Filtering communities in word co-occurrence networks to foster the emergence of meaning

Anna Béranger, Nicolas Dugué, Simon Guillot, Thibault Prouteau

Filtering communities in word co-occurrence networks to foster the emergence of meaning

Anna Béranger, Nicolas Dugué, Simon Guillot, and Thibault Prouteau

Université du Mans, Laboratoire d'Informatique de l'Université du Mans (LIUM), avenue Olivier Messiaen, 72000 Le Mans, France

Abstract. With `SINr`, we introduced a way to design graph and word embeddings based on community detection. Contrary to deep learning approaches, this approach does not require much compute and was proven to be at the state-of-the-art for interpretability in the context of word embeddings. In this paper, we investigate how filtering communities detected on word co-occurrence networks can improve performances of the approach. Community detection algorithms tend to uncover communities whose size follows a power-law distribution. Naturally, the number of activations per dimensions in `SINr` follows a power-law: a few dimensions are activated by many words, and many dimensions are activated by a few words. By filtering this distribution, removing part of its head and tail, we show improvement on intrinsic evaluation of the embedding while dividing their dimensionality by five. In addition, we show that these results are stable through several runs, thus defining a subset of distinctive features to describe a given corpus.

Keywords: Word co-occurrence networks, community detection, word embedding, linguistics, interpretability.

1 Introduction

In the field of Natural Language Processing (NLP), one of the main challenges is to represent the meaning of words into vectors, these vectors then being used as input to classification systems in order to solve various tasks such as part-of-speech tagging, named entity recognition, machine translation, etc. Vectors that represent words are commonly designated as word embeddings: the meaning of words is embedded in a small latent space with dense vectors. Approaches to train such vectors are based on the distributional hypothesis. Harris defines this hypothesis, writing that "linguistic items with similar distributions have similar meanings". To train word embedding, one thus need to estimate these distributions using word co-occurrences from large corpora. The seminal approaches to train word embeddings are actually based on word co-occurrences matrix factorization [19, 20, 16]. Other popular approaches such as `Word2vec` [18] use neural networks to build lexical representations, thus approximating matrix factorization methods [15]. Finally, transformer-based approaches [17] and large language

models based on these architectures [14] have demonstrated impressive performances, being able to contextualize word representations according to words' occurrences.

`SINr` (Sparse Interpretable Node Representations), the approach we introduced in Prouteau et al. [21] is based on those recent progresses in NLP. However, it pursues an alternative path. Instead of focusing on performance, the method aims to train interpretable word vectors frugally. To do so, it leverages the distributional hypothesis in a rather direct manner, using word co-occurrence networks. Communities are then detected and considered as dimensions of the latent space. The word vector is finally extracted using the connectivity of its representing nodes to the communities extracted, in line with Harris' claim: we consider that words with similar distributions of links through communities have similar meanings. The model has proven its low compute requirement, it is indeed based on the Louvain algorithm [5]. Furthermore, it was shown that dimensions of this model are mostly interpretable, the model being on-par with `SPINE` [23], a competing state-of-the-art approach for interpretability [22]. Indeed, dimensions are not abstract as in conventional approaches: they are the communities uncovered, tangible groups of words [22]. Even though `SINr` does not focus strictly on performance, it is still an important goal. On the similarity task, the approach is on-par with `SPINE`, but slightly under-performs when compared to `Word2vec`.

In this paper, we show how to foster community filtering to significantly improve performances of `SINr`, allowing to catch up with `Word2vec` while preserving its interpretability and low compute properties, even lowering its memory footprint. We first describe `SINr` in more details Section 2. In Section 3, we analyze dimensions' activations and how they seem related to the distribution of community sizes. Using this analysis, we propose our community filtering method based on dimension activation to improve the model. Then, in Section 4, we detail the textual corpora used to train the word embedding approaches considered, and the similarity task that we use to evaluate model performances. We finally detail the results by showing that filtering communities by removing part of the head and tail of the dimensions activations distribution allows a significant improvement in results while reducing models' memory footprint.

2 `SINr`: interpretable word vectors based on communities

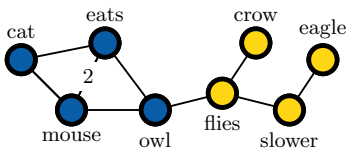
In this section, we first detail the `SINr` approach as we introduced it in [21]. As far as we know, it is the first graph-based approach to train word embeddings, but like other approaches, it is based on the distributional hypothesis, adjusting Harris' formula by claiming that words with similar distributions of links through communities have similar meanings.

We start with an undirected weighted network, the word co-occurrence network, where words are vertices, and edges between nodes represent co-occurrences of words inside a sentence, and at a distance at most of w. Textual corpora were preprocessed to keep only meaningful words (see Fig 1a). Weights are associated to edges and represent the number of co-occurrences. We then filter our word

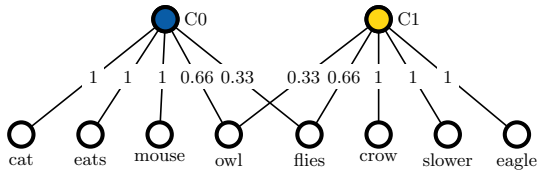
co-occurrence network by setting to 0 the edges weights whose co-occurrence is not significant, according to the PMI (the ratio of the probability of nodes u and v co-occurring together divided by their probability of occurrence). This is very similar to the construction of a co-occurrence matrix, showing that `SINr` is related to the matrix factorization approaches such as [19, 16]. In matrix factorization, the next step is to factorize the matrix to reduce dimensionality and get dense vectors. In `SINr`, we use community detection with the Louvain algorithm to group words together and get the interpretable dimensions of our latent space (see Fig 1b). We use the multiscale γ parameter [13] and set it to 60 in this paper: it allows uncovering small, thus consistent communities. We eventually calculate the distribution of the weighted degrees of each node through communities to get word embeddings, such as described in Figure 1c,1d. This distribution is computed according to the Node Recall introduced in [9] and defined as follows. Given a vertex u, a partition of the vertices $\mathcal{C} = \{C_0, \ldots, C_j\}$, and C_i the i_{th} community so that $1 \leq i \leq j$, the node recall of u considering the i^{th} community is : $\text{NR}_i(u) = \frac{d_{C_i}(u)}{d(u)}$ with $d_{C_i}(u) = \sum_{v \in C_i} W_{uv}$. To refine these vectors, we show in [10] that we can keep, for each vector, the 50 highest values, improving interpretability and lightening up the model.

Corpus before preprocessing: *My cat eats mice. Your owl eats mice. This owl flies. A crow flies slower than an eagle.*
After preprocessing: *cat eats mouse. owl eats mouse. owl flies. crow flies slower eagle.*

(a) Text corpus.



(b) A graph $G = (V, E, W)$ partitioned in two communities.



(c) Bipartite projection of G into graph $G\prime = (\top, \bot, E, W)$ along the communities. Weight on the edges is based on NR, the proportion of the weighted degree of each node related to the community.

	C0	*C1*
cat	1	0
eats	1	0
mouse	1	0
owl	0.66	0.33
flies	0.33	0.66
crow	0	1
eagle	0	1
slower	0	1

(d) Adjacency matrix of $G\prime$, each row is a `SINr` embedding.

Fig. 1: `SINr`: words are represented based on the communities they are linked to.

3 SINr-filtered: sampling communities using activations

First, community sizes tend to follow a power-law distribution [8]. We use the γ multiscale resolution parameter [13] to uncover small consistent communities, but there are still a few communities bigger than the rest as shown in Figure 2. In this figure, we applied SINr to two corpora, BNC and UkWac which are described in details in Section 4.

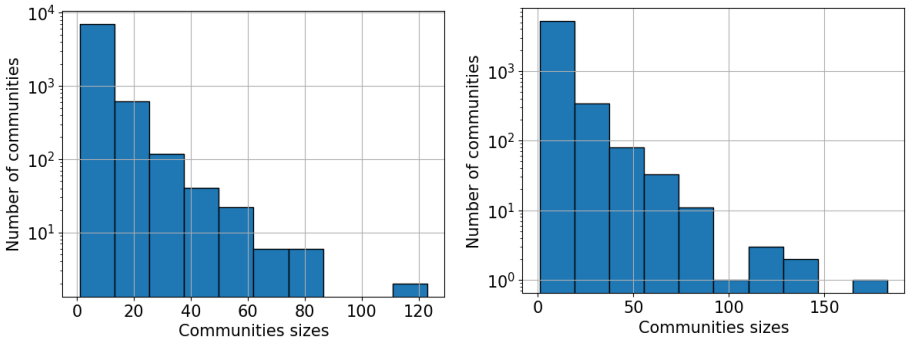


Fig. 2: Distribution of community sizes on BNC (left) and UkWac (right) corpora. The ordinate axis is in logarithmic scale.

Furthermore, linguists have studied the distribution of words occurrences and co-occurrences in corpora. It has been shown that it follows Zipf's law [3], which is consistent with a power-law [1], commonly observed in complex networks. It means that some words would co-occur far more than others with the rest of the vocabulary, and a lot of them may co-occur very few with the rest of the vocabulary. Because communities are made of words following this power-law, and because communities sizes also follow a power-law, some communities may be much more connected with the rest of the graph than others, and some of them may be quite isolated. Let us recall that dimensions of our SINr model correspond to these communities. Thus, if we say that a dimension is activated by a word if this dimension's value is greater than 0 for this word vector, we expect the activation of dimensions by the word vectors to follow a power-law.

As we can see in Figure 3, the number of activations (non-zero values) follows a distribution that looks like a power-law. This is critical to understand how the model works and how we can improve its performances while reducing its memory footprint.

As stated by Subramanian et al. [23], having dimensions activated by a large part of the vocabulary is not compatible with interpretability, which is a focus of our work: a dimension is interpretable if the set of words that activate it is consistent. Such heavily activated dimensions may be based on communities gathering very frequent words that appear in very different contexts, thus being

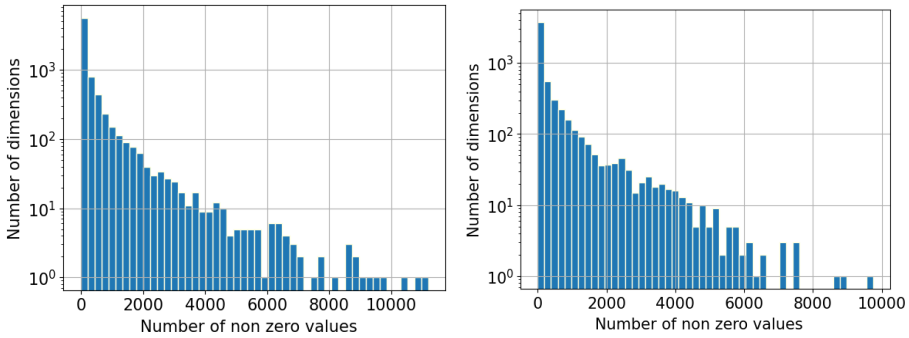


Fig. 3: Distribution of the number of activations per dimension on `BNC` (left) and `UkWac` (right) corpora. The ordinate axis is in logarithmic scale.

not consistent topically. We thus propose to remove these dimensions from the model, and we will see in Section 4 that it improves model performances.

We can also see that many dimensions are actually activated by only a few words. While these dimensions may be useful for very specific topics, they may not be useful for most of the vocabulary. They may also be noisy dimensions that penalize performances. We also propose to remove these dimensions from the models, and we will demonstrate in Section 4 that it notably reduces the memory footprint of the model (dividing the number of dimensions by 5) while preserving performances.

4 Experiments and results

4.1 Experimental setup

Task and baseline. We consider an intrinsic task to evaluate the performance of embeddings: the similarity task. In this task, we consider five datasets to evaluate our models: `WS353` [2], `MEN` [6], `SCWS` [12] and `SimLex` [11] split in `SimLex999` and `SimLex665`. Each dataset is made up of pairs of words associated with a similarity score that is a mean of ratings given by humans. The first three datasets comprise pairs of words, both representing word *similarity* (approximately synonymy, or at least substitutability, like "cat" and "feline") and word *relatedness* (much broader, encompasses pairs like "cup" and "coffee"). However, datasets differ regarding the parts of speech they include: `WS353` only includes nouns, while `MEN` and `SCWS` include nouns, verbs and adjectives. Lastly, `SCWS` is designed to evaluate contextual representations, scores representing word similarity in different contexts: "bank" can be scored high with both "river" and "money" depending on the phrastic context presented to annotators. `SimLex`, on the other hand, is a dataset specialized on word *similarity*, which is more restrictive regarding the substitutability criterion, and allows for hyperonym-hyponym pairs

to be closer than co-hyponyms. For example, in this dataset "father" and "parent" are rated at 7.07 on a scale of 10 regarding similarity, while "dad" and "mother" are at 3.55. Furthermore, `SimLex` does not rely on frequency information from a reference corpus to select its word pairs, it thus includes rarer words than `WS353`, `MEN` or `SCWS`. This dataset is of particular interest due to its difficulty, and its split with regard to parts of speech : `SimLex999` being the whole dataset with noun-noun, adjective-adjective and verb-verb pairs, and `SimLex665` being the noun subset. This split allows us to determine the validity of our modeling on different word categories, which probably follow different distributions of contexts.

Similarities in embedding spaces using cosine similarity are supposed to be correlated with human similarities, as shown in Figure 4. Correlation is computed with Spearman's definition: the closer to 1, the better. In order to assess the performances of our model, we also consider `Word2vec`, one of the most popular approaches to train word embeddings. We do not consider more recent state-of-the-art approaches that allow to get better performances because our approach focuses on interpretability and low compute. Eventually, it may be used in more complex architectures, such as transformers,

w_1	w_2	human rating		cosine_sim(w_1, w_2)
tiger	cat	7.35	Spearman Correlation ×	0.73
plane	car	6.31		0.65
drink	mother	2.85		0.20
forest	graveyard	1.85		0.12

Fig. 4: Example of word similarity rating from the `MEN` dataset and cosine similarity between vectors.

Corpora. We perform our evaluation on two text corpora :

- the written part of `BNC` [7], a collection of journalistic, academic and fictional texts totaling 100 million tokens ;
- `UkWac` [4], a cleaned crawled corpus from the .uk internet domain with over 2 billion tokens.

Both of these corpora are classic for the similarity tasks. Their size, their content, and the way they are constituted is very different. By reproducing the experiments on such different corpora, we aim to show that the results may be generalized to any collection.

Preprocessing and models parameters. The text corpora are preprocessed with `spaCy` to improve the quality of cooccurrence information and reduce the vocabulary to be covered by the models. The text is tokenized and lemmatized, named entities are chunked, words shorter than three characters, punctuation

and numerical characters are deleted. The minimum frequency to represent a type is set at 20 for BNC and 50 for UkWac. All models use a cooccurrence window of 5 words to the left and to the right of a target within sentence boundaries. Furthermore, as stated in Section 2, the multiscale resolution parameter γ is set at 60 for Louvain's community detection in SINr.

SINr-filtered approach. We first compute our SINr approach as described in Section 2. Then, by considering the distribution of activations per dimension, we explore the removal of the few very activated dimensions, and of the dimensions forming the long tail of this distribution. We explore the similarity performances regarding these removals and, in particular, the threshold used for these removals. The choice of thresholds is guided by performances on the word similarity evaluation task (see Figures 5, 6).

4.2 Results

Baseline. In order to assess the performance of SINr-filtered, we also provide results for Word2vec, a reference approach which is not interpretable, and SINr, the original approach without filters on the communities. Results are averaged over 10 runs for the whole section. Table 1 show that SINr-filtered is always better than the original version, and that it catches up with Word2vec, even outperforming this baseline on MEN and WS353.

	MEN		WS353		SCWS		SimLex999		SimLex665	
	BNC	UkWac	BNC	UkWac	BNC	UkWac	BNC	UkWac	BNC	UkWac
W2V	**.73**	**.75**	.64	.66	**.61**	**.64**	**.28**	**.34**	**.34**	**.37**
SINr	.67	.70	.63	.68	.56	.56	.20	.23	.28	.30
SINr-filtered	.72	**.75**	**.65**	**.70**	.58	.59	.25	.25	.30	.33

Table 1: Summary of the results of competing models and of SINr and its filtered version, SINr-filtered, introduced in this paper.

We then show the effects of filtering using the distribution of activations on the SINr performance regarding the similarity evaluation.

Filtering the distribution's head. As one can see in Figure 5, model performances varies a lot with regard to the filter threshold. On the right, when the threshold is set to 12 000, there is actually no filtering. Filtering more and more, moving the threshold to the left until the best threshold 4000, allows to gradually increase performances of models for both UkWac and BNC. Between 4000 and 2000, results are rather plateauing. After 2000, significant information is removed, leading to a decrease in performance. Using the 4000 threshold allows catching up with Word2vec's performances, our reference. Indeed, the 4000 filter allows a gain of 5 points in performance for the MEN dataset (from 0.67 to 0.72 for BNC and from 0.70

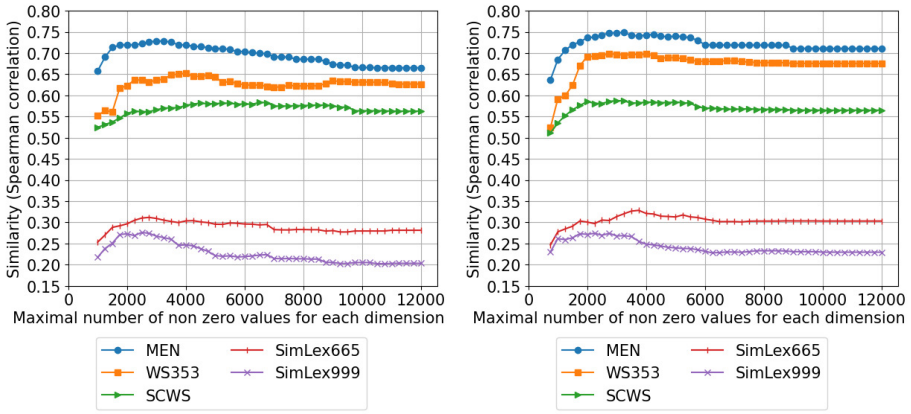


Fig. 5: Similarity on `BNC` (left) and `UkWac` (right) corpora. Dimensions **often** activated are removed according to the threshold in abscissa.

to 0.75 for `UkWac`), and a slight gain of 2 points for the `WS353` dataset (from 0.63 to 0.65 for `BNC`, from 0.68 to 0.70 on `UkWac`) and the `SCWS` dataset (from 0.56 to 0.58 for `BNC`, from 0.56 to 0.59 for `UkWac`). Such gains are statistically significant, and they are particularly interesting because they result from a simplification of the model, even if only 95 (resp. 90) dimensions are removed in average on the `BNC` (resp. `UkWac`) model using this filter. The `SimLex` dataset is much harder than the three others, for `SINr` but also for the reference model `Word2vec`. However, as one can see, filtering allows significant gain for `SimLex` also, especially for `SimLex999` on `BNC` (from 0.20 to 0.25), and the results are better between 2000 and 4000 thresholds like for the other datasets.

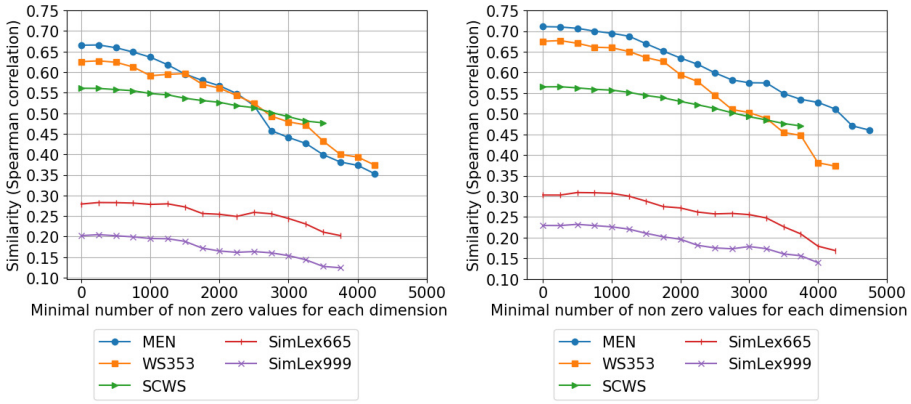


Fig. 6: Similarity on `BNC` (left) and `UkWac` (right) corpora. Dimensions **scarcely** activated are removed according to the threshold in abscissa.

Filtering the distribution's long tail. The effect of filtering the long tail of the distribution of activations is quite different, as one can see in Figure 6. At left, no filter is applied, and increasing the filter does not lead to any gain in performances. Still, it is interesting to notice that filtering dimensions with less than 500 activations does not lead to any significant loss in information, on the five similarity datasets used for evaluation. Indeed, it actually divides by 5 the number of dimensions of the model, reducing its number of dimensions from roughly 6600 to 1200 on average for `BNC`, and from 5700 to 1100 for `UkWac`, thus allowing to drastically improve its memory footprint!

Is filtering dimensions the same as filtering communities? As one can see in Figure 7, filters (more than 500 and less than 4000) applied using the number of dimensions have an expected effect on community distributions. Here, we assume that when a dimension is removed from the model, its community is also removed. We can see that removing dimensions with more than 4000 activations mostly removes big communities. This is especially the case for `UkWac` where the larger communities, those accounting for more than 150 words, are removed. For `BNC`, removed communities are not the largest, their size ranges from 60 to 80 words mostly. Similarly, removing dimensions with less than 500 activations tend to remove small communities. Still, most of the smallest communities are kept in the model while some larger are removed. These observations show that, even if the number of activations of a dimension and the size of its community are related, using the distribution of activations is different from using the distribution of the community sizes, showing the relevance of our approach.

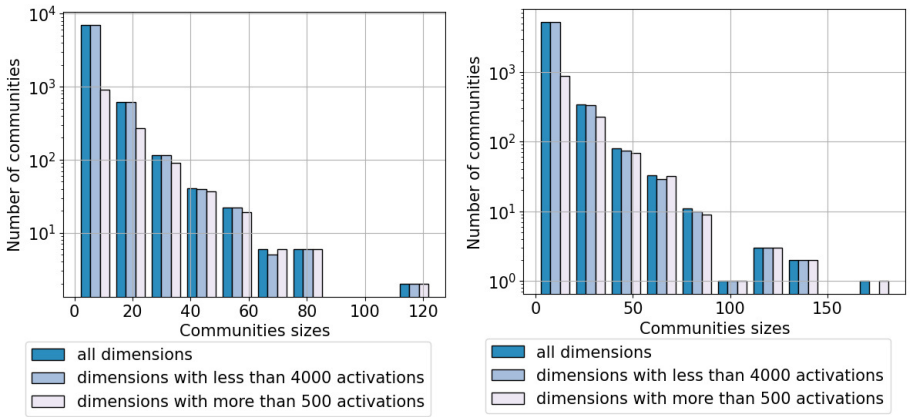


Fig. 7: Filtering effects on the community sizes distribution for `BNC` (left) and `UkWac` (right).

Singling out the subset of distinctive contexts. One may wonder if from one run of `SINr` to another, by filtering dimensions with more than 500 and less than 4000 activations, the vocabulary that forms communities that are kept is the same. It is surprising to notice that it is mostly the case: roughly 80% of the vocabulary kept is actually the same over ten runs when considering `BNC` and `UkWac` separately. However, this set is not the same from one corpus to another, only 35% of the vocabulary kept is actually common to `BNC` and `UkWac`. It seems to mean that these respective subsets of the vocabulary are essential to describe the meaning of words in these respective corpora. Those results, combined to the similarity improvements, point towards the notion that our filtering approach discriminates a subset of the dimensions that is the best fit to describe a given corpus. However, the evaluation results solely give insight on the subset of the lexicon covered by the similarity datasets, a subset that is heavily biased toward nouns, and especially frequent concrete nouns.

5 Conclusion

`SINr` is a graph-based approach to train word embedding which requires low compute and whose results are interpretable. In this paper, we show that we can significantly improve model performances and reduce its memory footprint by filtering its dimensions. Indeed, filtering the most activated dimensions allows gaining a few points on the similarity task for each dataset considered, showing that these dimensions are actually the bearer of noise into the model. This gain allows `SINr` performing on-par with `Word2vec`. Furthermore, filtering-out dimensions that are the least activated allows dividing the number of dimensions by 5 while preserving performances. We show that these filters relying on activations of the dimensions are somehow correlated with community sizes, but not completely, showing their relevance. Finally, we demonstrate that the vocabulary of communities that correspond to dimensions that are not filtered remains the same from one run of `SINr` to the other. We plan to experiment on other corpora but also on downstream tasks to confirm the ability of these results to generalize in a variety of contexts. Furthermore, it would be particularly interesting to test the ability of these filtered embeddings to model the meaning of very specialized vocabulary, to evaluate if removing dimensions affects the representation of these words.

Acknowledgements

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

References

1. Lada Adamic. Unzipping zipf's law. *Nature*, 474(7350):164–165, 2011.

2. Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *NAACL*, pages 19–27, June 2009.
3. Marco Baroni. 39 distributions in text. In *Corpus linguistics: An international handbook*, volume 2, pages 803–822. Mouton de Gruyter, 2005.
4. Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, 2009.
5. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.: Theory Exp.*, page P10008, 2008.
6. Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47, jan 2014.
7. BNC Consortium. British national corpus, XML edition, 2007. Oxford Text Archive.
8. Vinh Loc Dao, Cécile Bothorel, and Philippe Lenca. Community structure: A comparative evaluation of community detection methods. *Network Science*, 8(1):1–41, 2020.
9. Nicolas Dugué, Jean-Charles Lamirel, and Anthony Perez. Bringing a feature selection metric from machine learning to complex networks. In *Complex Networks and Their Applications*, pages 107–118, 2019.
10. Simon Guillot, Thibault Prouteau, and Nicolas Dugué. Sparser is better: one step closer to word embedding interpretability. In *International Workshop on Computational Semantics*, 2023.
11. Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *CoRR*, abs/1408.3456, 2014.
12. Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. Improving word representations via global context and multiple word prototypes. In *ACL*, pages 873–882, 2012.
13. Renaud Lambiotte. Multi-scale modularity and dynamics in complex networks. In *Dynamics On and Of Complex Networks, Volume 2*, pages 125–141. Springer New York, 2013.
14. Julien Launay, Elena Tommasone, Baptiste Pannier, François Boniface, Amélie Chatelain, Alessandro Cappelli, Iacopo Poli, and Djamé Seddah. Pagnol: An extra-large french generative model. *arXiv preprint arXiv:2110.08554*, 2021.
15. Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
16. Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *ACL*, 3:211–225, 2015.
17. Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. *arXiv:1911.03894*, 2019.
18. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.
19. Brian Murphy, Partha Talukdar, and Tom Mitchell. Learning effective and interpretable semantic models using non-negative sparse embedding. In *COLING*, pages 1933–1950, 2012.
20. Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

21. Thibault Prouteau, Victor Connes, Nicolas Dugué, Anthony Perez, Jean-Charles Lamirel, Nathalie Camelin, and Sylvain Meignier. Sinr: fast computing of sparse interpretable node representations is not a sin! In *IDA*, pages 325–337, 2021.
22. Thibault Prouteau, Nicolas Dugué, Nathalie Camelin, and Sylvain Meignier. Are embedding spaces interpretable? results of an intrusion detection evaluation on a large french corpus. In *LREC*, 2022.
23. Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. In *AAAI*, 2018.