# Neural Vector Conceptualization for Word Vector Space Interpretation

**Robert Schwarzenberg*, Lisa Raithel*, David Harbecke**
German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
{firstname.lastname}@dfki.de

## Abstract

Distributed word vector spaces are considered hard to interpret which hinders the understanding of natural language processing (NLP) models. In this work, we introduce a new method to interpret arbitrary samples from a word vector space. To this end, we train a neural model to conceptualize word vectors, which means that it activates higher order concepts it recognizes in a given vector. Contrary to prior approaches, our model operates in the original vector space and is capable of learning non-linear relations between word vectors and concepts. Furthermore, we show that it produces considerably less entropic concept activation profiles than the popular cosine similarity.

## 1 Introduction

In the vast majority of state-of-the-art NLP models, as for instance in translation models (Bojar et al., 2018) or text classifiers (Howard and Ruder, 2018), language is represented in distributed vector spaces. Using distributed representations comes at the price of low interpretability as they are generally considered uninterpretable, without further means (Levy and Goldberg, 2014; Montavon et al., 2018). In this work, we address this lack of interpretability with *neural vector conceptualization* (NVC), a neural mapping from a word vector space to a concept space (e.g. "chair" should activate the concept "furniture").

Using concepts to interpret distributed vector representations of language is inspired by the finding that "humans understand languages through multi-step cognitive processes which involves building rich models of the world and making multi-level generalizations from the input text" (Shalaby and Zadrozny, 2019). We are not the first, however, to utilize concepts for this purpose. Koç et al. (2018), for instance, modify the objective function of GloVe (Pennington et al., 2014) to align semantic concepts with word vector dimensions to create an interpretable space. Their method does not, however, offer an interpretation of vectors in the original space.

Senel et al. (2018), in contrast, do offer an interpretation of the original space: They propose a mapping of word vector dimensions to concepts. This mapping, however, is linear and consequently, their method is incapable of modeling non-linear relations.

Our method offers an interpretation of the original space and is capable of modeling non-linear relations between the word and the concept space. Furthermore, arguably, we interpret vectors similar to how a neural NLP model would, because a neural NLP model lies at the heart of our method. In addition, by design, our model is able to conceptualize random continuous samples, drawn from the word vector space.

This is particularly important as word vectors are sparse in their vector space and vectors without a word representative do not have intrinsic meaning. This hinders adapting methods from vision, such as activation maximization (Simonyan et al., 2013) or generative adversarial networks (Goodfellow et al., 2014), as in NLP these methods potentially produce vectors without word representations.

For introspection, one could map any vector onto its nearest neighbor with a word representative. However, nearest neighbor search does not necessarily find the closest semantic representative in the vector space (Schnabel et al., 2015). Moreover, we show that concept activation profiles produced with nearest neighbor search tend to be considerably more entropic than the activation profiles our method returns.

* Shared first authorship.

## 2 Method

For NVC, we propose to train a neural model to map word vectors onto associated concepts. More formally, the model should learn a meaningful mapping

$$f : \mathbb{R}^d \rightarrow \mathbb{R}^{|C|} \tag{1}$$

where $d$ denotes the number of word vector dimensions and $C$ is a set of concepts. The training objective should be a multi-label classification to account for instances that belong to more than one concept (e.g. “chair” should also activate “seat”).

For the training, we need to make two basic choices:

1. We need a ground truth concept knowledge base that provides the concepts a training instance should activate and
2. we need to choose a model architecture appropriate for the task.

In the following, we motivate our choices.

### 2.1 Ground Truth Concept Knowledgebase

As a ground truth concept knowledge base we chose the Microsoft Concept Graph (MCG), which is built on top of Probase, for the following reasons:

1. Wu et al. (2012) convincingly argue that with Probase they built a universal taxonomy that is more comprehensive than other existing candidates, such as for example, *Freebase* (Bollacker et al., 2008).

2. Furthermore, Probase is huge. The core taxonomy contains about 5.38 million concepts, 12.5 million unique instances, and 85.1 million *isA* relations. This allows our model to illuminate the word vector space from many angles.

3. Instance-concept relations are probabilistic in the MCG: For (instance, concept) tuples a $rep$ score can be retrieved. The $rep$ score describes the “representativeness” of an instance for a concept, and vice versa. According to the MCG, for example, the instance “chair” is a few thousand times more representative for the concept “furniture” than is the instance “car.” During training, we exploit the $rep$ scores to retrieve representative target concepts for a training instance.

The scores are based on the notion of Basic Level Concepts (BLC) which were first introduced by Rosch et al. (1976), as part of Prototype Theory. A basic level concept is a concept on which all people of the same culture consciously or unconsciously agree. For instance, according to Prototype Theory, most humans would categorize a “wood frog” simply as a “frog.” “Wood frog” is a representative instance of the concept “frog.”

Aiming to provide an approach to the computation of the BLC of an instance $i$ in the MCG, Wang et al. (2015) combine pointwise mutual information (PMI) with co-occurrence counts of concept $c$ and instance $i$. The authors compute the “representativeness” of an instance $i$ for a concept $c$ as

$$rep(i, c) = P(c|i) \cdot P(i|c). \tag{2}$$

By taking the logarithm of the $rep$ score, we can isolate the involvement of PMI:

$$log\ rep(i, c) - log\ P(i, c) = PMI(i, c). \tag{3}$$

In doing so, the authors boost concepts in the middle of the taxonomy (the basic level concepts) while reducing extreme values leading to super- or subordinate concepts. To find the BLC of a single instance, Wang et al. (2015) maximize over the $rep$ value of all concepts associated with $i$.

To train our model, for a training instance $i$, we collect all concepts for which $rep(i, c)$[1] is above a certain threshold and use them as the target labels for $i$. We discard concepts that have very few instances above a threshold $rep$ value in the graph.

### 2.2 Model

During training, the model repeatedly receives a word vector instance as input and a multi-hot vector retrieved from the MCG as the target concept vector. Thus, it must identify concepts encoded in the word vector.

We do not see any sequentiality or recurrence in this task which is why we discarded recurrent and Transformer candidate models. Concerning convolutional networks, we disregard small receptive fields because dimensional adjacency is semantically irrelevant in word vectors. However, any convolutional network with a receptive field over the whole input vector is equivalent to a fully-connected (FC) feed-forward network. Thus, we ultimately trained an FC feed-forward network to conceptualize vectors.

[1] We computed the rep values ourselves as we only acquired a count-based version of the graph.

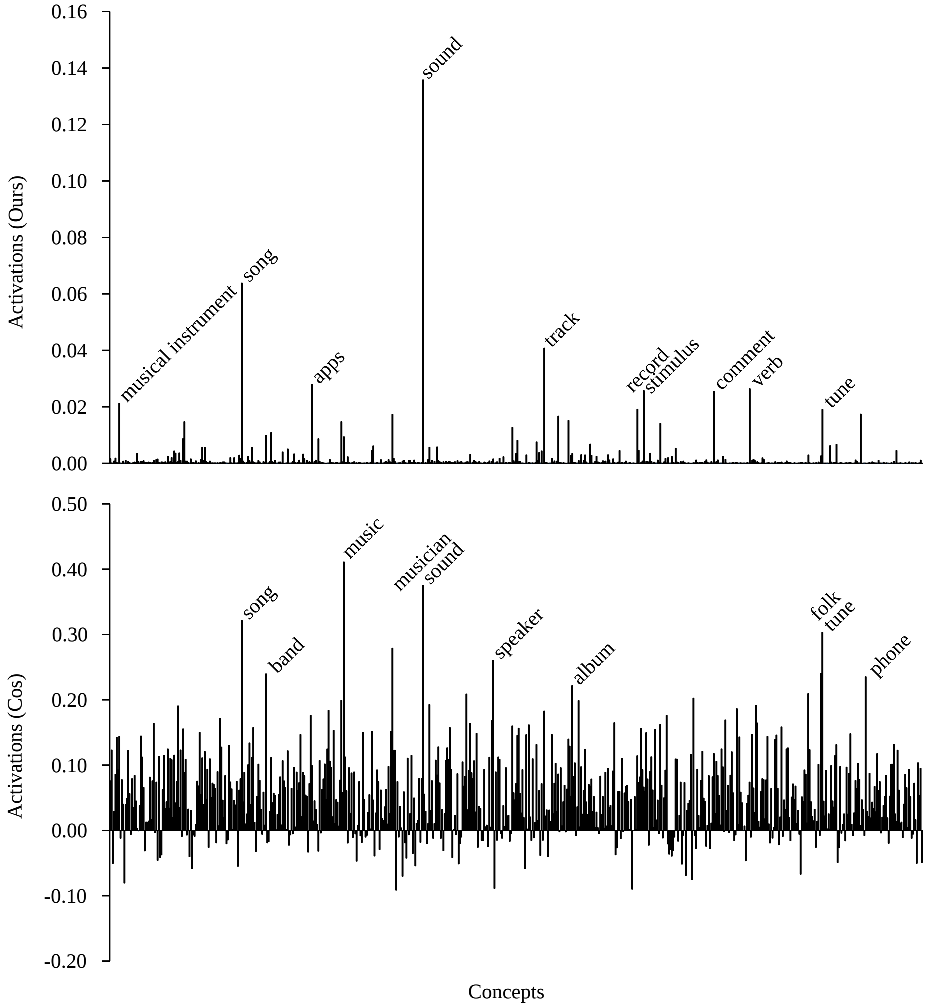

Figure 1: Vector interpretations of the word vector of "listening" with 637 concepts. Top: Neural vector conceptualization (our method, 10 highest activations labelled). Bottom: Cosine similarity (baseline, 10 highest activations labelled). Both activation profiles are unnormalized.

## 3 Experiments

For a proof of concept, we chose the word2vec embedding (Mikolov et al., 2013) as the word vector space to interpret. Recently, contextualized representations, like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), received increased attention. Nevertheless, well-established global representations, such as word2vec remain highly relevant: ELMo still benefits from using global embeddings as additional input and BERT trains its own global token embedding space.

The word2vec model and the MCG are based on different corpora. As a consequence of using data from two different sources, we sometimes needed to modify MCG instances to match the word2vec vocabulary.

We filtered the MCG for concepts that have at least 100 instances with a *rep* value of at least $-10$. This leaves 637 concepts with an average of 184 instances per concept and gives a class imbalance of 524 negative samples for every positive sample.

With the obtained data, we trained a three-layer FC network to map word vectors onto their concepts in the MCG. The model returns independent sigmoid activations for each concept. We trained with categorical cross entropy and applied weights regularization with a factor of $10^{-7}$. For all experiments, we optimized parameters with the ADAM optimizer (Kingma and Ba, 2015).[2]

To estimate task complexity, Table 1 lists the precision, recall and $F_1$ scores that our model achieved on a fixed, randomly sampled test set that

[2] Our experiments are open source and can be replicated out of the box: https://github.com/dfki-nlp/nvc.

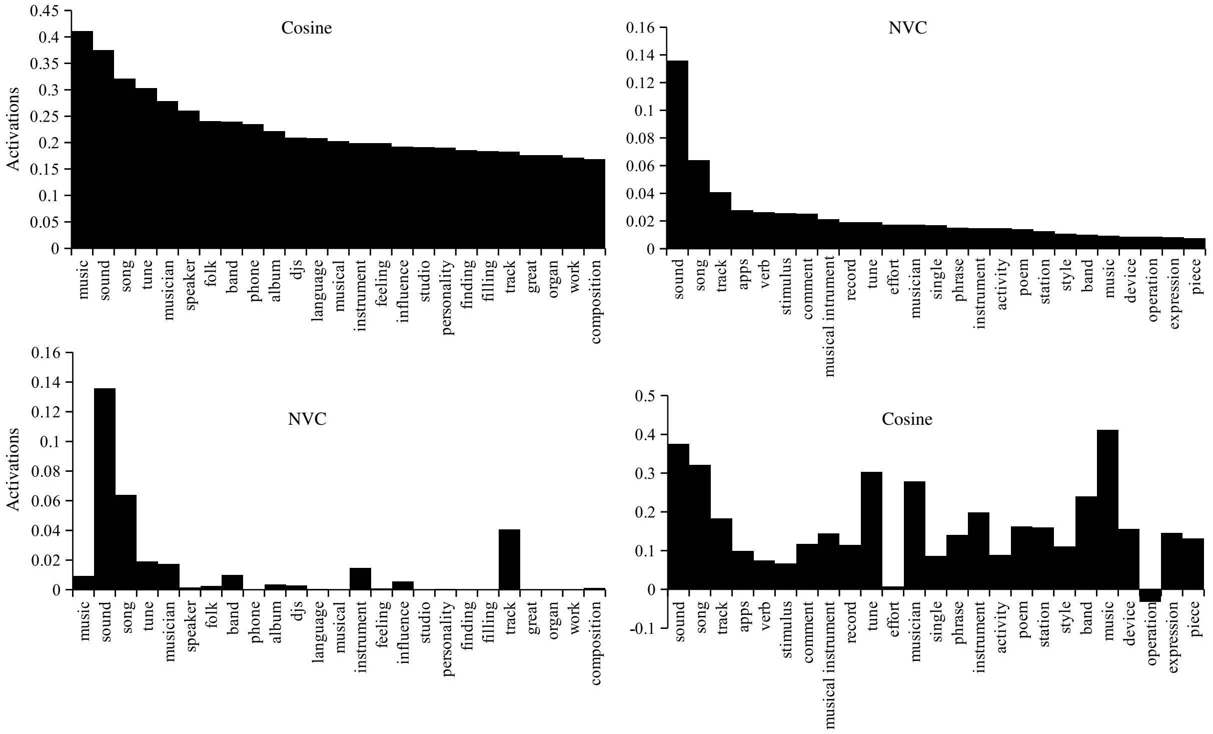

Figure 2: Concept activations for the instance "listening." Upper left: Top 25 concepts according to cosine similarity. Bottom left: NVC activations of the same cosine top 25 concepts. Upper right: Top 25 concepts according to NVC. Bottom right: Cosine activations of the same NVC top 25 concepts.

contained 10 % of the data. The table contains the weighted average scores accomplished for all concepts as well as the scores the model achieved for selected individual concepts, grouped semantically.

Fig. 1 juxtaposes the NVC and the baseline activation profile of the word vector of "listening", which was not encountered during training. Several other NVCs can be found in the appendix (see Figs. 3, 4 and 5) as well as selected concept activations of continuous samples (see Fig. 6).

While Fig. 1 shows a global perspective of the activation profiles, Fig. 2 zooms in on the top 25 concepts, activated by the baseline method (first column) and our method (second column).

| | P | R | F | S |
|---|---|---|---|---|
| all concepts | 0.43 | 0.16 | 0.22 | 9766 |
| province | 0.81 | 0.81 | 0.81 | 36 |
| district | 0.79 | 0.62 | 0.69 | 78 |
| island | 0.96 | 0.38 | 0.54 | 64 |
| locality | 0.5 | 0.03 | 0.06 | 29 |
| location | 0 | 0 | 0 | 14 |
| choreographer | 0.85 | 0.69 | 0.76 | 16 |
| composer | 0.8 | 0.66 | 0.72 | 61 |
| artist | 0.57 | 0.36 | 0.44 | 70 |
| legend | 0 | 0 | 0 | 33 |
| dish | 0 | 0 | 0 | 34 |
| meal | 0 | 0 | 0 | 17 |
| delicacy | 0 | 0 | 0 | 11 |
| salad | 0 | 0 | 0 | 9 |

Table 1: Precision (P), recall (R), $F_1$ Score (F), and support (S) for all 637 concepts ($F_1$ Score weighted by support) and selected individual concepts. Class membership was determined by an activation threshold of 0.5.

## 4 Discussion

The weighted classification $F_1$ score is 0.22 which suggests that the task is complex, probably due to the highly imbalanced data set. According to Table 1, however, $F_1$ scores vary significantly along individual concepts. While we observe a high score for *province*, our model has difficulties classifying *location*s, for instance. The same trend can be observed for *choreographer*s and *legend*s. What we see reflected in this table is the sharpness of concept boundaries. Arguably, the definition of a *province* is sharper than that of *location*. The same is true for *choreographer* and *legend*. We assume that the more precise a concept boundary, the higher the classification performance tends to be. We cannot, however, offer an explanation for the poor classification performance on some other concepts, such as the last ones in Table 1.

Fig. 1 (top) shows the NVC of "listening" with the top ten peaks labelled. For Table 1, a class membership was determined by an activation threshold of $0.5$ of the relevant output neuron. Fig. 1 (top), however, illustrates that the model activates many meaningful concepts beneath this threshold and thus $0.5$ might not be appropriate to determine class membership.

Some of the peaks are also reflected in the bottom plot of Fig. 1, which depicts the activation profile of the cosine similarity baseline method. The most notable difference between our method and the baseline is that the latter produces much more entropic activation profiles. It is less selective than NVC as NVC deactivates many concepts.

Fig. 2 (first column) shows that NVC indeed deactivates unrelated concepts, such as *personality*, *finding*, *filling*, *great*, and *work* that, according to cosine similarity, are close to the instance "listening." *Speaker*, *phone*, and *organ* arguably are reasonable concepts and yet deactivated by NVC but NVC replaces them with more meaningful concepts, as can be seen in the upper right plot in Fig. 2. Note that, contrary to NVC, the baseline method is not able to deactivate concepts that have close vectors in the word vector space, nor is it able to activate concepts that have vectors that are far from the input vector. Overall, a manual analysis suggests that the top 25 NVC concepts are more fitting than the top 25 cosine concepts.

## 5 Related Work

Concept knowledge bases such as the MCG exist because concepts are powerful abstractions of natural language instances that have been used for many downstream tasks, such as text classification (Song et al., 2011), ad-query similarity and query similarity (Kim et al., 2013), document similarity (Song and Roth, 2015), and semantic relatedness (Bekkali and Lachkar, 2019). The approaches mentioned above all implement some form of text conceptualization (TC).

TC models the probability $P(c|I)$ of a concept $c$ being reflected in a set of observed natural language instances $I$ (Song et al., 2011; Shalaby and Zadrozny, 2019). This is also the objective function of the model we train and our interpretability method can thus be understood as an implementation of TC.

Furthermore, besides the methods already discussed in the introduction, there is more research into the interpretability of language representations. Adi et al. (2017), for instance, also use auxiliary prediction tasks to analyse vector representations. However, they work on sentence level, not word level. Moreover, instead of retrieving concepts, they probe sentence length, word content conservation and word order conservation in the representation.

An approach similar to ours was introduced by Sommerauer and Fokkens (2018). The authors investigate the kind of semantic information encoded in word vectors. To this end, they train a classifier that recognizes whether word vectors carry specific semantic properties, some of which can be regarded as concepts.

## 6 Conclusion & Future Work

We introduced neural vector conceptualization as a means of interpreting continuous samples from a word vector space. We demonstrated that our method produces considerably less entropic concept activation profiles than the cosine similarity measure. For an input word vector, NVC activated meaningful concepts and deactivated unrelated ones, even if they were close in the word vector space.

Contrary to prior methods, by design, NVC operates in the original language space and is capable of modeling non-linear relations between language instances and concepts. Furthermore, our method is flexible: At the heart of it lies a neural NLP model that we trained on an instance-concept ground truth that could be replaced by another one.

In the future, we would like to extend NVC to contextualized representations. We consider this non-trivial because it may not be possible to directly apply the current instance-concept ground truth to contextualized instances, in particular if they are represented by sub-word embeddings.

## Acknowledgements

This research was partially supported by the German Federal Ministry of Education and Research through the project DEEPLEE (01IW17001). We would also like to thank the anonymous reviewers for their feedback and Leonhard Hennig for data and feedback.

## References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference of Learning Representations (ICLR)*.

Mohammed Bekkali and Abdelmonaime Lachkar. 2019. An effective short text conceptualization based on new short text similarity. *Social Network Analysis and Mining*, 9(1):1.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in neural information processing systems*, pages 2672–2680.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Dongwoo Kim, Haixun Wang, and Alice Oh. 2013. Context-dependent conceptualization. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Diederick P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Aykut Koç, Lutfi Kerem Senel, İhsan Utlu, and Haldun M. Ozaktas. 2018. Imparting Interpretability to Word Embeddings while Preserving Semantic Structure. *arXiv:1807.07279*.

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. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781*.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

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. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237.

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.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307. Association for Computational Linguistics.

Lutfi Kerem Senel, Ihsan Utlu, Veysel Yucesoy, Aykut Koc, and Tolga Cukur. 2018. Semantic Structure and Interpretability of Word Embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1769–1779.

Walid Shalaby and Wlodek Zadrozny. 2019. Learning concept embeddings for dataless classification via efficient bag-of-concepts densification. *Knowledge and Information Systems*, pages 1–24.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034*.

Pia Sommerauer and Antske Fokkens. 2018. Firearms and tigers are dangerous, kitchen knives and zebras are not: Testing whether word embeddings can tell. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Yangqiu Song and Dan Roth. 2015. Unsupervised sparse vector densification for short text similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1275–1280. Association for Computational Linguistics.

Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. 2011. Short text conceptualization using a probabilistic knowledgebase. In *Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2330–2336.

Zhongyuan Wang, Haixun Wang, Ji-Rong Wen, and Yanghua Xiao. 2015. An Inference Approach to Basic Level of Categorization. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15*, pages 653–662. ACM Press.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A Probabilistic Taxonomy for Text Understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 481–492. ACM.

# A NVCs

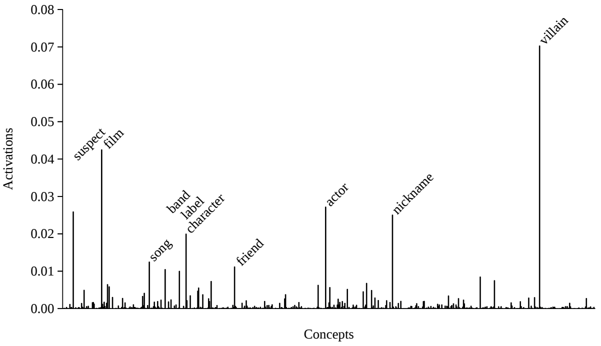

Figure 3: NVC of the word vector for "mafioso" (the instance was not encountered during training).

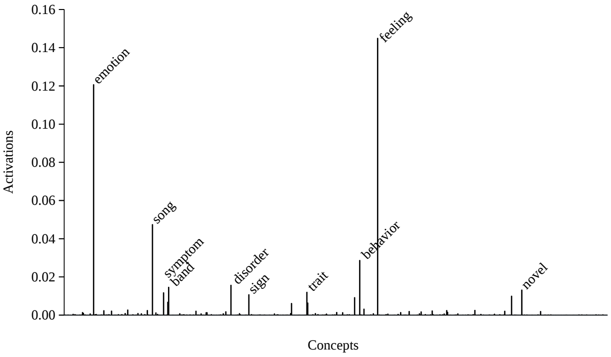

Figure 4: NVC of the word vector for "Jealousy" (the instance was not encountered during training).

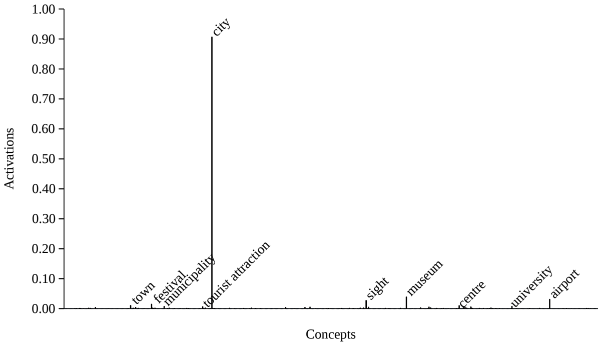

Figure 5: NVC of the word vector for "Berlin" (the instance was not encountered during training).

# B Concept Activations for Continuous Samples

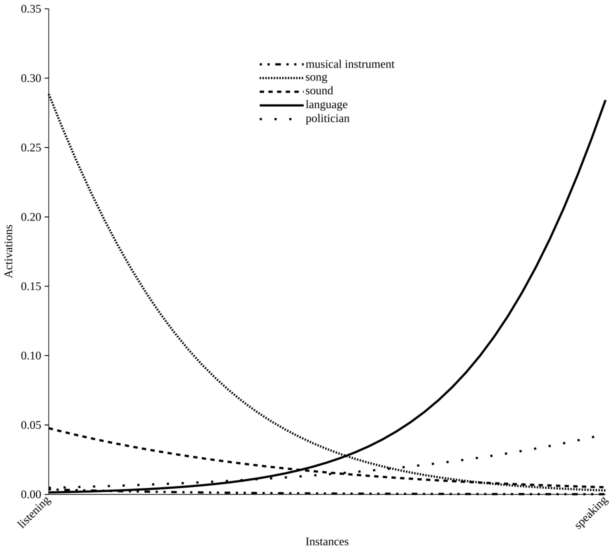

Figure 6: Concept activations of five selected concepts of word vectors sampled on the path between the instances "listening" and "speaking". Note the steady, non-oscillating paths between the instances.