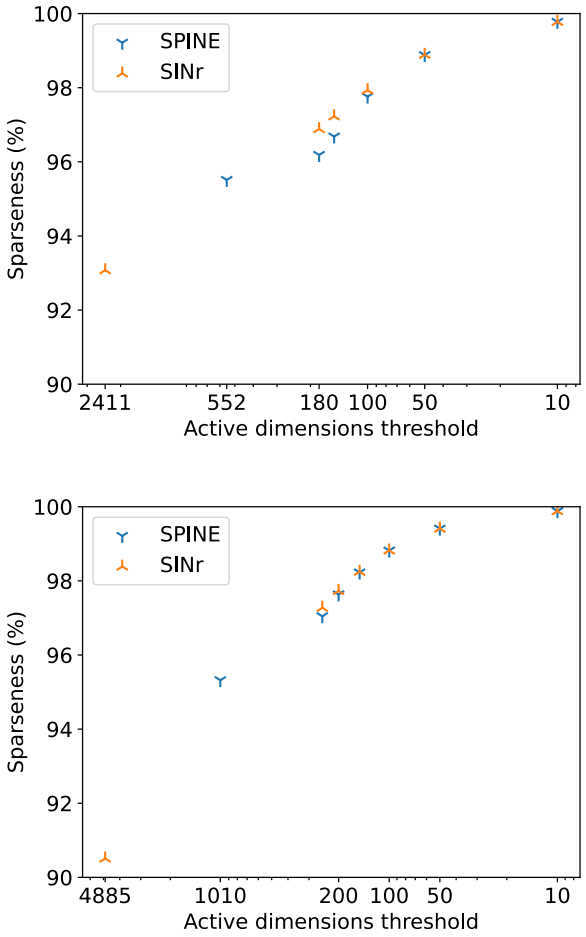


Figure 1: Sparseness of SPINE and SINr according to the maximum number of activated dimensions per vector on OANC (top) and BNC (bottom). First data point of each model is sparseness before sparsification.

Experimental framework. We introduce an experimental framework allowing to evaluate word embedding interpretability. We first consider a performance-sparsity compromise. Our hypothesis is that sparse vectors are both more interpretable and psycholinguistically plausible. To control sparseness, we introduce our sparsification method: from each embedding model, we keep only the k top strongest dimensions by value in each vector—k is in range $250 - 10$. Components not in the top k for the vector are set to zero. Figure 1 presents the sparseness of SPINE and SINr with regard to the active dimensions threshold. In the case of Word2Vec, we keep the top k dimensions out of the absolute values from the vectors.

In our second setup, we study the impact of switching to binary vectors. The binarization step is straightforward, we simply replace all non-zero values in each sparsified and unsparsified model by 1 as in Faruqui et al. (2015).

To evaluate the quality of the representations after sparsification and binarization, we use the word similarity evaluation—the correlation between the cosine similarity of words in our model and similarity rated by humans. Selected datasets model a variety of relations : MEN (Bruni et al., 2014), WS353 (Agirre et al., 2009), SCWS (Huang et al., 2012). To evaluate the stability of vectors produced by SPINE and SINr, our second criterion to interpretability, we learn 10 models and present the averaged results.

As similarity datasets are mostly available in English, we use the *British National Corpus* (BNC) (Consortium, 2007) and the text part of the *Open American National Corpus* (OANC) (Nancy et al.,

BNC	MEN		WS353		SCWS	
	$\overline{Spearman}$	σ	$\overline{Spearman}$	σ	$\overline{Spearman}$	σ
Word2Vec	$0,72$	$0,002$	$0,65$	$0,005$	$0,57$	$0,002$
SPINE	$0,65$	$0,006$	$0,57$	$0,01$	$0,60$	$0,004$
SINr	$0,66$	$0,0006$	$0,62$	$0,002$	$0,54$	$0,001$
OANC	**MEN**		**WS353**		**SCWS**	
	$\overline{Spearman}$	σ	$\overline{Spearman}$	σ	$\overline{Spearman}$	σ
Word2Vec	$0,43$	$0,002$	$0,50$	$0,005$	$0,46$	$0,003$
SPINE	$0,36$	$0,009$	$0,43$	$0,01$	$0,39$	$0,01$
SINr	$0,39$	$0,0008$	$0,44$	$0,002$	$0,39$	$0,002$

Table 2: Stability results for the word similarity evaluation on BNC (top), and OANC (bottom). Average Pearson correlation coefficient and standard deviation σ over 10 runs.

2011) to train our models. BNC contains 100 million tokens and OANC 11 million. Both corpus are composite in domain and genres. Those relatively small corpora, considering the standards in natural language processing, are chosen because documented corpora allow for finer interpretations of dimensions. Text preprocessing was performed using spaCy : tokenization with named-entity chunking, deletion of words shorter than three characters, of punctuation and of numerical characters. The minimum frequency for a type is set to 20. After preprocessing, OANC contains 20,814 types and roughly 4 million tokens, 58,687 types and 40 million tokens for BNC.

5 Results

Stability. The first property we consider with regards to interpretability is the stability of the models trained. This experiment is twofold, it allows to show whether methods are stable and also sets reference values for the similarity evaluation prior to sparsifying. Each model was run ten times on the same data with the same hyperparameters.

As reported in Table 2, the three models achieve scores in close ranges, with all models showing some degree if variability, their standard deviation being non-zero across ten runs. While `Word2Vec` and `SINr` seem more stable than `SPINE`, the overall observed variability on the small samples of the vocabulary present in the similarity datasets hinders reproducibility and is a flaw to the three model's interpretability.

Impact of sparsity on similarity. Results presented Figure 3 show the Pearson correlation scores on the similarity evaluation with regard to the number of components activated. The similarity scores are given with regard to the maximum number of top values kept in each vector according to our sparsification procedure. First, the three models achieve comparable results to those reported Table 2 up until 50 dimensions. More surprisingly, sparsifying `SINr` embeddings seems to improve performances. Sparsification may filter out noise from the base `SINr` model. Subsequently, there is not necessarily a trade-off between sparseness and efficiency. Furthermore, the fact that results remain satisfactory on our `Word2Vec` control model despite the sparsification is an unexpected behavior and is interesting with regard to how the semantic information is organized in its vectors.

In order to approach the sparsity objective of 10 dimensions presented Section 1, the experiment is also conducted at this level. Although we observe an overall drop in performance and especially for `Word2Vec`, a significant part of the semantic information is retained within these ten dimensions. Indeed, they allow to solve at least partially the similarity task. Even though the usefulness of this representation for downstream tasks can be discussed, it still allows to build interpretable word vectors despite the drop in performance. The low number of active dimensions render these models compatible with theoretical models leveraging semantic features, thus paving the way for new empirical opportunities.

Impact of binarization on similarity. Results presented Figure 3 follow the same display than sparsity results except that all models are binarized. Overall, we observe drops in performance across all models but to drastically varying extents. While `SPINE` and `SINr` lose some semantic information compared to the sparsified weighted models, they tend to retain performances of the same magnitude. This is especially true for models trained on BNC, considering that the models trained on OANC show bigger drops in word similarity performance. On the other hand, overall `Word2Vec` performances crumble with binarized vectors. This result is to be expected since `Word2Vec` is a dense model.

We can observe a common pattern across all models, where performance of binarized embeddings increases with sparsification until 100 or 50 activated dimensions. Binarizing while maintaining a lot of active dimensions flattens the hierarchy between components with strong values and others with low activations, thus otherwise very weak activations may gain weight in the vector as a result of binarization. In this case, the sparsification may remove noise from representations, by restoring a hierarchy between the few strong dimensions, activated with a 1 value, and the others set to 0. This denoising behavior resulting from sparsification seems common to binarized models, and weighted `SINr`.

6 Discussion

Our results show that there is not necessarily a trade-off between interpretability and performance. On the contrary, stability and increased sparseness of interpretable models can even improve results. At thresholds close to what is described in psycholinguistics, performances may remain accept-

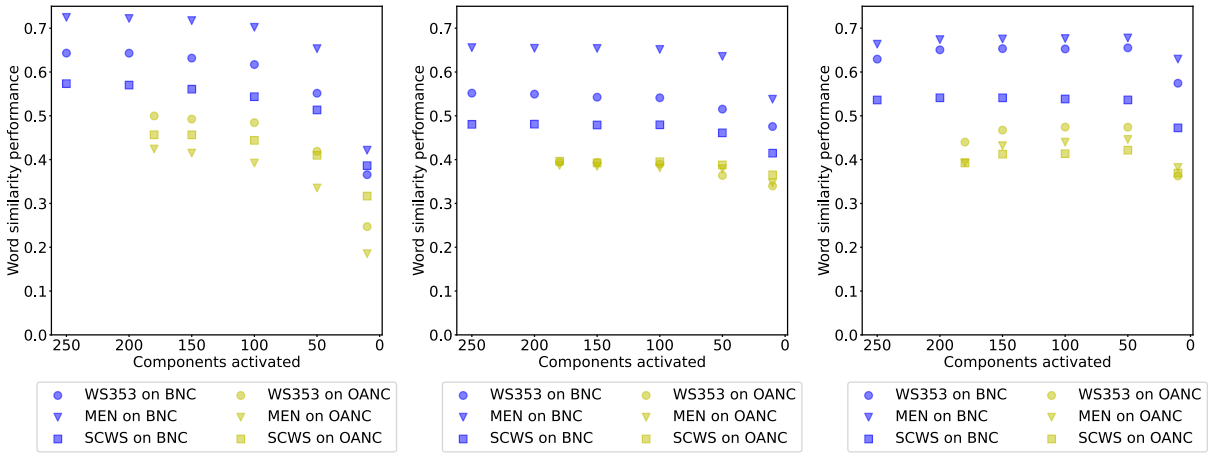


Figure 2: Word similarity performance (Pearson correlation) against maximum number of activated dimensions per vector for `Word2Vec` (left), `SPINE` (middle) and `SINr` (right). Performances on OANC are reported in yellow, and performances on BNC in blue.

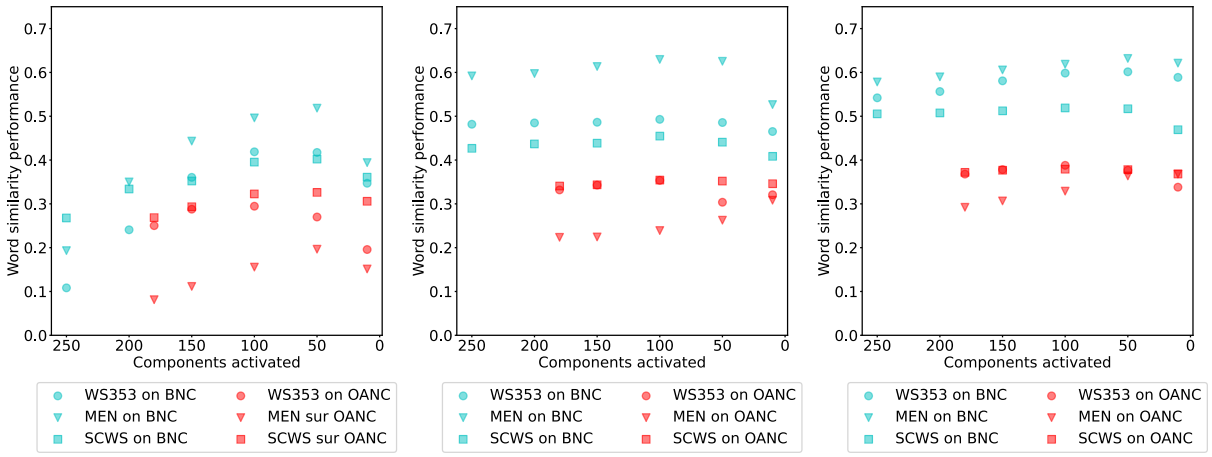
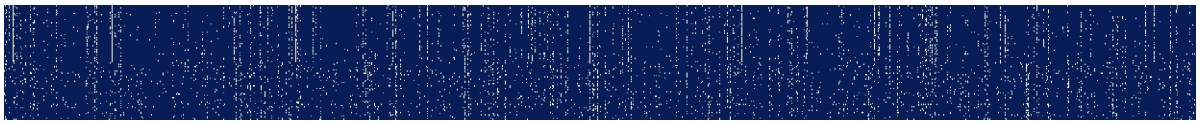


Figure 3: Word similarity performance (Pearson correlation) on binary models against maximum number of activated dimensions per vector for `Word2Vec` (left), `SPINE` (middle) and `SINr` (right). Performances on OANC are reported in magenta, and performances on BNC in cyan.

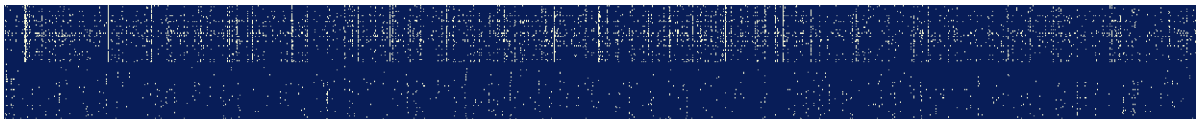
able considering the number of dimensions activated. Interpretability is hard to visualize without a set objective. In the discussion ensuing, we illustrate the interpretability of models through visualizations on selected items.

Interpretability of the dimension. Interpretability of the dimensions can be assessed after conducting a *word intrusion* evaluation with humans, both `SPINE` and `SINr`'s dimension interpretability have been previously evaluated without prior sparsification (Subramanian et al., 2018; Prouteau et al., 2022). The goal is to evaluate whether dimensions are interpretable—words with highest values on a dimension should be related. We present Table 1 top dimensions for three words as a glimpse into how interpretable dimensions of `SPINE` and `SINr` are in comparison with `Word2Vec`. As in previous evaluations, `Word2Vec` does not exhibit dimensions with related terms. If we consider the term "*insulin*", words on the first three strongest dimensions in the vectors are all related to medical conditions or biological functions. The word "*mint*" presents interesting dimensions, for `SPINE`, the first two dimensions are related to food and ingredients, the third one is less interpretable as one has trouble linking "*spadefoot*", a frog specie to "*dial*". `SINr` captures the polysemous nature of the word "*mint*" with top dimensions unrelated with one another. The first one is most probably related to mint as an aromatic, meanwhile, the sec-

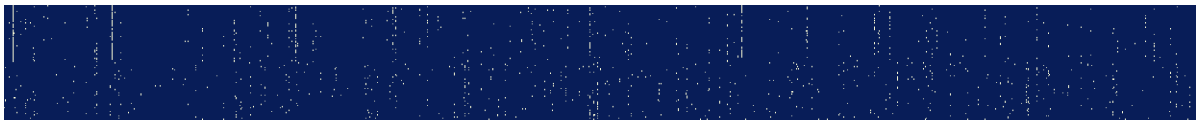
(a) SPINE



(b) SINr



(c) SPINE 100 active dimensions



(d) SINr 100 active dimensions

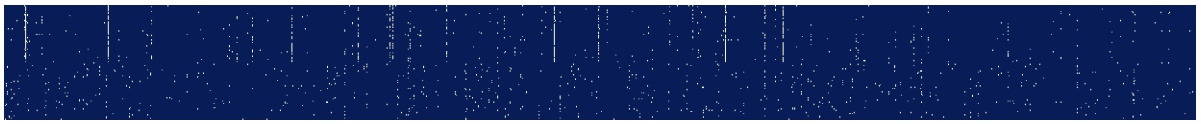


Figure 4: Shared dimensions across 50 most and least similar words to "*mint*" in SPINE and SINr. The models are trained on BNC both without sparsification, and with a threshold set to 100 dimensions on BNC. The top half of each figure represents the most similar words and the bottom half the least similar words.

ond one as the adjective describing guitars in mint condition, and the third one as a verb, to mint, in the sense of producing and managing currency. The same analysis can be drawn for the word "*oxygen*" where the use in the medical field is represented alongside chemical characteristics.

Interpretability of the vector. We evaluated increasingly sparsified word embeddings with the hypothesis that fewer features makes interpreting words vectors themselves easier. Our evaluations show that this gain in interpretability is not necessarily at the cost of model performances, the sparsification of representation can even increase performances up to a certain sparseness level. The following paragraphs aim to illustrate interpretability at the word vector level.

We present Figure 4 the distribution of values in the 50 most (top of each figure) and least similar (bottom of each figure) words to "*mint*" for SPINE (a; c) and SINr (b; d) on BNC. Lines appearing vertically across figures show shared dimensions between vectors in the embedding space. The first two figures (a; b) represent the shared features in the model prior to sparsification. SPINE presents vertical lines spanning most similar and least similar vectors, the embeddings seemingly share a large number of dimensions. SINr, on the other hand, exhibits a clear distinction between most and least similar words. One can clearly see shared dimensions among close neighbors of SINr for the word "*mint*". These first two distributions need to be compared with the distributions observed after sparsifying the vectors (c; d). At the 100 active dimensions sparsity setup, SINr seems to display more shared dimensions than SPINE for the word "*mint*". We assume that the performance gain in the similarity task observed for SINr Figure 2 is due to a process of noise reduction induced by the sparsification of the model.

The interesting results on similarity evaluation showed by sparsified interpretable models seems to indicate that the most important part of the semantic information is stored in the few strongest components of each vector. This observation allows us to analyze these models through the lens of our constrained version of interpretability di-

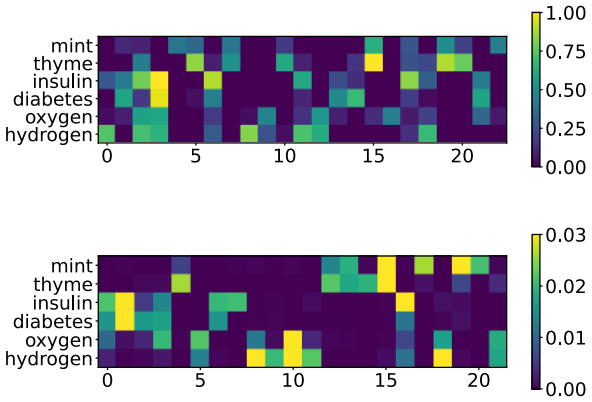


Figure 5: Word vectors on the set of top 5 shared dimensions for "*mint*", "*insulin*" and "*oxygen*" and their respective closest neighbors for SPINE (top) and SINr (bottom) on BNC.

rected towards the interpretation of word vectors. A speaker might want to interpret word embeddings by composing the meaning of a word with a limited subset of the features that describe it. In this case, the stability of the models becomes an increasingly important issue. Indeed, interpreting dimensions amounts to finding a consistency to a set of words that strongly interpret a dimension. However, interpreting a word vector relies both on this consistency and the strength of the activation of each dimension for a given vector. Thereby, even subtle variations in the representation across runs may induce different interpretations.

Binary representations. Our last experiment aims to quantify the benefit of weighted features over binary features. Considering results Figure 4, it appears that a significant part of the semantic information for sparse interpretable models is encoded in the mere activation of a dimension by a vector. Binarity is a means of reducing time and memory complexity of semantic models and is undoubtedly beneficial in embedded applications with low latency requirements or low resource hardware. We observe with Figure 5 that a SINr weighted model tends to have fewer and more strongly activated dimensions than a SPINE weighted model, which makes the former more alike binarized representations. This property facilitates the interpretation at the vector level: for example, dimensions 12 to 15 are strongly activated for "*mint*" and "*thyme*", and not at all for the other words, in the SINr representation. Recognizing the similarity of "*mint*" and "*thyme*", and their opposition to the other words, is easier when there is a clear gap between a strong activation and no activation of the dimension considered, like in a binarized vector.

Taking a step back, the comparison between weighted and binarized vectors performances allow us to pinpoint where the information is encoded. A significant part of the semantic information is stored in the activation of a few dimensions for each word vector, but the dimensions weights are needed to reach the most competitive performances. This assessment is coherent with the theoretical paradigm shift mentioned Section 2. Furthermore, it appears that, while binarizing embeddings represents a cost in performance, sparsifying them is not necessarily a trade-off. In some cases, it might even be beneficial.

7 Conclusion

Previously, the interpretability of embedding spaces focused mainly on dimension, this work redefined interpretability from the vector standpoint. We state that stability of the models and sparsity are necessary conditions to intepretability. Constraining on sparsity echoes psycholinguistic plausibility, it is essential to find semantic coherence within dimension of the embedding space but also to describe a word with a limited set of these dimensions. We hypothesize that vectors constrained following this protocol are interpretable by a speaker, since it becomes possible to manipulate this small number of dimensions in working memory.

Interpretable word embedding models achieve good results on the intrinsic word similarity evaluation task even with higher sparseness levels. SINr even benefits from being sparsified. Furthermore, we show through examples that dimensions remain interpretable even on sparsified vectors and that, indeed words that are close in the embedding space are represented by a common set of dimensions. Lastly, we show that real-valued vectors are a slight improvement upon binary representation.

These results allow to reconsider the interpretability performance for distributed representations. A following step would be to conceive an evaluation framework to measure vector-level interpretability, allowing us to investigate if and how speakers would make sense of interpretable word vectors. Such models also open up new perspectives in which theoretical models describing the lexicon benefit from semantic features of word embeddings. In the field of semantic drift detection,

it would also allow to easily characterize the drift by keeping track of the few dimensions at stake.

Acknowledgments

The work has been funded by the ANR project DIGING (ANR-21-CE23-0010).

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Vincent D. Blondel, Jean Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47.

BNC Consortium. 2007. British national corpus, XML edition. Oxford Text Archive.

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

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Association for Computational Linguistics*, 9:160–175.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. 2015. Sparse overcomplete word vector representations. *arXiv preprint arXiv:1506.02004*.

P. Garrard, M. A. Lambon Ralph, J. R. Hodges, and K. Patterson. 2001. Prototypicality, distinctiveness, and intercorrelation: Analyses of the semantic attributes of living and nonliving concepts. *Cognitive Neuropsychology*, 18(2):125–174.

Ward H. Goodenough. 1956. Componential analysis and the study of meaning. *Language*, 32(1):195.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Empirical Methods in Natural Language Processing*, page 2733–2743.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.

Ray Jackendoff. 1983. *Semantic and Cognition*. MIT Press.

Jerrold J. Katz and Jerry A. Fodor. 1963. The structure of a semantic theory. *Language*, 39(2):170–210.

Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 2, page 2177–2185.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris Mcnorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547.

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

George A. Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Conference on Computational Linguistics*, pages 1933–1950.

Ide Nancy, Reppen Randi, and Suderman Keith. 2011. The open anc (oanc). ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

Samarth Navali, Praneet Sherki, Ramesh Inturi, and Vanraj Vala. 2020. Word Embedding Binarization with Semantic Information Preservation. In *International Conference on Computational Linguistics*, pages 1256–1265.

Stephen E. Palmer. 1977. Hierarchical structure in perceptual representation. *Cognitive Psychology*, 9(4):441–474.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543.

Lloyd Peterson and Margaret Jean Peterson. 1959. Short-term retention of individual verbal items. *Journal of Experimental Psychology*, 58(3):193.

Bénédicte Pierrejean. 2020. *Qualitative Evaluation of Word Embeddings: Investigating the Instability in Neural-Based Models*. Ph.D. thesis, Université Toulouse 2 - Jean Jaurès.

Bernard Pottier. 1963. *Recherches sur l'analyse sémantique en linguistique et en traduction mécanique*. Publications linguistiques de la Faculté des lettres et sciences humaines de Nancy.

Thibault Prouteau, Victor Connes, Nicolas Dugué, Anthony Perez, Jean-Charles Lamirel, Nathalie Camelin, and Sylvain Meignier. 2021. SINr: Fast Computing of Sparse Interpretable Node Representations is not a Sin! In *Intelligent Data Analysis*, 12695, pages 325–337.

Thibault Prouteau, Nicolas Dugué, Nathalie Camelin, and Sylvain Meignier. 2022. Are embedding spaces interpretable? results of an intrusion detection evaluation on a large french corpus. In *Language Ressources and Evaluation Conference*.

François Rastier. 2009. Principes et conditions de la sémantique componentielle. In *Sémantique interprétative*, Formes sémiotiques, pages 17–37. Presses Universitaires de France.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *European Chapter of the Association for Computational Linguistics*, pages 3363–3377.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Eleanor Rosch. 1975. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104:192–233.

Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-Braem. 1976. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215.

Lutfi Kerem Senel, Ihsan Utlu, Veysel Yucesoy, Aykut Ko.c, and Tolga Cukur. 2018. Semantic structure and interpretability of word embeddings. *Transactions on Audio, Speech, and Language Processing*, 26(10):1769–1779.

Jamin Shin, Andrea Madotto, and Pascale Fung. 2018. Interpreting word embeddings with eigenvector analysis. *Advances in Neural Information Processing Systems*, 32.

Robert R. Sokal and Charles Duncan Michener. 1958. A statistical method for evaluating systematic relationships. *University of Kansas science bulletin*, 38:1409–1438.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. 2018. Spine: Sparse interpretable neural embeddings. In *AAAI conference on artificial intelligence*, volume 32.

Julien Tissier, Christophe Gravier, and Amaury Habrard. 2019. Near-lossless Binarization of Word Embeddings. *AAAI Conference on Artificial Intelligence*, 33(01):7104–7111.

Muhammad Rehman Zafar and Naimul Khan. 2021. Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, 3(3):525–541.