Sequential Modelling in Vector Space

<u>Benyou Wang</u>, Emanuele Di Buccio and Massimo Melucci

University of Padova, Italy

11th Italian Information Retrieval Workshop, September 13-15, 2021, Bari, Italy

Embed Discrete Objects in Vector Space

Two examples

- **Word embedding**
- User/item embedding

Learn implicit features that could be adaptively updated during training

Word Embedding

- **Prediction-based method [1,2]**
  - e.g., using neural networks to predict central/neighboring words

- **Count-based method [3]**
  - e.g., decompose PPMI matrices

[1] Bengio et.al. A Neural Probabilistic Language Model. JMLR 2003

[2] Mikolov et.al. Efficient Estimation of Word Representations in Vector Space. NIPS 2013.

[3] Pennington et.al. GloVe: Global Vectors for Word Representation. EMNLP 2014

Sequential aspects to model

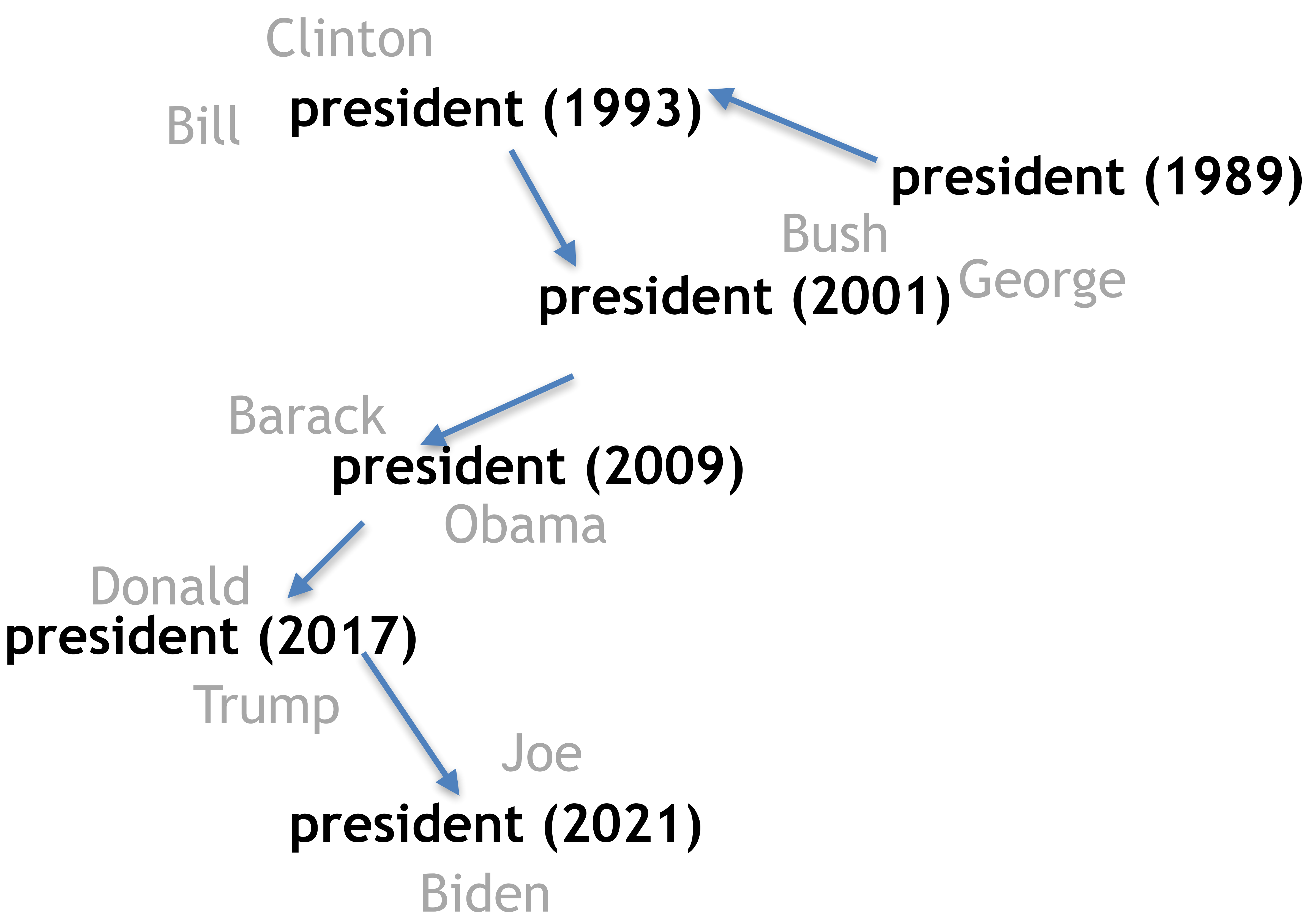
- Position
  - Encode word order in neural networks (e.g., Transformer [1]) [2]

- Temporal Evolution
  - Individual words may change their meaning over time
  - Existing solutions, e.g., Dynamic Word Embeddings

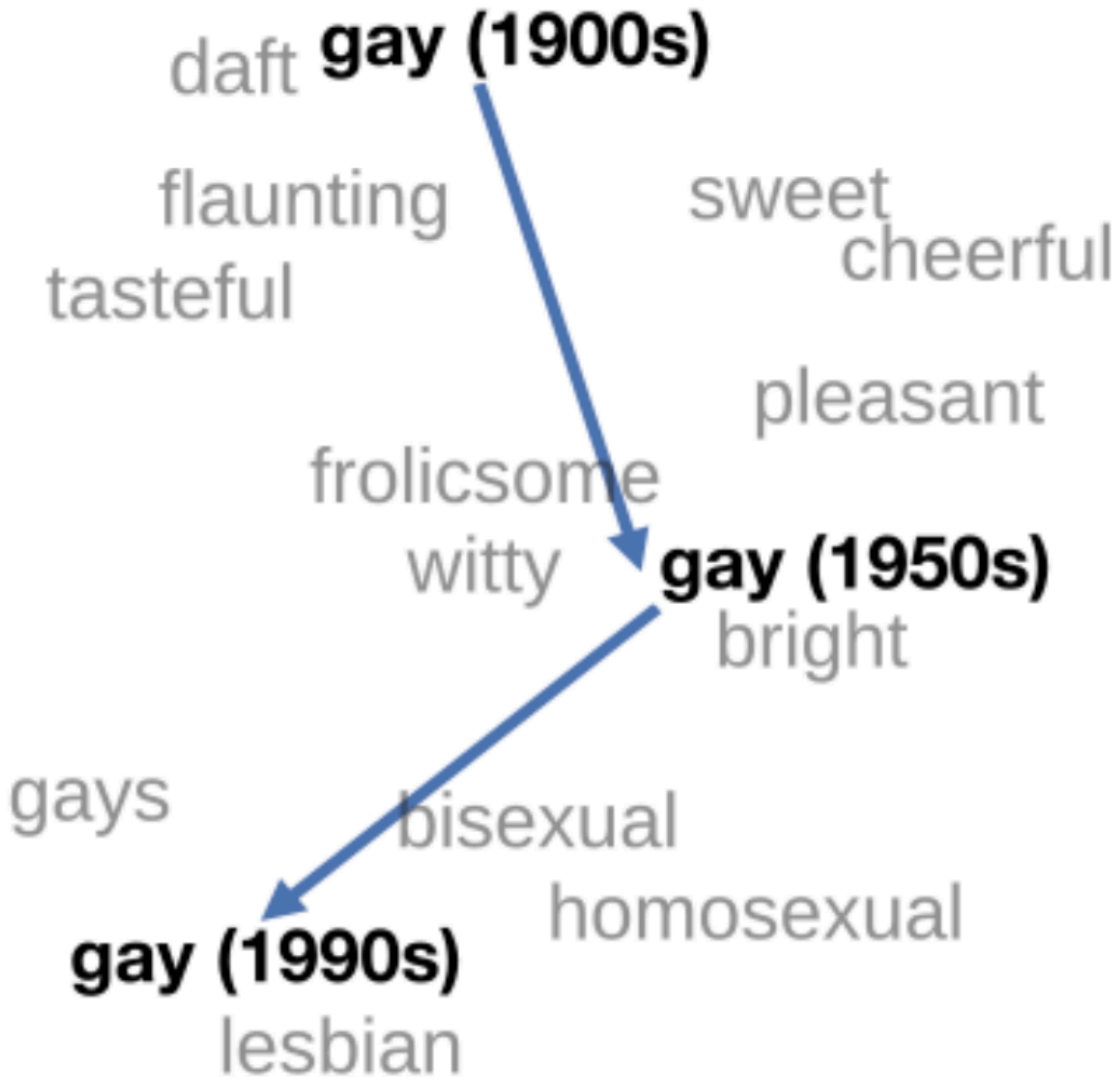
[1] Vaswani et.al. Attention is all you need, NIPS 2017

[2] Benyou Wang et.al. Encoding word order in complex embeddings

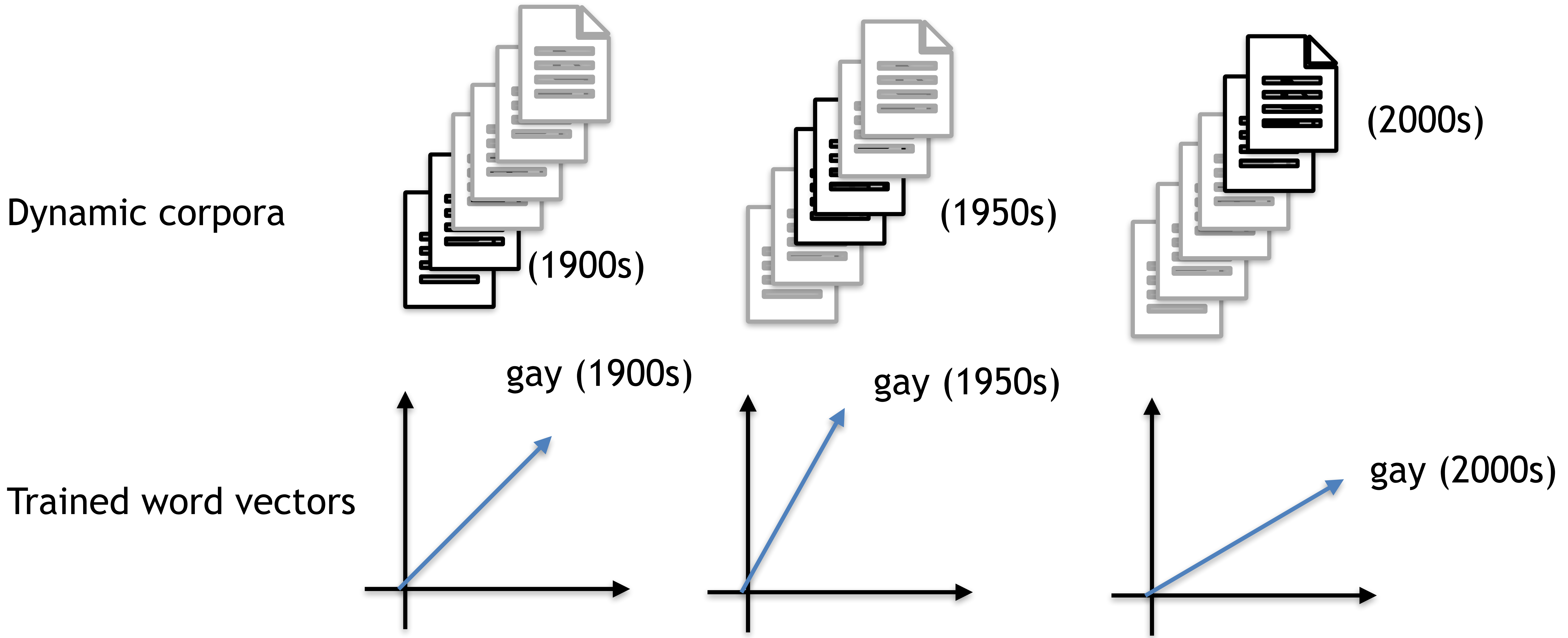
Example 1: short-term evolution



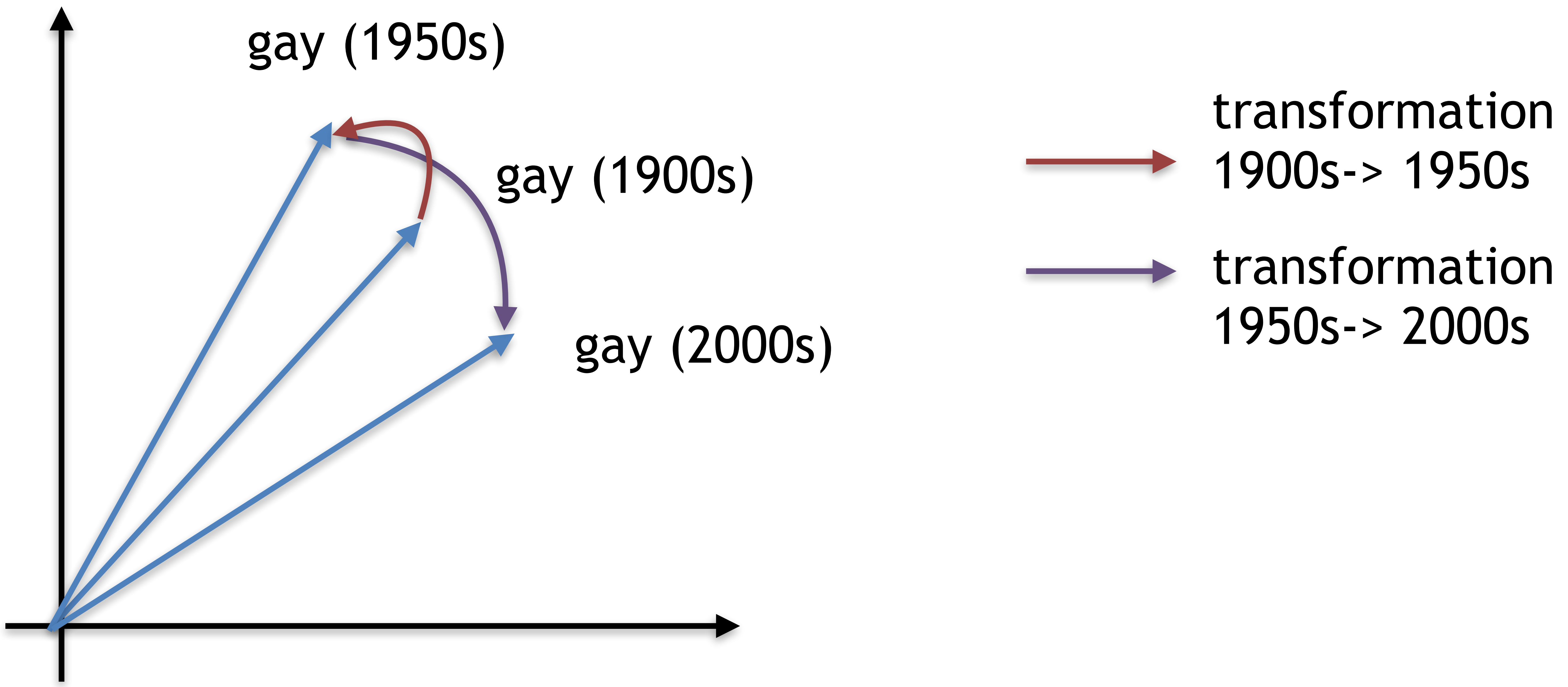
Example 2: long-term evolution



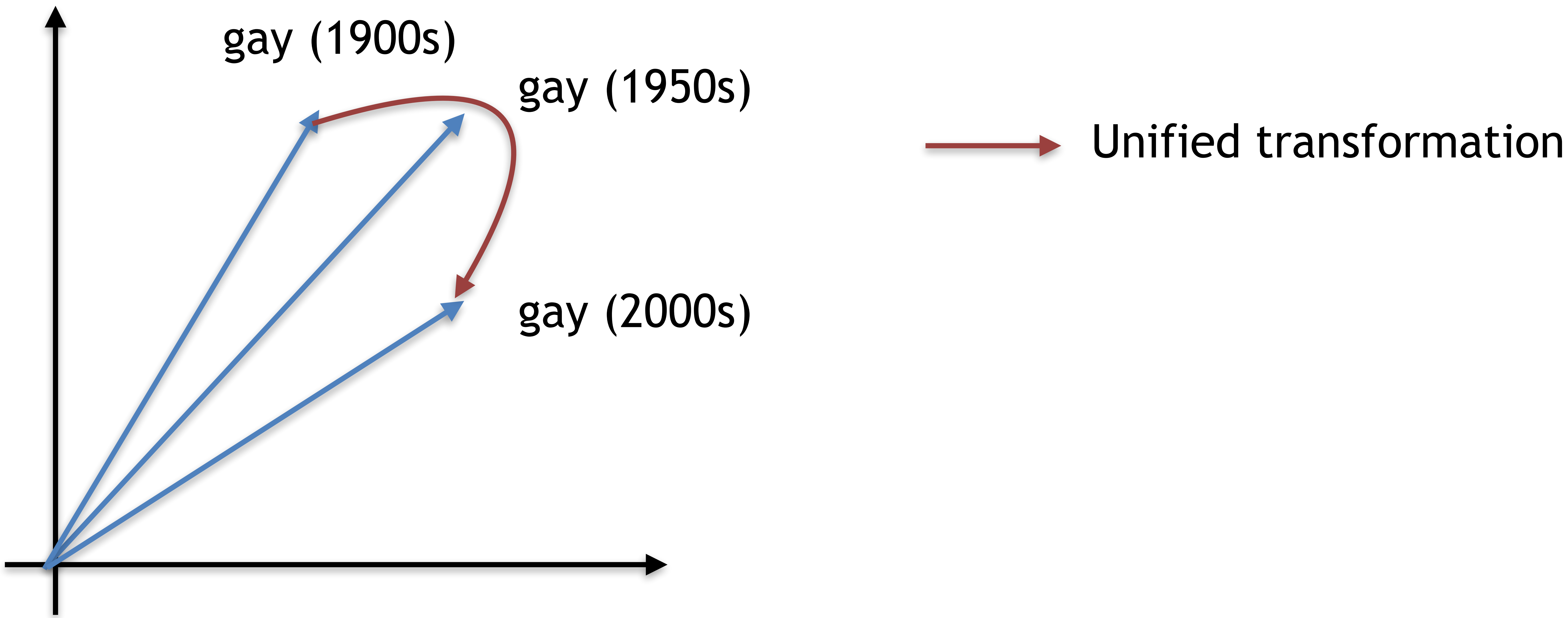
Train and Align Paradigm



Previous one-hop assumption



Our approach

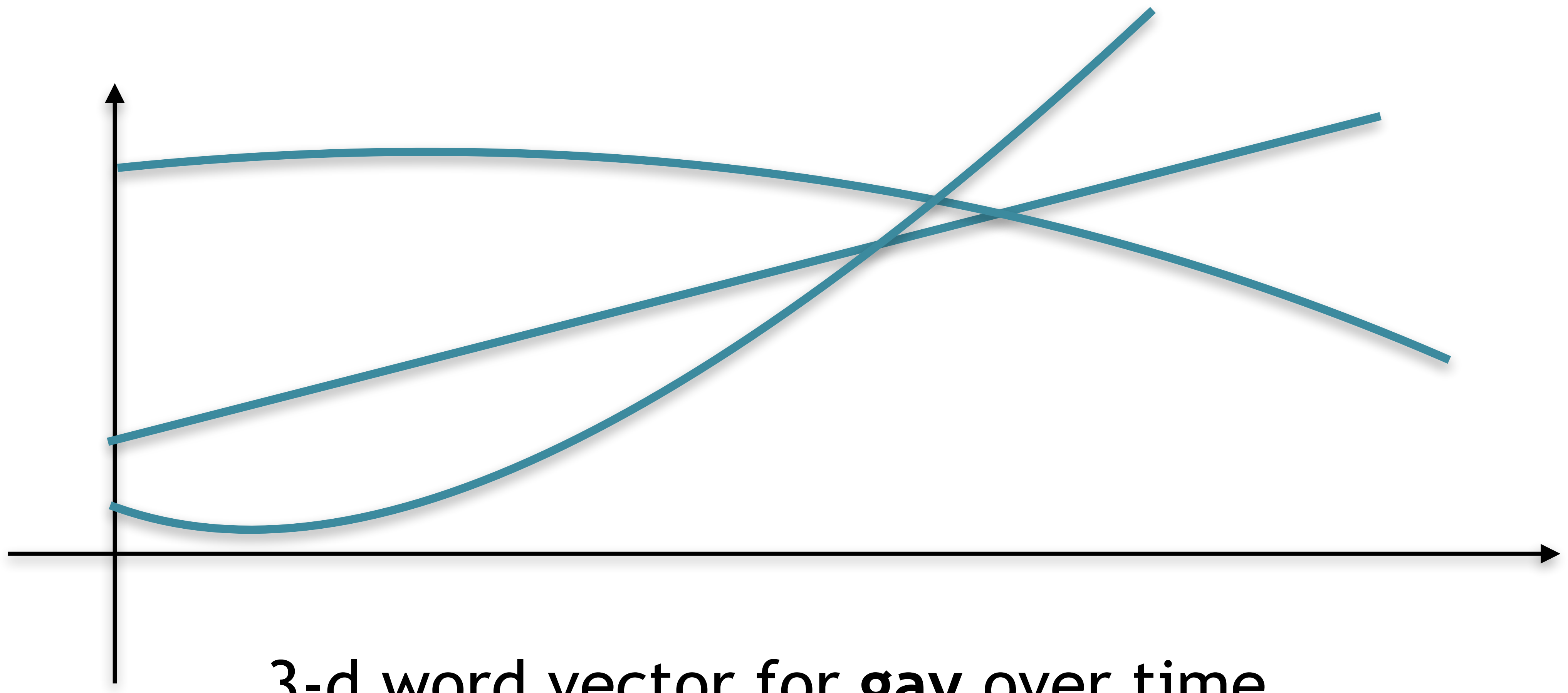


Modeling Word as Functions

Treating time as a continuous variable [4] induces a new formalization (Word2Fun)

$$f:(N) \rightarrow G\{g; g: N \rightarrow R^k\}$$

Word index   Time index



3-d word vector for **gay** over time

<u>Question:</u> ***Which functions should we use?***

[4] Alex Rosenfeld, Katrin Erk. Deep Neural Models of Semantic Shift. NAACL 2018

Approximation of Word Meaning Evolution

Here we define a **temporal word embedding**

$$f(\cdot\,,\cdot\,) : (\mathbb{N}, \mathbb{R}) \rightarrow \mathbb{R}^D$$

that maps a word w_i in time t as a N-dimensional vector $f(i, t) \in \mathbb{R}^D$. $f_i(t)$ is a function over t.

Approximation of Word Meaning Evolution

Here we define a **temporal word embedding**

$$f(\cdot,\cdot):(\mathbb{N},\mathbb{R})\rightarrow\mathbb{R}^D$$

that maps a word w_i in time t as a N-dimensional vector $f(i,t)\in\mathbb{R}^D$. $f_i(t)$ is a function over t.

We also define a static word embedding for alignment, also called a compass [1].

$$g(\cdot):\mathbb{N}\rightarrow\mathbb{R}^D$$

[1] Valerio Di Carlo et.al. Training Temporal Word Embeddings with a Compass. AAAI 2019

Approximation of Word Meaning Evolution

Here we define a **temporal word embedding**

$$f(\cdot, \cdot) : (\mathbb{N}, \mathbb{R}) \rightarrow \mathbb{R}^D$$

that maps a word w_i in time t as a N-dimensional vector $f(i, t) \in \mathbb{R}^D$. $f_i(t)$ is a function over t.

We also define a static word embedding for alignment, also called a compass [1].

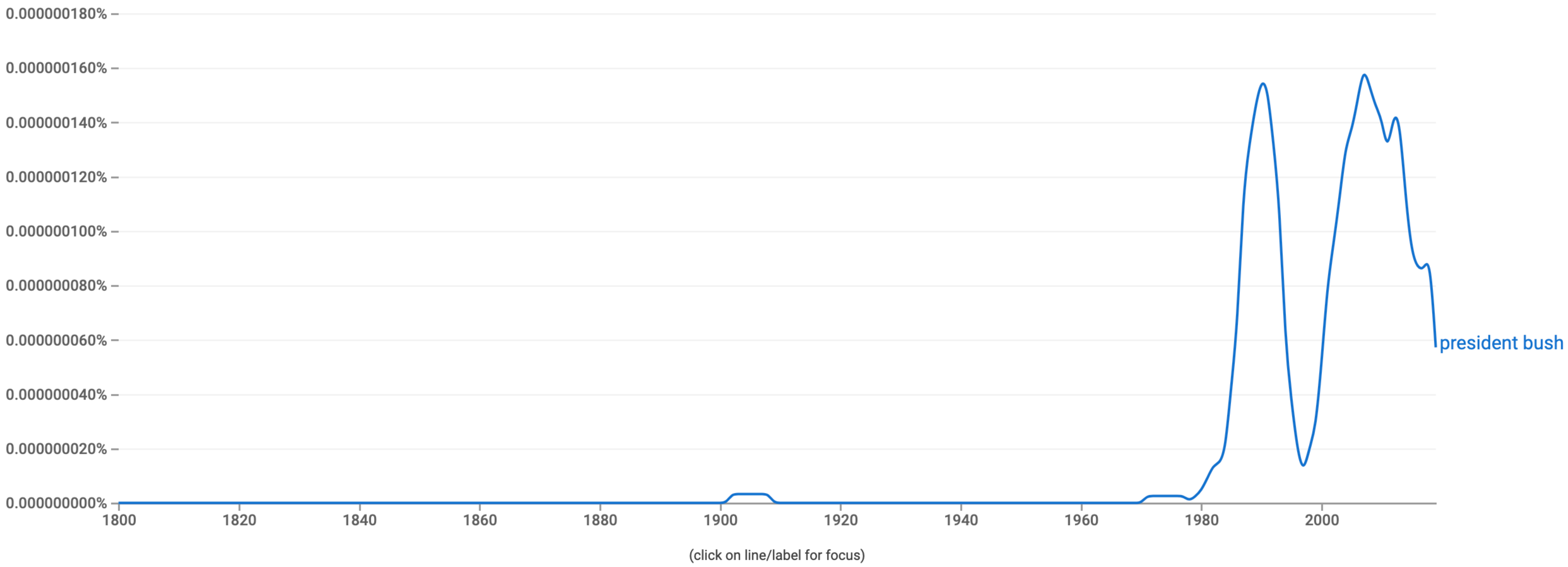
$$g(\cdot) : \mathbb{N} \rightarrow \mathbb{R}^D$$

A dot product between them should approximate their PPMI over time.

$$f_i(t)g(j)^T \propto PPMI_{i,j}(t)$$

[1] Valerio Di Carlo et.al. Training Temporal Word Embeddings with a Compass. AAAI 2019

Between-word relatedness over Time



evolving relatedness between “**president**” and “**bush**” may be *highly-nonlinear*

The result is from https://books.google.com/ngrams

Approximation of Word Meaning Evolution

Here we define a **temporal word embedding**

$$f(\cdot, \cdot) : (\mathbb{N}, \mathbb{R}) \rightarrow \mathbb{R}^D$$

that maps a word w_i in time t as a N-dimensional vector $f(i, t) \in \mathbb{R}^D$. $f_i(t)$ is a function over t.

We also define a static word embedding for alignment, also called a compass [1].

$$g(\cdot) : \mathbb{N} \rightarrow \mathbb{R}^D$$

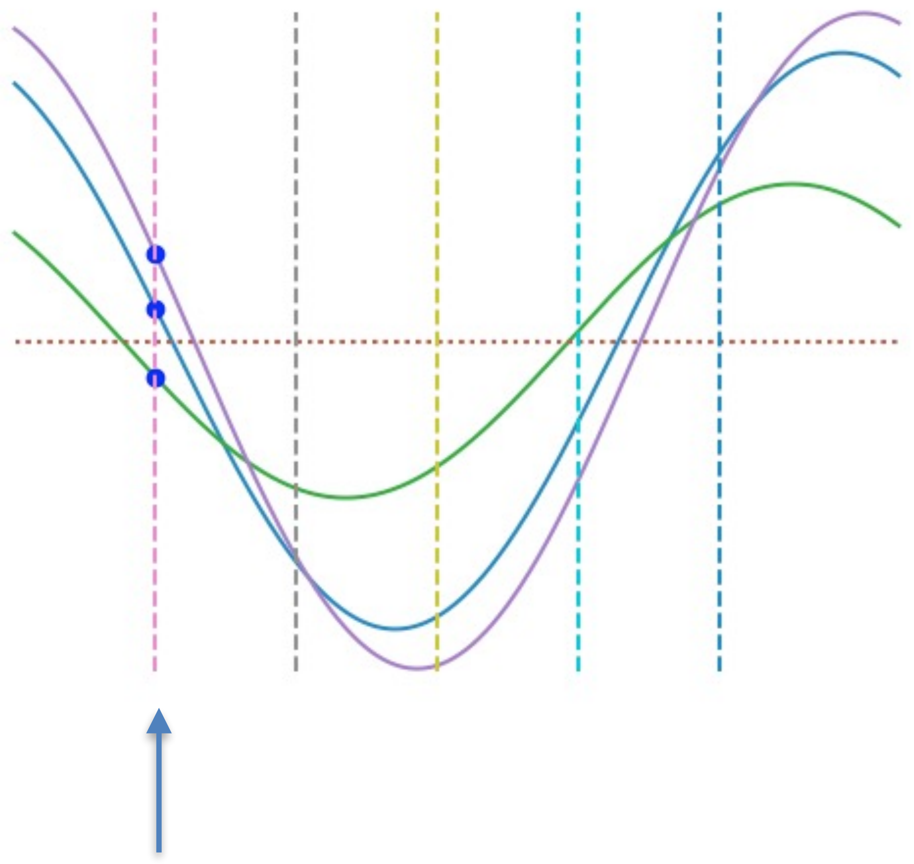
A dot product between them should approximate their PPMI over time.

$$f_i(t)g(j)^T \propto PPMI_{i,j}(t)$$

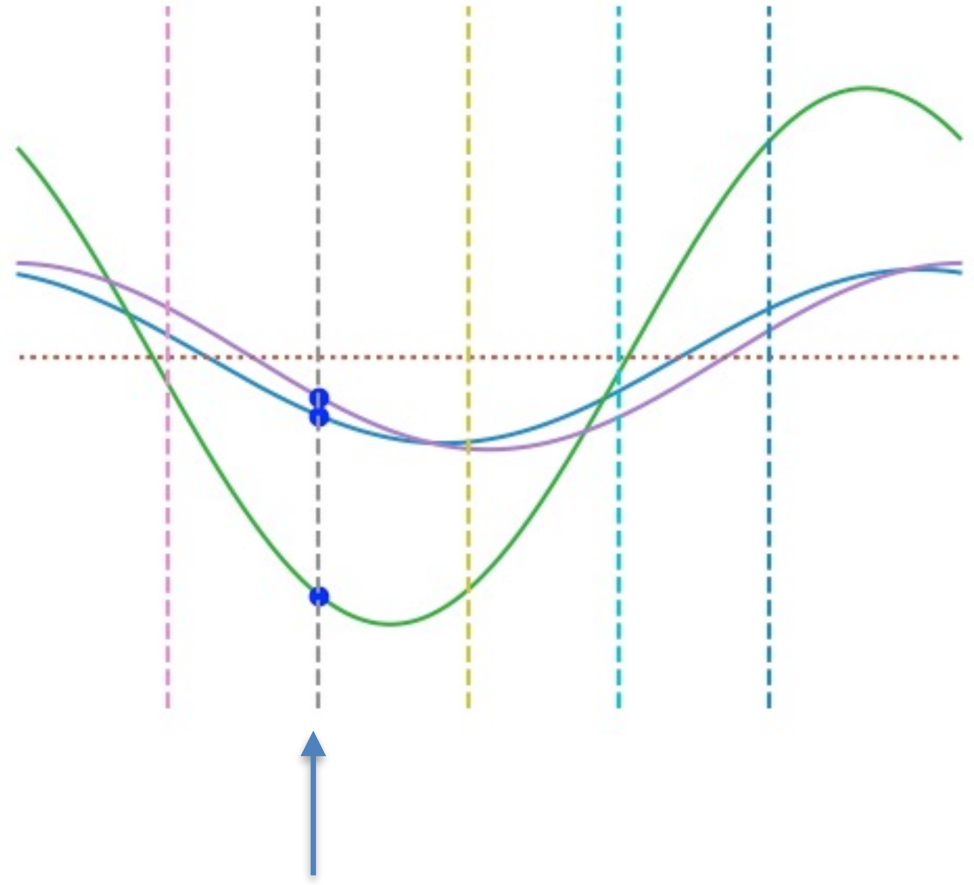
When $f_i(t)$ is formalised as a **sinusoidal** function. $f(i, t)g(j)^T$ is proved to **approximate any continuous functions** thanks to the **Weierstrass Approximation theorem**.

[1] Valerio Di Carlo et.al. Training Temporal Word Embeddings with a Compass. AAAI 2019

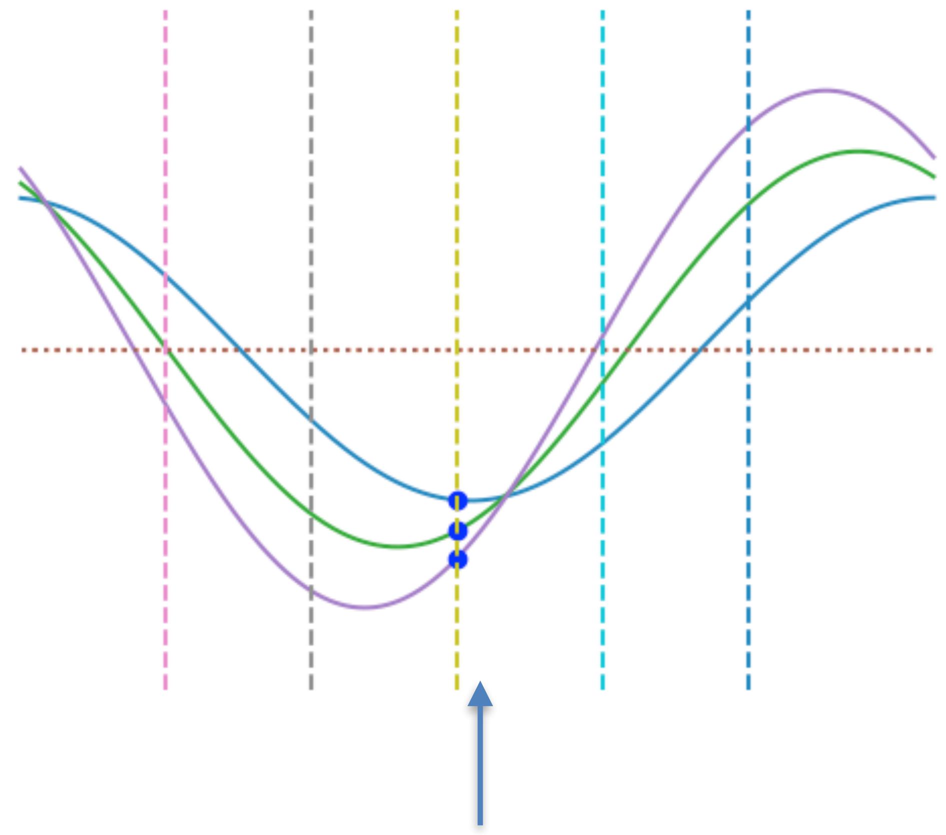
Word2Fun (examples)



gay in 1910s



cheerful in 1920s



homosexual in 1930s

Experimental Evaluation

Table 3: Experimental results of Time-aware word clustering.

Method	10 Clusters		15 Clusters		20 Clusters	
	NMI	F_β	NMI	F_β	NMI	F_β
Global/static word vector [16]	0.6736	0.6163	0.6867	0.7147	0.6713	0.7214
Transformed Word2Vec [14]	0.5175	0.4584	0.5221	0.5072	0.5130	0.5373
Aligned Word2Vec [9]	0.6580	0.6530	0.6618	0.7115	0.6386	0.7187
Dynamic Word2Vec [26]	0.7175	0.6949	0.7162	0.7515	0.6906	0.7585
Compass aligned Word2Vec [6]	0.5191	0.3750	0.5062	0.4051	0.5077	0.4331
Word2Fun linear	0.1676	0.1813	0.2826	0.3035	0.2473	0.2932
Word2Fun I (Time2Fun)	0.1703	0.1783	0.2691	0.2680	0.2842	0.2649
Word2Fun II	**0.7281**	**0.7147**	**0.7181**	0.7645	**0.7012**	0.7616
Word2Fun III	0.7233	0.7080	0.7086	**0.7701**	0.6980	**0.7630**
Word2Fun IV	0.7111	0.6913	0.7023	0.7451	0.6823	0.7602

Time-aware word clustering

Table 4: Experimental results of temporal analogy in *test1*

Method	MRR	P@1	P@3	P@5	P@10
Global/static Word2Vec [16]	0.3560	0.2664	0.4210	0.4774	0.5612
Transformed Word2Vec [14]	0.0920	0.0500	0.1168	0.1482	0.1910
Aligned Word2Vec [9]	0.1582	0.1066	0.1814	0.2241	0.2953
Dynamic Word2Vec [26]	0.4222	0.3306	0.4854	0.5488	0.6191
Compass aligned Word2Vec [6]	**0.481**	**0.404**	**0.534**	0.582	0.636
Word2Fun linear	0.3016	0.2649	0.3255	0.3426	0.3630
Word2Fun I (Time2Fun)	0.3735	0.2646	0.4300	0.4955	0.5874
Word2Fun II	0.4061	0.2756	0.4916	0.5614	0.6434
Word2Fun III	0.4354	0.3076	0.5330	**0.5837**	**0.6647**
Word2Fun IV	0.4208	0.2954	0.5076	0.5715	0.6470

Temporal analogy test1

Table 5: Experimental results of temporal analogy in *test2*

Method	MRR	P@1	P@3	P@5	P@10
Global/static Word2Vec [16]	0.0472	0.0000	0.0787	0.0787	0.2022
Transformed Word2Vec [14]	0.0664	0.0404	0.0764	0.0989	0.1438
Aligned Word2Vec [9]	0.0500	0.0225	0.0517	0.0787	0.1416
Dynamic Word2Vec [26]	0.1444	0.0764	0.1596	0.2202	0.3820
Compass Aligned Word Embedding [6]	0.1361	0.0749	0.1918	0.2904	0.3918
Word2Fun linear	0.0425	0.0137	0.0384	0.0630	0.1014
Word2Fun I (Time2Fun)	0.0992	0.0000	0.1315	0.1726	0.2849
Word2Fun II	0.1194	0.0358	0.1075	0.2219	0.3863
Word2Fun III	**0.1824**	**0.0795**	**0.1973**	**0.2932**	**0.4164**
Word2Fun IV	0.1536	0.0548	0.1562	0.2411	0.3918

Temporal analogy test2

Table 6: Semantic change detection. Baselines in the first group are implemented by this work.

models	Pearson	Spearman
Global/static Word2Vec [16]	nan	nan
Transformed Word2Vec [14]	0.0727	0.0865
Aligned Word2Vec [9]	0.3333	0.3083
Dynamic Word2Vec [26]	0.2727	0.2877
Compass aligned word embedding [6]	0.3199	0.2567
Word2Fun linear	-0.1200	-0.0790
Word2Fun I (Time2Fun)	0.3925	0.4550
Word2Fun II	0.4478	**0.5038**
Word2Fun III	**0.5355**	0.4057
Word2Fun IV	0.4483	0.3578
multilingual BERT [20] (SemEval-2020 1st)	-	0.436
ensemble between aligned Word2Vec and BERT [18] (SemEval-2020 2nd)	-	0.422

Semantic change detection

Case study

word	1900s	1920s	1940s	1960s	1980s	2000s
frolicsome	**0.5230**	0.3574	0.2802	0.1511	0.1649	0.1992
playful	0.4094	0.3757	**0.4268**	0.3298	0.2425	0.2839
debonair	0.3840	0.4705	**0.5523**	0.4597	0.2243	0.3547
activists	0.2319	0.2430	0.0892	0.2894	**0.4698**	0.4072
homosexuality	-0.1435	-0.0274	0.1209	0.2605	0.3242	**0.3727**

Word similarity to “gay” over time

Acknowledgments

This work is supported by the Quantum Access and Retrieval Theory (QUARTZ) project, which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 721321.