Google research

Unsupervised Morphology Induction using Word Embeddings

Radu Soricut, Franz J. Och*

NAACL 2015

*now at Human Longevity Inc.

Word Embeddings

- vocabulary V, embedding function e: $V \rightarrow \mathbf{R}^n$
- vector space encodes semantic similarity
  - $e(\text{car}) \simeq e(\text{automobile})$, $e(\text{car}) \neq e(\text{seahorse})$
- vector space encodes compositionality
  - semantic: $e(\text{king}) - e(\text{man}) + e(\text{woman}) \simeq e(\text{queen})$
  - syntactic: e(cars) - e(car) + e(fireman) ≃ e(firemen)
- vector space encodes syntactic/semantic transformations
  - anti+ $\simeq e(\text{anticoruption}) - e(\text{corruption})$

SkipGram Embeddings [Mikolov et al., 2013]

British scientists recreated Down ’s syndrome in mice in order to study the disease and develop new treatments .

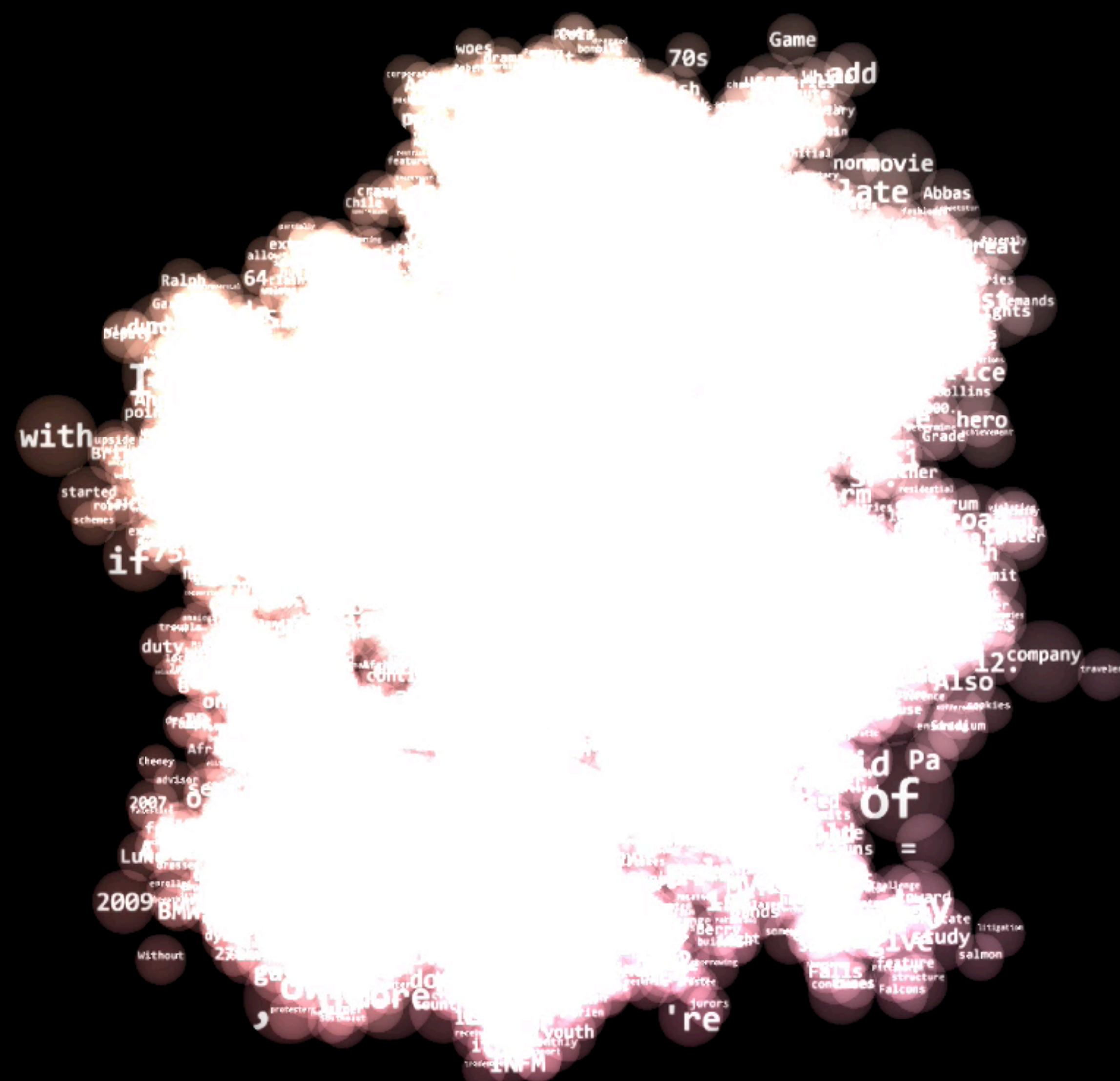
SkipGram Embeddings [Mikolov et al., 2013]

British scientists recreated Down 's syndrome in mice in order to study the disease and develop new treatments .

$$\arg\max_{v_w, v_c} \Big(\sum_{(w,c) \in D_w} \log \sigma(v_w \cdot v_c) + \sum_{(w,c) \in \overline{D_w}} \log \sigma(-v_w \cdot v_c) \Big)$$

$$\sigma(x) = \frac{1}{1 + e^{x}}$$

Train on large monolingual corpus


Game
70s
add
movie
Abbas
hero
company
Also
of
with
if
2009
duty
're
youth
salmon
study

SkipGram Embeddings: Introduction

size: 8,869 tupl

king - man + w

evaluation: acc

(over entire voc

size: 10,675 tuples

cars - car + fireman ≈ firemen

evaluation: acc% closest poin

(over entire vocabulary)

size: 2003 pairs

is a celebrated ?

an English ?roc

eval: Spearman

(against 10-hun

size: 2034 pairs

belligerence ≈ hostility 8.7

amorphous ≈ inorganic 1.9

eval: Spearman corr.

(against 10-human score avg.)

Model	Compositionality (Acc%)		Similarity (Spearman ρ)	
	Semantic	Syntactic	Stanford-C	Stanford-RW
SkipGram (Wikipedia 1Bw)	76.7	68.3	66.3	35.8

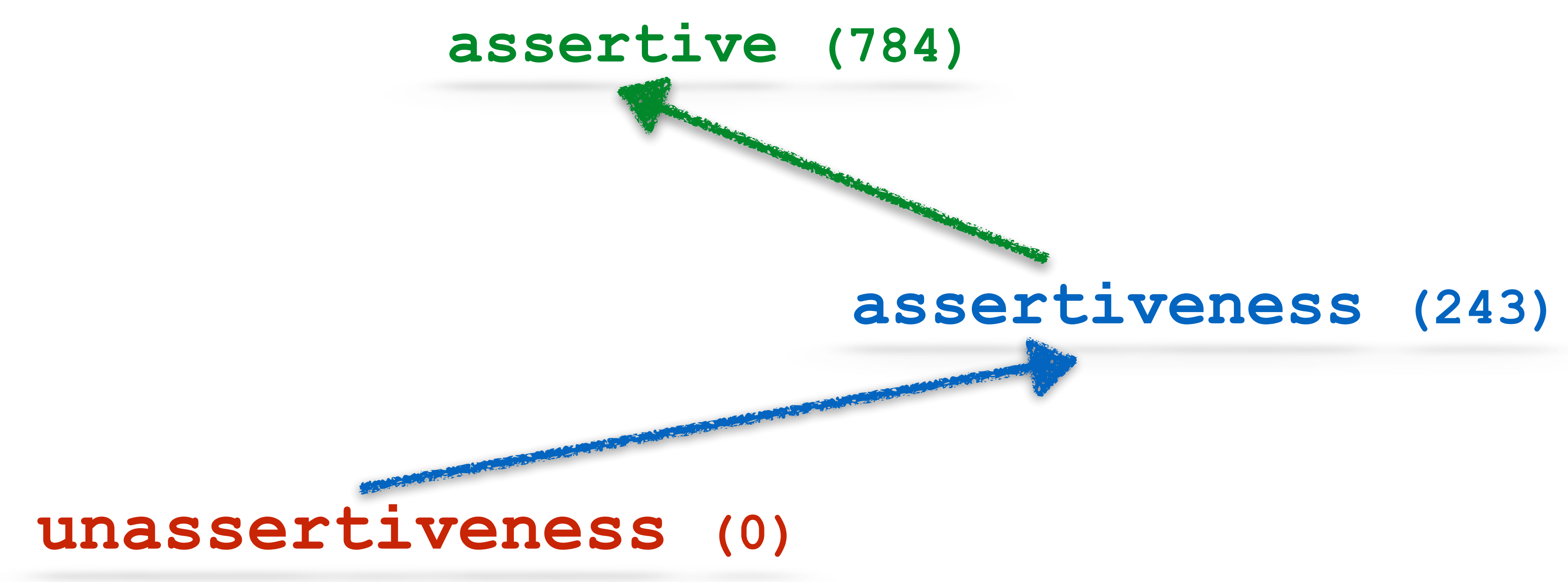
$$\arg\max_{v_w, v_c} \Big(\sum_{(w,c)\in D_w} \log \sigma(v_w \cdot v_c) + \sum_{(w,c)\in \overline{D_w}} \log \sigma(-v_w \cdot v_c) \Big)$$

$$\sigma(x) = \frac{1}{1+e^{x}}$$

Unsupervised Morphology Induction

Q: What do we want?

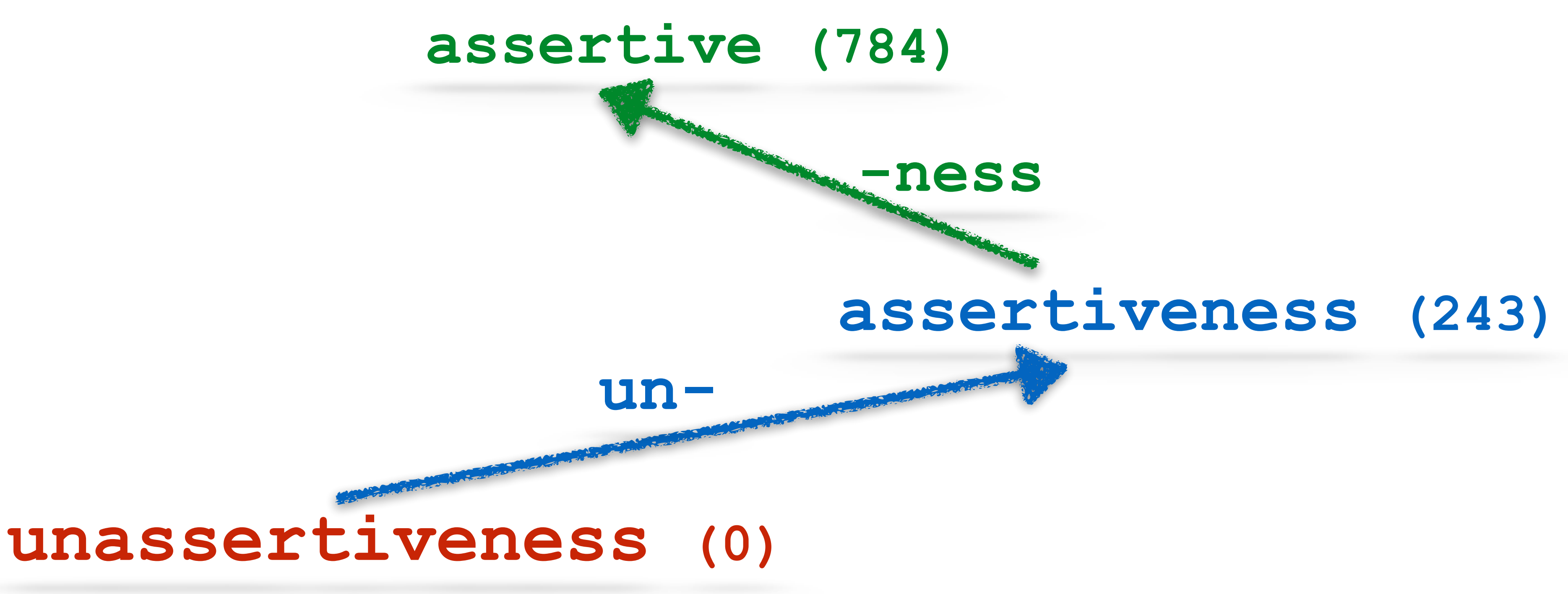
A: We want *high-quality* embeddings for all words (even ones outside *V*)



Unsupervised Morphology Induction

Q: What do we want?

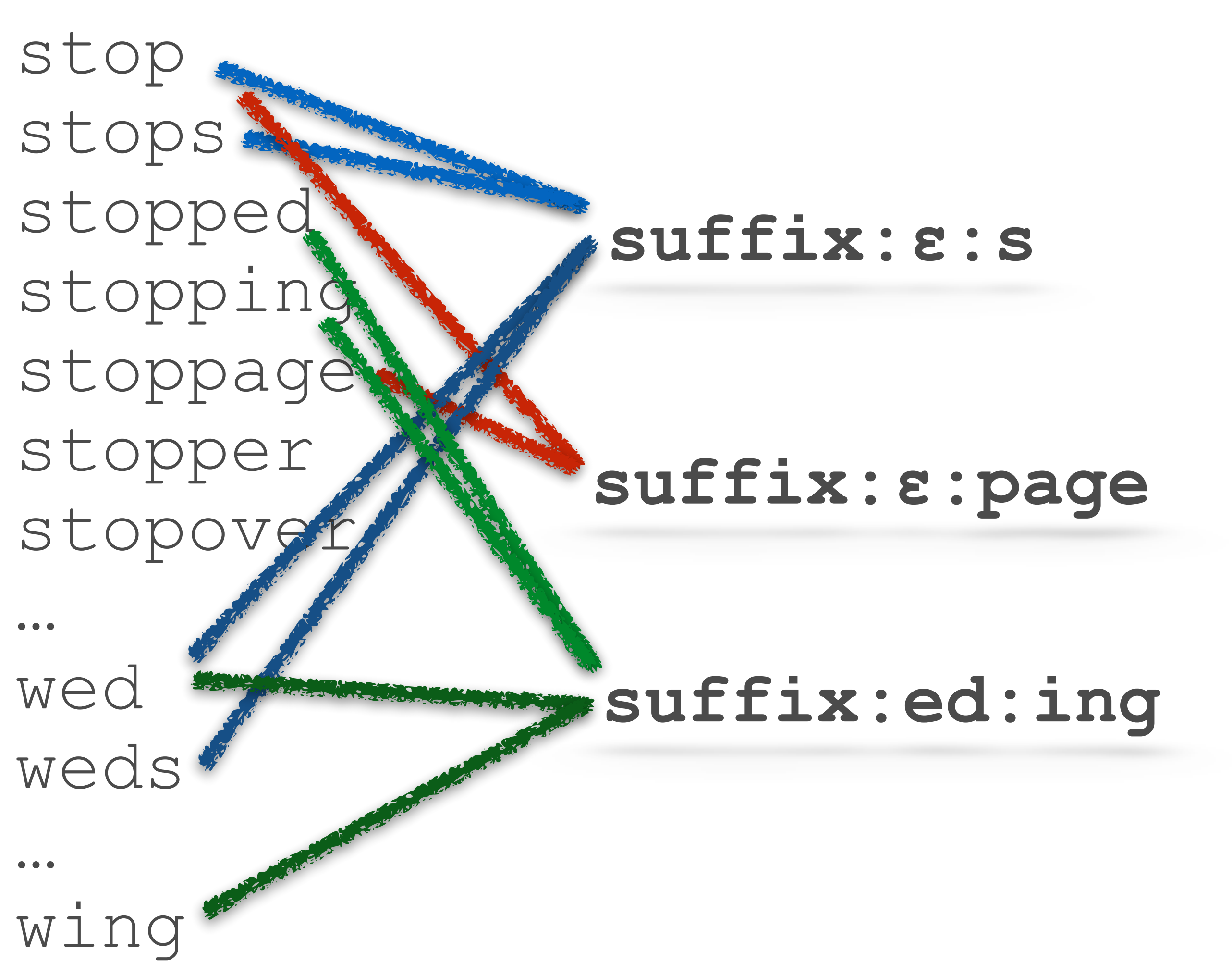
A: We want *morphology-based transformations* that can accurately analyze words (even ones unseen at training time)



Unsupervised Morphology Induction: Algorithm

Steps:

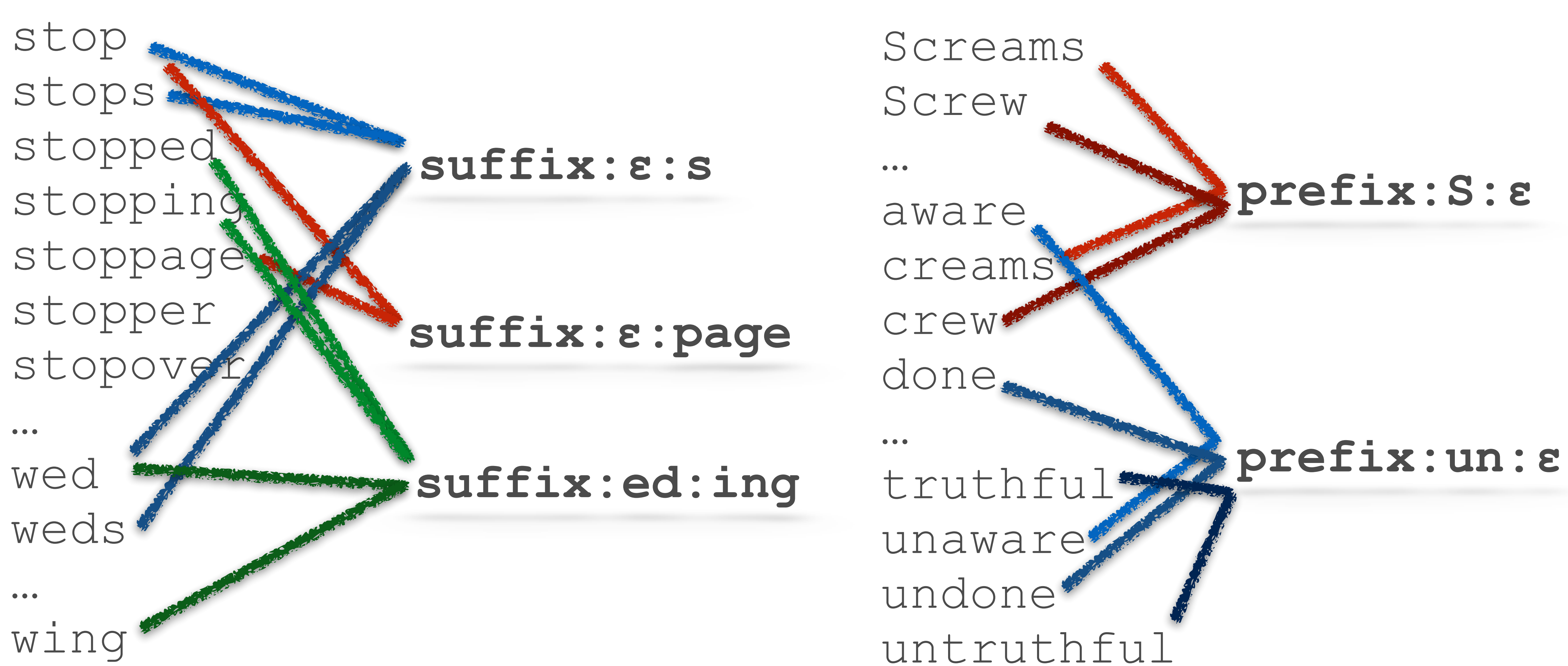
1. From *V*, extract candidates for morphological rules (prefix & suffix only)



Unsupervised Morphology Induction: Algorithm

Steps:

1. From *V*, extract candidates for morphological rules (prefix & suffix only)



Unsupervised Morphology Induction: Algorithm

Steps:

2. Query against embedding space: *morphology does not shift meaning*

suffix:ed:ing

adored adorned affected …
blamed blitzed blogged …
stayed stepped stopped …
weaned wed wedged whirled

rank(blamed → blaming) = 1
rank(stopped → stopping) = 2
rank(wed → wing) = 28609

prefix:ε:S

aura aux ave …
canned cans car care …
crape cream creams …
miles mitten mothers …

rank(care → Scare) = 57778
rank(cream → Scream) = 9434
rank(miles → Smiles) = 18800

Unsupervised Morphology Induction: Algorithm

Steps:

2. Query against embedding space: *morphology does not shift meaning*

```
prefix:un:ε

unabated unable unabridged…
unaware unbalance unbeaten…
undoing undone undoubted…
untrusted untrustworthy…
```

rank(unaware → aware) = 1

rank(undone → done) = 129

Unsupervised Morphology Induction: Algorithm

Steps:

2. Query against embedding space: ~~*morphology does not shift meaning*~~

morphology shifts meaning consistently

prefix:un:ε

```
unabated unable unabridged…
unaware unbalance unbeaten…
undoing undone undoubted…
untrusted untrustworthy…
```

rank(unaware → aware) = 0

rank(undone → done) = 129

↑un-

```
    clear - unclear
delivered - undelivered
 truthful - untruthful
```

rank(undone + ↑un- → done) = 4

Unsupervised Morphology Induction: Algorithm

Steps:

3. Extract candidate rules using embedding-based stats

	Candidate Rule	Direction	#Correct	#Total	Acc10
Bad	**suffix:h:a**	↑Teh	1	449	0.4%
	suffix:o:es	↑Tono	7	688	1.0%
	prefix:D:W	↑Daring	9	675	1.3%
	…				
Good	**prefix:un:ε**	↑undelivered	166	994	23.3%
	suffix:ed:ing	↑procured	2138	4714	56.2%
	…				
	suffix:ating:ate	↑formulating	255	395	74.7%
	suffix:sed:zed	↑victimised	153	186	90.9%

Unsupervised Morphology Induction: Algorithm

Steps:

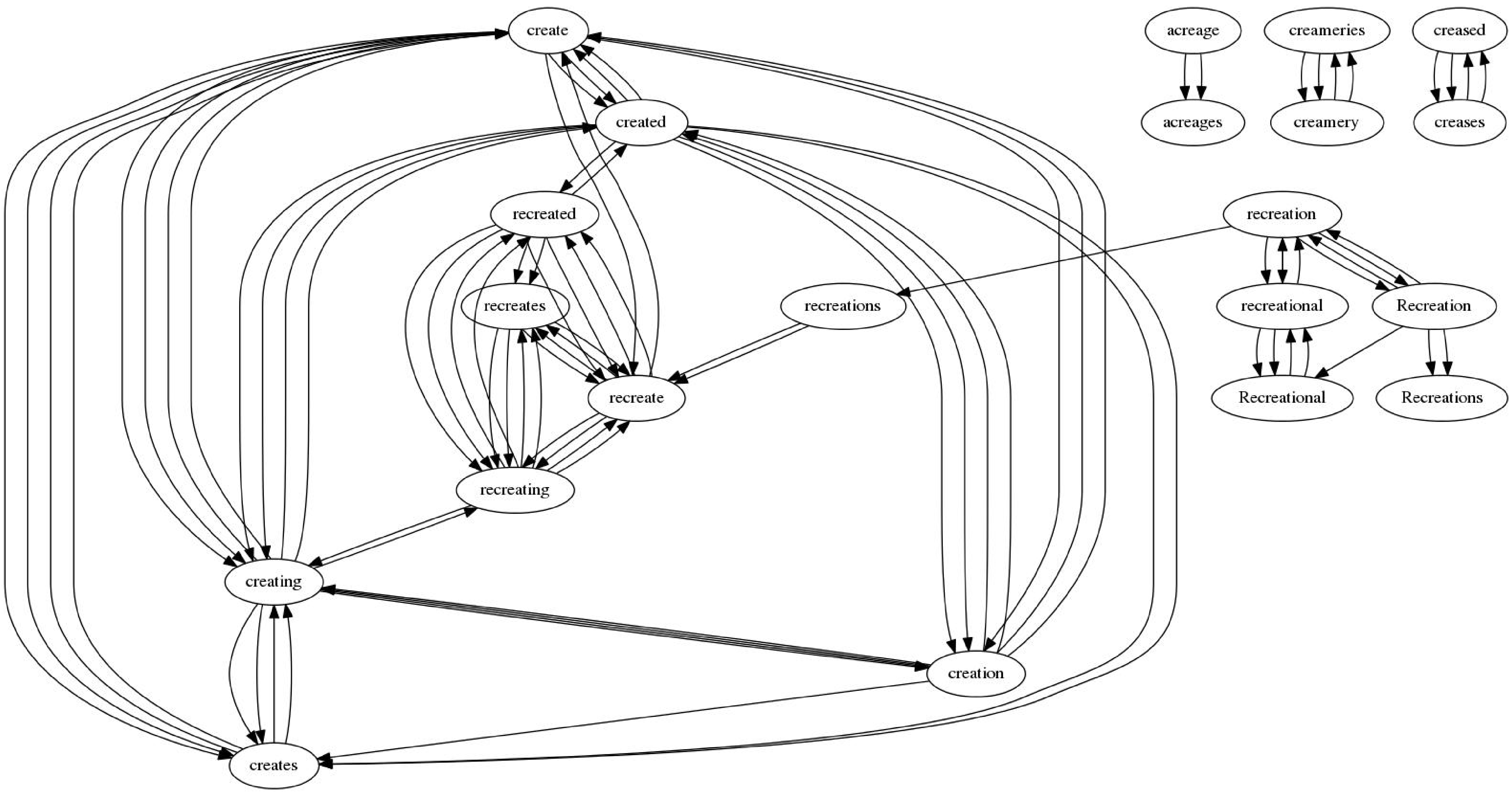
4. Use rules to extract lexicalized, weighted morphological transformations

Start	Rule + Direction = Transformation	End	Cosine	Rank
...				
recreations	suffix:ions:e + ↑investigations	recreate	0.69	1
recreations	suffix:tions:te + ↑investigations	recreate	0.70	1
recreations	suffix:ions:ed + ↑delineations	recreated	0.51	29
recreations	suffix:ions:ing + ↑reconstruction	recreating	0.72	1
...				
unaware	prefix:un:ε + ↑uncivilized	aware	0.77	1
unaware	prefix:un:ε + ↑undelivered	aware	0.63	7

Unsupervised Morphology Induction: Algorithm

Output (I): labeled, weighted, cyclic, directed multigraph G^{V}_{Morph}

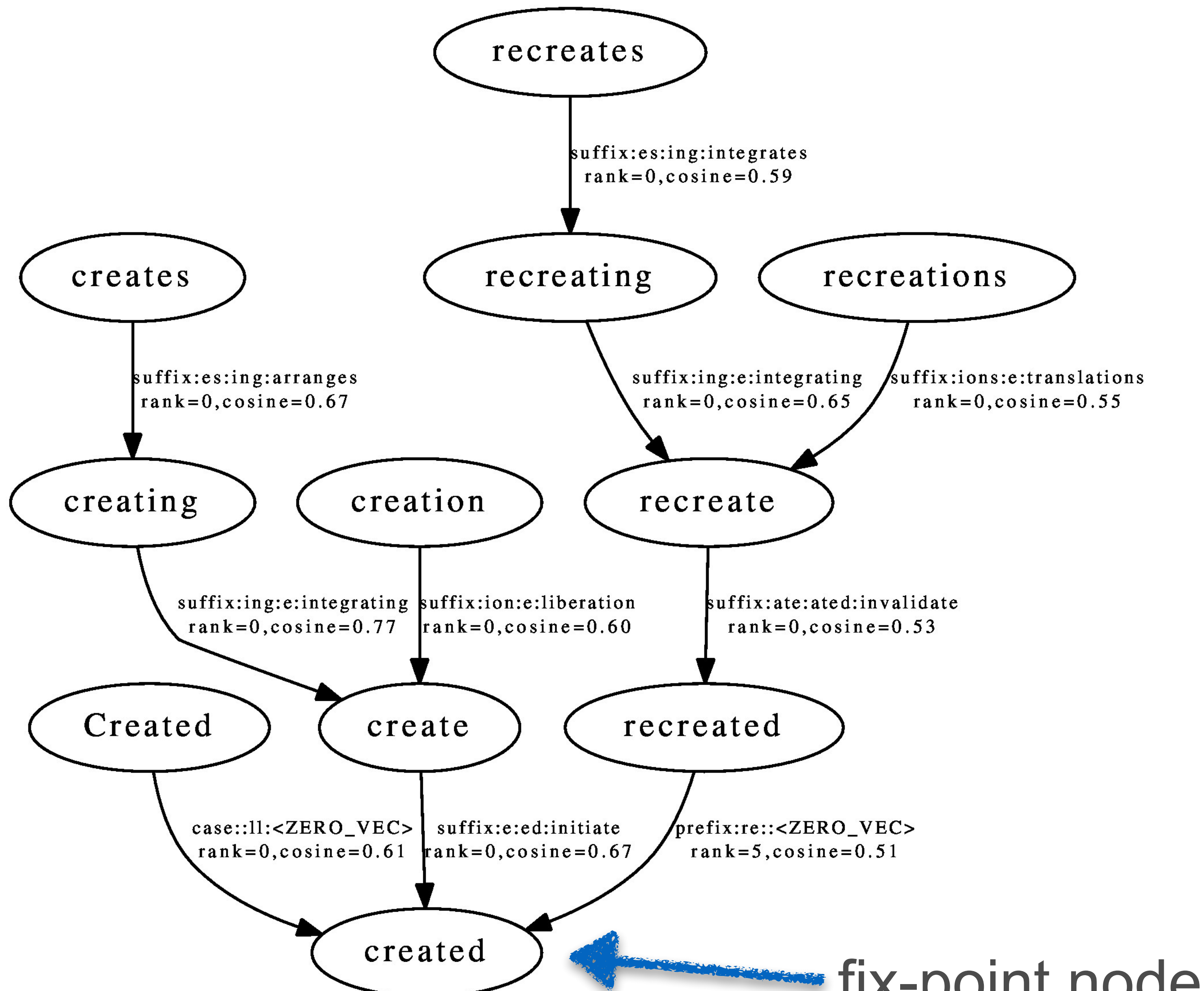
- words are nodes, morphological transformations are (weighted) edges



Unsupervised Morphology Induction: Algorithm

Output (II): labeled, weighted, acyclic, directed graph D^{V}_{Morph}

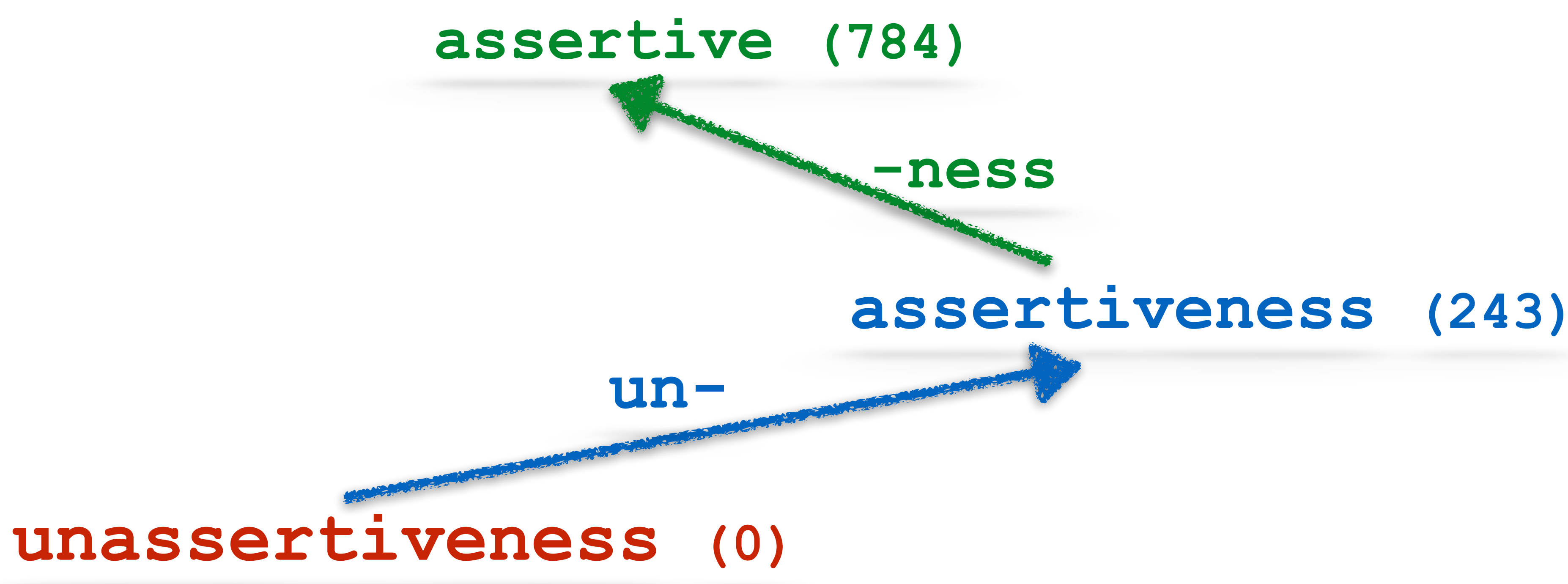
- words are nodes, morphological mappings are weighted edges



Unsupervised Morphology Induction

Q: What do we want?

A: We want *morphology-based transformations* that can accurately analyze words (even ones unseen at training time)



Unsupervised Morphology Induction

Basic algorithm: embedding words outside *V*

Outside Wikipedia (1B tokens, |*V*| = 4.3M)

animalize (0)

balminess (0)

caesarism (0)

containerful (0)

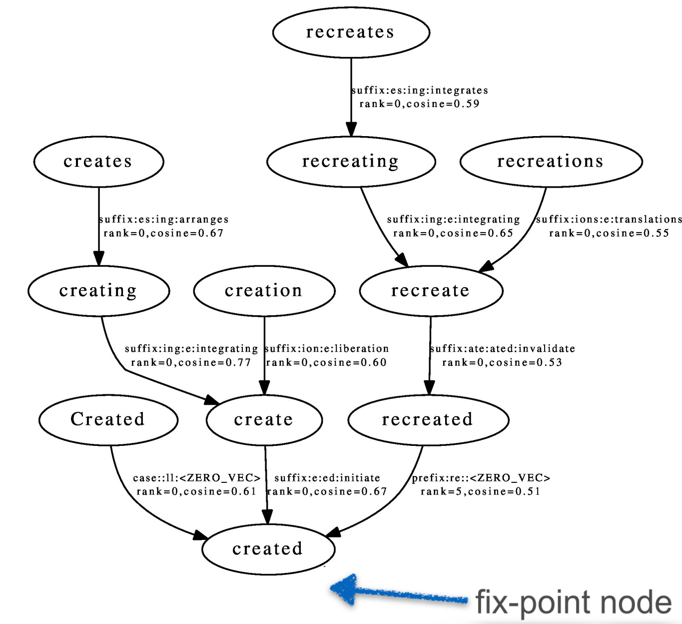
nonindulgent (0)

unassertiveness (0)

Unsupervised Morphology Induction: Algorithm

Analyze words outside V

1. Train time: extract and count all paths ending in a “fix-point” from the directed acyclic graph D^{V}_{Morph}
   - each path is called a “rule sequence”



rule sequence	count
`suffix:s:ε`	3119
`suffix:ed:ε`	687
`suffix:ing:ed`	412
`prefix:un:ε`	207
`suffix:ness:ε`	162
`suffix:ness:ly`	25
`suffix:y:ier,suffix:er:ness`	10
`prefix:un:ε,suffix:ed:ing`	5

Unsupervised Morphology Induction: Algorithm

Google

Analyze words outside *V*

2. Run time: apply each rule sequence in descending order of counts
   - if rule fires, check that result has count > 0 and in-degree > 0
   - stop at first winner

	rule sequence	count	
`unassertiveness(0)`	`suffix:s:ε`	3119	`unassertivenes(0)`
	`suffix:ed:ε`	687	
	`suffix:ing:ed`	412	
`unassertiveness(0)`	`prefix:un:ε`	207	`assertiveness(243)`
	`suffix:ness:ε`	162	
	`suffix:ness:ly`	25	
	`suffix:y:ier,suffix:er:ness`	10	
	`prefix:un:ε,suffix:ed:ing`	5	

`unassertiveness = assertiveness + ↑un+`

Unsupervised Morphology Induction: Evaluation

A: We want *morphology-based transformations* that can accurately analyze words unseen at training time

assertiveness (243)

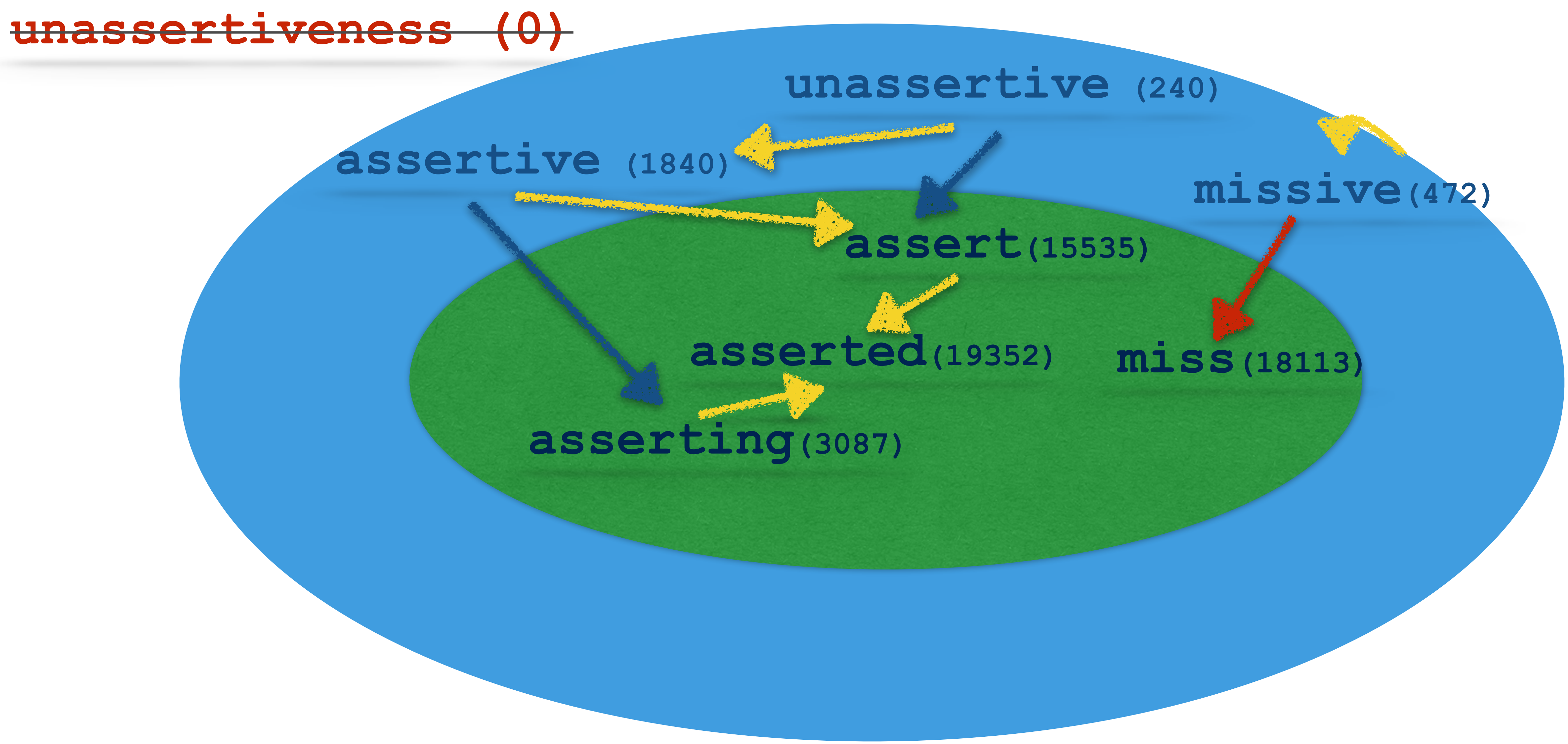
un-

unassertiveness (0)

Language	|Tokens|	|V|	$\lvert G^{V}_{Morph} \rvert$	$\lvert D^{V}_{Morph} \rvert$
EN	1.1b	1.2m	780k	75,823
DE	1.2b	2.9m	3.7m	169,017
FR	1.5b	1.2m	1.8m	92,145
ES	566m	941k	2.2m	82,379
RO	1.7b	963k	3.8m	141,642
AR	453m	624k	2.4m	114,246
UZ	850m	2.0m	5.6m	194,717

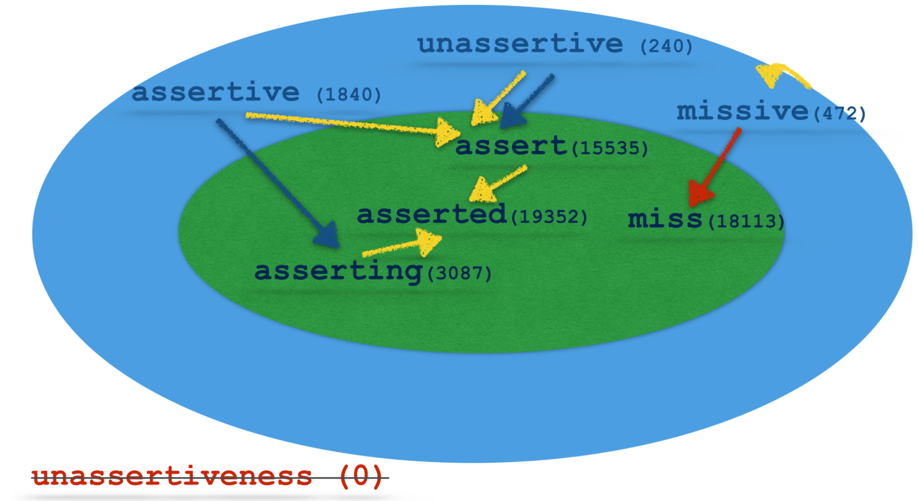
Unsupervised Morphology Induction: Evaluation

Evaluate OOV analysis using low-count words



Unsupervised Morphology Induction: Evaluation

Evaluate OOV analysis using rare words



Language	$\lvert V_{[1000,2000)} \rvert$		Accuracy	
	Have analysis	Don't have analysis	Have analysis	Don't have analysis
EN	3421	10617	89.7%	89.6%
DE	10778	21234	90.8%	93.1%
FR	6435	9807	90.3%	90.4%
ES	5724	7412	91.1%	90.3%
RO	11905	9254	86.5%	85.3%
AR	7913	5202	92.4%	69.0%
UZ	11772	9027	81.3%	84.1%

Unsupervised Morphology Induction: Evaluation

Q: What do we want?

A: We want *morphology-based transformations* that can accurately analyze words (even ones unseen at training time)

Key result

Improved word similarity judgment: unknown, low-count, high-count words

- evaluation on Stanford Rare Word similarity dataset (RW-EN)
- evaluation of similarity datasets on various languages (RG-DE)

Unsupervised Morphology Induction: Evaluation

Training Setup

	Language	Train Set	|Tokens|	|V|	$\|G^V_{Morph}\|$	$\|D^V_{Morph}\|$
Small	EN	Wiki-EN	1.1b	1.2m	780k	75,823
	DE	WMT-DE	1.2b	2.9m	3.7m	169,017
Large	EN	News-EN	120b	1.0m	2.9m	98,268
	DE	News-DE	20b	1.8m	6.7m	351,980

Unsupervised Morphology Induction: Evaluation

Evaluation on similarity datasets (RG-DE, RW-EN)

size: 2034 pairs

impossibilities	unattainableness	8.8
deregulating	liberation	8.0
baseness	unworthiness	4.0
transmigrating	born	1.1

Language	Train Set	|Tokens|	|V|	$\|G^V_{Morph}\|$	$\|D^V_{Morph}\|$
EN	Wiki-EN	1.1b	1.2m	780k	75,823
DE	WMT-DE	1.2b	2.9m	3.7m	169,017
EN	News-EN	120b	1.0m	2.9m	98,268
DE	News-DE	20b	1.8m	6.7m	351,980

	RW-EN Testset			
	|Unembedded|		Spearman ρ	
System	Wiki-EN	News-EN	Wiki-EN	News-EN
SkipGram	78	177	35.8	44.7
SkipGram+Morph	1	0	41.8	52.0

+9

+7

Unsupervised Morphology Induction: Evaluation

Evaluation on similarity datasets (RG-DE, RW-EN)

size: 65 pairs

Edelstein	Juwel	3.8
Autogramm	Unterschrift	3.5
Irrenhaus	Friedhof	0.3
Kraftfahrzeug	Magier	0.0

Language	Train Set	|Tokens|	|V|	$\|G^V_{Morph}\|$	$\|D^V_{Morph}\|$
EN	Wiki-EN	1.1b	1.2m	780k	75,823
DE	WMT-DE	1.2b	2.9m	3.7m	169,017
EN	News-EN	120b	1.0m	2.9m	98,268
DE	News-DE	20b	1.8m	6.7m	351,980

	RW-EN Testset			
	|Unembedded|		Spearman ρ	
System	Wiki-EN	News-EN	Wiki-EN	News-EN
SkipGram	80	177	35.8	44.7
SkipGram+Morph	1	0	41.8	52.0

+9 +7

	RG-DE Testset			
	|Unembedded|		Spearman ρ	
System	WMT-DE	News-DE	WMT-DE	News-DE
SkipGram	0	20	62.4	62.1
SkipGram+Morph	0	0	64.1	69.1

+0 +7

Conclusions

1. Method for inducing morphological transformations between words
   - from scratch, unsupervised, language agnostic
2. Provides morphology-based structure over embedding spaces
3. Provides high-quality embeddings for out-of-vocabulary and low-count morphological variants

Next steps

- Going beyond suffix & prefix morphology
    - nothing in the approach prevents from extending it
- Use it for improved Machine Translation
    - quick and painless morphological analysis on source side
    - generate morphological variants on target side (even new ones!)
- Use it for improved Information Retrieval

Thank you