

# Parallax: Visualizing and Understanding the Semantics of Embedding Spaces via Algebraic Formulae

Piero Molino

Uber AI Labs

San Francisco, CA, USA

piero@uber.com

Yang Wang

Uber Technologies Inc.

San Francisco, CA, USA

gnavvy@uber.com rivulet.zhang@gmail.com

Jiawei Zhang\*

Facebook

Menlo Park, CA, USA

## Abstract

Embeddings are a fundamental component of many modern machine learning and natural language processing models. Understanding them and visualizing them is essential for gathering insights about the information they capture and the behavior of the models. In this paper, we introduce Parallax<sup>1</sup>, a tool explicitly designed for this task. Parallax allows the user to use both state-of-the-art embedding analysis methods (PCA and t-SNE) and a simple yet effective task-oriented approach where users can explicitly define the axes of the projection through algebraic formulae. In this approach, embeddings are projected into a semantically meaningful subspace, which enhances interpretability and allows for more fine-grained analysis. We demonstrate<sup>2</sup> the power of the tool and the proposed methodology through a series of case studies and a user study.

## 1 Introduction

Learning representations is an important part of modern machine learning and natural language processing. These representations are often real-valued vectors also called embeddings and are obtained both as byproducts of supervised learning or as the direct goal of unsupervised methods. Independently of how the embeddings are learned, there is much value in understanding what information they capture, how they relate to each other and how the data they are learned from influences them. A better understanding of the embedded space may lead to a better understanding of the data, of the problem and the behavior of the model, and may lead to critical insights in improving such models. Because of their high-dimensional nature, they are hard to visualize effectively.

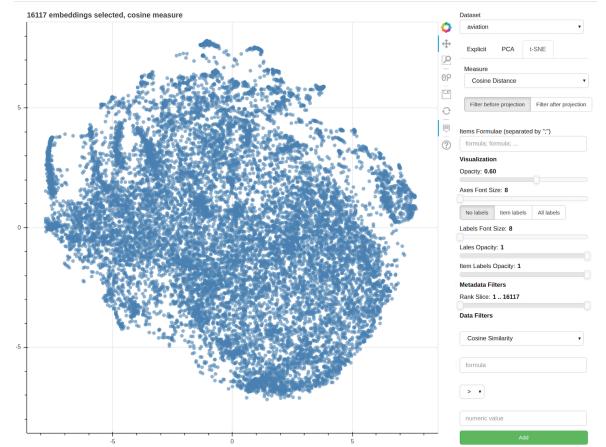


Figure 1: Screenshot of Parallax.

In this paper, we introduce Parallax, a tool for visualizing embedding spaces. The most widely adopted projection techniques (Principal Component Analysis (PCA) (Pearson, 1901) and t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton, 2008)) are available in Parallax. They are useful for obtaining an overall view of the embedding space, but they have a few shortcomings: 1) projections may not preserve distance in the original space, 2) they are not comparable across models and 3) do not provide interpretable axes, preventing more detailed analysis and understanding.

PCA projects embeddings on a lower dimensional space that has the directions of the highest variance in the dataset as axes. Those dimensions do not carry any interpretable meaning, so by visualizing the first two dimensions of a PCA projection, the only insight obtainable is semantic relatedness (Budanitsky and Hirst, 2006) between points by observing their relative closeness, and therefore, topical clusters can be identified. Moreover, as the directions of highest variance differ from embedding space to embedding space, the projections are incompatible among different em-

\*Work done while at Purdue University

<sup>1</sup><https://github.com/uber-research/parallax>

<sup>2</sup><https://youtu.be/CSkJGVsFPIg>

beddings spaces, and this makes them incomparable, a common issue among dimensionality reduction techniques.

t-SNE, differently from PCA, optimizes a loss that encourages embeddings that are in their respective close neighborhoods in the original high-dimensional space to be close in the lower dimensional projection space. t-SNE projections visually approximate better the original embedding space and topical clusters are more clearly distinguishable, but do not solve the issue of comparability of two different sets of embeddings, nor do they solve the lack of interpretability of the axes or allow for fine-grained inspection.

For these reasons, there is value in mapping embeddings into a more specific, controllable and interpretable semantic space. In this paper, a new and simple method to inspect, explore and debug embedding spaces at a fine-grained level is proposed. This technique is made available in Parallax alongside PCA and t-SNE for goal-oriented analysis of the embedding spaces. It consists of explicitly defining the axes of projection through formulae in vector algebra that use embedding labels as atoms. Explicit axis definition assigns interpretable and fine-grained semantics to the axes of projection. This makes it possible to analyze in detail how embeddings relate to each other with respect to interpretable dimensions of variability, as carefully crafted formulas can map (to a certain extent) to semantically meaningful portions of the space. The explicit axes definition also allows for the comparison of embeddings obtained from different datasets, as long as they have common labels and are equally normalized.

We demonstrate three visualizations that Parallax provides for analyzing subspaces of interest of embedding spaces and a set of example case studies including bias detection, polysemy analysis and fine-grained embedding analysis, but additional ones, like diachronic analysis and the analysis of representations obtained through graph learning or any other means, may be performed as easily. Moreover, the proposed visualizations can be used for debugging purposes and, in general, for obtaining a better understanding of the embedding spaces learned by different models and representation learning approaches.

The main contribution of this work lies in 1) the tool itself, as Parallax enables researchers in the fields of machine learning, computational

linguistics, natural language processing, social sciences and digital humanities to perform exploratory analysis and better understand the semantics of their embeddings. and 2) the use of explicit user-defined algebraic formulae as axes for projecting embedding spaces into semantically-meaningful subspaces that when visualized provide interpretable axes, which allows more fine grained task-oriented analysis and comparison across corpora.

We show how this methodology can be widely used through a series of case studies on well known models and data, and furthermore, we validate its usefulness for goal-oriented analysis through a user study.

## 2 Parallax

Parallax, is a tool that allows to visualize embedding spaces through the common PCA and t-SNE techniques and through the explicit definition of projection axes through algebraic formulae, which is particularly useful for goal oriented analysis. Parallax interface, shown in Figure 1, presents a plot on the left side (scatter or polar) and controls on the right side that allow users to define parameters of the projection (what measure to use, values for the hyperparameters, the formulae for the axes in case of explicit axes projections are selected, etc.) and additional filtering and visualization parameters. Filtering parameters define logic rules applied to embeddings metadata to decide which of them should be visualized, e.g., the user can decide to visualize only the most frequent words or only verbs if metadata about part-of-speech tags is made available. Filters on the embeddings themselves can also be defined, e.g., the user can decide to visualize only the embeddings with cosine similarity above 0.5 to the embedding of “horse”.

In particular, Parallax’s capability of explicitly defining axes is useful for goal-oriented analyses, e.g., when the user has a specific analysis goal in mind, like detecting bias in the embeddings space. Goals are defined in terms of dimensions of variability (axes of projection) and items to visualize (all the embeddings that are projected, after filtering). In the case of a few dimensions of variability (up to three) and potentially many items of interest, a Cartesian view is ideal. Each axis is the vector obtained by evaluating the algebraic formula it is associated with, and the coordinates displayed are similarities or distances of the items with re-

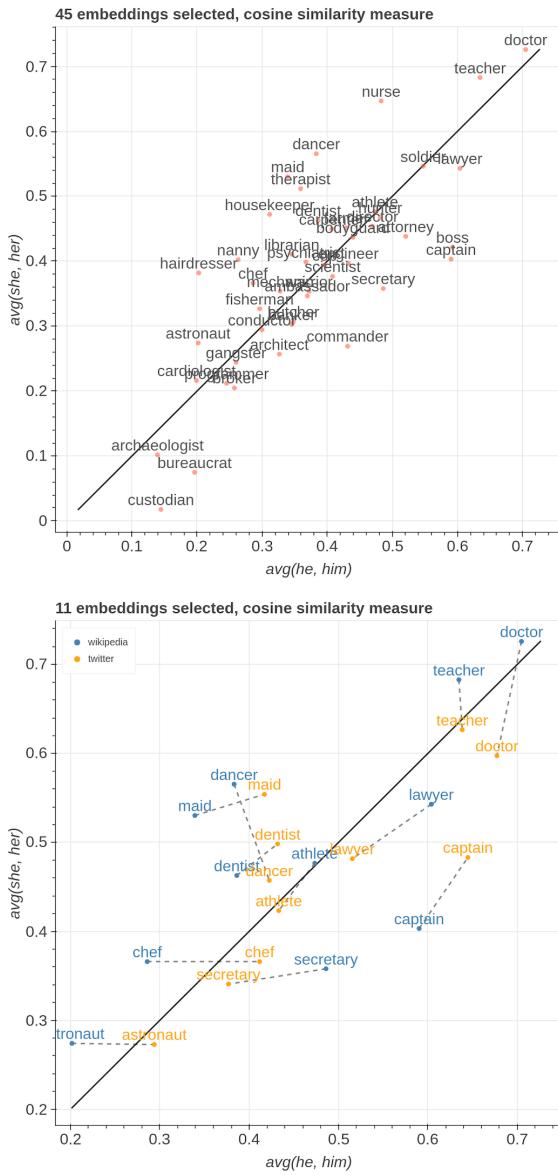


Figure 2: In the top we show professions plotted on “male” and “female” axes in *Wikipedia* embeddings. In the bottom we show their comparison in *Wikipedia* and *Twitter* datasets.

spect to each axis. Figure 2 shows an example of a bi-dimensional Cartesian view. In the case where the goal is defined in terms of many dimensions of variability, a polar view is preferred. The polar view can visualize many more axes by showing them in a circle, but it is limited in the number of items it can display, as each item will be displayed as a polygon with each vertex lying on a different axis and too many overlapping polygons would make the visualization cluttered. Figure 6 shows an example of a five-dimensional polar view.

The use of explicit axes allows for interpretable comparison of different embedding spaces, trained on different corpora or on the same corpora but with different models, or even trained on two different time slices of the same corpora. The only requirement for embedding spaces to be comparable is that they contain embeddings for all labels present in the formulae defining the axes. Moreover, embeddings in the two spaces do not need to be of the same dimension, but they need to be normalized. Items will now have two sets of coordinates, one for each embedding space, and thus they will be displayed as lines. Short lines are interpreted as items being embedded similarly in the subspaces defined by the axes in both embedding spaces, while long lines are interpreted as really different locations in the subspaces, and their direction gives insight on how items shift in the two subspaces. Those two embedding spaces could be, for instance, embeddings trained on a clean corpus like Wikipedia as opposed to a noisy corpus like tweets from Twitter, or the two corpora could be two different time slices of the same corpus, in order to compare how words changed over time. The bottom side of Figure 2 shows an example of how to use the Cartesian comparison view to compare embeddings in two datasets.

### 3 Case Studies

Parallax can be used fruitfully in many analysis tasks in linguistics, digital humanities, in social studies based on empirical methods, and can also be used by researchers in computational linguistics and machine learning to inspect, debug and ultimately better understand the representations learned by their models.

In this section, a few goal-oriented use cases are presented, but Parallax’s flexibility allows for many others. We used 50-dimensional publicly available GloVe (Pennington et al., 2014) embeddings

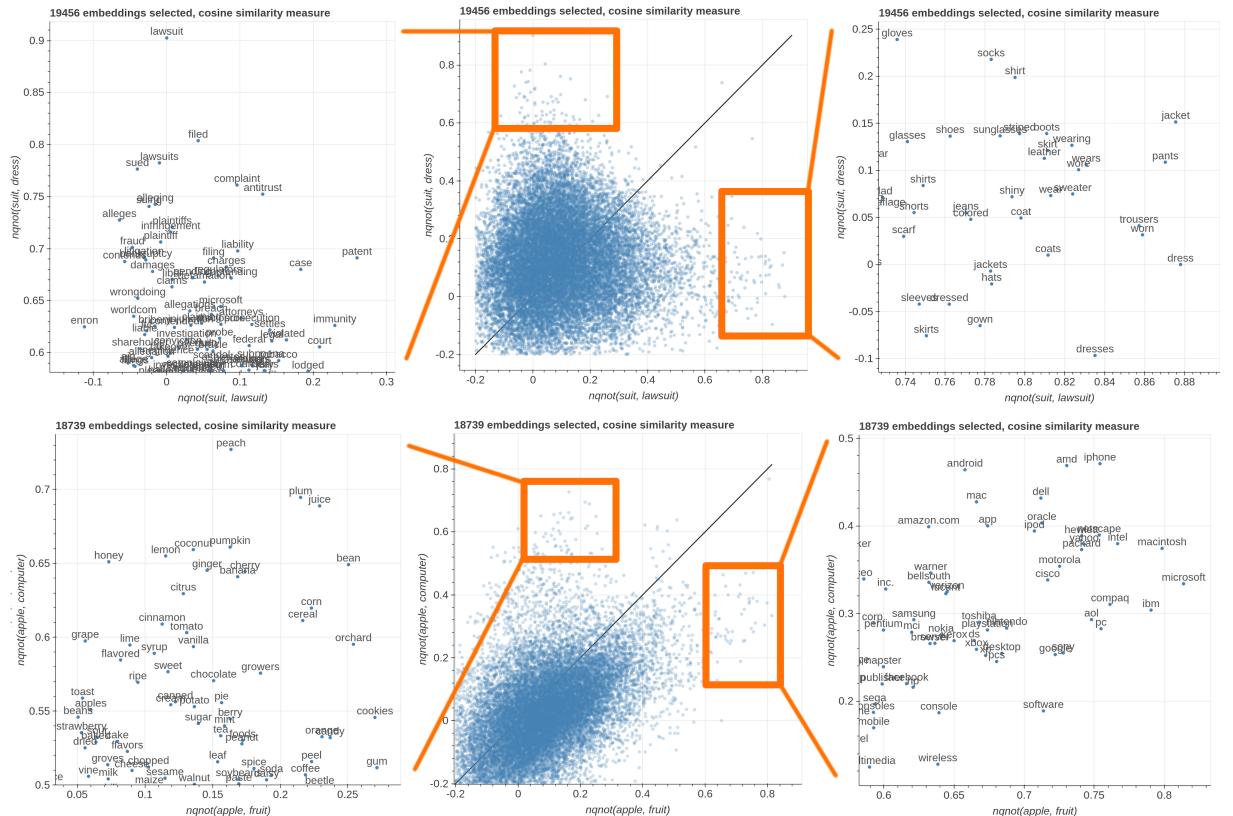


Figure 3: In the top a plot of embeddings in *Wikipedia* with *suit* negated with respect to *lawsuit* and *dress* respectively as axes. In the bottom a plot of *apple* negated with respect to *fruit* and *computer*.

trained on Wikipedia and Gigaword 5 summing to 6 billion tokens (for short *Wikipedia*) and 2 billion tweets containing 27 billion tokens (*Twitter*).

**Bias detection** The task of bias detection is to identify, and in some cases correct for, bias in data that is reflected in the embeddings trained on such data. Studies have shown how embeddings incorporate gender and ethnic biases ((Garg et al., 2018; Bolukbasi et al., 2016; Islam et al., 2017)), while other studies focused on warping spaces in order to de-bias the resulting embeddings ((Bolukbasi et al., 2016; Zhao et al., 2017)). We show how our proposed methodology can help visualize biases.

To visualize gender bias with respect to professions, the goal is defined with the formulae  $\text{avg}(he, him)$  and  $\text{avg}(she, her)$  as two dimensions of variability, in a similar vein to (Garg et al., 2018). A subset of the professions used by (Bolukbasi et al., 2016) is selected as items and cosine similarity is adopted as the measure for the projection. The Cartesian view visualizing *Wikipedia* embeddings is shown in the left of Figure 2. *Nurse*, *dancer*, and *maid* are the professions closer to the “female” axis, while *boss*, *captain*, and *commander* end up closer to the “male” axis.

The Cartesian comparison view comparing the embeddings trained on *Wikipedia* and *Twitter* is shown in the right side of Figure 2. Only the embeddings with a line length above 0.05 are displayed. The most interesting words in this visualization are the ones that shift the most in the direction of negative slope. In this case, *chef* and *doctor* are closer to the “male” axis in *Twitter* than in *Wikipedia*, while *dancer* and *secretary* are closer to the bisector in *Twitter* than in *Wikipedia*.

Additional analysis of how words tend to shift in the two embedding spaces would be needed in order to derive provable conclusions about the significance of the shift, for instance through a permutation test with respect to all possible pairs, but the visualization can help inform the most promising words to perform the test on.

**Polysemy analysis** Methods for representing words with multiple vectors by clustering contexts have been proposed (Huang et al., 2012; Nee-lakantan et al., 2014), but widely used pre-trained vectors conflate meanings in the same embedding.

**Widdows (2003)** showed how using a binary orthonormalization operator that has ties with the quantum logic *not* operator it is possible to remove

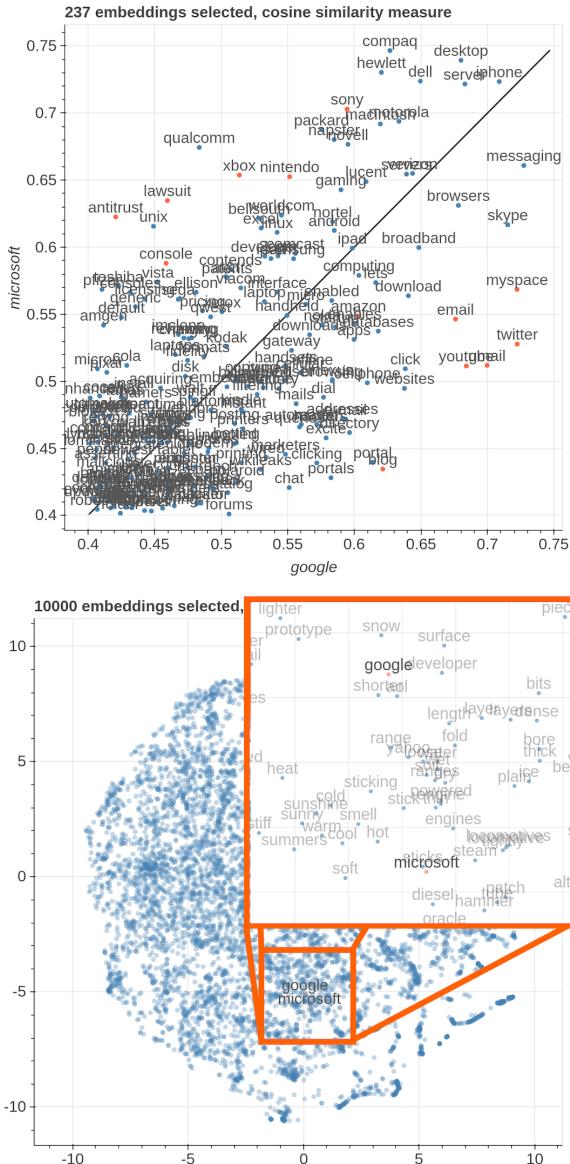


Figure 4: The top figure is a fine-grained comparison of the subspace on the axis *google* and *microsoft* in *Wikipedia*, the bottom one is the t-SNE counterpart.

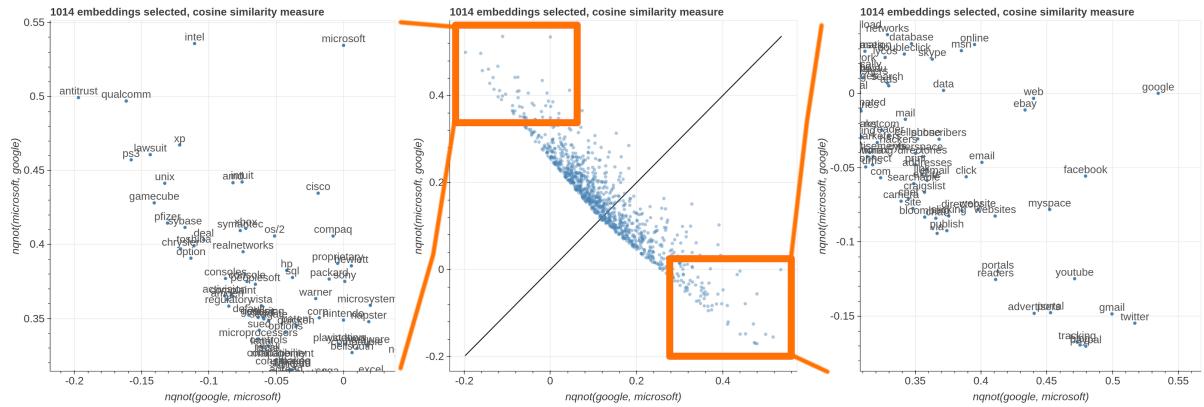


Figure 5: Fine-grained comparison of the subspace on the axis  $nqnot(google, microsoft)$  and  $nqnot(microsoft, google)$  in Wikipedia.

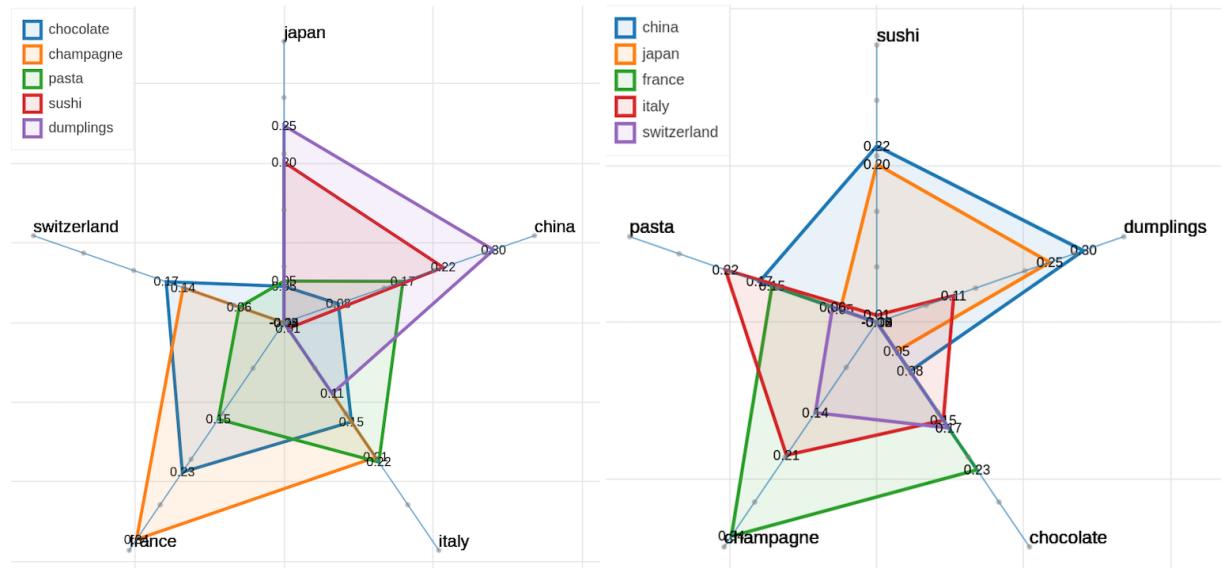


Figure 6: Two polar views of countries and foods in Wikipedia.

part of the conflated meaning from the embedding of a polysemous word. The authors define the operator  $nqnot(a, b) = a - \frac{a \cdot b}{|b|^2}b$  and we show with a comparison plot how it can help distinguish the different meanings of a word.

For illustrative purposes, we choose the same polysemous word used by (Widdows, 2003), *suit*, and use the *nqnot* operator to orthonormalize with respect to *lawsuit* and *dress*, the two main meanings used as dimensions of variability. The items in our goal are the 20,000 most frequent words in the *Wikipedia* embedding space removing stop-words. In Figure 3, we show the overall plot and we zoom on the items that are closer to each axis. Words closer to the axis negating *lawsuit* are all related to dresses and the act of wearing something, while words closer to the axis negating *dress* are related to law.

We chose another polysemous word, *apple*, and orthonormalized with respect to *fruit* and *computer*. In the bottom of Figure 3 words that have a higher similarity with respect to the first axis are all tech related, while the ones that have a higher similarity with respect to the second axis are mostly other fruits or food.

Both examples confirm the ability of the *nqnot* operator to disentangle multiple meanings from polysemous embeddings and show how the proposed visualizations are able to show it clearly.

**Fine-grained embedding analysis** We consider embeddings that are close to be semantically related, but even close embeddings may have nuances that distinguish them. When projecting in two dimensions through PCA or t-SNE we are conflating a multidimensional notion of similarity to a bi-dimensional one, losing the fine-grained distinctions. The Cartesian view allows for a more fine-grained visualization that emphasizes nuances that could otherwise go unnoticed.

To demonstrate this capability, we select as dimensions of variability single words in close vicinity to each other in the *Wikipedia* embedding space: *google* and *microsoft*, as *google* is the closest word to *microsoft* and *microsoft* is the 3<sup>rd</sup> closest word to *google*. As items, we pick the 30,000 most frequent words removing stop-words and remove the 500 most frequent words (as they are too generic) and keeping only the words that have a cosine similarity of at least 0.4 with both *google* and *microsoft* and a cosine similarity below 0.75 with respect to *google + microsoft*, as we are in-

terested in the most polarized words.

The left side of Figure 4 shows how even if those embeddings are close to each other, it is easy to identify peculiar words (highlighted with red dots). The ones that relate to web companies and services (*twitter*, *youtube*, *myspace*) are much closer to the *google* axis. Words related to both legal issues (*lawsuit*, *antitrust*) and videogames (*ps3*, *nintendo*, *xbox*) and traditional IT companies are closer to the *microsoft* axis.

In Figure 5 we obtain similar results by using *google* and *microsoft* orthonormalized with respect to each other as axes. The top left and the bottom right corners are the most interesting ones, as they contain terms that are related to one word after having negated the other. The pattern that emerges is similar to the one highlighted in the left side of Figure 4, but now also operating systems terms (*unix*, *os/2*) appear in the *microsoft* corner, while *advertisement* and *tracking* appear in the *google* corner.

For contrast, the t-SNE projection is shown in the right side of Figure 4: it is hard to appreciate the similarities and differences among those embeddings other than seeing them being close in the projected space. This confirms on one hand that the notion of similarity between terms in an embedding space hides many nuances that are captured in those representations, and on the other hand, that the proposed methodology enables for a more detailed inspection of the embedded space.

Multi-dimensional similarity nuances can be visualized using the polar view. In Figure 6, we show how to use Parallax to visualize a small number of items on more than two axes, specifically five food-related items compared over five countries' axes. The most typical food from a specific country is the closest to the country axis, with *sushi* being predominantly close to *Japan* and *China*, *dumplings* being close to both Asian countries and *Italy*, *pasta* being predominantly closer to *Italy*, *chocolate* being close to European countries and *champagne* being closer to *France* and *Italy*. This same approach could be also be used for bias detection among different ethnicities, for instance, where the axes are concepts capturing the notion of ethnicity and items could be adjectives, or the two could be swapped, depending.

**Analogy** The analogy task introduced by (Mikolov et al., 2013) is often used as an example to show the amount of information encoded in

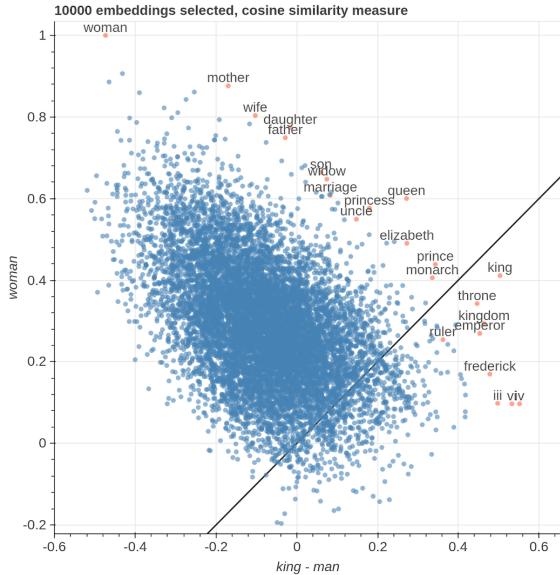


Figure 7: king-man vs woman

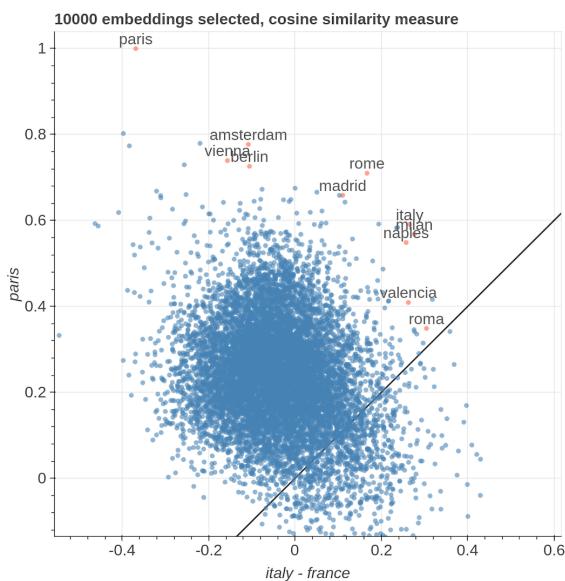


Figure 8: italy-france vs paris

the dimensions of the embedding space and the fact that linear relationships emerge from training embedding models. In this section we show how using explicit axes makes it really easy to analyze more in detail the space of linear relationships.

Two examples are presented.

In Figure 7 we show a cartesian plot where the axes are 'king-man' and 'woman' respectively. The bisector line depicted is the direction of the sum, meaning that words that are far off in the direction (closer to both axes) are to be considered good candidates to solve the analogy. In this case the words that are further in the direction are 'king' and 'queen', but usually when computing scores in these tasks, words already present in the analogy are omitted, so, removing 'king', 'queen' remain as the best candidate for the analogy. In the plot we highlight several words in the same band perpendicular to the bisector, meaning that all those words are similarly good candidates for the analogy. Within the band, some words are closer to the 'king-man' axis (interpretable as the concept of 'royalty'), for instance: 'frederick', 'emperor', 'ruler', 'throne', but also roman numerals like 'iii', 'iv' and 'vi'. Other words within the band are close to the 'queen' axis: 'mother', 'wife', 'daughter' and 'father'.

In Figure 8 we show a cartesian plot where the axes are 'italy-france' and 'paris' respectively. The same interpretation of the bisector and the bands of the previous example hold also in this case. 'rome' is the word further off in both directions, while words closer to the 'paris' axis tend to be other European capitals: 'amsterdam', 'vienna', 'berlin'. Words closer to the 'italy-rome' axis (interpretable as "italianness") are mostly Italian cities: 'italy', 'naples', 'milan', 'valencia'.

This way of visualizing analogies provides a richer understanding of the embedding space and a characterization of why some words appear in the analogy ranking list.

**Paradigmatic versus Syntagmatic** Depending on the definition of context embeddings methods can learn to a different notion of relatedness in the similarity of the vectors that are obtained. Even with a specific notion of context, such as a window of occurrence, the model architecture can lead to obtain representations that encode different word relationships. For instance, the GloVe (Pennington et al., 2014) algorithm trains two sets of vectors, one for the rows of the co-occurrence matrix

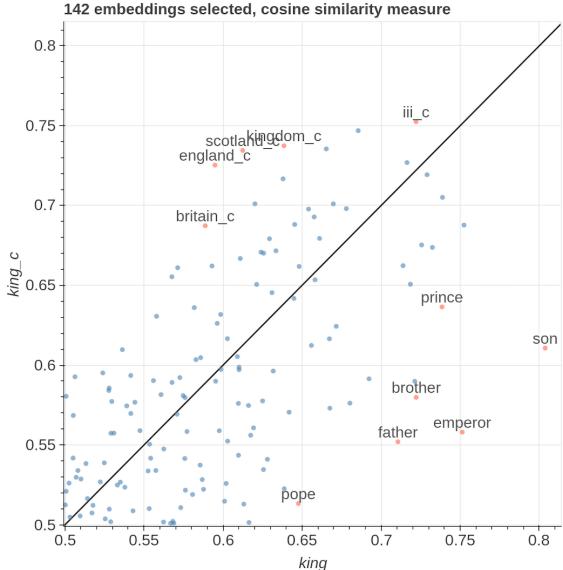


Figure 9: paradigmatic vs syntagmatic

and one for the columns. Those two sets of vectors will likely contain different information as embeddings of row words will be similar for words that co-occur with the same contexts and are then substitutable with each other (paradigmatic relationship), while embeddings of the column words will be similar to each other if they appear co-occur together in the contexts (syntagmatic relationship). At the end of the training of a GloVe model, the two sets of vectors are summed together, obfuscating their different nature. We modified the GloVe training procedure to save the two embeddings set without summing them and trained a model on the text8 dataset as a proof of concept.

In Figure ?? we show a cartesian plot using the word ‘king’ (the embedding of the row word) and ‘king\_c’ (the embedding of the row word) as axes. Note that ‘\_c’ at the end of a word denotes an embedding of a column, it is just a simple way to differentiate among the two sets of vectors. By looking at the words that are closer to the ‘king’ axis, ‘emperor’, ‘prince’, ‘pope’, ‘father’ and ‘son’ it is evident that they are indeed words in a paradigmatic relationship with ‘king’ as they are substitutable. On the other hand, looking at the words closer to the ‘king\_c’ axis, the words ‘scotland\_c’, ‘kingdom\_c’, ‘england\_c’, ‘britain\_c’ and ‘iiii\_c’ appear. Those words are clearly in a syntagmatic relationship with ‘king\_c’ as they will probably appear in the same window of text like ‘the King of England’ or ‘King George III’.

With the use of explicit axes this phenomenon

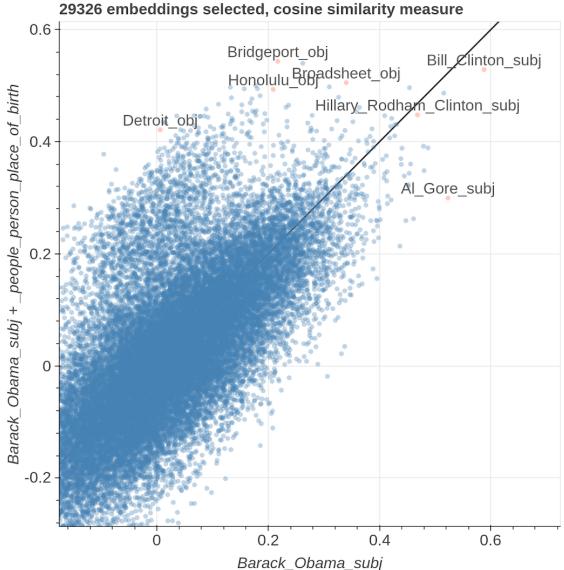


Figure 10: knowledge bases

is made clear and visible.

**Knowledge Base Embeddings** Another area where embeddings are trained is automatic construction of Knowledge Bases (KBs). In this thread of works, representations of entities in the KB and their relationships are encoded, with the aim to use those representations to automatically infer missing data and in some cases populate a probabilistic KB.

Many approaches have been proposed, the one we use here in order to obtain some insights through our proposed visualizations is StarSpace (Wu et al., 2018), trained on the FB15K dataset (Bordes et al., 2013), a subset of FreeBase. In this algorithm different embeddings are obtained for each subject, predicate and object by training to predict the object from the subject and the predicate. Given the triple  $\langle \text{Barack\_Obama}, \text{place\_of\_birth}, \text{Honolulu} \rangle$ , the model will minimize the distance between  $e_{\text{subj}}(\text{Barack\_Obama}) + e_{\text{pred}}(\text{place\_of\_birth})$  and  $e_{\text{obj}}(\text{Honolulu})$ , where  $e_{\text{subj}}$ ,  $e_{\text{pred}}$  and  $e_{\text{obj}}$  are functions that map identifiers of entities and predicates to their embeddings when they appear as subject, predicate and object respectively. This also implies that the same entity may have distinct subject and object embeddings.

In Figure 10 we show a cartesian plot using the entity ‘Barack\_Obama\_subj’ (the embedding of the entity Barack\_Obama when it appears as subject) and ‘Barack\_Obama\_subj + person\_person\_place\_of\_birth’ (the of the embedding

Accuracy	Factor	$F_{(1,91)}$	p-value
Projection × Task	Projection	46.11	0.000***
	Task	1.709	0.194
	Projection × Task	3.452	0.066
Projection × Obfuscation	Projection	57.73	0.000***
	Obfuscation	23.93	0.000***
	Projection × Obf	5.731	0.019*

Table 1: Two-way ANOVA analyses of Task (Commonality vs. Polarization) and Obfuscation (Obfuscated vs. Non-obfuscated) over Projection (Explicit Formulae vs. t-SNE).

of the entity Barack\_Obama and the predicate person\_person\_place\_of\_birth that denotes the relationship between people and their place of birth) as axes. Ideally the second axis should be close to the place of birth of Barack Obama, as that is what the model was optimized for. From the visualization it is apparent how the embeddings close to the Barack\_Obama\_subj axis below or close to the bisector are mostly other American democratic presidents or candidate presidents, while the ones with higher cosine similarity with respect to the vertical axis are mostly cities. The correct object, Honolulu, is among the top ranked ones, although not the one with the highest similarity to the embedding of the vertical axis.

## 4 User Study

We conducted a user study to find out if and how visualizations using user-defined semantically meaningful algebraic formulae help users achieve their analysis goals. What we are not testing for is the projection quality itself, as in PCA and t-SNE it is obtained algorithmically, while in our case it is explicitly defined by the user. We formalized the research questions as: Q1) Does Explicit Formulae outperform t-SNE in goal-oriented tasks? Q2) Which visualization do users prefer?

To answer these questions we invited twelve subjects among data scientists and machine learning researchers, all acquainted with interpreting dimensionality reduction results. We defined two types of tasks, namely Commonality and Polarization, in which subjects were given a visualization together with a pair of words (used as axes in Explicit Formulae or highlighted with a big font and red dot in case of t-SNE). We asked the subjects to identify either common or polarized words w.r.t. the two provided ones. The provided pairs were: banana & strawberry, google & microsoft, nerd &

geek, book & magazine. The test subjects were given a list of eight questions, four per task type, and their proposed lists of five words are compared with a gold standard provided by a committee of two computational linguistics experts. The tasks are fully randomized within the subject to prevent from learning effects. In addition, we obfuscated half of our questions by replacing the words with a random numeric ID to prevent prior knowledge from affecting the judgment. We track the *accuracy* of the subjects by calculating the number of words provided that are present in the gold standard set, and we also collected an overall *preference* for either visualizations.

As reported in Table 1, two-way ANOVA tests revealed significant differences in accuracy for the factor of Projection and t-SNE against both Task and Obfuscation, which is a strong indicator that the proposed Explicit Formulae method outperforms t-SNE in terms of accuracy in both Commonality and Polarization tasks. We also observed significant differences in Obfuscation: subjects tend to have better accuracy when the words are not obfuscated. We run post-hoc t-tests that confirmed how the accuracy of Explicit Formulae on Non-obfuscated is significantly better than Obfuscated, which in turn is significantly better than t-SNE Non-obfuscated, which is significantly better than t-SNE Obfuscated. Concerning Preference, nine out of all twelve (75%) subjects chose Explicit Formulae over t-SNE. In conclusion, our answers to the research questions are that (Q1) Explicit Formulae leads to better accuracy in goal-oriented tasks, (Q2) users prefer Explicit Formulae over t-SNE.

## 5 Related Work

**Embedding methods and applications.** Several methods for learning embeddings from symbolic data have been recently proposed (Pennington et al., 2014; Mikolov et al., 2013; Mnih and Kavukcuoglu, 2013; Lebret and Collobert, 2014; Ji et al., 2016; Rudolph et al., 2016; Nickel et al., 2016). The learned representations have been used for a variety of tasks like recommendation (Barkan and Koenigstein, 2016), link prediction on graphs (Grover and Leskovec, 2016), discovery of drug-drug interaction (Abdelaziz et al., 2017) and many more. In particular, positive results in learning embeddings for words using a surrogate prediction task (Mikolov et al., 2013)

started the resurgence of interest in those methods, while a substantial body of research from the distributional semantics community using count and matrix factorization based methods (Deerwester et al., 1990; Baroni and Lenci, 2010; Kanerva et al., 2000; Levy and Goldberg, 2014; Biemann and Riedl, 2013) was previously developed. Refer to (Lenci, 2018) for a comprehensive overview. In some of those papers, explicit axes are used to visualize portions of the embedding space in an ad-hoc fashion.

In their recent paper, (Heimerl and Gleicher, 2018) extracted a list of routinely conducted tasks where embeddings are employed in visual analytics for NLP, such as *compare concepts*, *finding analogies*, and *predict contexts*. iVisClustering (Lee et al., 2012) represents topic clusters as their most representative keywords and displays them as a 2D scatter plot and a set of linked visualization components supporting interactively constructing topic hierarchies. ConceptVector (Park et al., 2018) makes use of multiple keyword sets to encode the relevance scores of documents and topics: positive words, negative words, and irrelevant words. It allows users to select and build a concept iteratively. (Liu et al., 2018) display pairs of analogous words obtained through analogy by projecting them on a 2D plane obtained through a PCA and an SVM to find the plane that separates words on the two sides of the analogy. Besides word embeddings, visualization has been used to understand topic modeling (Chuang et al., 2012) and how topic models evolve over time (Havre et al., 2002). Compared to existing literature, our work allows for more fine-grained direct control over the conceptual axes and the filtering logic, allowing users to: 1) define concepts based on explicit algebraic formulae beyond single keywords, 2) filter depending on metadata, 3) perform multidimensional projections beyond the common 2D scatter plot view using the polar view, and 4) perform comparisons between embeddings from different data sources. Those features are absent in other proposed tools.

## 6 Conclusions

In this work, we presented Parallax, a tool for embedding visualization, and a simple methodology for projecting embeddings into lower-dimensional semantically-meaningful subspaces through explicit algebraic formulae. We showed how

this approach allows goal-oriented analyses and more fine-grained and cross-dataset comparisons through a series of case studies and a user study.

## Acknowledgments

The authors want to thank Antonio Vergari, Eli Bingham, Fritz Obermayer, Gaetano Rossiello, Pasquale Minervini, Lezhi Li, Zoubin Ghahramani and Peter Dayan for the fruitful conversations and opinions that lead to improvement of this work.

## References

- Ibrahim Abdelaziz, Achille Fokoue, Oktie Hassan-zadeh, Ping Zhang, and Mohammad Sadoghi. 2017. Large-scale structural and textual similarity-based mining of knowledge graph to predict drug-drug interactions. *J. Web Sem.*, 44:104–117.
- Oren Barkan and Noam Koenigstein. 2016. ITEM2VEC: neural item embedding for collaborative filtering. In *MLSP*, pages 1–6. IEEE.
- Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Comput. Ling.*, 36(4):673–721.
- Chris Biemann and Martin Riedl. 2013. Text: now in 2d! A framework for lexical expansion with contextual similarity. *J. Language Modelling*, 1(1):55–95.
- Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NIPS*, pages 4349–4357.
- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2787–2795.
- Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Ling.*, 32(1):13–47.
- Jason Chuang, Christopher D. Manning, and Jeffrey Heer. 2012. Termite: visualization techniques for assessing textual topic models. In *AVI*, pages 74–77.
- Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

- Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*.
- Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864. ACM.
- Susan Havre, Elizabeth G. Hetzler, Paul Whitney, and Lucy T. Nowell. 2002. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. Vis. Comput. Graph.*, 8(1):9–20.
- Florian Heimerl and Michael Gleicher. 2018. Interactive analysis of word vector embeddings. *Comput. Graph. Forum*, 37(3):253–265.
- Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *ACL*.
- Aylin Caliskan Islam, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora necessarily contain human biases. *Science*, 356:183–186.
- Shihao Ji, Hyokun Yun, Pinar Yanardag, Shin Matushima, and S. V. N. Vishwanathan. 2016. Worrank: Learning word embeddings via robust ranking. In *EMNLP*, pages 658–668.
- Pentti Kanerva, Jan Kristoferson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In *Cognitive Science Society*, pages 103–6.
- Rémi Lebret and Ronan Collobert. 2014. Word embeddings through hellinger PCA. In *EACL*, pages 482–490.
- Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John T. Stasko, and Haesun Park. 2012. ivisclustering: An interactive visual document clustering via topic modeling. *Comput. Graph. Forum*, 31(3):1155–1164.
- Alessandro Lenci. 2018. Distributional models of word meaning. *Annual Review of Linguistics*, 4(1):151–171.
- Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *CoNLL*, pages 171–180.
- Shusen Liu, Peer-Timo Bremer, Jayaraman J. Thiagarajan, Vivek Srikumar, Bei Wang, Yarden Livnat, and Valerio Pascucci. 2018. Visual exploration of semantic relationships in neural word embeddings. *IEEE Trans. Vis. Comput. Graph.*, 24(1):553–562.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.
- Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*, pages 2265–2273.
- Arvind Neelakantan, Jeevan Shankar, Alexandre Paschos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *EMNLP*, pages 1059–1069.
- Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.
- Deok Gun Park, Seungyeon Kim, Jurim Lee, Jaegul Choo, Nicholas Diakopoulos, and Niklas Elmquist. 2018. Conceptvector: Text visual analytics via interactive lexicon building using word embedding. *IEEE Trans. Vis. Comput. Graph.*, 24(1):361–370.
- K. Pearson. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.
- Maja R. Rudolph, Francisco J. R. Ruiz, Stephan Mandt, and David M. Blei. 2016. Exponential family embeddings. In *NIPS*, pages 478–486.
- Dominic Widdows. 2003. Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In *ACL*, pages 136–143.
- Ledell Yu Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2018. Starspace: Embed all the things! In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5569–5577. AAAI Press.
- Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, pages 2979–2989.

## A Appendix

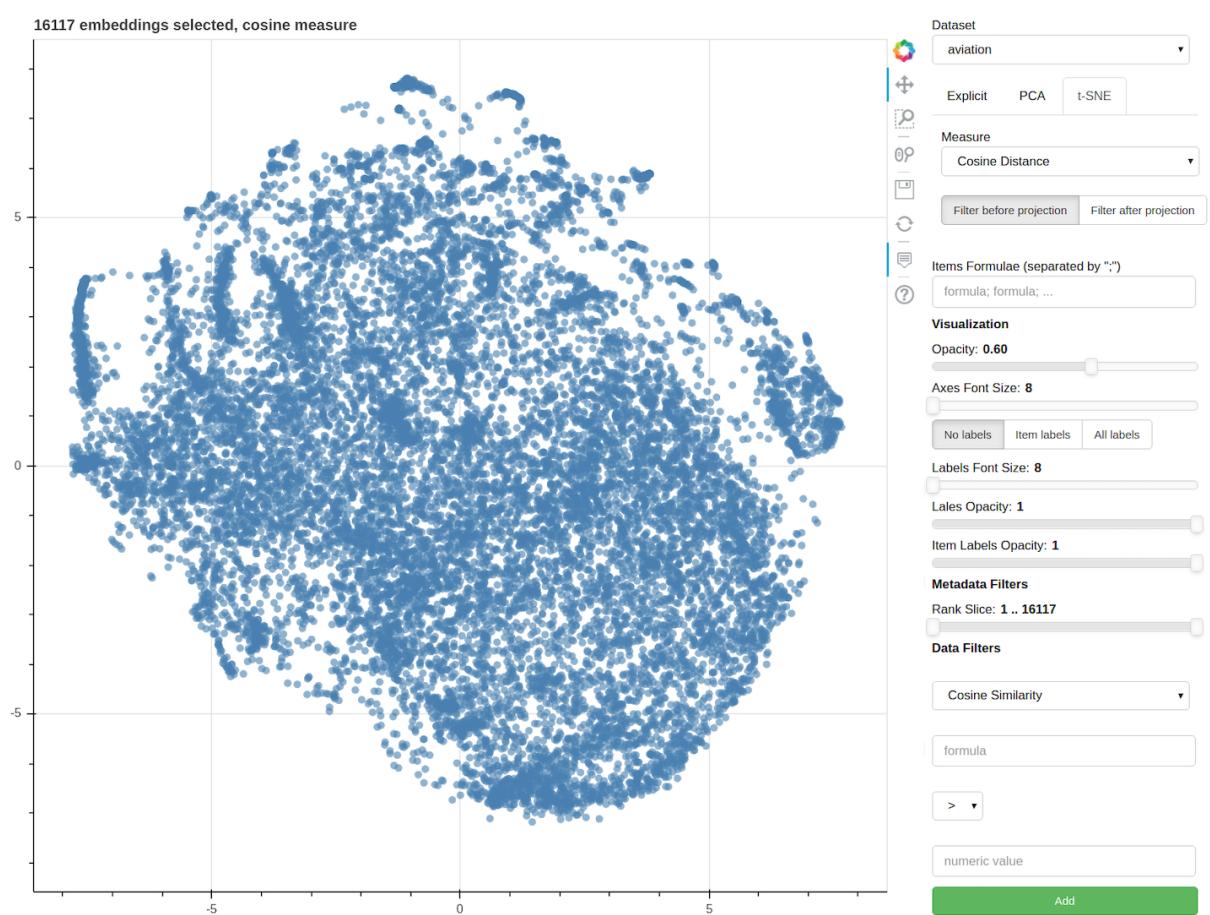


Figure 11: Screenshot of Parallax.

**45 embeddings selected, cosine similarity measure**

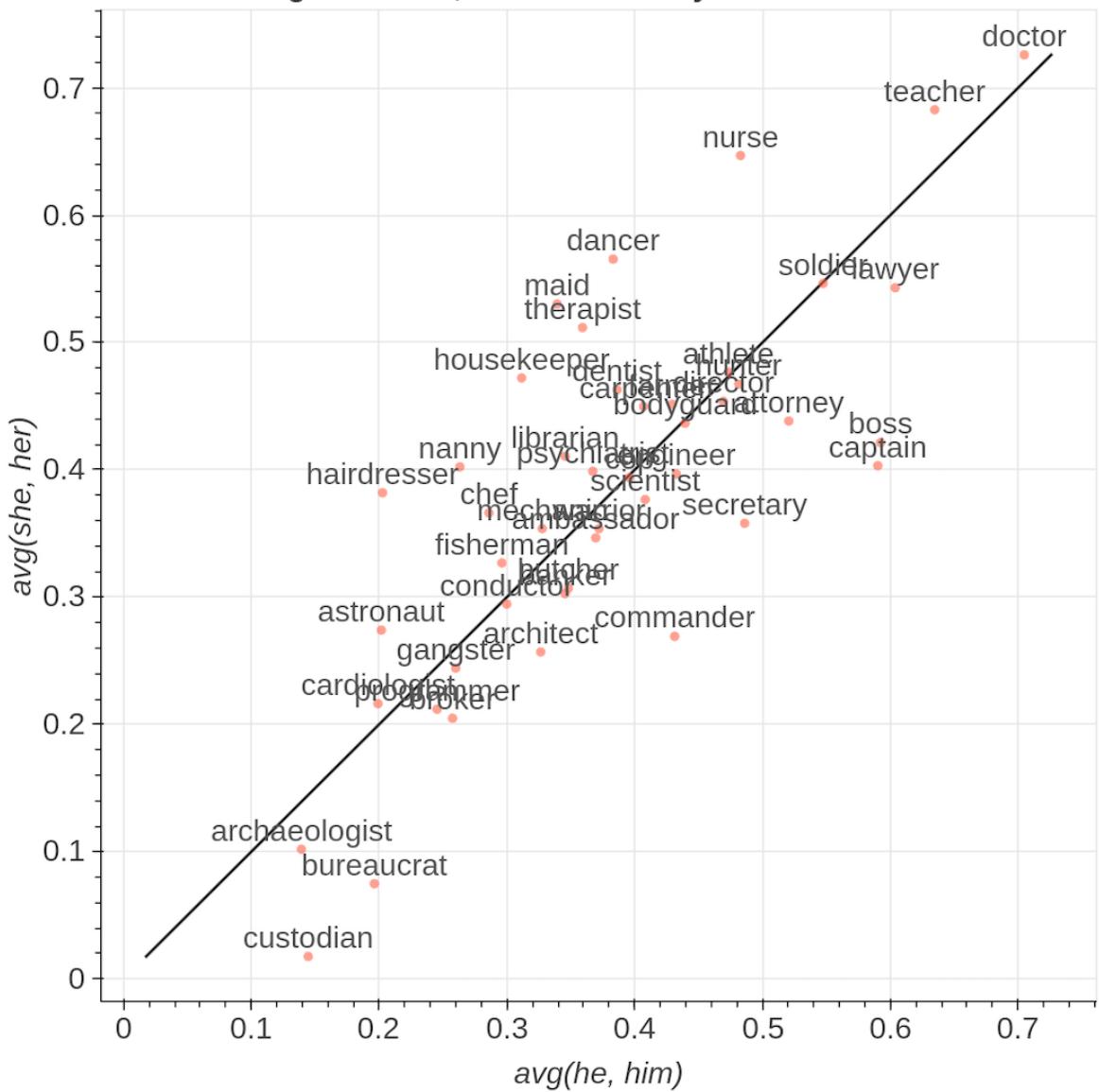


Figure 12: Professions plotted on “male” and “female” axes in *Wikipedia* embeddings.

### 11 embeddings selected, cosine similarity measure

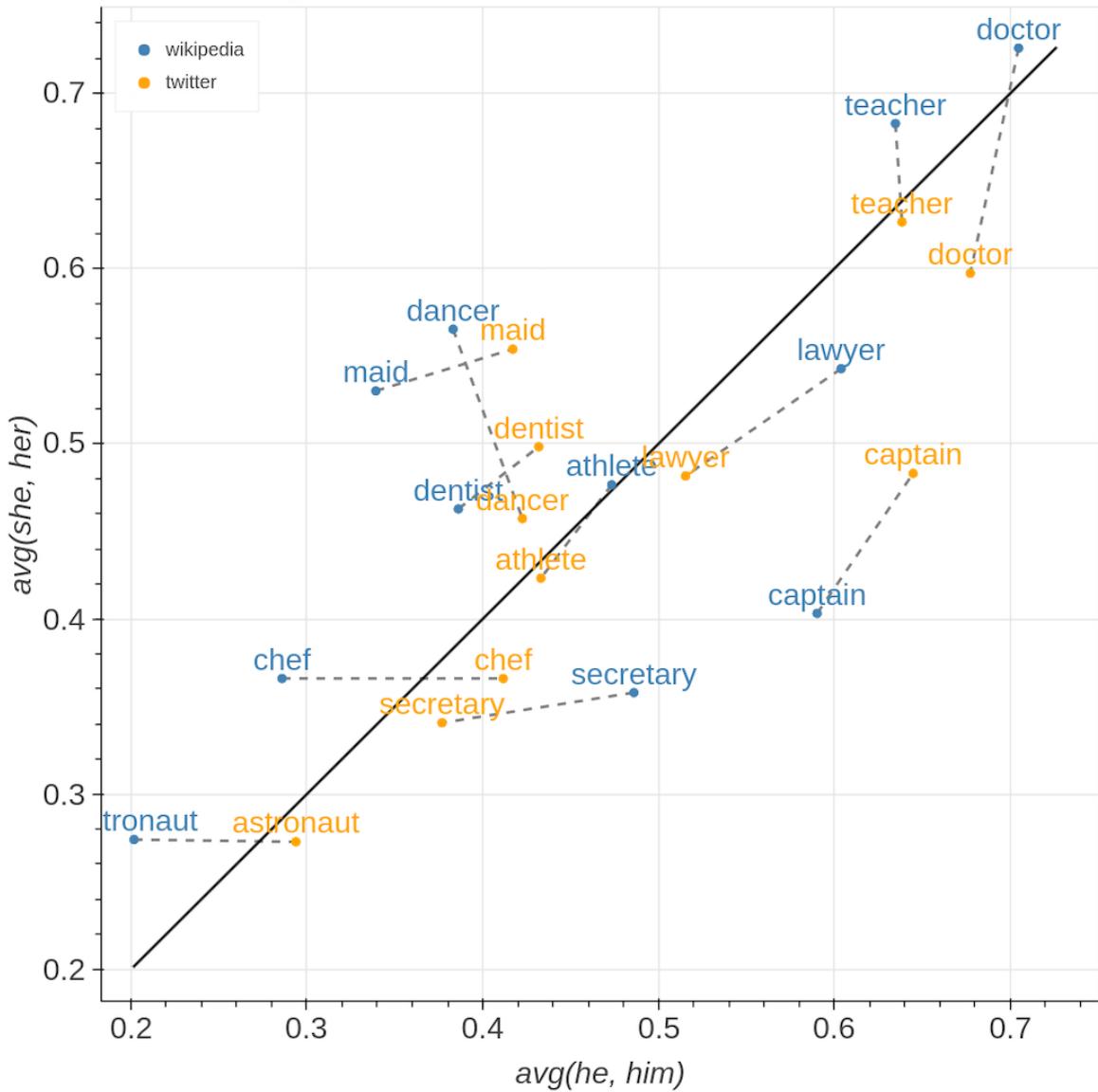


Figure 13: Professions plotted on “male” and “female” axes in *Wikpedia* and *Twitter* embeddings.

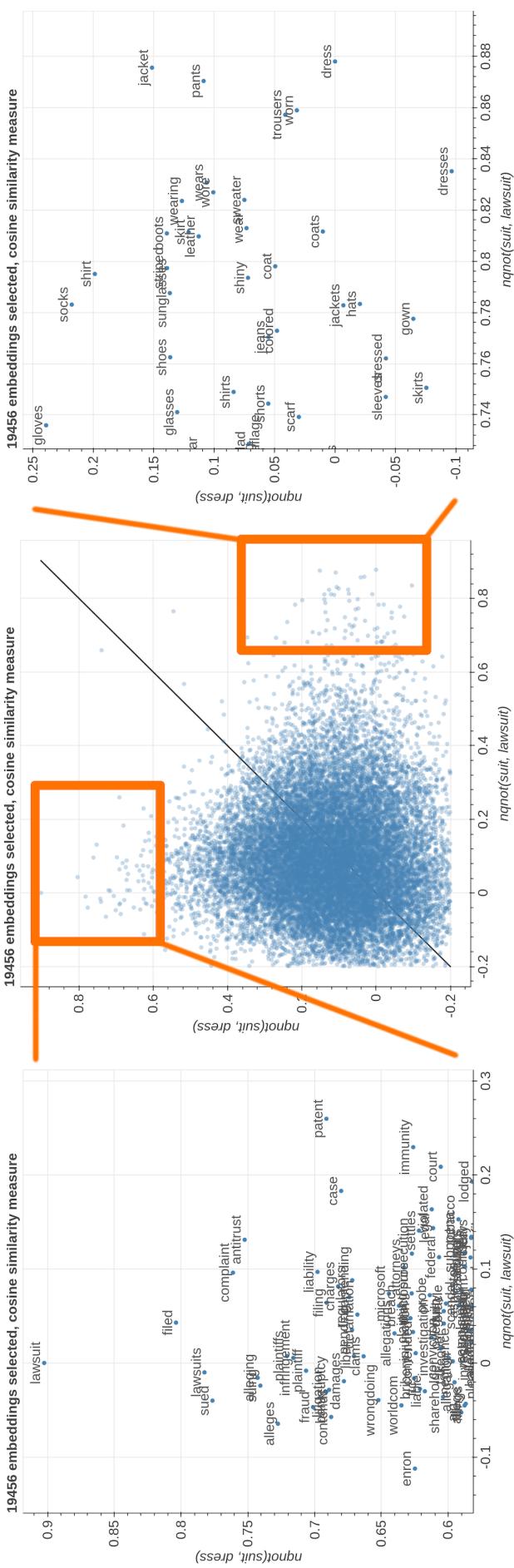


Figure 14: Plot of embeddings in Wikipedia with *suit* negated with respect to *lawsuit* and *dress* respectively as axes.

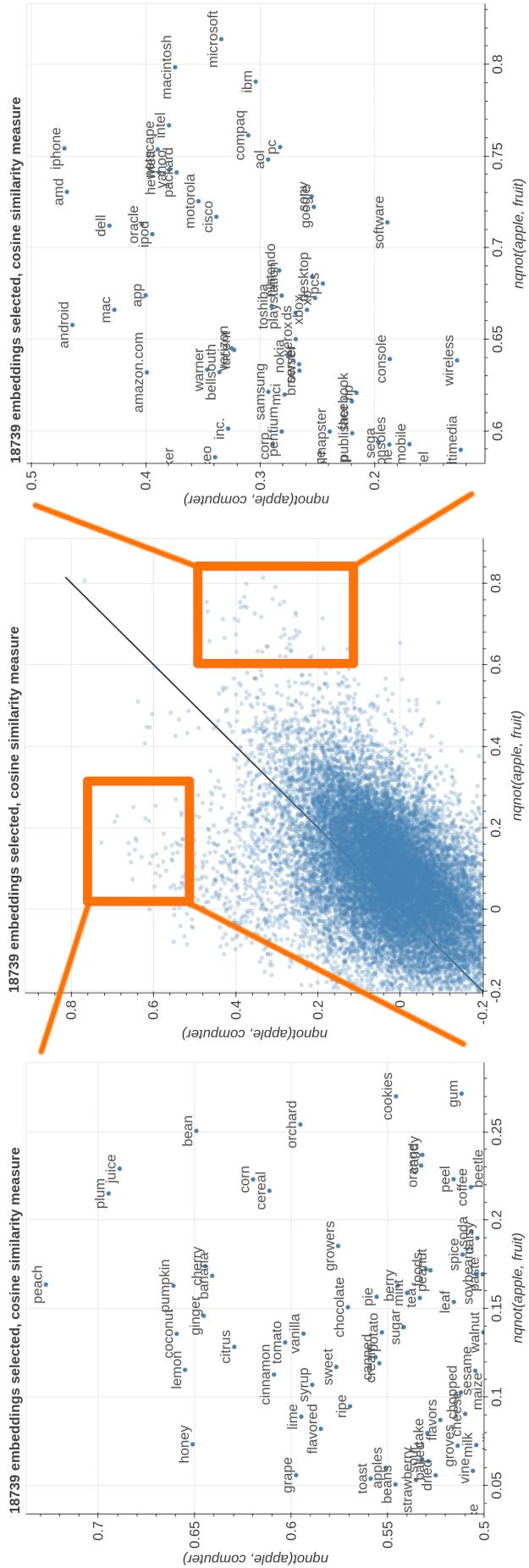


Figure 15: Plot of embeddings in Wikipedia with *apple* negated with respect to *fruit* and *computer* respectively as axes.

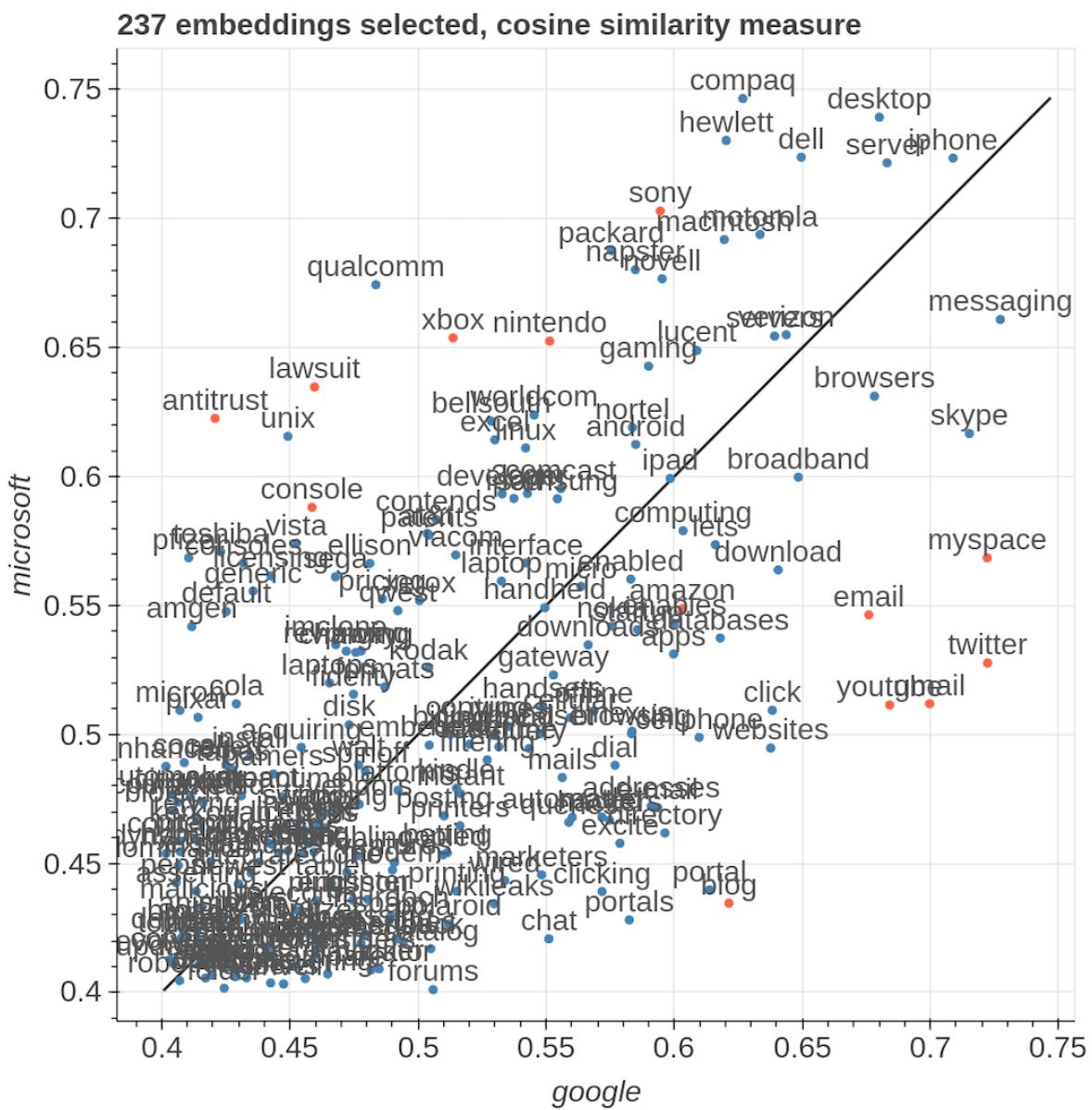


Figure 16: Fine-grained comparison of the subspace on the axis *google* and *microsoft* in *Wikipedia*.

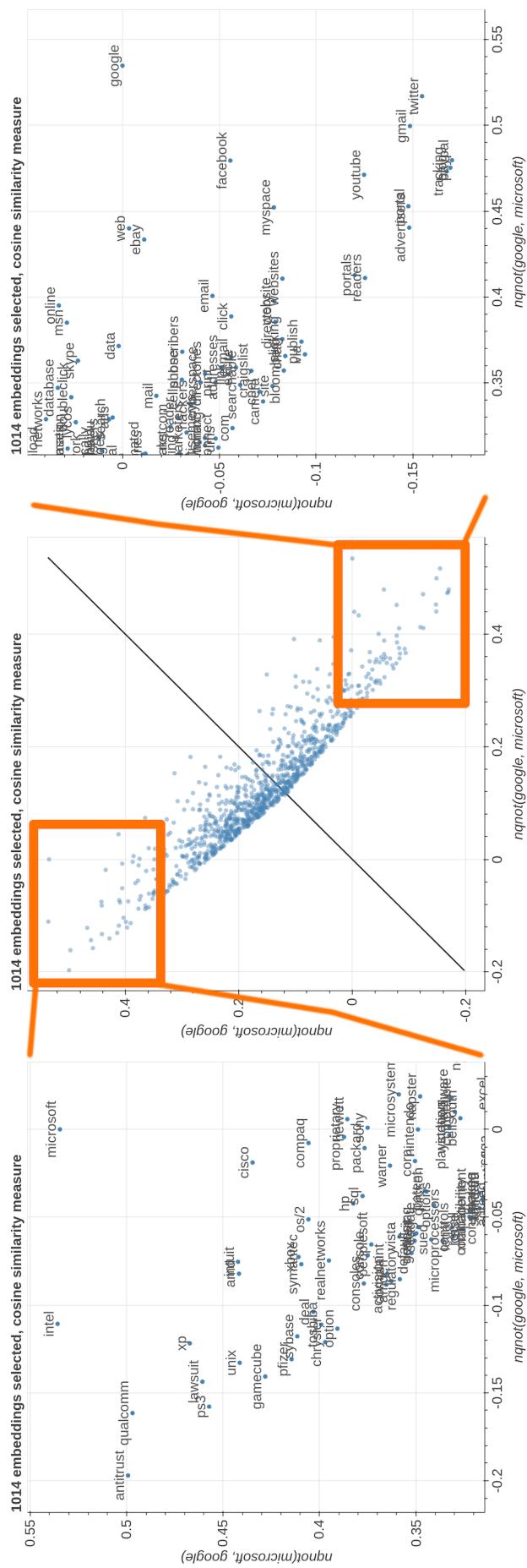


Figure 17: Fine-grained comparison of the subspace on the axis  $nqnot(google, microsoft)$  and  $nqnot(microsoft, google)$  in Wikipedia.

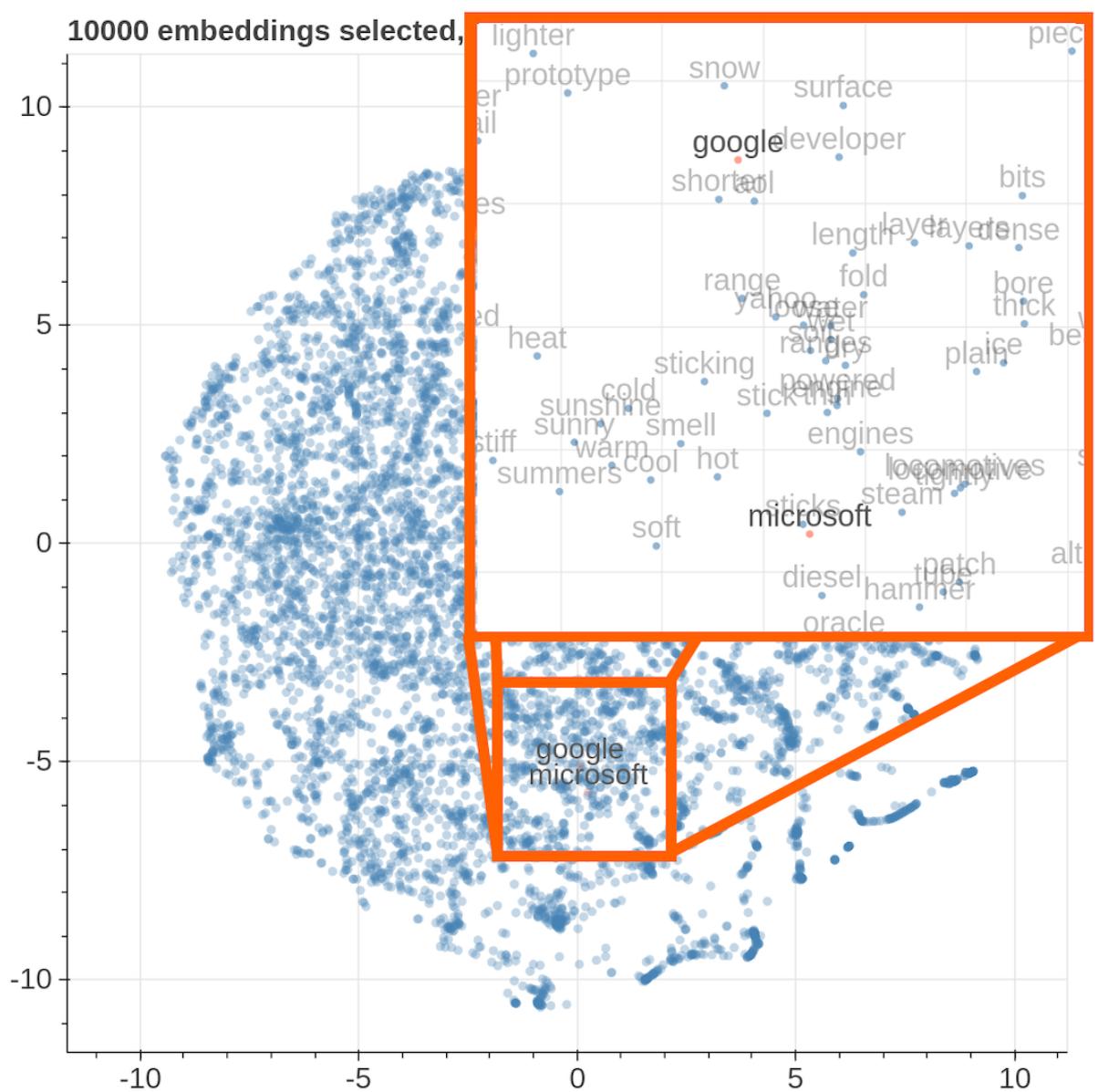


Figure 18: t-SNE visualization of *google* and *microsoft* in *Wikipedia*.

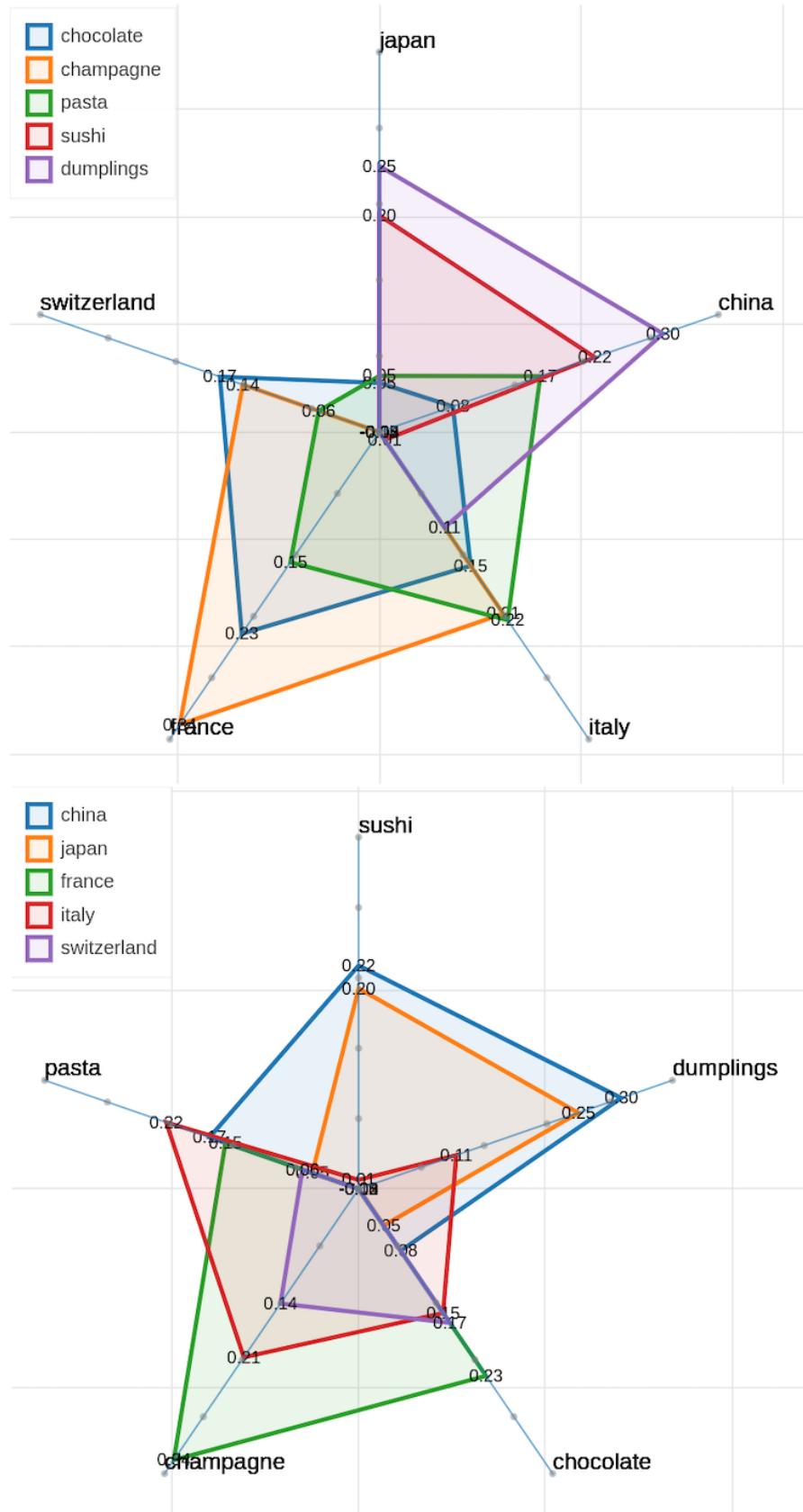


Figure 19: Two polar view of countries and foods in *Wikipedia*.

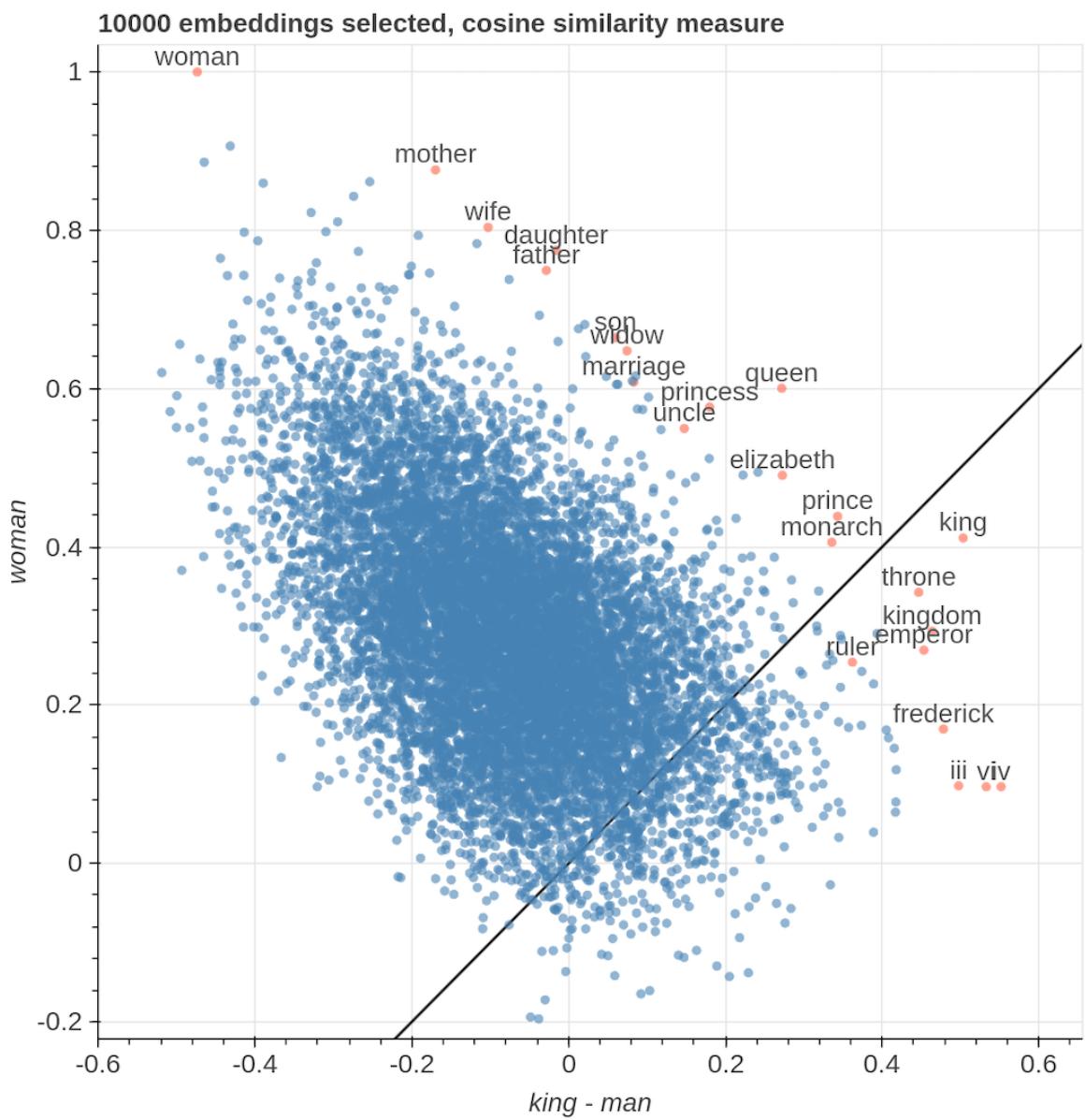


Figure 20: king-man vs woman

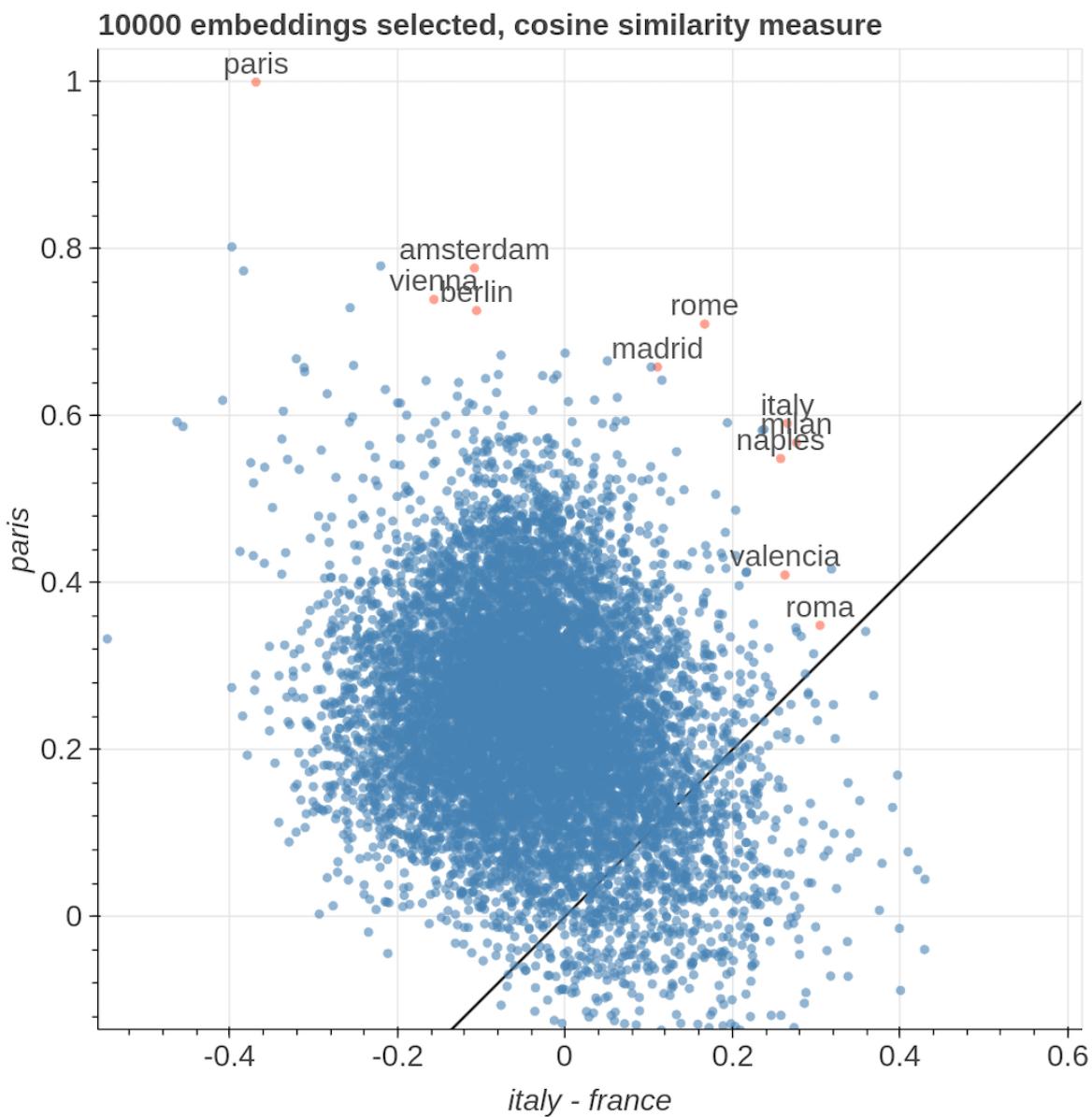


Figure 21: italy-france vs paris

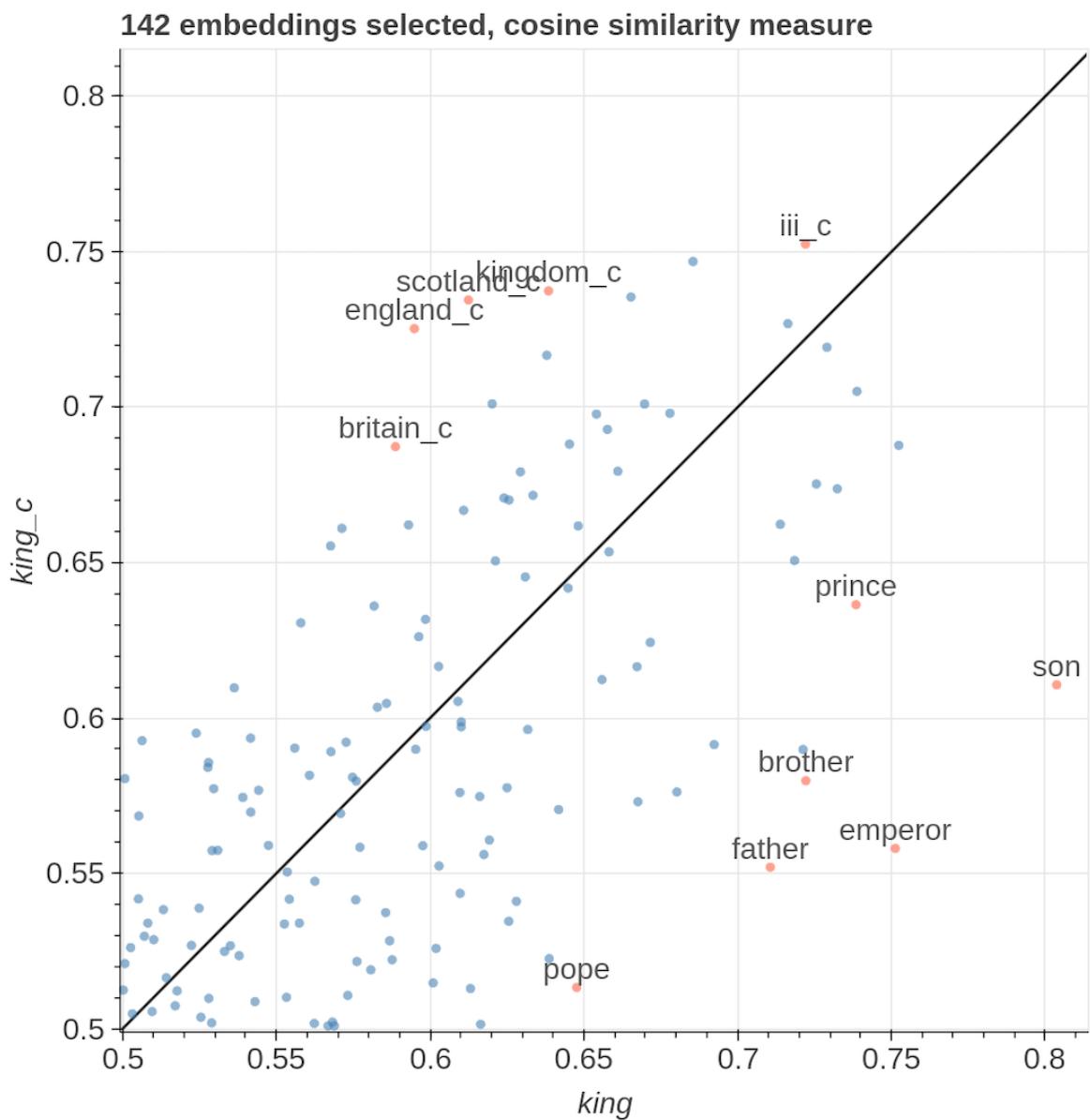


Figure 22: paradigmatic vs syntagmatic

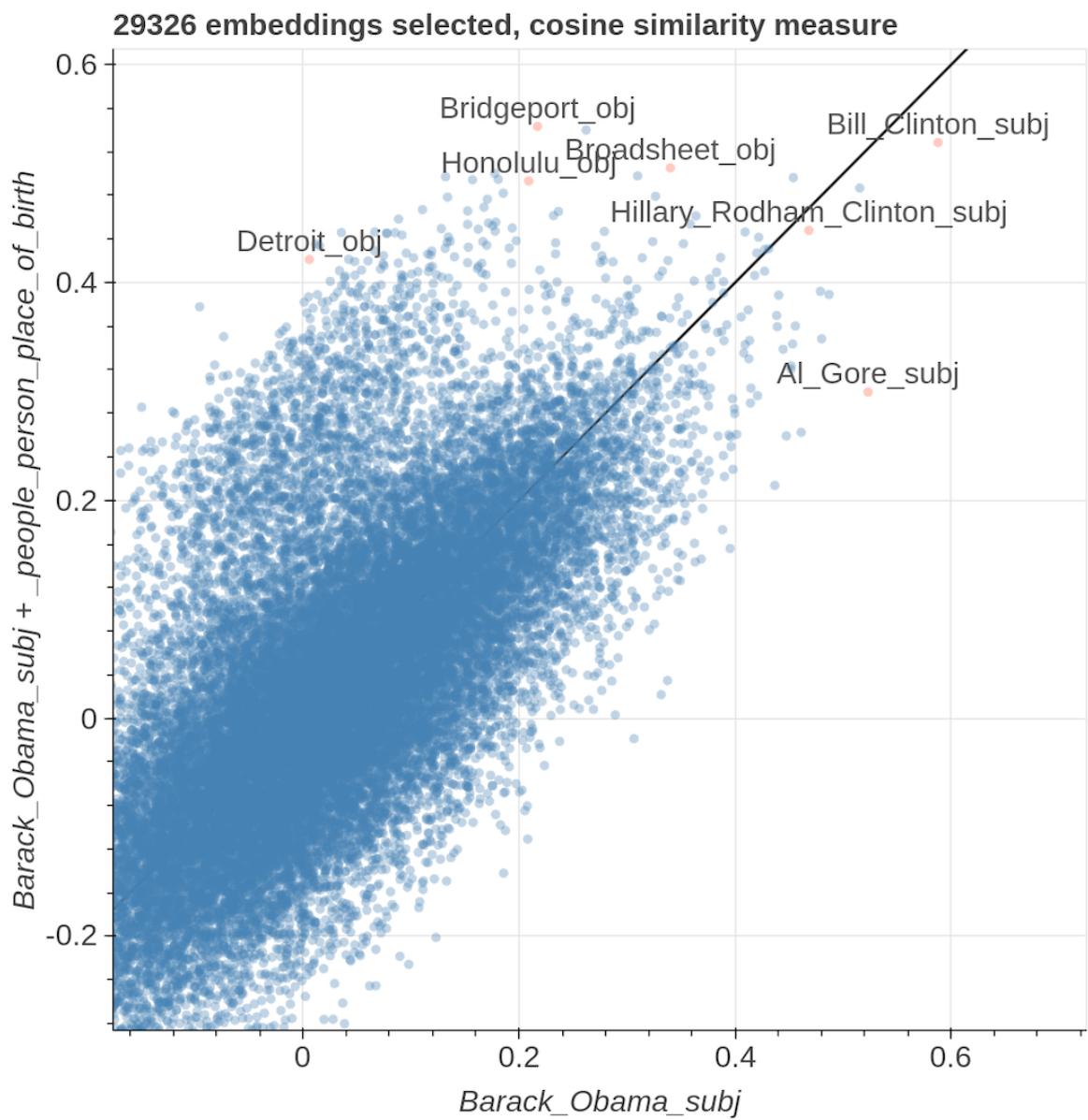


Figure 23: knowledge bases