Word embeddings analogies and paraphrases: proofs and open problems

•••

Mircea Petrache – UC Chile
ReLeLa, 11 January 2023

CENIA CENTRO NACIONAL DE INTELIGENCIA ARTIFICIAL
FACULTAD DE MATEMÁTICAS PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
Instituto de Ingeniería Matemática y Computacional UC

Plan of talk:

1. Structure(s) of language
2. Word2vec, GloVe background (we focus on W2V)
3. Explanations about linear analogies in Word2Vec
4. Debiasing or “lipstick on pigs”?
5. (Hierarchies – is there a hyperbolic structure?)

- Questions/discussions/ideas

Some principles (later translated to math)

- Firth (1957): the meaning of a word is defined by "***the company it keeps***".
- Languages have structure. Idea 1: ***Zipf's law.*** $f(r) \propto \frac{1}{r^{\alpha}}$

  [Nice 2014 survey and experiments to test conjectures, focusing on language: link]

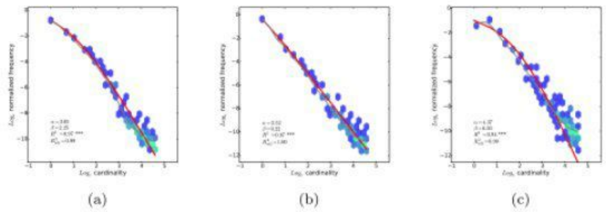


Figure 3: Power law frequencies for number words ("one", "two", "three", etc.) in English (a), Russian (b) and Italian (c) using data from Google (Lin et al., 2012). Note that here the x-axis is ordered by cardinality, *not* frequency rank, although these two coincide. Additionally, decades ("ten", "twenty", "thirty", etc.) were removed from this analysis due to unusually high frequency from their approximate usage. Here and in all plots the red line is the fit of (2) and the gray line is a LOESS.

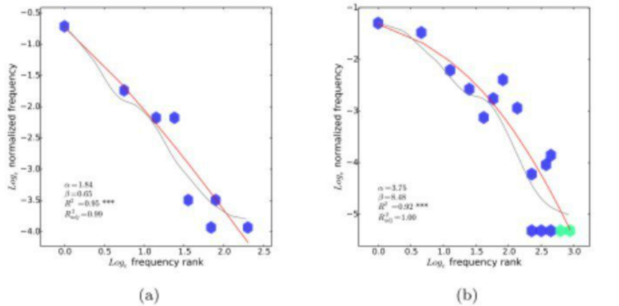
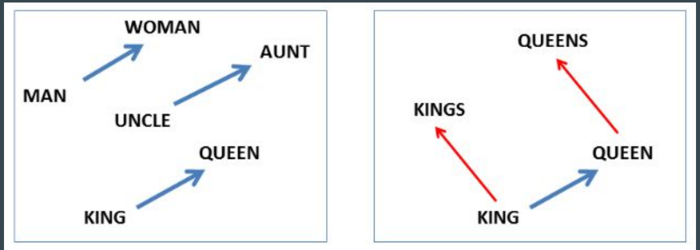
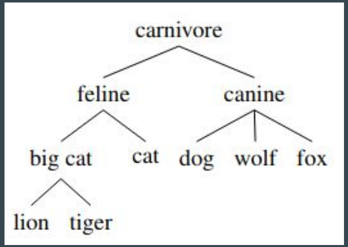


Figure 4: Distributions for taboo words for (a) sex (gerunds) and (b) feces.

Some principles (later translated to math)

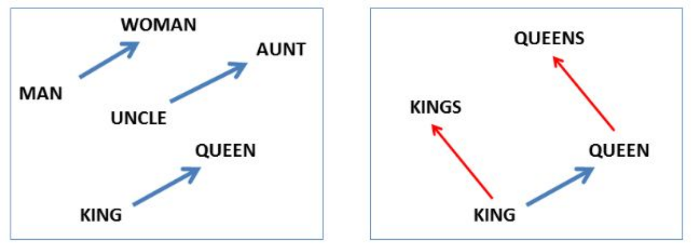
- Firth (1957): the meaning of a word is defined by "***the company it keeps***".
- Languages have structure: ***Zipf's law.*** $f(r) \propto \frac{1}{r^\alpha}$
- Geometry:
  - Spatial-like structure (analogies and more)
  - Hierarchical structure (entailment)





Text vectorization: linear algebra gives analogies

- Word2Vec and others – learn context-dependent probab.
- We get a dictionary-sized vector for each word.
- The result works remarkably like euclidean space !!

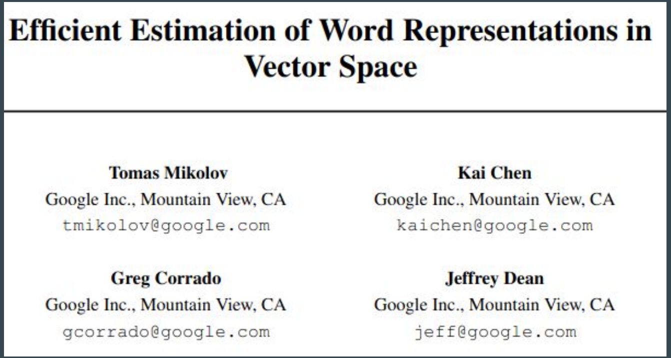


1. How far does this go?

2. What is the principle/theory behind it?

Mikolov et al. – Word2vec and Skip-gram with neg. sampling (SGNG)

Word2Vec (2013) link

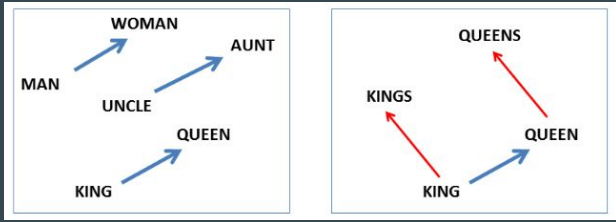

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com



Linguistic similarity (2013) link

Mikolov et al. – Word2vec and Skip-gram with neg. sampling (SGNG)

Skip-Gram assumes that the conditional probability of each possible set of words in a window around a context word c factorizes as the product of the respective conditional probabilities:

$$p(w_{-\Delta}, \ldots, w_{\Delta} | c) = \prod_{\substack{\delta=-\Delta \\ \delta \neq 0}}^{\Delta} p(w_{\delta} | c).$$

Average log probability

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

$$J = -\sum_{\substack{i \in \text{corpus} \\ j \in \text{context}(i)}} \log Q_{ij}$$

$$Q_{ij} = \frac{\exp(w_i^T \tilde{w}_j)}{\sum_{k=1}^{V} \exp(w_i^T \tilde{w}_k)}$$

(Word-context prob. Q_{ij} → softmax of vectors)

SGNG: replace log(Q_{ij}) by adding k more ***negative samples*** from (empirical) noise:

$$\log \sigma({v'_{w_O}}^{\top} v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^{\top} v_{w_I}) \right]$$

[Paper: Distributed representations (2013) link]

GloVe

GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning
Computer Science Department, Stanford University, Stanford, CA 94305
jpennin@stanford.edu, richard@socher.org, manning@stanford.edu

GloVe (2014) link: "local context windows → global co-occurrence counts"

$$P_{ij} = P(j|i) = \frac{\sharp\{w_j \text{ in context } w_i\}}{\sharp\{w_i\}} = \frac{X_{ij}}{X_i}$$

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

General form to start with..

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}},$$

Imposing linearity..

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

Imposing invariance to relabeling..

All these allow final choice F=exp, and we can set

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$$

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

History: GloVe

GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning
Computer Science Department, Stanford University, Stanford, CA 94305
jpennin@stanford.edu, richard@socher.org, manning@stanford.edu

GloVe (2014) link: "local context windows → global co-occurrence counts"

$$P_{ij} = P(j|i) = \frac{\sharp\{w_j \text{ in context } w_i\}}{\sharp\{w_i\}} = \frac{X_{ij}}{X_i}$$

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

General form to start with..

$$F\left((w_i - w \ldots\right.$$

$$F\left((w_i - w_j)^T \ldots\right.$$

Skip-gram obj. f. in this notation:

$$J = -\sum_{i=1}^{V} X_i \sum_{j=1}^{V} P_{ij} \log Q_{ij} = \sum_{i=1}^{V} X_i H(P_i, Q_i)$$

All these allow final choice F=exp, and we can set

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$$

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

History: Implicit factorization (2014) link

Neural Word Embedding as Implicit Matrix Factorization

Omer Levy
Department of Computer Science
Bar-Ilan University
omerlevy@gmail.com

Yoav Goldberg
Department of Computer Science
Bar-Ilan University
yoav.goldberg@gmail.com

SGNG loss in another notation:

$$\ell = \sum_{w \in V_W} \sum_{c \in V_C} \#(w,c) \left(\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D} \left[\log \sigma(-\vec{w} \cdot \vec{c}_N)\right]\right)$$

Now for each (w,c) we optimize (try opt. in $x = \vec{w} \cdot \vec{c}$)

$$\ell(w,c) = \#(w,c) \log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \#(w) \cdot \frac{\#(c)}{|D|} \log \sigma(-\vec{w} \cdot \vec{c})$$

Obtain this!

$$\vec{w} \cdot \vec{c} = \log \left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)} \cdot \frac{1}{k} \right) = \log \left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)} \right) - \log k$$

$$M_{ij}^{\text{SGNS}} = W_i \cdot C_j = \vec{w}_i \cdot \vec{c}_j = PMI(w_i, c_j) - \log k$$

Message: ***Pointwise mutual information matrix* M is factorized by SGNG***

(* : shifted)

Explaining analogy – Arora et al. 2016 (link)

RAND-WALK: A latent variable model approach to word embeddings

Sanjeev Arora Yuanzhi Li Yingyu Liang Tengyu Ma Andrej Risteski *

PMI matrix is found to be closely approximated by a low rank matrix: there exist word vectors in say 300 dimensions, which is much smaller than the number of words in the dictionary, such that

$$\langle v_w, v_{w'} \rangle \approx \mathrm{PMI}(w, w') \tag{1.1}$$

They obtain this with error bounds, assuming some modelling ansatz on the data, such as $\Pr[w \text{ emitted at time } t \mid c_t] \propto \exp(\langle c_t, v_w \rangle)$

Explaining analogy – Gittens et al. 2017 (link)

Skip-Gram – Zipf + Uniform = Vector Additivity

Alex Gittens
Dept. of Computer Science
Rensselaer Polytechnic Institute
gittea@rpi.edu

Dimitris Achlioptas
Dept. of Computer Science
UC Santa Cruz
optas@soe.ucsc.edu

Michael W. Mahoney
ICSI and Dept. of Statistics
UC Berkeley
mmahoney@stat.berkeley.edu

A natural way of capturing the compositionality of words is to say that the *set* of context words $c_1, \ldots, c_m$ has the same meaning as the single word c if for every other word w,

$$p(w|c_1, \ldots, c_m) = p(w|c) \ .$$

→ Paraphrase for C: $\arg\min_{c \in V} \mathrm{D_{KL}}(p(\cdot|C) \,|\, p(\cdot|c))$

A1. For every word c, there exists Z_c such that for every word w,

$$p(w|c) = \frac{1}{Z_c} \exp(\mathbf{u}_c^T \mathbf{v}_w) \ . \qquad (5)$$

A2. For every set of words $C = \{c_1, c_2, \ldots, c_m\}$, there exists Z_C such that for every word w,

$$p(w|C) = \frac{p(w)^{1-m}}{Z_C} \prod_{i=1}^{m} p(w|c_i) \ . \qquad (6)$$

Theorem 1. *In every word model that satisfies A1 and A2, for every set of words $C = \{c_1, \ldots, c_m\}$, any paraphrase c of C satisfies*

$$\sum_{w \in V} p(w|c)\mathbf{v}_w = \sum_{w \in V} p(w|C)\mathbf{v}_w \ . \qquad (7)$$

Theorem 2. *In every word model that satisfies A1, A2, and where $p(w) = 1/|V|$ for every $w \in V$, the paraphrase of $C = \{c_1, \ldots, c_m\}$ is*

$$\mathbf{u}_1 + \ldots + \mathbf{u}_m \ .$$

Zipf law says this is false!

- "if we pre-manipulate words to make Zipf weaker, we'll get better additivity"

Explaining analogy 2019 (link)

Analogies Explained: Towards Understanding Word Embeddings

Carl Allen [1] Timothy Hospedales [1]

They remove "shift" in the PMI factorization.

$$\mathbf{w}_i^\top \mathbf{c}_j = \mathrm{PMI}(w_i, c_j) \quad \text{or} \quad \mathbf{W}^\top \mathbf{C} = \mathbf{PMI}$$

A1. $\mathbf{C}$ *has full row rank.*

A2. *Letting* $\mathbf{M}_k$ *denote the* k^{th} *column of factored matrix* $\mathbf{M} \in \mathbb{R}^{n\times n}$, *the projection* $f : \mathbb{R}^n \to \mathbb{R}^d$, $f(\mathbf{M}_i) = \mathbf{w}_i$ *is approximately homomorphic with respect to addition, i.e.* $f(\mathbf{M}_i + \mathbf{M}_j) \approx f(\mathbf{M}_i) + f(\mathbf{M}_j)$.

A3. $p(\mathcal{W}) > 0, \ \forall \mathcal{W} \subseteq \mathcal{E}, |\mathcal{W}| < l$,

where (throughout) "$|\mathcal{W}| < l$" means $|\mathcal{W}|$ *sufficiently less* than l.

Lemma 1. *For any word* $w_* \in \mathcal{E}$ *and word set* $\mathcal{W} \subseteq \mathcal{E}$, $|\mathcal{W}| < l$:

$$\mathrm{PMI}_* = \sum_{w_i \in \mathcal{W}} \mathrm{PMI}_i + \boldsymbol{\rho}^{\mathcal{W}, w_*} + \boldsymbol{\sigma}^{\mathcal{W}} - \tau^{\mathcal{W}} \mathbf{1}, \qquad (5)$$

where $\mathrm{PMI}_\bullet$ *is the column of* $\mathbf{PMI}$ *corresponding to* $w_\bullet \in \mathcal{E}$, $\mathbf{1} \in \mathbb{R}^n$ *is a vector of 1s, and error terms* $\boldsymbol{\sigma}_j^{\mathcal{W}} = \log \frac{p(\mathcal{W}|c_j)}{\prod_i p(w_i|c_j)}$ *and* $\tau^{\mathcal{W}} = \log \frac{p(\mathcal{W})}{\prod_i p(w_i)}$.

Theorem 1 (Paraphrase). *For any word* $w_* \in \mathcal{E}$ *and word set* $\mathcal{W} \subseteq \mathcal{E}$, $|\mathcal{W}| < l$:

$$\mathbf{w}_* = \mathbf{w}_{\mathcal{W}} + \mathbf{C}^\dagger (\boldsymbol{\rho}^{\mathcal{W}, w_*} + \boldsymbol{\sigma}^{\mathcal{W}} - \tau^{\mathcal{W}} \mathbf{1}), \qquad (6)$$

where $\mathbf{w}_{\mathcal{W}} = \sum_{w_i \in \mathcal{W}} \mathbf{w}_i$.

Proof. Multiply (5) by $\mathbf{C}^\dagger$. □

Paraphrase error:

$$\boldsymbol{\rho}_j^{\mathcal{W}, \mathcal{W}_*} = \log \frac{p(c_j|\mathcal{W}_*)}{p(c_j|\mathcal{W})}, \ c_j \in \mathcal{E}$$

$$\mathbf{x} = \mathbf{p}^i + \mathbf{p}^j = \log \frac{p(\mathcal{E}|w_i)}{p(\mathcal{E})} + \log \frac{p(\mathcal{E}|w_j)}{p(\mathcal{E})}$$

$$= \underbrace{\log \frac{p(\mathcal{E}|w_i, w_j)}{p(\mathcal{E})}}_{\mathbf{p}^{i,j}} - \underbrace{\log \frac{p(w_i, w_j|\mathcal{E})}{p(w_i|\mathcal{E}) p(w_j|\mathcal{E})}}_{\boldsymbol{\sigma}^{ij}} + \underbrace{\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}_{\tau^{ij}} = \mathbf{p}^{i,j} - \boldsymbol{\sigma}^{ij} + \tau^{ij}\mathbf{1},$$

Explaining analogy 2019 (link)

Analogies Explained: Towards Understanding Word Embeddings

Carl Allen[1] Timothy Hospedales[1]

They remove "shift" in the PMI factorization.

$$\mathbf{w}_i^\top \mathbf{c}_j = \mathrm{PMI}(w_i, c_j) \quad \text{or} \quad \mathbf{W}^\top \mathbf{C} = \mathbf{PMI}$$

A1. $\mathbf{C}$ *has full row rank.*

A2. *Letting* $\mathbf{M}_k$ *denote the* k^{th} *column of factored matrix* $\mathbf{M} \in \mathbb{R}^{n\times n}$, *the projection* $f:\mathbb{R}^n \to \mathbb{R}^d$, $f(\mathbf{M}_i) = \mathbf{w}_i$ *is approximately homomorphic with respect to addition, i.e.* $f(\mathbf{M}_i + \mathbf{M}_j) \approx f(\mathbf{M}_i) + f(\mathbf{M}_j)$.

A3. $p(\mathcal{W}) > 0, \ \forall \mathcal{W} \subseteq \mathcal{E}, |\mathcal{W}| < l$,

where (throughout) "$|\mathcal{W}| < l$" means $|\mathcal{W}|$ *sufficiently less* than l.

Paraphrase error defined as:

$$\boldsymbol{\rho}_j^{\mathcal{W},\mathcal{W}_*} = \log \frac{p(c_j|\mathcal{W}_*)}{p(c_j|\mathcal{W})}, \ c_j \in \mathcal{E}$$

Error to "linear" generalized paraphrase:

Theorem 2 (Generalised Paraphrase). *For any word sets* $\mathcal{W}, \mathcal{W}_* \subseteq \mathcal{E}, |\mathcal{W}|, |\mathcal{W}_*| < l$:

$$\mathbf{w}_{\mathcal{W}_*} = \mathbf{w}_{\mathcal{W}} + \mathbf{C}^\dagger(\boldsymbol{\rho}^{\mathcal{W},\mathcal{W}_*} + \boldsymbol{\sigma}^{\mathcal{W}} - \boldsymbol{\sigma}^{\mathcal{W}_*} - (\tau^{\mathcal{W}} - \tau^{\mathcal{W}_*})\mathbf{1}).$$

Proof. Multiply (10) by $\mathbf{C}^\dagger$. □

Note that $|\mathcal{W}_*| = 1$ recovers Lem 1 and Thm 1. With analogies in mind, we restate Thm 2 as:

Corollary 2.1. *For any words* $w_x, w_{x^*} \in \mathcal{E}$ *and word sets* $\mathcal{W}^+, \mathcal{W}^- \subseteq \mathcal{E}, |\mathcal{W}^+|, |\mathcal{W}^-| < l - 1$:

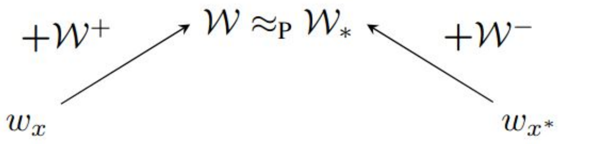
$$\mathbf{w}_{x^*} = \mathbf{w}_x + \mathbf{w}_{\mathcal{W}^+} - \mathbf{w}_{\mathcal{W}^-} + \mathbf{C}^\dagger(\boldsymbol{\rho}^{\mathcal{W},\mathcal{W}_*} + \boldsymbol{\sigma}^{\mathcal{W}} - \boldsymbol{\sigma}^{\mathcal{W}_*} - (\tau^{\mathcal{W}} - \tau^{\mathcal{W}_*})\mathbf{1}), \tag{11}$$

where $\mathcal{W} = \{w_x\} \cup \mathcal{W}^+$, $\mathcal{W}_* = \{w_{x^*}\} \cup \mathcal{W}^-$.

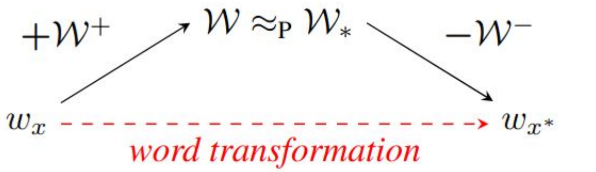
Proof. Set $\mathcal{W} = \{w_x\} \cup \mathcal{W}^+$, $\mathcal{W}_* = \{w_{x^*}\} \cup \mathcal{W}^-$ in Thm 2. □

Explaining analogy 2019 (link)

$$\text{“}w_a \text{ is to } w_{a^*} \text{ as } w_b \text{ is to } w_{b^*}\text{”} \iff \begin{matrix} w_a \xrightarrow[\mathcal{W}^-]{\mathcal{W}^+} w_{a^*} \\ \wedge \\ w_b \xrightarrow[\mathcal{W}^-]{\mathcal{W}^+} w_{b^*} \end{matrix} \iff \begin{matrix} \{w_a, \mathcal{W}^+\} \approx_{\text{P}} \{w_{a^*}, \mathcal{W}^-\} \\ \wedge \\ \{w_b, \mathcal{W}^+\} \approx_{\text{P}} \{w_{b^*}, \mathcal{W}^-\} \end{matrix} \implies \begin{matrix} \mathbf{w}_{a^*} - \mathbf{w}_a \\ \approx \\ \mathbf{w}_{b^*} - \mathbf{w}_b \end{matrix}$$

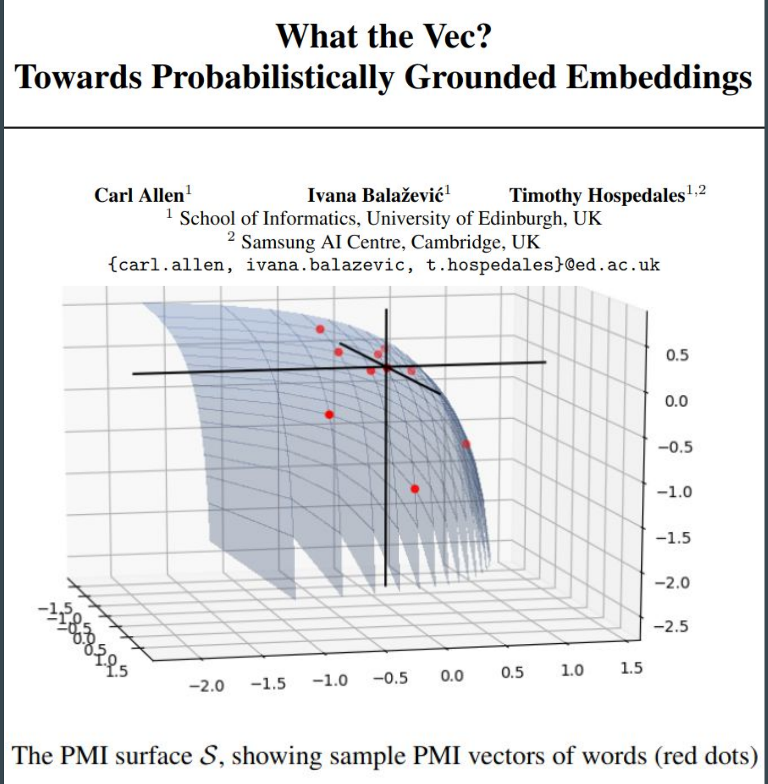


(a) Adding context to each of w_x and w_{x^*} to reach a paraphrase.



(b) Adding and subtracting context to *transform* w_x to w_{x^*}.

Explaining analogy 2 2019 (link2)


What the Vec?
Towards Probabilistically Grounded Embeddings

Carl Allen[1] **Ivana Balažević**[1] **Timothy Hospedales**[1,2]
[1] School of Informatics, University of Edinburgh, UK
[2] Samsung AI Centre, Cambridge, UK
{carl.allen, ivana.balazevic, t.hospedales}@ed.ac.uk

The PMI surface $\mathcal{S}$, showing sample PMI vectors of words (red dots)

Bolukbasi et al 2016 (link)

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

Tolga Bolukbasi[1], Kai-Wei Chang[2], James Zou[2], Venkatesh Saligrama[1,2], Adam Kalai[2]
[1]Boston University, 8 Saint Mary's Street, Boston, MA
[2]Microsoft Research New England, 1 Memorial Drive, Cambridge, MA
tolgab@bu.edu, kw@kwchang.net, jamesyzou@gmail.com, srv@bu.edu, adam.kalai@microsoft.com

$$\text{DirectBias}_c = \frac{1}{|N|} \sum_{w \in N} |\cos(\vec{w}, g)|^c$$

$$w = w_g + w_\perp$$

$$\beta(w, v) = \left(w \cdot v - \frac{w_\perp \cdot v_\perp}{\|w_\perp\|_2 \|v_\perp\|_2}\right) \Big/ w \cdot v$$

1)

$$\vec{w} := (\vec{w} - \vec{w}_B)/\|\vec{w} - \vec{w}_B\|.$$

$$\begin{aligned} \mu &:= \sum_{w \in E} w/|E| \\ \nu &:= \mu - \mu_B \\ \text{For each } w \in E,\quad \vec{w} &:= \nu + \sqrt{1 - \|\nu\|^2} \frac{\vec{w}_B - \mu_B}{\|\vec{w}_B - \mu_B\|} \end{aligned}$$

2) $$\min_T \|(TW)^T(TW) - W^T W\|_F^2 + \lambda \|(TN)^T(TB)\|_F^2$$

Zhao et al. 2018 (link)
Kaneko Bollegala 2019 (link)

Learning Gender-Neutral Word Embeddings

Jieyu Zhao **Yichao Zhou** **Zeyu Li** **Wei Wang** **Kai-Wei Chang**
University of California, Los Angeles
{jyzhao, yz, zyli, weiwang, kwchang}@cs.ucla.edu

Gender-preserving Debiasing for Pre-trained Word Embeddings

Masahiro Kaneko
Tokyo Metropolitan University, Japan
kaneko-masahiro@ed.tmu.ac.jp

Danushka Bollegala
University of Liverpool, UK
danushka@liverpool.ac.uk

$$J = J_G + \lambda_d J_D + \lambda_e J_E,$$

$$w = [w^{(a)}; w^{(g)}]. \quad w^{(a)} \in \mathbb{R}^{d-k} \quad w^{(g)} \in \mathbb{R}^k$$

$$J_G = \sum_{i,j=1}^{V} f(X_{i,j}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{i,j} \right)^2$$

→ usual GloVe objective

$$J_D^{L1} = - \left\| \sum_{w \in \Omega_M} w^{(g)} - \sum_{w \in \Omega_F} w^{(g)} \right\|_1$$

→ increase gap between male/female clouds (?)

$$J_D^{L2} = \sum_{w \in \Omega_M} \left\| \beta_1 \boldsymbol{e} - w^{(g)} \right\|_2^2 + \sum_{w \in \Omega_F} \left\| \beta_2 \boldsymbol{e} - w^{(g)} \right\|_2^2$$

where $\boldsymbol{e} \in \mathcal{R}^k$ is a vector of all ones. β_1 and β_2 can be arbitrary values, and we set them to be 1 and -1, respectively.

→ make gender part fixed (?)

$$J_E = \sum_{w \in \Omega_N} \left(v_g^T w^{(a)} \right)^2$$

$$v_g = \frac{1}{|\Omega'|} \sum_{(w_m, w_f) \in \Omega'} (w_m^{(a)} - w_f^{(a)}),$$

where Ω' is a set of predefined gender word pairs.

→ retain neutral words non-gender part

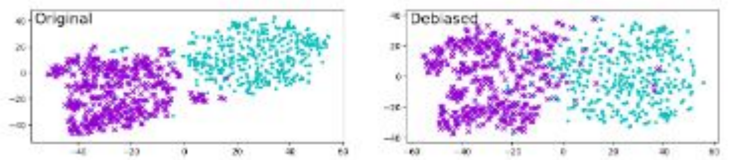
Gonen Goldberg 2019 (link)

Lipstick on a Pig:
Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them

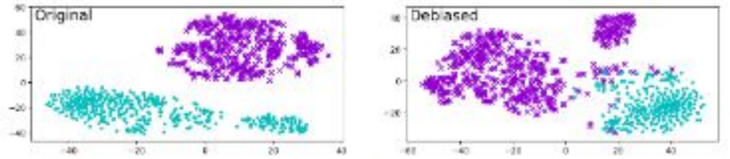
Hila Gonen[1] and **Yoav Goldberg**[1,2]
[1]Department of Computer Science, Bar-Ilan University
[2]Allen Institute for Artificial Intelligence
{hilagnn,yoav.goldberg}@gmail.com

Key observation:

- most word pairs maintain previous similarity
- words with a specific bias still grouped together
- Implicit bias remains



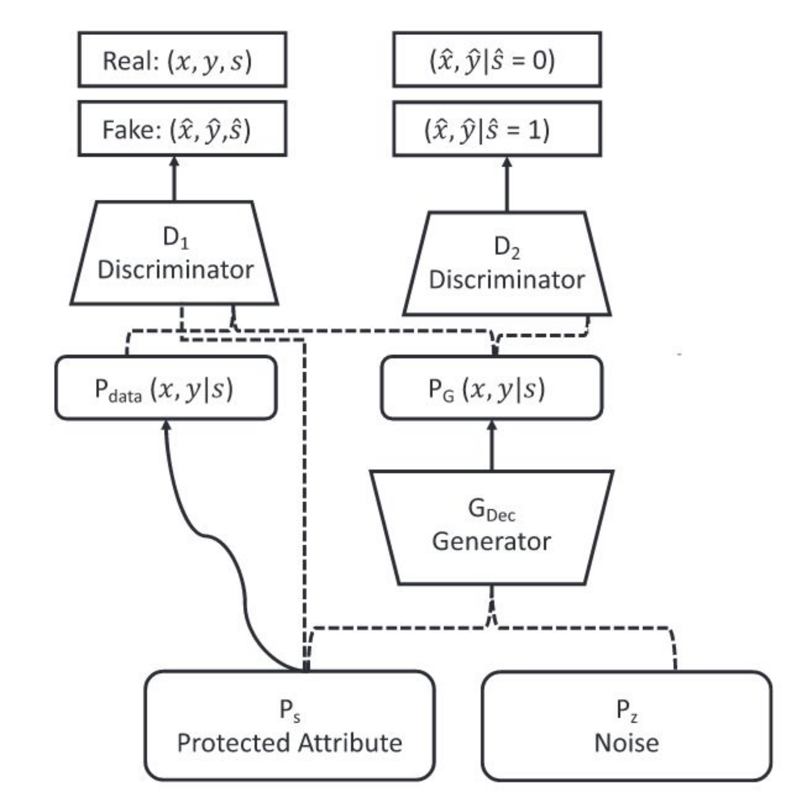
(a) Clustering for HARD-DEBIASED embedding, before (left hand-side) and after (right hand-side) debiasing.



(b) Clustering for GN-GLOVE embedding, before (left hand-side) and after (right hand-side) debiasing.

Figure 1: Clustering the 1,000 most biased words, before and after debiasing, for both models.

Xu et al. 2018 (link)



FairGAN: Fairness-aware Generative Adversarial Networks

Depeng Xu
University of Arkansas
depengxu@uark.edu

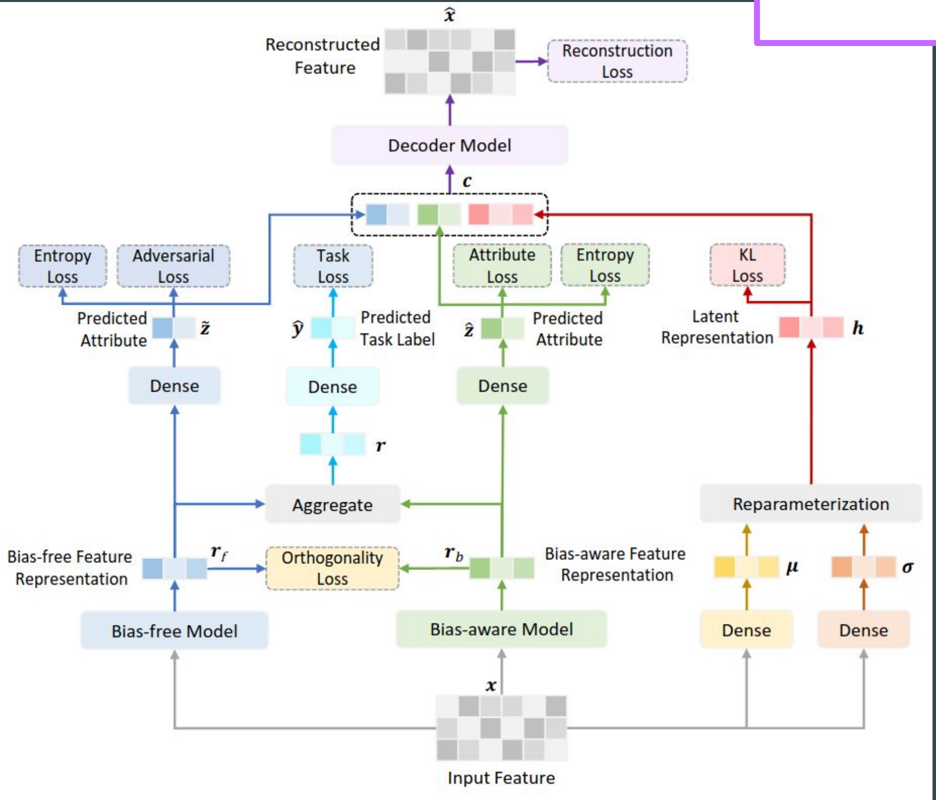
Shuhan Yuan
University of Arkansas
sy005@uark.edu

Lu Zhang
University of Arkansas
lz006@uark.edu

Xintao Wu
University of Arkansas
xintaowu@uark.edu

How is it not still a pig?

Wu et al. 2022 (link)




Semi-FairVAE: Semi-supervised Fair Representation Learning with Adversarial Variational Autoencoder

Chuhan Wu[1], Fangzhao Wu[2], Tao Qi[1], Yongfeng Huang[1]
[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084
[2]Microsoft Research Asia, Beijing 100080, China
{wuchuhan15,wufangzhao,taoqi.qt}@gmail.com,yfhuang@tsinghua.edu.cn

How is it not still a pig?

Hyperbolic GloVe: Tifrea et al. 2018 (link)

POINCARÉ GLOVE: HYPERBOLIC WORD EMBEDDINGS

Alexandru Țifrea*, **Gary Bécigneul***, **Octavian-Eugen Ganea***
Department of Computer Science
ETH Zürich, Switzerland
tifreaa@ethz.ch,{gary.becigneul,octavian.ganea}@inf.ethz.ch

Gaussian embedding for text: Vilnis McCallum 2015 (link)

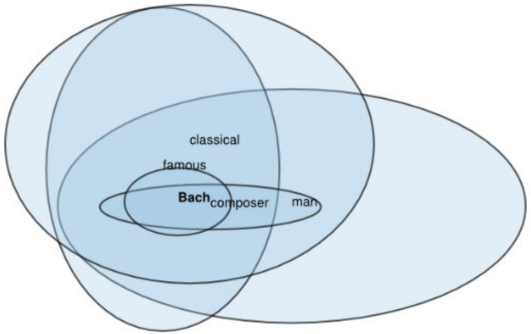


Figure 1: Learned diagonal variances, as used in evaluation (Section 6), for each word, with the first letter of each word indicating the position of its mean. We project onto generalized eigenvectors between the mixture means and variance of query word *Bach*. Nearby words to *Bach* are other composers e.g. *Mozart*, which lead to similar pictures.

WORD REPRESENTATIONS VIA GAUSSIAN EMBEDDING

Luke Vilnis, Andrew McCallum
School of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
luke@cs.umass.edu, mccallum@cs.umass.edu

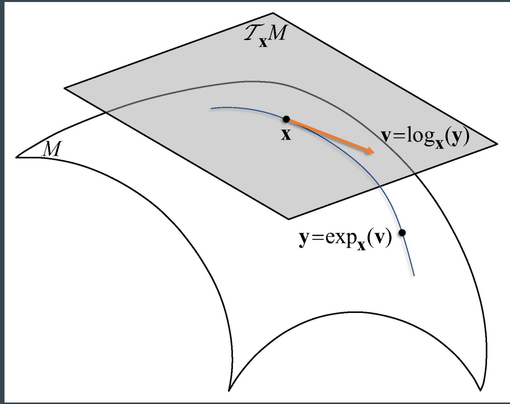
Entailment
→ Gaussian Fisher distance
→ Hyperbolic space distance

$$d_F\left(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma')\right) = \sqrt{\sum_{i=1}^{n} 2d_{\mathbb{H}^2}\left((\mu_i/\sqrt{2},\sigma_i),(\mu_i'/\sqrt{2},\sigma_i')\right)^2}$$

$$\mathrm{KL}(P(\theta+d\theta)||P(\theta)) = (1/2)\sum_{ij} g_{ij}\,d\theta^i d\theta^j + \mathcal{O}(\|d\theta\|^3)$$

Hyperbolic space: Accommodates well trees (euclidean has huge problem)

Hyperbolic Neural Networks (very sketchy)



NN operations transferred from tangent space

$$f^{\otimes_c}(\mathbf{x}) := \exp_{\mathbf{0}}^c(f(\log_{\mathbf{0}}^c(\mathbf{x})))$$

$$M^{\otimes_c}(\mathbf{x}) = (1/\sqrt{c})\tanh\left(\frac{\|\mathbf{Mx}\|}{\|\mathbf{x}\|}\tanh^{-1}(\sqrt{c}\|\mathbf{x}\|)\right)\frac{\mathbf{Mx}}{\|\mathbf{Mx}\|}$$

$$\mathbf{x} \oplus_c \mathbf{b} = \exp_{\mathbf{x}}^c(P_{\mathbf{0}\to\mathbf{x}}^c(\log_{\mathbf{0}}^c(\mathbf{b})))$$

Hyperbolic space: Accommodates well trees (euclidean has huge problem)