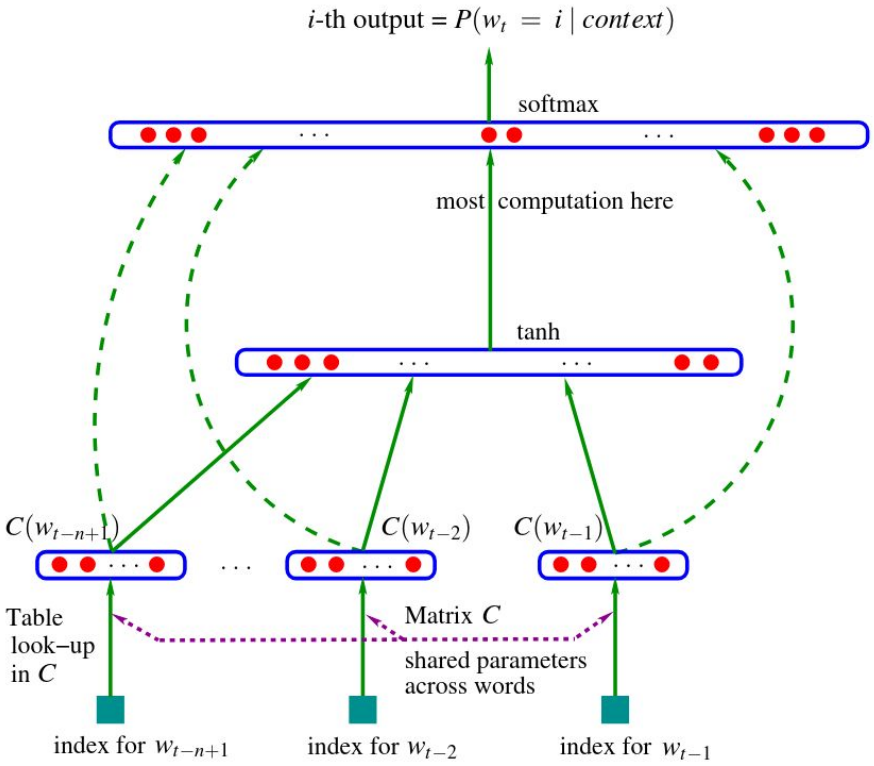
word2vec

“Distributed representations of words and phrases and their compositionality.”
&
“Efficient Estimation of Word Representations in Vector Space”
by Mikolov et al. in 2013

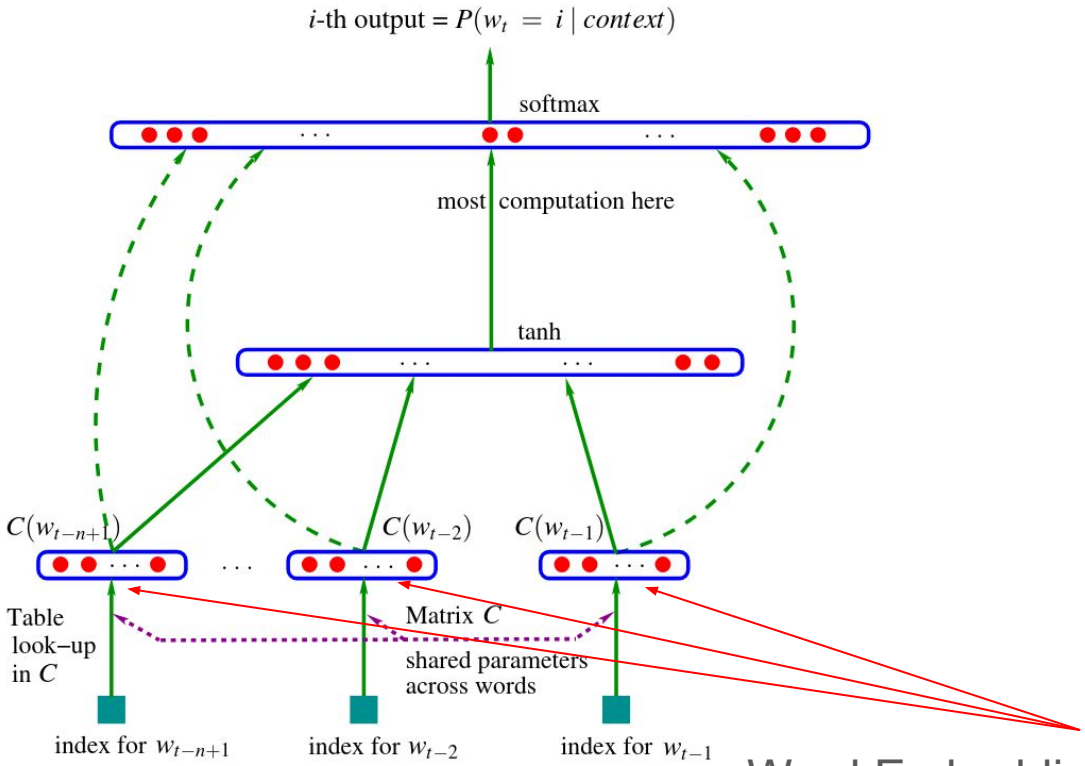
Presenters: Sandro Paval, Peter Preinesberger

Last Time: the dawn of neural language models!



Last Time: the dawn of neural language models …

Bengio et al.
in 2003 (!)



Word Embeddings in Matrix C

What's the deal with word2vec?

- Word embeddings ("distributed representations of words") for LM were already an idea for 10 years when word2vec was introduced …
- … so what was new in this work?

Greatly Simplified Training Tasks
(skip-gram & CBOW)

=> enables the use of much more training data

Direct Focus on Quality of Word Embeddings

=> (very) interesting syntactic and semantic relationships are discovered in embedding space

What's the deal with word2vec?

- Word embeddings ("distributed representations of words") for LM were already an idea for 10 years when word2vec was introduced …
- … so what was new in this work?

Greatly Simplified Training Tasks
(skip-gram & CBOW)

=> enables the use of much more training data

Direct Focus on Quality of Word Embeddings

=> (very) interesting syntactic and semantic relationships are discovered in embedding space

Last time (again) … in Bengio et al. 2003:

quences seen in the training set. We propose to fight the curse of dimensionality by **learning a distributed representation for words** which allows ea

rds forming an already seen sentence. Training such large models (with millions of parameters) within a reasonable time is itself a significant challenge. We report on exp

ulti-layer neural networks. Another contribution of this paper concerns the challenge of training such very large neural networks (with millions of parameters) for very large data sets (with millions or tens of millions of examples). Finally, an

s of relative frequencies. The main computational bottleneck with the neural implementation is the computation of the activations of the output layer.

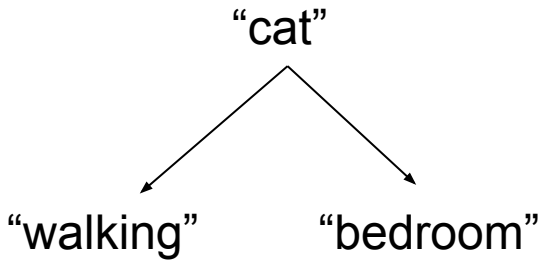
3. Parallel Implementation

Getting Down the Costs: Skip-Gram Objective

Previously (modeling of the conditional language distribution):

Input	**Label**
“The” “cat” “is” “walking” “in” “the”	“bedroom”

Now, instead, predict the context of a word:



Predict the context, but how exactly?

maximize:

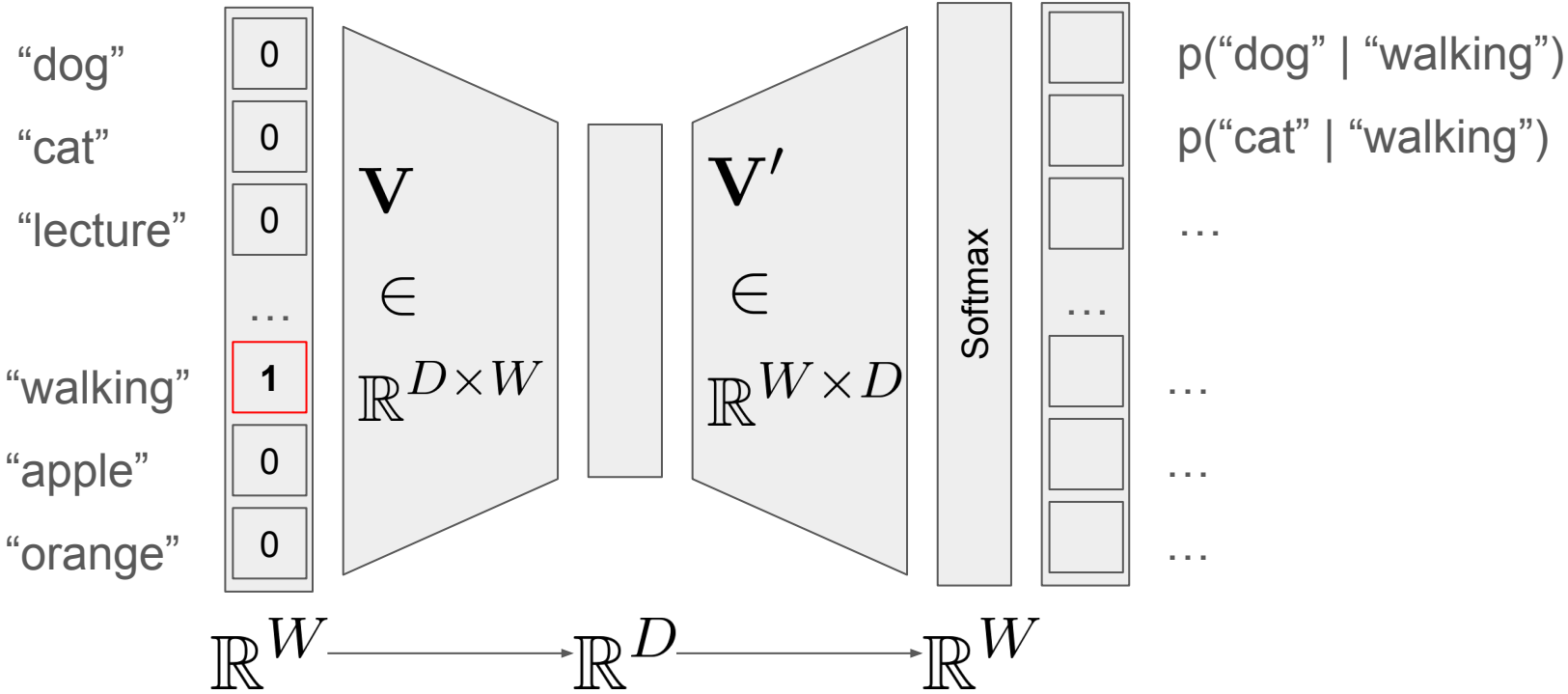
$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

1. For every position in the input corpus

2. maxmize the sum of seeing it's context

log(p("cat" | "walking")) + log(p("bedroom" | "walking"))

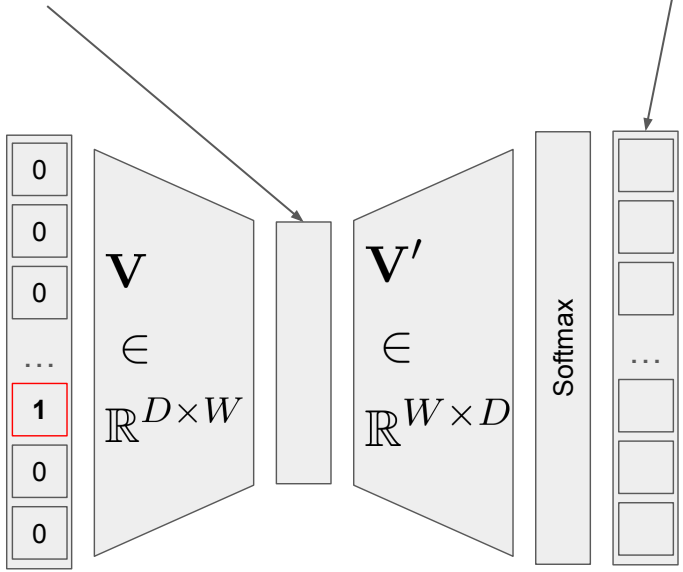
What does that look like?



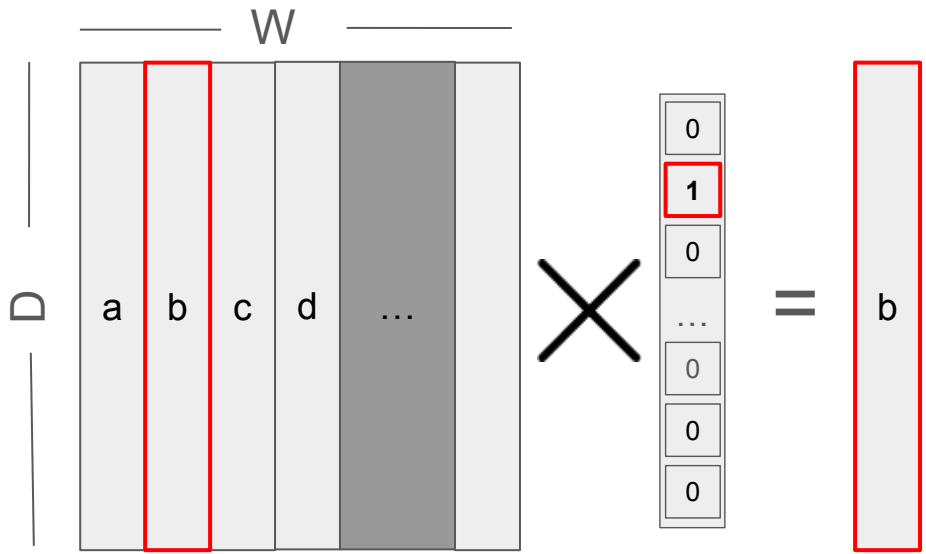
You Shall Know a Word by the Company it Keeps …

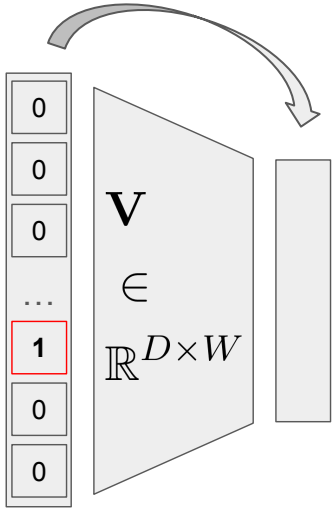
1. Words that have similar “company” have similar “labels”

2. So they need to be close-by in embedding space



Why that helps with computations …



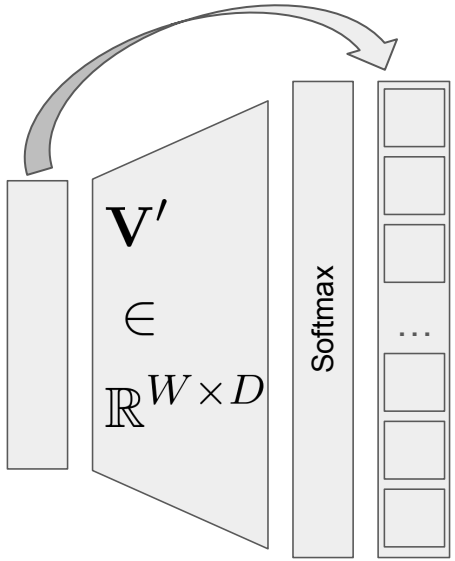


=> Input Space to Embedding is really easy, because of sparse input

… also V contains the embeddings directly after training

The other part is a bit less obvious …

- For each $p(w_{t+j}|w_t)$ in training, we need to evaluate the **ENTIRE** last layer (because of normalization in softmax) ⇒ there's quite a few values there …
- Tricks offered in the paper:
  - Hierarchical Softmax (reduce from cost W to log(W))
  - Negative Sampling
    - instead of caring about all words which aren't in the context (which are many), we take a random subsample of words which aren't in the context
    - task is now: separate the true context word from those negative samples



tl;dl

- task: predict context of words (instead of predict next word from sequence)
- simple linear architecture for individual probabilities
- some tricks to make it faster (negative sampling)
- you can now afford to train with much more training data, improving embedding quality

What's the deal with word2vec?

- Word embeddings ("distributed representations of words") for LM were already an idea for 10 years when word2vec was introduced …
- … so what was new in this work?

Greatly Simplified Training Tasks
(skip-gram & CBOW)

=> enables the use of much more training data

Direct Focus on Quality of Word Embeddings

=> (very) interesting syntactic and semantic relationships are discovered in embedding space

Word Embedding Evaluation until then

TABLE 1. JUDGED SYNONYMY OF THEME PAIRS

cord	smile	0.02	hill	woodland	1.48
rooster	voyage	0.04	car	journey	1.55
noon	string	0.04	cemetery	mound	1.69
fruit	furnace	0.05	glass	jewel	1.78
autograph	shore	0.06	magician	oracle	1.82
automobile	wizard	0.11	crane	implement	2.37
mound	stove	0.14	brother	lad	2.41
grin	implement	0.18	sage	wizard	2.46
asylum	fruit	0.19	oracle	sage	2.61
asylum	monk	0.39	bird	crane	2.63
graveyard	madhouse	0.42	bird	cock	2.63
glass	magician	0.44	food	fruit	2.69
boy	rooster	0.44	brother	monk	2.74
cushion	jewel	0.45	asylum	madhouse	3.04
monk	slave	0.57	furnace	stove	3.11
asylum	cemetery	0.79	magician	wizard	3.21
coast	forest	0.85	hill	mound	3.29
grin	lad	0.88	cord	string	3.41
shore	woodland	0.90	glass	tumbler	3.45
monk	oracle	0.91	grin	smile	3.46
boy	sage	0.96	serf	slave	3.46
automobile	cushion	0.97	journey	voyage	3.58
mound	shore	0.97	autograph	signature	3.59
lad	wizard	0.99	coast	shore	3.60
forest	graveyard	1.00	forest	woodland	3.65
food	rooster	1.09	implement	tool	3.66
cemetery	woodland	1.18	cock	rooster	3.68
shore	voyage	1.22	boy	lad	3.82
bird	woodland	1.24	cushion	pillow	3.84
coast	hill	1.26	cemetery	graveyard	3.88
furnace	implement	1.37	automobile	car	3.92
crane	rooster	1.41	midday	noon	3.94
			gem	jewel	3.94

Contextual Correlates of Synonymy

“Experimental corroboration was obtained for the hypothesis that the proportion of words common to the **contexts of word A** and to the **contexts of word B** is a function of **the degree to which A and B are similar in meaning**. …”

similar context -> similar output -> similar weights

Cup - Food	2.071
Cup - Liquid	2.000
Cup - Artifact	2.000

A:	B:
Husky - Stone	Husky - Corgi
Husky - Printer	Husky - Pitbull
Husky - Cat	Husky - Cat

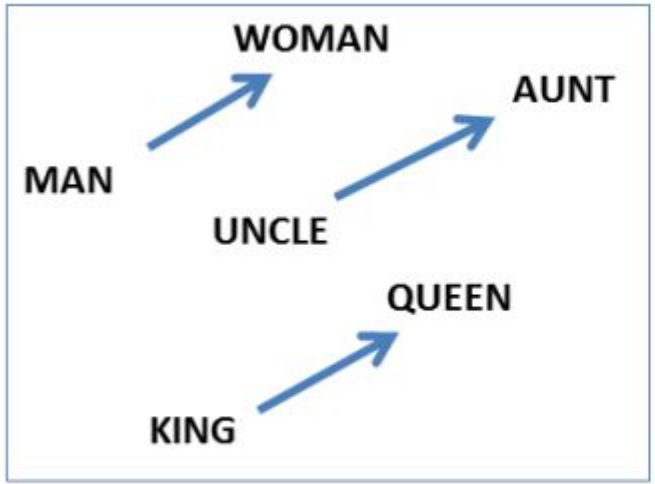
Tiger - Cat	2.840
Tiger - Animal	2.840
Tiger - Feline	5.071
Tiger - Mammal	5.923

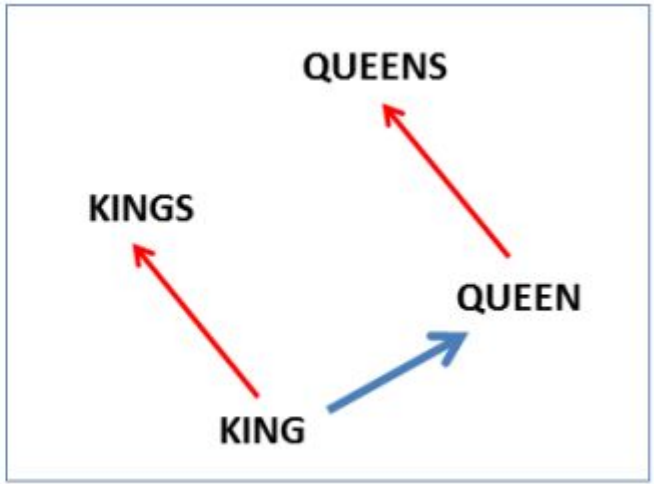
Mouse - Rat

Mouse - Keyboard

How to quantify more complex similarity?

What is the female version of a King?





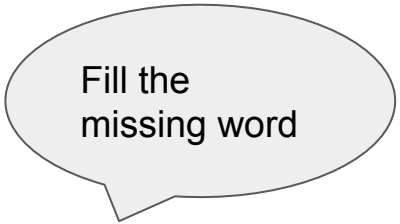
vec(“King”) - vec(“Man”) + vec(“Woman”)

Big - Bigger: Small - ? Syntactic

Germany - Berlin: France - ? Semantic

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza



Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

What about phrases?

New York Times, LA Lakers, Würzburger Panthers

Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

Model (training time)	Redmond	Havel	ninjutsu	graffiti	capitulate
Collobert (50d) (2 months)	conyers lubbock keene	plauen dzerzhinsky osterreich	reiki kohona karate	cheesecake gossip dioramas	abdicate accede rearm
Turian (200d) (few weeks)	McCarthy Alston Cousins	Jewell Arzu Ovitz	- - -	gunfire emotion impunity	- - -
Mnih (100d) (7 days)	Podhurst Harlang Agarwal	Pontiff Pinochet Rodionov	- - -	anaesthetics monkeys Jews	Mavericks planning hesitated
Skip-Phrase (1000d, 1 day)	Redmond Wash. Redmond Washington Microsoft	Vaclav Havel president Vaclav Havel Velvet Revolution	ninja martial arts swordsmanship	spray paint grafitti taggers	capitulation capitulated capitulating

Table 6: Examples of the closest tokens given various well known models and the Skip-gram model trained on phrases using over 30 billion training words. An empty cell means that the word was not in the vocabulary.