

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1

# Lecture 4: Document Representation

Pilsung Kang

School of Industrial Management Engineering

Korea University

# AGENDA

01 Bag of Words

---

02 Word Weighting

---

03 N-Grams

---

04 Distributed Representations

---

05 R Exercise

---

# What We Have Done So Far...

## Collecting Text Data

Cornell University Library  
We gratefully acknowledge support from the Simons Foundation and member institutions

arXiv.org > search

Search or Article ID All papers

(Help | Advanced search)

### arXiv.org Search Results

Back to Search form | Next 25 results

The URL for this search is [http://arxiv.org:443/find/all/1/all:+EXACT+text\\_mining/0/1/0/all/0](http://arxiv.org:443/find/all/1/all:+EXACT+text_mining/0/1/0/all/0)

Showing results 1 through 25 (of 168 total) for all:"text mining"

1. arXiv:1703.05692 [pdf, other]  
**OncScore: a novel, Internet-based tool to assess the oncogenic potential of genes**  
Rocco Piazza, Daniele Ramazzotti, Roberta Spinelli, Alessandra Pirola, Luca De Sano, Pierangelo Ferrari, Vera Magistroni, Nicoletta Cordani, Nitesh Sharma, Carlo Gambacorti-Passerni  
Subjects: Genomics (q-bio.GN); Quantitative Methods (q-bio.QM)

2. arXiv:1703.04213 [pdf, other]  
**MetaPAD: Meta Pattern Discovery from Massive Text Corpora**  
Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M. Kaplan, Timothy P. Hanratty, Jiawei Han  
Comments: 9 pages  
Subjects: Computation and Language (cs.CL)

3. arXiv:1703.02819 [pdf, other]  
**Introduction to Formal Concept Analysis and Its Applications in Information Retrieval and Related Fields**  
Dmitry I. Ignatov  
Journal-ref: RussIR 2014, Nizhniy Novgorod, Russia, CCIS vol. 505, Springer 42-141  
Subjects: Information Retrieval (cs.IR); Artificial Intelligence (cs.AI); Computation and Language (cs.CL); Discrete Mathematics (cs.DM); Machine Learning (stat.ML)

4. arXiv:1702.07117 [pdf, other]  
**LTSg: Latent Topical Skip-Gram for Mutually Learning Topic Model and Vector Representations**  
Jianan Law, Hankui Zhuo, Junhua He, Erhu Rong (Dept. of Computer Science, Sun Yat-Sen University, GuangZhou, China.)  
Subjects: Computation and Language (cs.CL)

5. arXiv:1702.03519 [pdf, ps, other]  
**A Technical Report: Entity Extraction using Both Character-based and Token-based Similarity**  
Zeyi Wen, Dong Deng, Rui Zhang, Kotagiri Ramamohanarao  
Comments: 12 pages, 6 figures, technical report  
Subjects: Databases (cs.DB)



The complicated, evolving landscape of cancer

Mining textual patterns in news, tweets, papers, and

This paper is a tutorial on Formal Concept Analysis (FCA) and its applications. FCA is an applied branch of Lattice Theory, a mathematical discipline which enables formalisation of concepts as basic units of human thinking and analysing data in the object-attribute form. Originated in early 80s, during the last three decades, it became a popular human-centred tool for knowledge representation and data analysis with numerous applications. Since the tutorial was specially prepared for RuSSIR 2014, the covered FCA topics include Information Retrieval with a focus on visualisation aspects, Machine Learning, Data Mining and Knowledge Discovery, Text Mining and several others.

pattern quality assessment function, which avoids costly dependency parsing and generates high-quality patterns; (2) it identifies and groups synonymous meta patterns from multiple facets---their types, contexts, and extractions; and (3) it examines type distributions of entities in the instances extracted by each group of patterns, and looks for appropriate type levels to make discovered patterns precise. Experiments demonstrate that our proposed framework discovers high-quality typed textual patterns efficiently from different genres of massive corpora and facilitates information extraction.

# What We Have Done So Far...

## Preprocessing with some NLP techniques

The complicated, evolving landscape of cancer

Mining textual patterns in news, tweets, papers, and

This paper is a tutorial on Formal Concept Analysis (FCA) and its applications. FCA is an applied branch of Lattice Theory, a mathematical discipline which enables formalisation of concepts as basic units of human thinking and analysing data in the object-attribute form. Originated in early 80s, during the last three decades, it became a popular human-centred tool for knowledge representation and data analysis with numerous applications. Since the tutorial was specially prepared for RuSSIR 2014, the covered FCA topics include Information Retrieval with a focus on visualisation aspects, Machine Learning, Data Mining and Knowledge Discovery, Text Mining and several others.

pattern quality assessment function, which avoids costly dependency parsing and generates high-quality patterns; (2) it identifies and groups synonymous meta patterns from multiple facets---their types, contexts, and extractions; and (3) it examines type distributions of entities in the instances extracted by each group of patterns, and looks for appropriate type levels to make discovered patterns precise. Experiments demonstrate that our proposed framework discovers high-quality typed textual patterns efficiently from different genres of massive corpora and facilitates information extraction.

the complic evolv landscap of cancer mutat pose a

formine textual pattern in news tweet paper and mani

list oth pric tex to a dep too the base on curv and curat da oncosco oncosco priorit o

# What We Will Do...

Transform unstructured data into structured data

the complic evolv landscap of cancer mutat pose a  
form mine textual pattern in news tweet paper and mani  
list oth  
prior tex  
to a dep  
too the  
base on  
curv and  
curat da  
oncosco  
oncosco  
priorit o

this paper is a tutori on formal concept analysi fca and it applic fca is an appli branch of lattic theori a mathemat disciplin which enabl formalis of concept as basic unit of human think and analys data in the objectattribut form origin in earli s dure the last three decad it becam a popular humancentr tool for knowledg represent and data analysi with numer applic sinc the tutori was special prepar for russir the cover fca topic includ inform retriev with a focus on visualis aspect machin learn data mine and knowledg discoveri text mine and sever other

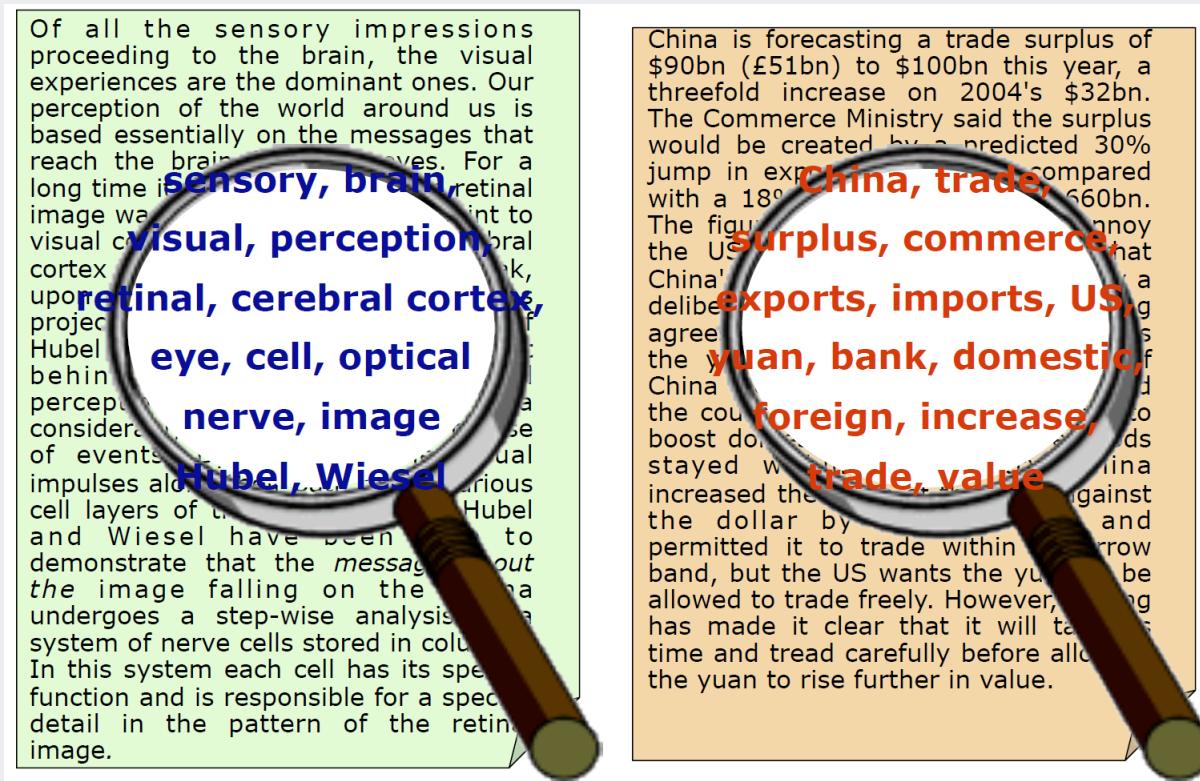


	Var 1	Var 2	...	...	Var P
Doc 1					
Doc 2					
Doc 3					
...					
...					
...					
Doc D					

# Bag of Words: Motivation

- Document Representation

- ✓ How to represent a document in a structured way?
- ✓ How to convert a unstructured text into a matrix form to apply those machine learning algorithms based on a vector space?



# Bag of Words: Idea

박은정 (2016)

- Bag-of-words
    - ✓ Simplifying representation method for documents where a text is represented as a vector of an unordered collection of words
    - ✓ Consider words as atomic symbols, represented in the discrete space

Ex:

```
five_random_documents = [sentences
    documents { 'i like this movie',
    'the movie hunger games is a trilogy movie',
    'jennifer lawrence is an excellent actor',
    'i would give the film an 8 out of 10',
    'you can observe some jaw-dropping cleverness' }]
```

# Bag of Words: Idea

- Bag-of-words: Term-Document Matrix

- ✓ Simplifying representation method for documents where a text is represented in a vector of an unordered collection of words

**S<sub>1</sub>:** John likes to watch movies. Mary likes too.

**S<sub>2</sub>:** John also likes to watch football game.

Binary representation

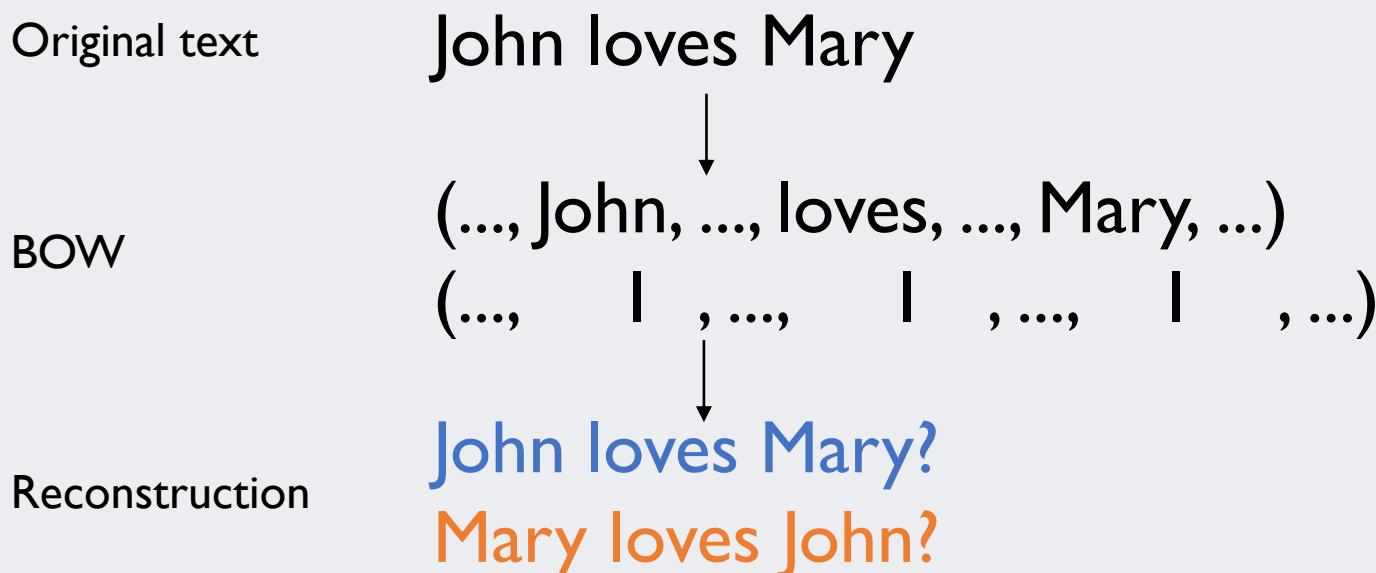
Word	S <sub>1</sub>	S <sub>2</sub>
John	1	1
Likes	1	1
To	1	1
Watch	1	1
Movies	1	0
Also	0	1
Football	0	1
Games	0	1
Mary	1	0
too	1	0

Frequency representation

Word	S <sub>1</sub>	S <sub>2</sub>
John	1	1
Likes	2	1
To	1	1
Watch	1	1
Movies	1	0
Also	0	1
Football	0	1
Games	0	1
Mary	1	0
too	1	0

# Bag of Words: Idea

- Bag of words Representation in a Vector Space
  - ✓ The contents can be inferred from the frequency of words
  - ✓ Vector representation **does not consider the ordering of words** in a document
    - Visual words = independent features
    - John is quicker than Mary = Mary is quicker than John in BOW representation
  - ✓ We cannot reconstruct the original text based on the term-document matrix



# Text Preprocessing

- Remove unnecessary information
  - ✓ They vs. they: different words in many systems
    - lower case is commonly used
  - ✓ Punctuation
    - Punctuations do not contain significant information → Remove them!
  - ✓ Numbers
    - Numbers are not critical in some domains but critical in other domains
    - Removing numbers should be carefully determined based on the domain for which a collection of text is about to be analyzed

# Stop Words

- What are stop words?

✓ Words that **do not carry any information**

- Mainly functional role
- Usually remove them to help the machine learning algorithms to perform better

✓ Natural language dependent

- English: a, about, above, across, after, again, against, all, also, etc.
- 한국어: ...습니다, ...로서(써), ...를 등

[Original text]

Information Systems Asia Web - provides research, IS-related commercial materials, interaction, **and even** research sponsorship **by** interested corporations **with a focus on** Asia Pacific region.

[After removing stop words]

Information Systems Asia Web provides research IS-related commercial materials interaction research sponsorship interested corporations focus Asia Pacific region

# Stop Words

- Example I: SMART stop words list

✓ SMART: System for the Mechanical Analysis and Retrieval of Text

- A total of 571 stop words

[1]	"a"	"a's"	"able"	"about"	"above"	"according"	"accordingly"	"across"	"actually"	"after"	"afterwards"
[12]	"again"	"against"	"ain't"	"all"	"allow"	"allows"	"almost"	"alone"	"along"	"already"	"also"
[23]	"although"	"always"	"am"	"among"	"amongst"	"an"	"and"	"another"	"any"	"anybody"	"anyhow"
[34]	"anyone"	"anything"	"anyway"	"anyways"	"anywhere"	"apart"	"appear"	"appreciate"	"appropriate"	"are"	"aren't"
[45]	"around"	"as"	"aside"	"ask"	"asking"	"associated"	"at"	"available"	"away"	"awfully"	"b"
[56]	"be"	"became"	"because"	"become"	"becomes"	"becoming"	"been"	"before"	"beforehand"	"behind"	"being"
[67]	"believe"	"below"	"beside"	"besides"	"best"	"better"	"between"	"beyond"	"both"	"brief"	"but"
[78]	"by"	"c"	"c'mon"	"c's"	"came"	"can"	"can't"	"cannot"	"cant"	"cause"	"causes"
[89]	"certain"	"certainly"	"changes"	"clearly"	"co"	"com"	"come"	"comes"	"concerning"	"consequently"	"consider"
[100]	"considering"	"contain"	"containing"	"contains"	"corresponding"	"could"	"couldn't"	"course"	"currently"	"d"	"definitely"
[111]	"described"	"despite"	"did"	"didn't"	"different"	"do"	"does"	"doesn't"	"doing"	"don't"	"done"
[122]	"down"	"downwards"	"during"	"e"	"each"	"edu"	"eg"	"eight"	"either"	"else"	"elsewhere"
[133]	"enough"	"entirely"	"especially"	"et"	"etc"	"even"	"ever"	"every"	"everybody"	"everyone"	"everything"
[144]	"everywhere"	"ex"	"exactly"	"example"	"except"	"f"	"far"	"few"	"fifth"	"first"	"five"
[155]	"followed"	"following"	"follows"	"for"	"former"	"formerly"	"forth"	"four"	"from"	"further"	"furthermore"
[166]	"g"	"get"	"gets"	"getting"	"given"	"gives"	"go"	"goes"	"going"	"gone"	"got"
[177]	"gotten"	"greetings"	"h"	"had"	"hadn't"	"happens"	"hardly"	"has"	"haven't"	"have"	"hereby"
[188]	"having"	"he"	"he's"	"hello"	"help"	"hence"	"her"	"here"	"hereafter"	"hither"	"how"
[199]	"herein"	"hereupon"	"hers"	"herself"	"hi"	"him"	"himself"	"his"	"hopefully"	"ignored"	"immediate"
[210]	"howbeit"	"however"	"i"	"i'd"	"i'll"	"i'm"	"i've"	"ie"	"if"	"instead"	"into"
[221]	"in"	"inasmuch"	"inc"	"indeed"	"indicate"	"indicated"	"indicates"	"inner"	"itself"	"j"	"just"
[232]	"inward"	"is"	"isn't"	"it"	"it'd"	"it'll"	"it's"	"its"	"last"	"lately"	"later"
[243]	"k"	"keep"	"keeps"	"kept"	"know"	"knows"	"known"	"known"	"liked"	"likely"	"little"
[254]	"latter"	"latterly"	"least"	"less"	"lest"	"let"	"let's"	"like"	"maybe"	"me"	"mean"
[265]	"look"	"looking"	"looks"	"ltd"	"m"	"mainly"	"many"	"may"	"must"	"my"	"myself"
[276]	"meanwhile"	"merely"	"might"	"more"	"moreover"	"most"	"mostly"	"much"	"needs"	"neither"	"never"
[287]	"n"	"name"	"namely"	"nd"	"near"	"nearly"	"necessary"	"need"	"noone"	"nor"	"normally"
[298]	"nevertheless"	"new"	"next"	"nine"	"no"	"nobody"	"non"	"none"	"off"	"often"	"oh"
[309]	"not"	"nothing"	"novel"	"now"	"nowhere"	"o"	"obviously"	"of"	"onto"	"or"	"other"
[320]	"ok"	"okay"	"old"	"on"	"once"	"one"	"ones"	"only"	"over"	"overall"	"own"
[331]	"others"	"otherwise"	"ought"	"our"	"ours"	"ourselves"	"out"	"outside"	"possible"	"presumably"	"probably"
[342]	"p"	"particular"	"particularly"	"per"	"perhaps"	"placed"	"please"	"plus"	"re"	"really"	"reasonably"
[353]	"provides"	"q"	"que"	"quite"	"qv"	"r"	"rather"	"rd"	"same"	"saw"	"say"
[364]	"regarding"	"regardless"	"regards"	"relatively"	"respectively"	"right"	"s"	"said"	"seem"	"seen"	"seen"
[375]	"saying"	"says"	"second"	"secondly"	"see"	"seeing"	"seem"	"seemed"	"several"	"shall"	"she"
[386]	"self"	"selves"	"sensible"	"sent"	"serious"	"seriously"	"seven"	"several"	"something"	"should"	"sometimes"
[397]	"shouldn't"	"since"	"six"	"so"	"some"	"somebody"	"somehow"	"someone"	"sub"	"such"	"sup"
[408]	"somewhat"	"somewhere"	"soon"	"sorry"	"specified"	"specify"	"specifying"	"still"	"than"	"thank"	"thanks"
[419]	"sure"	"t"	"t's"	"take"	"taken"	"tell"	"tends"	"thi"	"themselves"	"then"	"thence"
[430]	"thanx"	"that"	"that's"	"thats"	"the"	"their"	"theirs"	"them"	"these"	"they"	"they'd"
[441]	"there"	"there's"	"thereafter"	"thereby"	"therefore"	"therein"	"theres"	"thereupon"	"those"	"though"	"three"
[452]	"they'll"	"they're"	"they've"	"think"	"third"	"this"	"thorough"	"thoroughly"	"toward"	"towards"	"tried"
[463]	"through"	"throughout"	"thru"	"thus"	"to"	"together"	"too"	"took"	"under"	"unfortunately"	"unless"
[474]	"tries"	"truly"	"try"	"trying"	"twice"	"two"	"u"	"un"	"uses"	"using"	"using"
[485]	"unlikely"	"until"	"unto"	"up"	"upon"	"us"	"use"	"used"	"w"	"want"	"want"
[496]	"usually"	"uucp"	"v"	"value"	"various"	"very"	"via"	"viz"	"vs"	"welcome"	"well"
[507]	"wants"	"was"	"wasn't"	"way"	"we"	"we'd"	"we'll"	"we're"	"where"	"where's"	"where's"
[518]	"went"	"were"	"weren't"	"what"	"what's"	"whatever"	"when"	"whence"	"whichever"	"whither"	"who"
[529]	"whereafter"	"whereas"	"whereby"	"wherein"	"whereupon"	"wherever"	"whether"	"which"	"while"	"with"	"within"
[540]	"who's"	"whoever"	"whole"	"whom"	"whose"	"why"	"will"	"willing"	"with"	"with"	"you"
[551]	"without"	"won't"	"wonder"	"would"	"would"	"wouldn't"	"x"	"y"	"yes"	"yet"	"zero"
[562]	"you'd"	"you'll"	"you're"	"you've"	"your"	"yours"	"yourself"	"yourselves"	"z"		

# Stop Words

- Example 2: MySQL Stop words list

✓ <http://dev.mysql.com/doc/refman/5.1/en/fulltext-stopwords.html>

- A total of 543 stop words

a's	able	about	above	according	her	here	here's	hereafter	hereby	serious	seriously	seven	several	shall
accordingly	across	actually	after	afterwards	herein	hereupon	hers	herself	hi	she	should	shouldn't	since	six
again	against	ain't	all	allow	him	himself	his	hither	hopefully	so	some	somebody	somewhat	somewhere
allows	almost	alone	along	already	how	howbeit	however	i'd	i'll	something	sometime	sometimes	somewhat	somewhere
also	although	always	am	among	i'm	i've	ie	if	ignored	soon	sorry	specified	specify	specifying
amongst	an	and	another	any	immediate	in	inasmuch	inc	indeed	still	sub	such	sup	sure
anybody	anyhow	anyone	anything	anyway	indicate	indicated	indicates	inner	insofar	t's	take	taken	tell	tends
anyways	anywhere	apart	appear	appreciate	instead	into	inward	is	isn't	th	than	thank	thanks	thanx
appropriate	are	aren't	around	as	it	it'd	it'll	it's	its	that	that's	thats	the	their
aside	ask	asking	associated	at	itself	just	keep	keeps	kept	theirs	them	themselves	then	thence
available	away	awfully	be	became	know	known	knows	last	lately	there	there's	thereafter	thereby	therefore
because	become	becomes	becoming	been	later	latter	latterly	least	less	therein	theres	thereupon	these	they
before	beforehand	behind	being	believe	lest	let	let's	like	liked	they'd	they'll	they're	they've	think
below	beside	besides	best	better	likely	little	look	looking	looks	third	this	thorough	thoroughly	those
between	beyond	both	brief	but	ltd	mainly	many	may	maybe	though	three	through	throughout	thru
by	c'mon	c's	came	can	me	mean	meanwhile	merely	might	thus	to	together	too	took
can't	cannot	cant	cause	causes	more	moreover	most	mostly	much	toward	towards	tried	tries	truly
certain	certainly	changes	clearly	co	must	my	myself	name	namely	try	trying	twice	two	un
com	come	comes	concerning	consequently	nd	near	nearly	necessary	need	under	unfortunately	unless	unlikely	until
consider	considering	contain	containing	contains	needs	neither	never	nevertheless	new	unto	up	upon	us	use
corresponding	could	couldn't	course	currently	next	nine	no	nobody	non	used	useful	uses	using	usually
definitely	described	despite	did	didn't	none	noone	nor	normally	not	value	various	very	via	viz
different	do	does	doesn't	doing	nothing	novel	now	nowhere	obviously	vs	want	wants	was	wasn't
don't	done	down	downwards	during	of	off	often	oh	ok	way	we	we'd	we'll	we're
each	edu	eg	eight	either	okay	old	on	once	one	we've	welcome	well	went	were
else	elsewhere	enough	entirely	especially	ones	only	onto	or	other	weren't	what	what's	whatever	when
et	etc	even	ever	every	others	otherwise	ought	our	ours	whence	whenever	where	where's	whereafter
everybody	everyone	everything	everywhere	ex	ourselves	out	outside	over	overall	whereas	whereby	wherein	whereupon	wherever
exactly	example	except	far	few	own	particular	particularly	per	perhaps	whether	which	while	whither	who
fifth	first	five	followed	following	placed	please	plus	possible	presumably	who's	whoever	whole	whom	whose
follows	for	former	formerly	forth	probably	provides	que	quite	qv	why	will	willing	wish	with
four	from	further	furthermore	get	rather	rd	re	really	reasonably	within	without	won't	wonder	would
gets	getting	given	gives	go	regarding	regardless	regards	relatively	respectively	wouldn't	yes	yet	you	you'd
goes	going	gone	got	gotten	right	said	same	saw	say	you'll	you're	you've	your	yours
greetings	had	hadn't	happens	hardly	saying	says	second	secondly	see	yourself	yourselves	zero		
has	hasn't	have	haven't	having	seeing	seem	seemed	seeming	seems					
he	he's	hello	help	hence	seen	self	selves	sensible	sent					

# Stop Words

- Example 3: Stop words list in Korean

✓ <http://www.ranks.nl/stopwords/korean>

- A total of 677 stop words

아	어찌됐든	하기보다는	뿐만 아니라 다시 말하자면	까닭으로	할 생각이다	즈음하여	본대로	얼마간	너	총자
휴	그위에	차라리	만이 아니나 바꿔 말하면	이유만으로	하려고하다	다른	자	약간	너희	자기
아이구	게다가	하는 편이 낫다	만은 아니다 즉	이로 인하여	이리하여	다른 방면으로	이	다소	당신	자기집
아이쿠	점에서 보아	흐흐	막恸하고 구체적으로	그래서	그리하여	해봐요	이쪽	줄	어찌	자신
아이고	비추어 보아	놀라다	관계없이 말하자면	이 때문에	그렇게 함으	습니까	여기	조금	설마	우에 종합한것과
어	고려하면	상대적으로 말하	그치지 않다 시작하여	그러므로	로써	했어요	이것	다수	차라리	같이
나	허가될것이다	자연	그러나 시초에	그런 까닭에	하지만	말할것도 없고	이번	몇	할지언정	총적으로 보면
우리	될것이다	마치	그런데 이상	할 수 있다	일때	무를쓰고	이렇게 말하자면	얼마	할지라도	총적으로 말하면
저희	비교적	아니라면	하지만 허	결론을 냄 수 있	할때	개의치않고	이런	지만	할망정	총적으로
따라	좀	첫	듣간에 혁	다	앞에서	하는것만 못하다	이러한	허물며	할지언정	대로 하다
의해	보다더	그렇지 않으면	논하지 않다 허걱	으로 인하여	중에서	하는것이 낫다	이와 같은	또한	구토하다	으로서
을	비하면	그렇지 않다면	따지지 않다 반와같이	있다	보는데서	매	요만큼	그러나	게우다	참
를	시키다	안 그러면	설사 해도좋다	어떤것	으로써	매번	요만한 것	그렇지만	토하다	그만이다
에	허가하다	아니었다면	비록 해도된다	관계가 있다	로써	흘	얼마 안 되는 것	하지만	매쓰겁다	할 따름이다
의	할만하다	하든지	더라도 게다가	관련이 있다	까지	모	이만큼	이외에도	영 사람	옹
가	의해서	아니면	아니면 더구나	연관되다	해야한다	어느것	이 정도의	대해 말하자 텐	탕탕	
으로	연이어서	이라면	만 못하다 하물며	어떤것들	일것이다	어느	이렇게 많은 것	면	쳇	팡팡
로	이어서	좋아	하는 편이 낫 와르르	반드시	로써	이와 같다	이와 같다	뿐이다	의거하여	뚱뚱
에게	잇따라	알았어	다	항종일각	할줄알다	이때	근거하여	디듬에	근거하여	뚱뚱
뿐이다	뒤따라	하는것도	불문하고 꽉	갖고말하자면	할수있다	어디	이럴구나	반대로	의해	보라
의가하여	뒤이어	그만이다	황하여 펠링	그리하여	할수있다	할수있다	것과 같이	반대로 말하 따라	아이야	
근가하여	결국	어쩔수 없다	황해서 동안	하고있다는	할수있어	어느쪽	끼익	자면	휩입어	
입각하여	의지하여	하나	향히다 이래	하느니	일때	것과 같이	것과 같이	그와 반대로 그	아니	
기준으로	기대여	일	족으로 하고있었다	하면 할수록	할수있었어	어느쪽	끼익	이와 반대로 그	와아	
예하면	통하여	일반적으로	틈타 이었다	운운	한다면	어느년도	빼걱	바꾸어서 말 디름	응	
예를 들면	자마자	일단	이용하여 에서	이리이리하다	다	언제나	빼걱	하면	버금	아이
예를 들자면	더운데	한번연으로는	된다 로부터	부류의 사람들	한다만	언제나	와 같은 사람들	바꾸어서 한 번째로	챔나	
저	불구하고	오자마자	오르다 까지	부류의 사람들	다만	언젠가	부류의 사람들	다면	기타	
소인	얼마든지	이렇게되면	제외하고 예하면	왜나하면	등등	제	왜나하면	만약	첫번째로	
소생	마음대로	이와같다면	이 외에 했어요	다시말하면	간 해도	제	중의하나	그렇지않으	첫번째로	
저희	주저하지 않고	전부	이 밖에 해요	다음으로	그간	언젠가	중의하나	나마지는	나마지는	
지말고	곧	한마다	하여야 함께	에 달려 있다	그간	언젠가	오직	면	그중에서	
하지마	즉시	한황록	비로소 갈이	우리	그간	언젠가	오로지	까악	견지에서	
하지마라	비로	근거로	한다면 몰라 더불어	우리를	그간	언젠가	에 환하다	로	형식으로 쓰여	
다른	당장	하기에	도 마저	대해서	그간	언젠가	하기만 하면	딱	입장에서	
물론	하자마자	아울러	외에도 마저도	그간	그간	언젠가	도착하다	빼걱거리다	위해서	
또한	밖에 안된다	하지 않도록	이곳 양자	그간	그간	언젠가	도착하다	보드득	단지	
그리고	하면된다	않기 위해서	여기 모두	그간	그간	언젠가	까지 미치다	비걱거리다	위해되다	
비길수 없다	그래	이로기까지	부터 습니다	그간	그간	언젠가	도달하다	깨닫	하도록시키다	
해서는 안된	그럴지	이 되다	기점으로 가까스로	그간	그간	언젠가	정도에 이르다	응당	뿐만아니라	
다	요컨대	로 인하여	따라서 하려고하다	여	그간	언젠가	할 지경이다	해야한다	반대로	
						여	결과에 이르다	여러분	전후	
							관해서는	하고 있다	전자	
							하고 있다	하고 있다	전자	
							할 힘이 있다	한 후		

• • •

# AGENDA

01 Bag of Words

---

02 Word Weighting

---

03 N-Grams

---

04 Distributed Representations

---

05 R Exercise

---

# Word Weighting: Term-Frequency (TF)

Nayak & Raghavan (2014)

- Term frequency  $tf_{t,d}$

- ✓ The number of times that the term **t** occurs in the document **d**



	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

# Word Weighting: Term-Frequency (TF)

이재천, 김수경, 홍성연 (2015)

- Term frequency  $tf_{t,d}$ 
    - ✓ The more frequently occurs, the more important it is

## <산공 강의 상위 25%>



## <산공 강의 하위 25%>



# Word Weighting: Document Frequency (DF)

- Document frequency  $df_t$ 
  - ✓ The number of documents in which the term  $t$  appears.
- Issues on DF
  - ✓ Rare terms are more informative than frequent terms across the document collection
    - is, can, the, of, ...
  - ✓ Consider a term in the query that is rare in the collection (e.g., Pneumonoultramicroscopicsilicovolcanoconiosis (longest word in English, ))
  - ✓ A document containing this term is very likely to be relevant to the query
  - ✓ We should give a high weight for rare terms than common terms

# Word Weighting: Inverse Document Frequency (IDF)

- Inverse document frequency  $\text{idf}_t$

✓  $\text{idf}_t = \log_{10}(N/\text{df}_t)$

✓ We use  $\log(N/\text{df}_t)$  instead of  $N/\text{df}_t$  to “dampen” the effect of idf

- IDF example with  $N = 1$  million

term	$\text{df}_t$	$\text{idf}_t$
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

# Word Weighting: TF-IDF

- TF-IDF

- ✓ TF-IDF weight of a term is the product of its tf weight and its idf weight

$$TF - IDF(w) = \underline{tf(w)} \times \log\left(\frac{N}{df(w)}\right)$$

More important if the term occur more frequently in a document

More important if the term occur less frequently in the other document

- ✓ Best known weighting scheme in information retrieval
- ✓ Increases with the number of occurrences within a document
- ✓ Increases with the rarity of the term in the collection

# Word Weighting: TF-IDF

Nayak & Raghavan (2014)

- Example revisited

- ✓ Each document is now represented by a real-valued vector of tf-idf weights in  $\mathbb{R}^{|V|}$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0.35
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

- ✓ So, we have a  $|V|$ -dimensional vector space

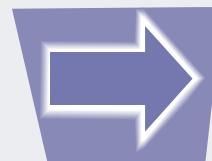
- Terms are axes of the space
- Documents are points or vectors in this space
- **Very high dimensional**: need to reduce the number of features!
- **Sparseness**: most entries are zero

# Word Weighting: TF-IDF

- TF-IDF Example

- ✓ Q1: Which term is the **most** important for the document 1?
- ✓ Q2: Which term is the **least** important for the document 1?

	Doc1	Doc2	Doc3
Term1	5	0	0
Term2	1	0	0
Term3	5	5	5
Term4	3	3	3
Term5	3	0	1



Doc1	TF	DF	IDF	TF-IDF
Term1	5	1	Log3	$5\log 3$
Term2	1	1	Log3	$1\log 3$
Term3	5	3	Log1	0
Term4	3	3	Log1	0
Term5	3	2	Log(3/2)	$3\log(3/2)$

Word weighting: Term 1 > Term 5 > Term 2 > Term 3 = Term 4

# TF Variants

Roelleke (2013)

- **TF Variants**

**Definition 2.1 TF Variants:**  $\text{TF}(t, d)$ .  $\text{TF}(t, d)$  is a quantification of the within-document term frequency,  $\text{tf}_d$ . The main variants are:

$$\text{tf}_d := \text{TF}_{\text{total}}(t, d) := \text{lf}_{\text{total}}(t, d) := \frac{n_L(t, d)}{N_L(d)} \quad (2.1)$$

$$\text{TF}_{\text{sum}}(t, d) := \text{lf}_{\text{sum}}(t, d) := \frac{n_L(t, d)}{n_L(d)} \quad \left( = \frac{\text{tf}_d}{\text{dl}} \right) \quad (2.2)$$

$$\text{TF}_{\text{max}}(t, d) := \text{lf}_{\text{max}}(t, d) := \frac{n_L(t, d)}{n_L(t_{\text{max}}, d)} \quad (2.3)$$

$$\text{TF}_{\text{log}}(t, d) := \text{lf}_{\text{log}}(t, d) := \log(1 + n_L(t, d)) \quad \left( = \log(1 + \text{tf}_d) \right) \quad (2.4)$$

$$\text{TF}_{\text{frac}, K}(t, d) := \text{lf}_{\text{frac}, K}(t, d) := \frac{n_L(t, d)}{n_L(t, d) + K_d} \quad \left( = \frac{\text{tf}_d}{\text{tf}_d + K_d} \right) \quad (2.5)$$

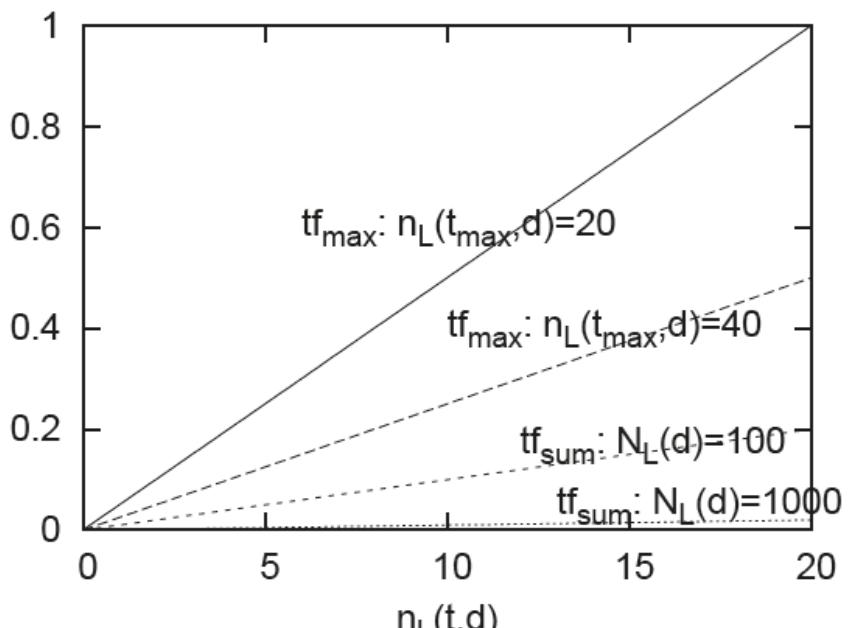
$$\text{TF}_{\text{BM25}, k_1, b}(t, d) := \text{lf}_{\text{BM25}, k_1, b}(t, d) := \frac{n_L(t, d)}{n_L(t, d) + k_1 \cdot (b \cdot \text{pivdl}(d, c) + (1 - b))} \quad (2.6)$$

- $K_d$ : (document length)/(average document length)

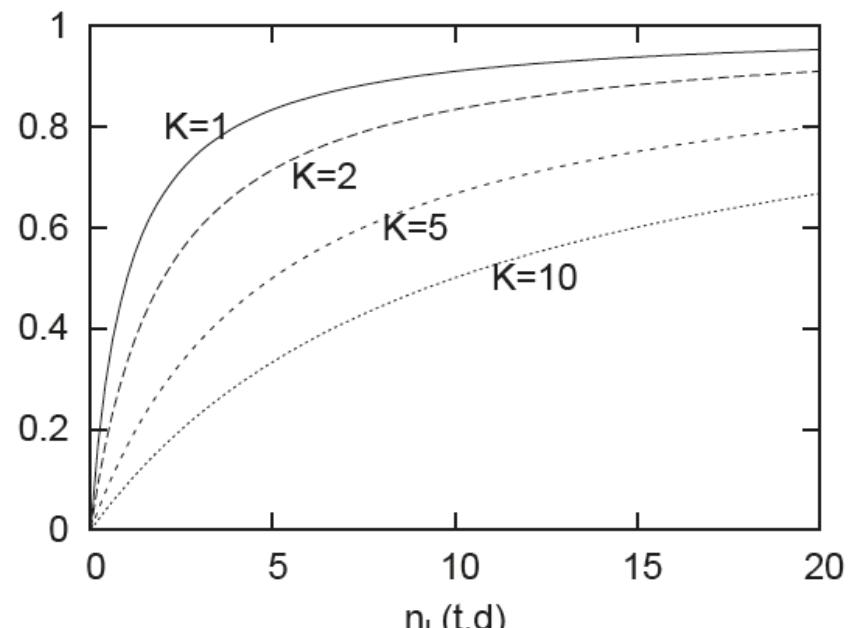
# TF Variants

Roelleke (2013)

- TF Variants



(a)  $\text{TF}_{\text{sum}}$  and  $\text{TF}_{\max}$



(b)  $\text{TF}_{\text{frac}}$

# DF & IDF Variants

Roelleke (2013)

- DF & IDF Variants

**Definition 2.3 DF Variants.** DF( $t, c$ ) is a quantification of the document frequency, df( $t, c$ ).  
The main variants are:

$$df(t, c) := df_{\text{total}}(t, c) := \frac{n_D(t, c)}{N_D(c)} \quad (2.18)$$

$$df_{\text{sum}}(t, c) := \frac{n_D(t, c)}{N_D(c)} \left( = \frac{df(t, c)}{N_D(c)} \right) \quad (2.19)$$

$$df_{\text{sum,smooth}}(t, c) := \frac{n_D(t, c) + 0.5}{N_D(c) + 1} \quad (2.20)$$

$$df_{\text{BIR}}(t, c) := \frac{n_D(t, c)}{N_D(c) - n_D(t, c)} \quad (2.21)$$

$$df_{\text{BIR,smooth}}(t, c) := \frac{n_D(t, c) + 0.5}{N_D(c) - n_D(t, c) + 0.5} \quad (2.22)$$

**Definition 2.4 IDF Variants.** IDF( $t, c$ ) is the negative logarithm of a DF quantification. The main variants are:

$$idf_{\text{total}}(t, c) := -\log df_{\text{total}}(t, c) \quad (2.23)$$

$$idf(t, c) := idf_{\text{sum}}(t, c) := -\log df_{\text{sum}}(t, c) \quad (2.24)$$

$$idf_{\text{sum,smooth}}(t, c) := -\log df_{\text{sum,smooth}}(t, c) \quad (2.25)$$

$$idf_{\text{BIR}}(t, c) := -\log df_{\text{BIR}}(t, c) \quad (2.26)$$

$$idf_{\text{BIR,smooth}}(t, c) := -\log df_{\text{BIR,smooth}}(t, c) \quad (2.27)$$

# TF-IDF Variants Summary

Roelleke (2013)

- The most commonly used TF-IDF in general

Term frequency	Document frequency	Normalization
n (natural) $tf_{t,d}$	n (no)      1	n (none)      1
I (logarithm) $1 + \log(tf_{t,d})$	t (idf) $\log \frac{N}{df_t}$	c (cosine) $\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (augmented) $0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf) $\max\{0, \log \frac{N - df_t}{df_t}\}$	u (pivoted unique) $1/u$
b (boolean) $\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$		b (byte size) $1/CharLength^\alpha, \alpha < 1$
L (log ave) $\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$		

# Effects of TF-IDF Variants

- Comparative Study (Paltoglou and Thelwall, 2010)
  - ✓ Task 1: Classification of 2,000 movie reviews: positive vs. negative
  - ✓ Task 2: Multi-Domain Sentiment Data set (MDSD)
    - Four different product types: books, electronics, DVDs, and kitchen appliances
    - 1,000 positive & 1,000 negative for each type, 8,000 in total

Term Frequency

Notation	Term frequency
n (natural)	$tf$
l (logarithm)	$1 + \log(tf)$
a (augmented)	$0.5 + \frac{0.5 \cdot tf}{\max_t(tf)}$
b (boolean)	$\begin{cases} 1, & tf > 0 \\ 0, & \text{otherwise} \end{cases}$
L (log ave)	$\frac{1+\log(tf)}{1+\log(\text{avg\_dl})}$
o (BM25)	$\frac{(k_1+1) \cdot tf}{k_1 \left( (1-b) + b \cdot \frac{df}{\text{avg\_dl}} \right) + tf}$

Inverse Document Frequency

Notation	Inverse Document Frequency
n (no)	1
t (idf)	$\log \frac{N}{df}$
p (prob idf)	$\log \frac{N-df}{df}$
k (BM25 idf)	$\log \frac{N-df+0.5}{df+0.5}$
$\Delta(t)$ (Delta idf)	$\log \frac{N_1 \cdot df_2}{N_2 \cdot df_1}$
$\Delta(t')$ (Delta smoothed idf)	$\log \frac{N_1 \cdot df_2 + 0.5}{N_2 \cdot df_1 + 0.5}$
$\Delta(p)$ (Delta prob idf)	$\log \frac{(N_1 - df_1) \cdot df_2}{df_1 \cdot (N_2 - df_2)}$
$\Delta(p')$ (Delta smoothed prob idf)	$\log \frac{(N_1 - df_1) \cdot df_2 + 0.5}{(N_2 - df_2) \cdot df_1 + 0.5}$
$\Delta(k)$ (Delta BM25 idf)	$\log \frac{(N_1 - df_1 + 0.5) \cdot df_2 + 0.5}{(N_2 - df_2 + 0.5) \cdot df_1 + 0.5}$

Normalization

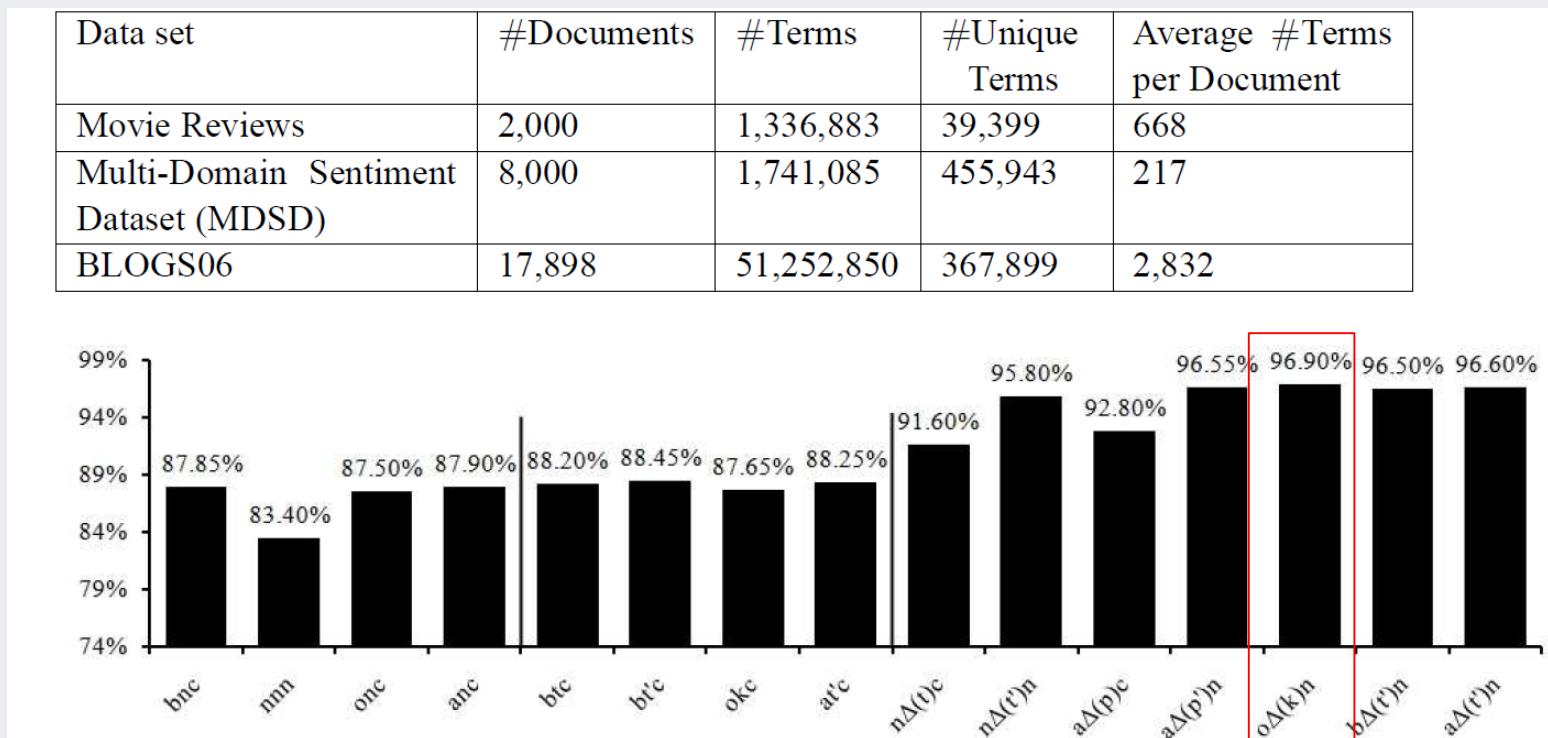
Notation	Normalization
n (none)	1
c (cosine)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_n^2}}$

# Effects of TF-IDF Variants

Paltoglou and Thelwall (2010)

- Experimental Result 1: Movie Reviews

- ✓ Base classifier: support vector machine (SVM)



$$o(\text{BM25}) = \frac{(k_1+1) \cdot tf}{k_1 \left( (1-b) + b \cdot \frac{dl}{avg\_dl} \right) + tf}$$

$$\Delta(k) \text{ (Delta BM25 idf)} = \log \frac{(N_1 - df_1 + 0.5) \cdot df_2 + 0.5}{(N_2 - df_2 + 0.5) \cdot df_1 + 0.5}$$

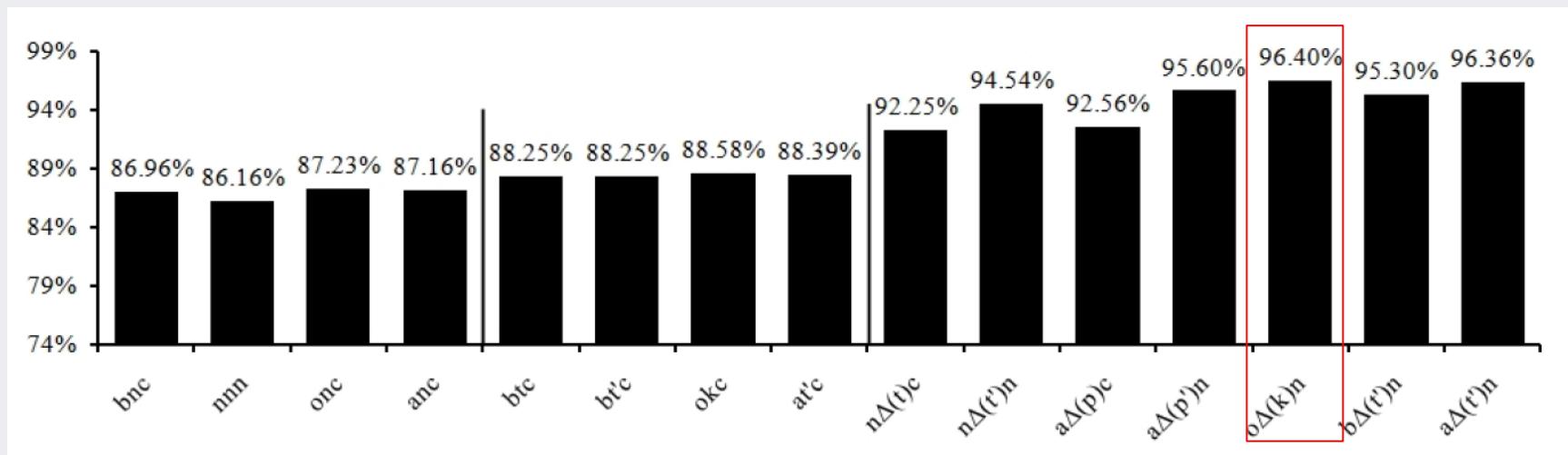
n (none)	1
----------	---

# Effects of TF-IDF Variants

Paltoglou and Thelwall (2010)

- Experimental Result 2: MDSD

- ✓ Base classifier: support vector machine (SVM)



$$o \text{ (BM25)} = \frac{(k_1+1) \cdot tf}{k_1 \left( (1-b) + b \cdot \frac{dl}{avg\_dl} \right) + tf}$$

$$\Delta(k) \text{ (Delta BM25 idf)} = \log \frac{(N_1 - df_1 + 0.5) \cdot df_2 + 0.5}{(N_2 - df_2 + 0.5) \cdot df_1 + 0.5}$$

n (none)	1
----------	---

# AGENDA

01 Bag of Words

---

02 Word Weighting

---

03 N-Grams

---

04 Distributed Representations

---

05 R Exercise

---

# N-Grams

- N-Gram-based Language Models in NLP

- ✓ Use the previous N-1 words in a sequence to predict the next word

$$P(w_n | w_{n-1}, w_{n-2}, \dots, w_1) = \frac{P(w_n, w_{n-1}, w_{n-2}, \dots, w_1)}{P(w_{n-1}, w_{n-2}, \dots, w_1)}$$

- ✓ Q) One of the hottest topics in artificial intelligence is deep \_\_\_\_\_

- blue vs. frying vs. learning ?

- N-Gram in Text Mining

- ✓ Some phrases are very useful in text clustering/categorization!

- Six sigma, supply chain management, big data, etc.

- ✓ Term-frequency for n-grams can be utilized.

- ✓ Domain-dependent.

# N-Grams

- Bigram example
- ✓ Total counts in a corpus

	<b>i</b>	<b>want</b>	<b>to</b>	<b>eat</b>	<b>chinese</b>	<b>food</b>	<b>lunch</b>	<b>spend</b>
<b>i</b>	5	827	0	9	0	0	0	2
<b>want</b>	2	0	608	1	6	6	5	1
<b>to</b>	2	0	4	686	2	0	6	211
<b>eat</b>	0	0	2	0	16	2	42	0
<b>chinese</b>	1	0	0	0	0	82	1	0
<b>food</b>	15	0	15	0	1	4	0	0
<b>lunch</b>	2	0	0	0	0	1	0	0
<b>spend</b>	1	0	1	0	0	0	0	0

# N-Grams

Furnkranz (1998)

- Empirical evaluation

- ✓ Data sets

- 20 newsgroup data set: 20,000 articles (1,000 for each category)
    - 21578 REUTERS newswire articles: 21,578 articles with 90 categories

- ✓ Classification algorithm: RIPPER

- Results for 20 newsgroup dataset

Pruning	<i>n</i> -grams	Error rate	CPU secs.	No. Features
set-of-words		$47.07 \pm 0.92$	n.a.	71,731
DF: 3	1	$46.18 \pm 0.94$	12686.12	36,534
	2	$45.28 \pm 0.51$	15288.32	113,716
TF: 5	3	$45.05 \pm 1.22$	15253.27	155,184
	4	$45.18 \pm 1.17$	14951.17	189,933
DF: 5	1	$45.51 \pm 0.83$	12948.31	22,573
	2	$45.34 \pm 0.68$	13280.73	44,893
TF: 10	3	$46.11 \pm 0.73$	12995.66	53,238
	4	$46.11 \pm 0.72$	13063.68	59,455
DF: 10	1	$45.88 \pm 0.89$	10627.10	13,805
	2	$45.53 \pm 0.86$	13080.32	20,295
TF: 20	3	$45.58 \pm 0.87$	11640.18	22,214
	4	$45.74 \pm 0.62$	11505.92	23,565

DF: 25	1	$48.23 \pm 0.69$	10676.43	n.a.
TF: 50	2	$48.97 \pm 1.15$	8870.05	n.a.
	3	$48.69 \pm 1.04$	10141.25	n.a.
	4	$48.36 \pm 1.01$	10436.58	n.a.
	5	$48.36 \pm 1.01$	10462.65	n.a.
DF: 50	1	$51.54 \pm 0.60$	8547.43	n.a.
	2	$49.71 \pm 0.53$	8164.27	n.a.
TF: 100	3	$51.21 \pm 1.26$	8079.59	n.a.
	4	$51.21 \pm 1.26$	8078.55	n.a.
	5	$51.21 \pm 1.26$	8147.75	n.a.
DF: 75	1	$52.59 \pm 0.71$	6609.05	n.a.
	2	$52.83 \pm 0.25$	6532.80	n.a.
TF: 150	3	$52.36 \pm 0.48$	6128.49	n.a.
	4	$52.36 \pm 0.48$	6128.49	n.a.
	5	$52.36 \pm 0.48$	6119.27	n.a.

# N-Grams

Furnkranz (1998)

- Results for 21578 REUTERS

- ✓ Classification accuracy is the highest with bigram features

Pruning	<i>n</i> -grams	Recall	Precision	F1	Accuracy	No. Features
set-of-words		76.71	83.42	79.92	99.5140	n.a.
	1	77.22	83.55	80.26	99.5211	9,673
DF: 3	2	80.34	82.03	81.18	99.5302	28,045
TF: 5	3	77.56	82.74	80.07	99.5130	38,646
	4	78.18	82.31	80.19	99.5130	45,876
	1	77.19	83.65	80.29	99.5221	6,332
DF: 5	2	80.05	82.06	81.04	99.5278	13,598
TF: 10	3	77.96	82.29	80.07	99.5106	17,708
	4	78.21	82.13	80.12	99.5106	20,468
	1	76.92	83.99	80.30	99.5241	4,068
DF: 10	2	79.06	82.04	80.52	99.5177	7,067
TF: 20	3	77.32	82.67	79.91	99.5096	8,759
	4	76.98	82.91	79.84	99.5096	9,907

# AGENDA

01 Bag of Words

---

02 Word Weighting

---

03 N-Grams

---

04 Distributed Representations

---

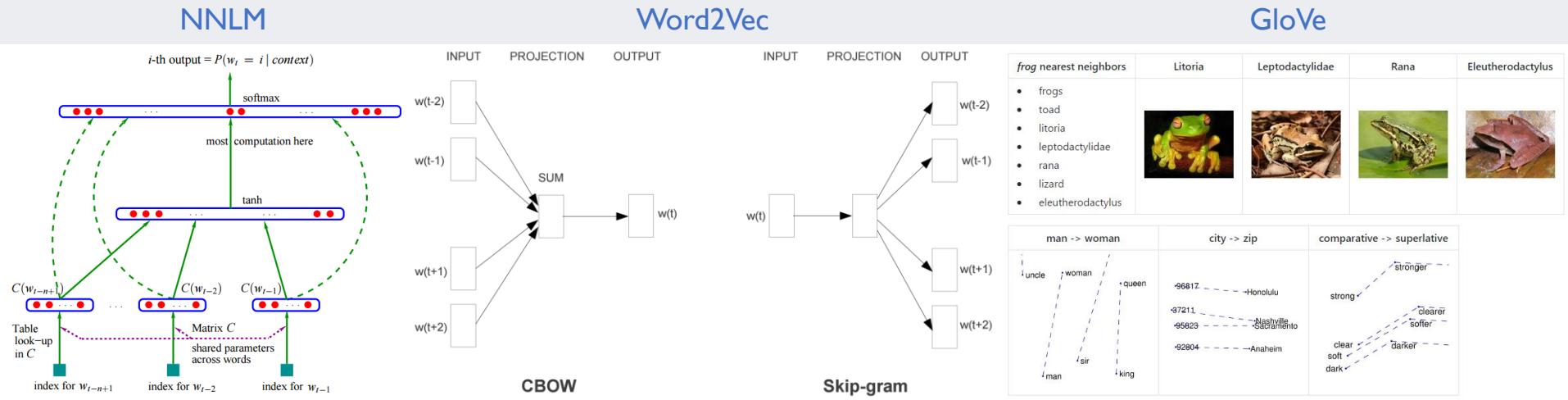
05 R Exercise

---

# Distributed Representation: Word Embedding

- Word Embedding

- ✓ The purpose of word embedding is to map the words in a language into a vector space so that semantically similar words are located close to each other.
- ✓ Hypothetically, the number of token in English is estimated about 13M, there exist a d (< 13M) dimensional optimal space that can embed the meaning of all words.



# Distributed Representation: Word Embedding

cs224d Lecture 2

- Word vectors: one-hot vector
  - The most simple & intuitive representation

$$w^{aardvark} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^a = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{at} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots w^{zebra} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

- Can make a vector representation, but similarities between words cannot be preserved.

motel [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND  
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0] = 0

$$(w^{hotel})^\top w^{motel} = (w^{hotel})^\top w^{cat} = 0$$

# Distributed Representation: Word Embedding

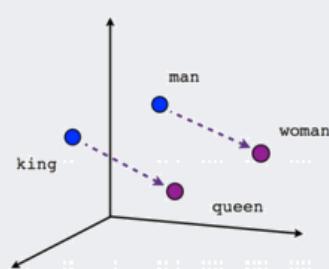
- Word vectors: distributed representation
  - ✓ A parameterized function mapping words in some language to a certain dimensional vectors

$$W : \text{words} \rightarrow \mathbb{R}^n$$

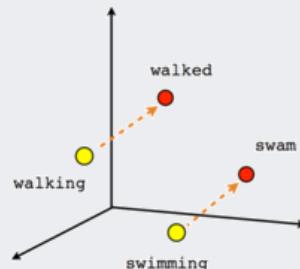
$$W(\text{"cat"}) = (0.2, -0.4, 0.7, \dots)$$

$$W(\text{"mat"}) = (0.0, 0.6, -0.1, \dots)$$

- Interesting feature of word embedding
  - ✓ Semantic relationship between words can be preserved



Male-Female



Verb tense



Country-Capital

# Neural Network Language Model (NNLM)

Bengio et al. (2003)

- Purpose

- ✓ Fighting the curse of dimensionality with distributed representations

- ✓ Associate with each word in the vocabulary a **distributed word feature vector** (a real valued vector in  $R^m$ ),
- ✓ Express the joint **probability function** of word sequences in terms of the feature vectors of these words in the sequence,
- ✓ Learn simultaneously the **word feature vectors** and the parameters of that **probability function**

# Neural Network Language Model (NNLM)

- Why it works?

- ✓ If we knew that dog and cat played similar roles (semantically and synthetically), and similarity for (the, a), (bedroom, room), (is, was), (running, walking), we could naturally generalize from

The cat is walking in the bedroom

to

A dog was running in a room

The cat is running is a room

A dog is walking in a bedroom

The dog was waling in the room

...

# Neural Network Language Model (NNLM)

Kim et al. (2016)

- Comparison with Count-based Language Models
  - ✓ Count-based Language Models

By the chain rule, any distribution can be factorized as

$$p(w_1, \dots, w_T) = \prod_{t=1}^T p(w_t | w_1, \dots, w_{t-1})$$

Count-based  $n$ -gram language models make a Markov assumption:

$$p(w_t | w_1, \dots, w_t) \approx p(w_t | w_{t-n}, \dots, w_{t-1})$$

Need smoothing to deal with rare  $n$ -grams.

# Neural Network Language Model (NNLM)

Kim et al. (2016)

- Comparison with Count-based Language Models

✓ NNLM

- Represent words as dense vectors in  $\mathbb{R}^n$  (word embeddings).

$\mathbf{w}_t \in \mathbb{R}^{|\mathcal{V}|}$  : One-hot representation of word  $\in \mathcal{V}$  at time  $t$   
 $\Rightarrow \mathbf{x}_t = \mathbf{X}\mathbf{w}_t$  : Word embedding ( $\mathbf{X} \in \mathbb{R}^{n \times |\mathcal{V}|}$ ,  $n < |\mathcal{V}|$ )

- Train a neural net that composes history to predict next word.

$$\begin{aligned} p(w_t = j | w_1, \dots, w_{t-1}) &= \frac{\exp(\mathbf{p}^j \cdot g(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}) + q^j)}{\sum_{j' \in \mathcal{V}} \exp(\mathbf{p}^{j'} \cdot g(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}) + q^{j'})} \\ &= \text{softmax}(\mathbf{P}g(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}) + \mathbf{q}) \end{aligned}$$

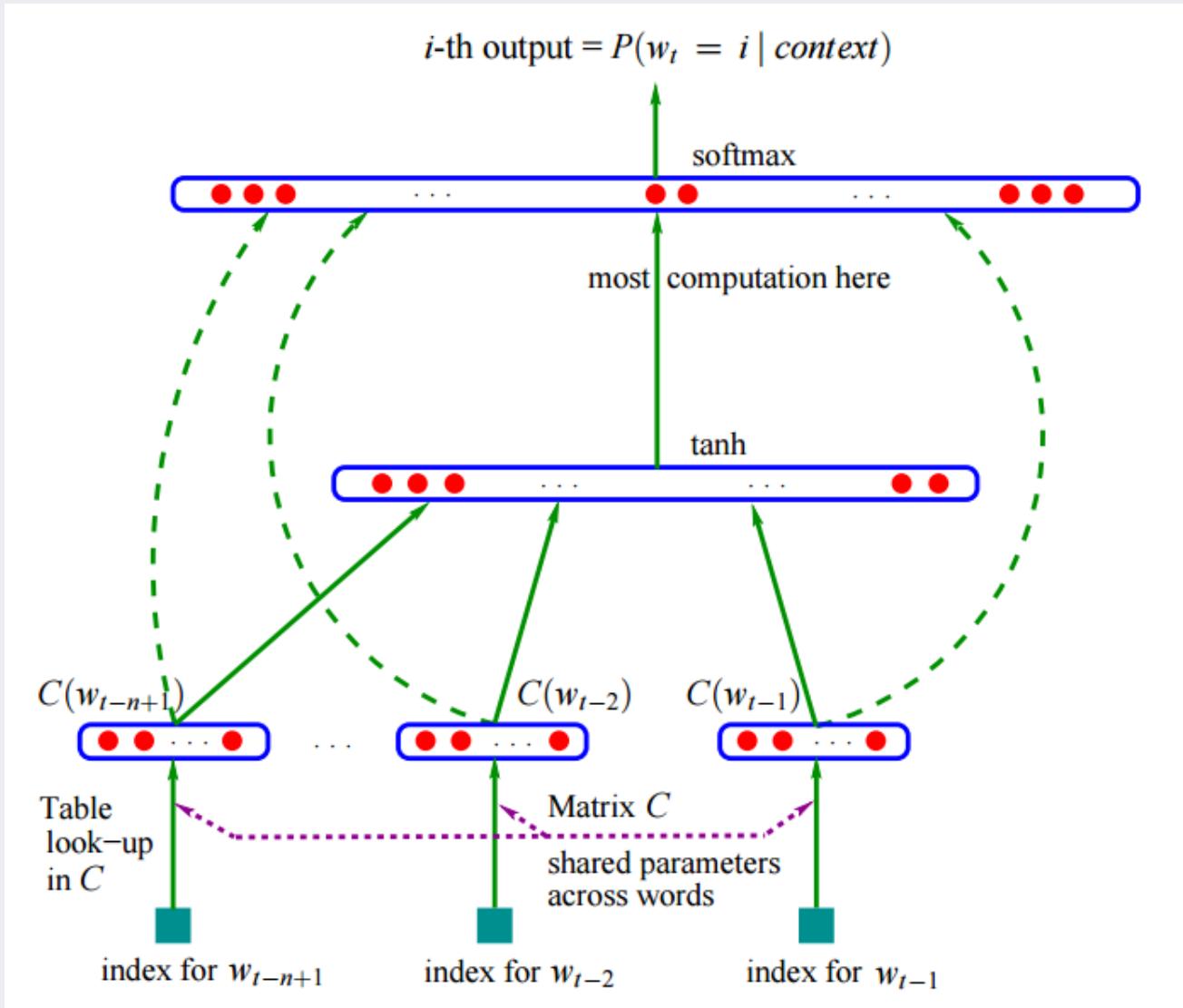
$\mathbf{p}^j \in \mathbb{R}^m$ ,  $q^j \in \mathbb{R}$  : Output word embedding/bias for word  $j \in \mathcal{V}$

$g$  : Composition function

# Neural Network Language Model (NNLM)

Bengio et al. (2003)

- Learning NNLM



# Neural Network Language Model (NNLM)

Bengio et al. (2003)

- Learning NNLM

- ✓ The objective is to learn a good model  $f(w_t, \dots w_{t-n+1}) = \hat{P}(w_t | w_1^{t-1})$ , in the sense that it gives high out-of-sample likelihood
- ✓ Two constraints
  - For any choice of  $w_1^{t-1}$ ,  $\sum_{i=1}^{|V|} f(i, w_{t-1} \dots w_{t-n+1}) = 1$
  - $f \geq 0$

# Neural Network Language Model (NNLM)

Bengio et al. (2003)

- Learning NNLM

- ✓ Decompose the function  $f(w_t, \dots w_{t-n+1}) = \hat{P}(w_t | w_1^{t-1})$  in two parts:
  - A mapping  $C$ , a.k.a the lookup table, from any element  $i$  of  $V$  to a real vector  $C(i) \in R^m$ , it represents the distributed feature vectors associated with each word in the vocabulary.  $C$  is represented by a  $|V| \times m$  matrix of free parameters
  - The probability function over words, expressed with  $C$ : a function  $g$  maps an input sequence of feature vectors for words in context,  $(C(w_{t-n+1}), \dots C(w_{t-1}))$ , to a conditional probability distribution over words in  $V$  for the next word  $w_t$ . The output of  $g$  is a vector whose  $i$ -th element estimates the probability  $\hat{P}(w_t = i | w_1^{t-1})$

# Neural Network Language Model (NNLM)

Bengio et al. (2003)

- Learning NNLM

$$f(i, w_{t-1} \cdots w_{t-n+1}) = g(i, C(w_{t-1}), \dots, C(w_{t-n+1}))$$

- ✓ The function  $f$  is a composition of these two mappings ( $C$  and  $g$ ), with  $C$  being shared across all the words in the context.
  - The parameters of the mapping  $C$  are simply the feature vectors themselves, represented by a  $|V| \times m$  matrix  $C$  whose row  $i$  is the feature vector  $C(i)$  for word  $i$
  - The function  $g$  may be implemented by a feed-forward or recurrent neural network or another parameterized function, with parameters  $\omega$ .
- ✓ Training is done by maximizing the penalized log-likelihood of the training corpus

$$L = \frac{1}{T} \sum_t \log f(i, w_{t-1} \cdots w_{t-n+1}; \theta) + R(\theta)$$

# Neural Network Language Model (NNLM)

Bengio et al. (2003)

- Learning NNLM

- ✓ The neural network has one hidden layer beyond the word features mapping, and optionally, direct connections from the word features to the output.
- ✓ Computation of the output layer

$$\hat{P}(w_t | w_{t-1}, \dots, w_{t-n+1}) = \frac{\exp(y_{w_t})}{\sum_i \exp(y_i)}$$

$$y = b + Wx + U \cdot \tanh(d + Hx)$$

- $W$  is optionally zero (no direct connections from the input to the output)
- $x$  is the word features layer activation vector  $x = (C(w_{t-1}), \dots, C(w_{t-n+1}))$
- $h$  is the number of hidden units

# Neural Network Language Model (NNLM)

Bengio et al. (2003)

- Learning NNLM

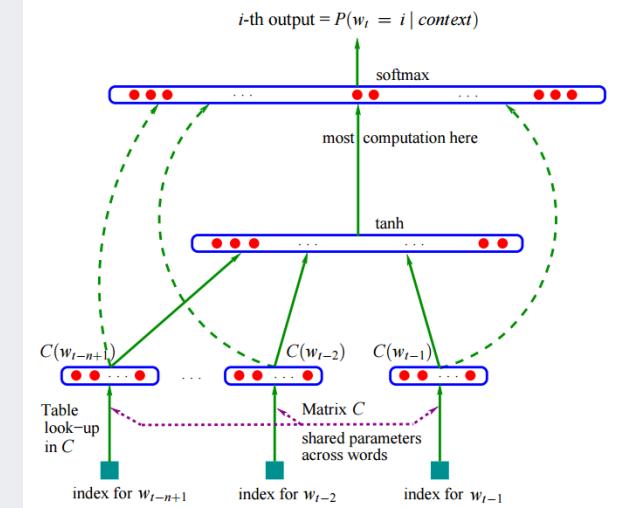
- ✓ The free parameters of the model

$$y = b + Wx + U \cdot \tanh(d + Hx)$$

- the output bias  $b$  ( $|V|$  elements)
- the hidden layer biases  $d$  (with  $h$  elements)
- the hidden-to-output weights  $U$  (a  $|V|$  by  $h$  matrix)
- the word features to output weights  $W$  (a  $|V|$  by  $(n-1)m$  matrix)
- the hidden layer weight  $H$  (a  $h$  by  $(n-1)m$  matrix)

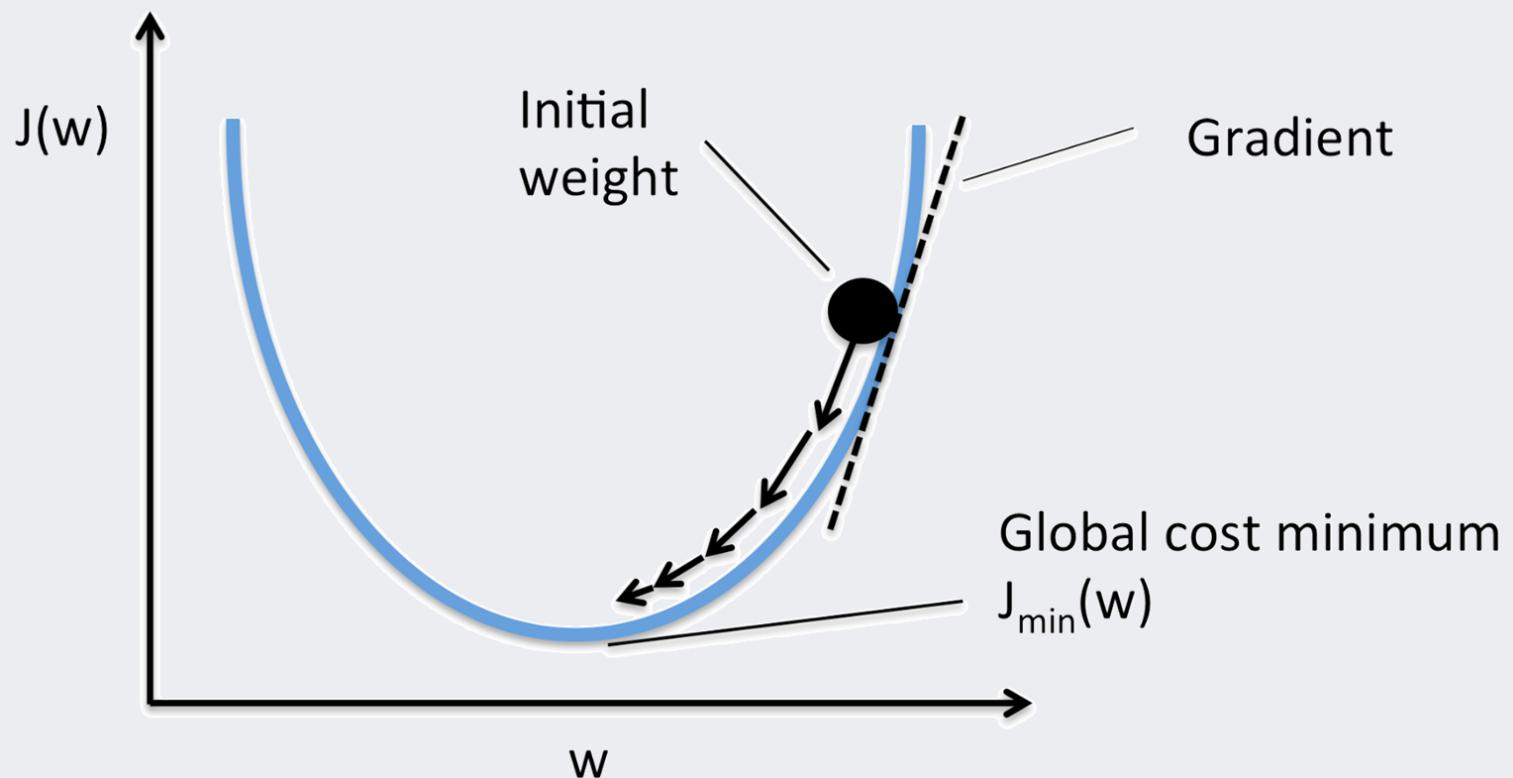
- ✓ Stochastic gradient ascent

$$\theta \leftarrow \theta + \varepsilon \frac{\partial \log \hat{P}(w_t | w_{t-1}, \dots, w_{t-n+1})}{\partial \theta}$$



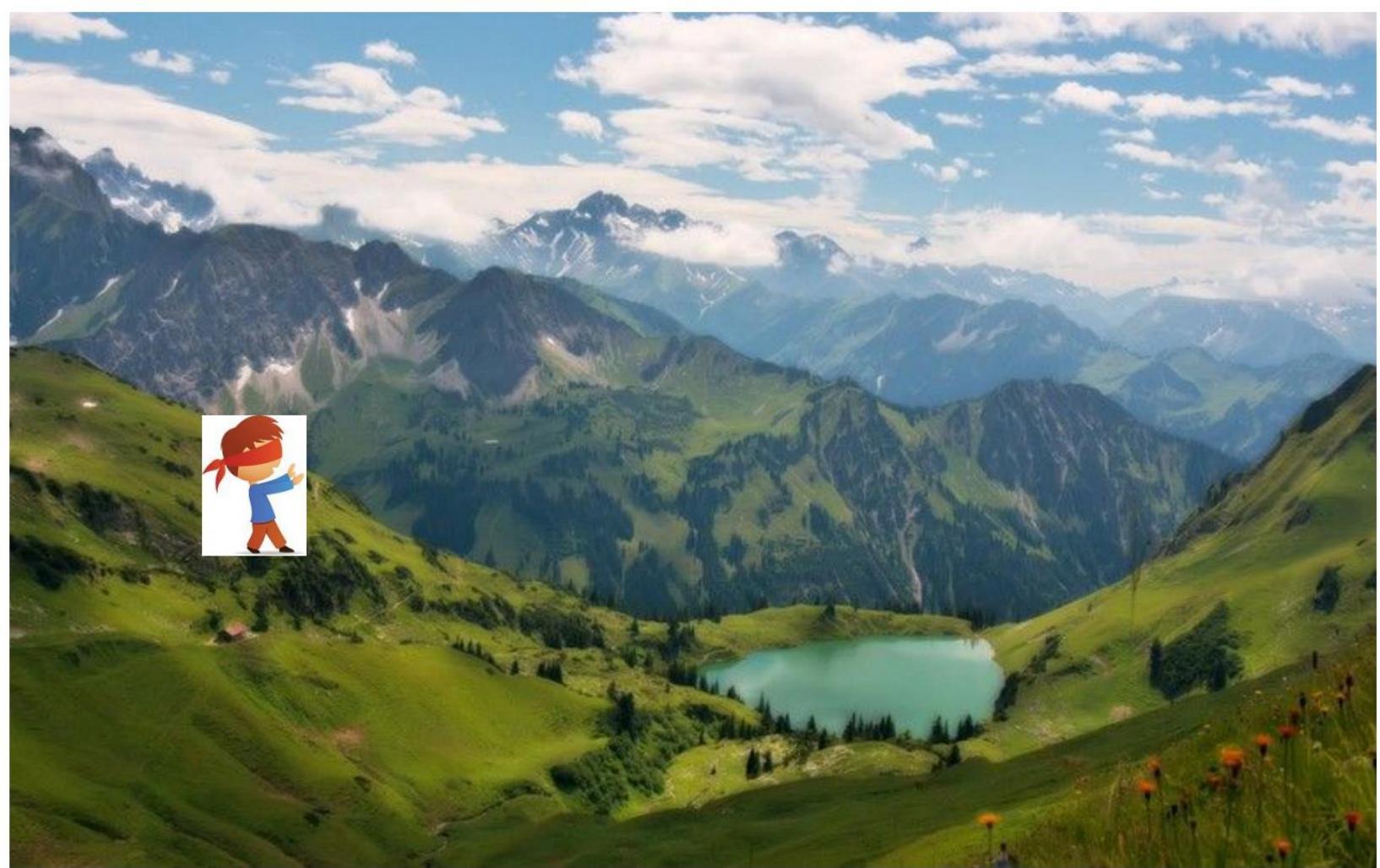
# Gradient Descent/Ascent

- Illustrative Example
  - ✓ Blue line: the value of the objective function w.r.t.  $w$
  - ✓ Black point: current position
  - ✓ Arrow: the direction that  $w$  should move toward to minimize the objective function



# Gradient Descent/Ascent

- Imagine that you are



# Gradient Descent/Ascent

- Theoretical Background

- ✓ Taylor expansion

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!} \Delta x + \frac{f''(x)}{2!} \Delta x^2 + \dots$$

- ✓ If the first derivative is not zero, we can decrease the function value by moving  $x$  toward the opposite direction of its first derivative

$$x_{new} = x_{old} - \eta f'(x), \text{ where } 0 < \eta < 1$$

- ✓ Then the function value of the new  $x$  is always smaller than that of the old  $x$

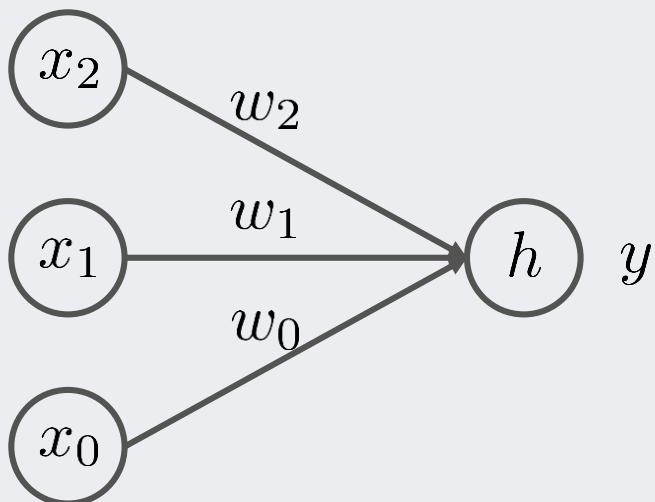
$$f(x_{new}) = f(x_{old} - \eta f'(x)) \cong f(x_{old}) - \eta |f'(x)|^2 < f(x_{old})$$



<https://www.youtube.com/watch?v=3d6DsjlBzJ4&t=322s>

# Gradient Descent/Ascent

- A Simple Example (Logistic Regression with two input variables)



$$h = \sum_{i=0}^2 w_i x_i$$

$$y = \frac{1}{1 + \exp(-h)}$$

- Let's define the squared loss function  $L = \frac{1}{2}(t - y)^2$
- How to find the gradient w.r.t.  $w$  or  $x$ ?

# Gradient Descent/Ascent

- Use chain rule

$$\frac{\partial L}{\partial y} = y - t$$

$$\frac{\partial y}{\partial h} = \frac{exp(-h)}{(1 + exp(-h))^2} = \frac{1}{1 + exp(-h)} \cdot \frac{exp(-h)}{1 + exp(-h)} = y(1 - y)$$

$$\frac{\partial h}{\partial w_i} = x_i \quad \frac{\partial h}{\partial x_i} = w_i$$

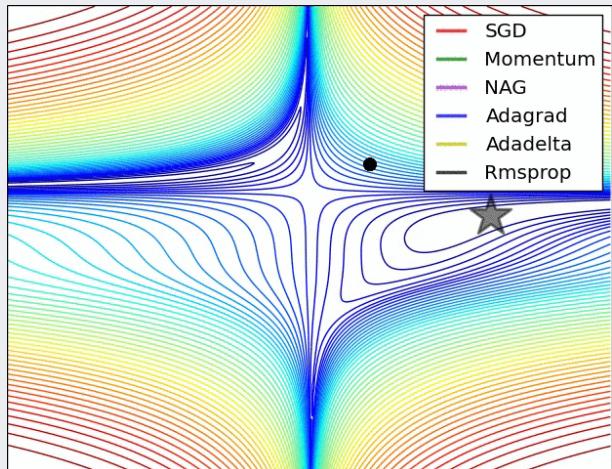
- Gradients for  $w$  and  $x$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial w_i} = (y - t) \cdot y(1 - y) \cdot x_i$$

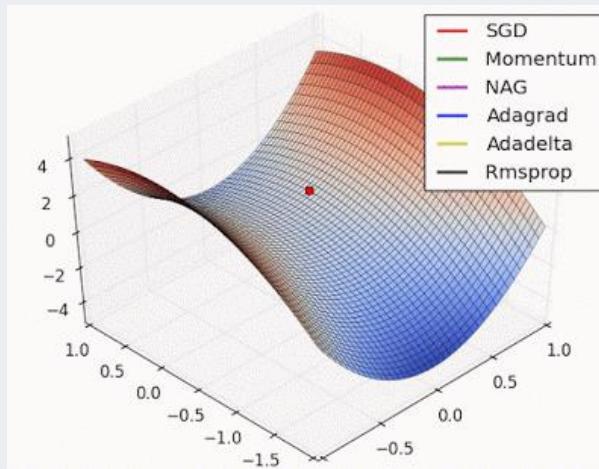
$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial x_i} = (y - t) \cdot y(1 - y) \cdot w_i$$

# Gradient Descent/Ascent

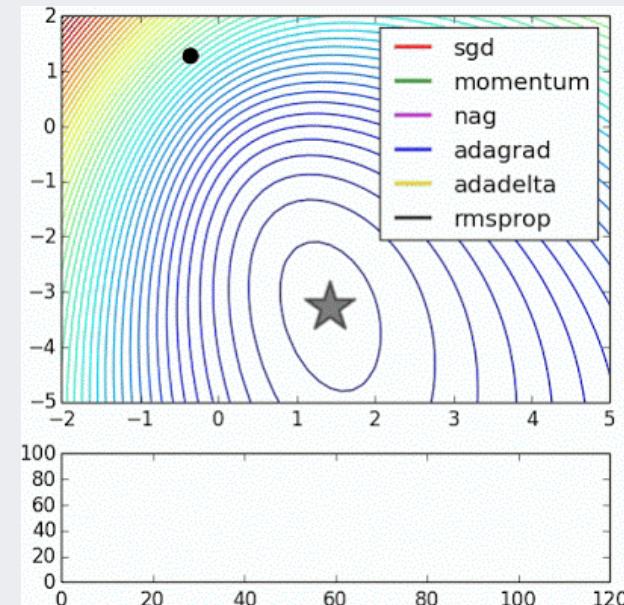
- Convergence



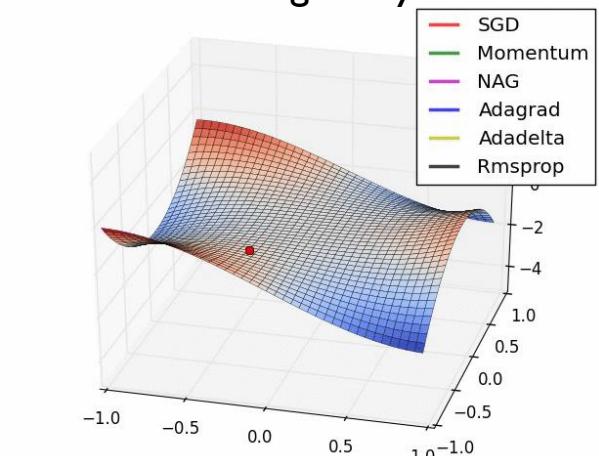
Beale's function



Long valley



Noisy moons



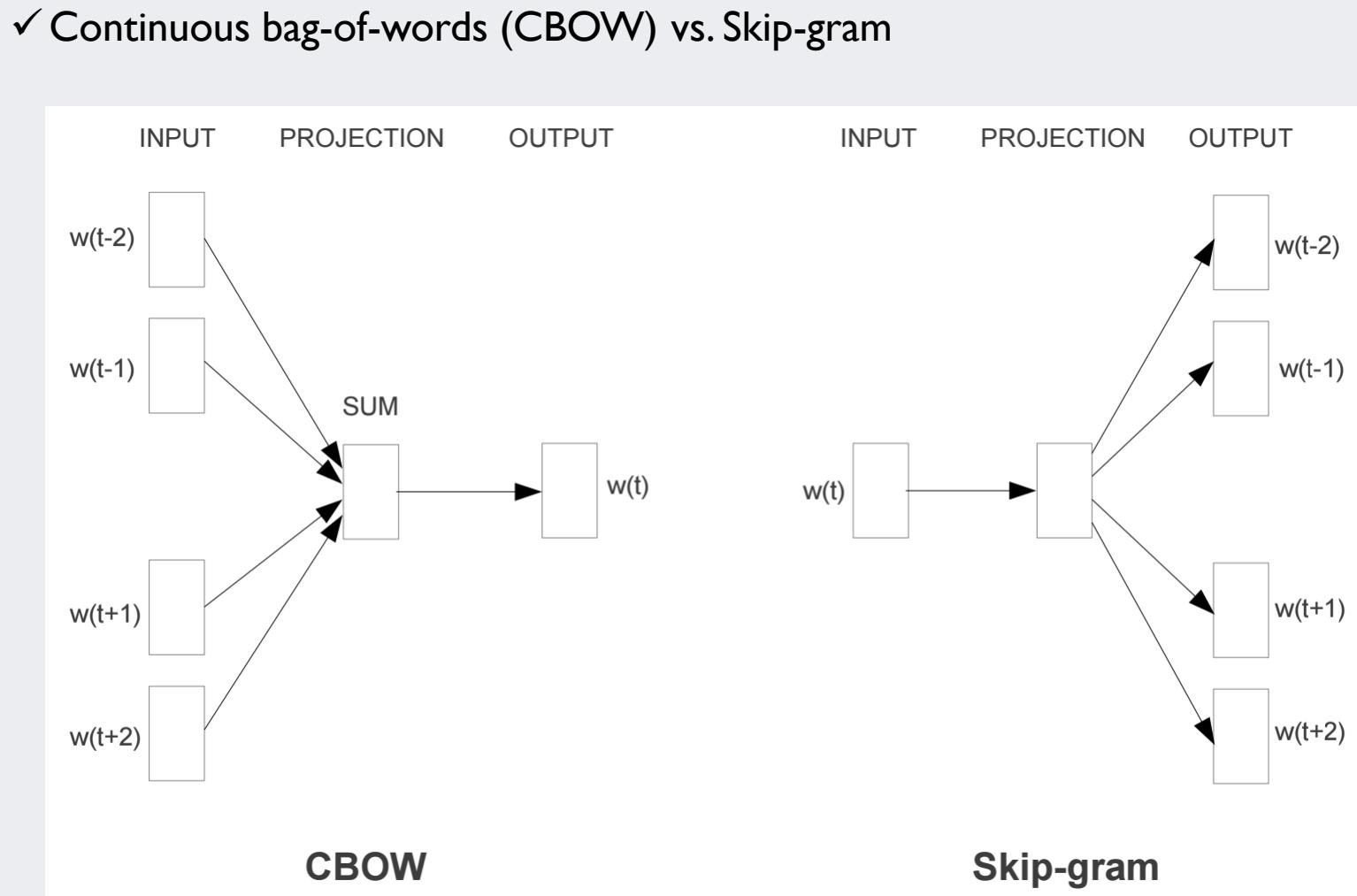
Saddle point



# Word2Vec

Mikolov et al. (2013)

- Two Architectures



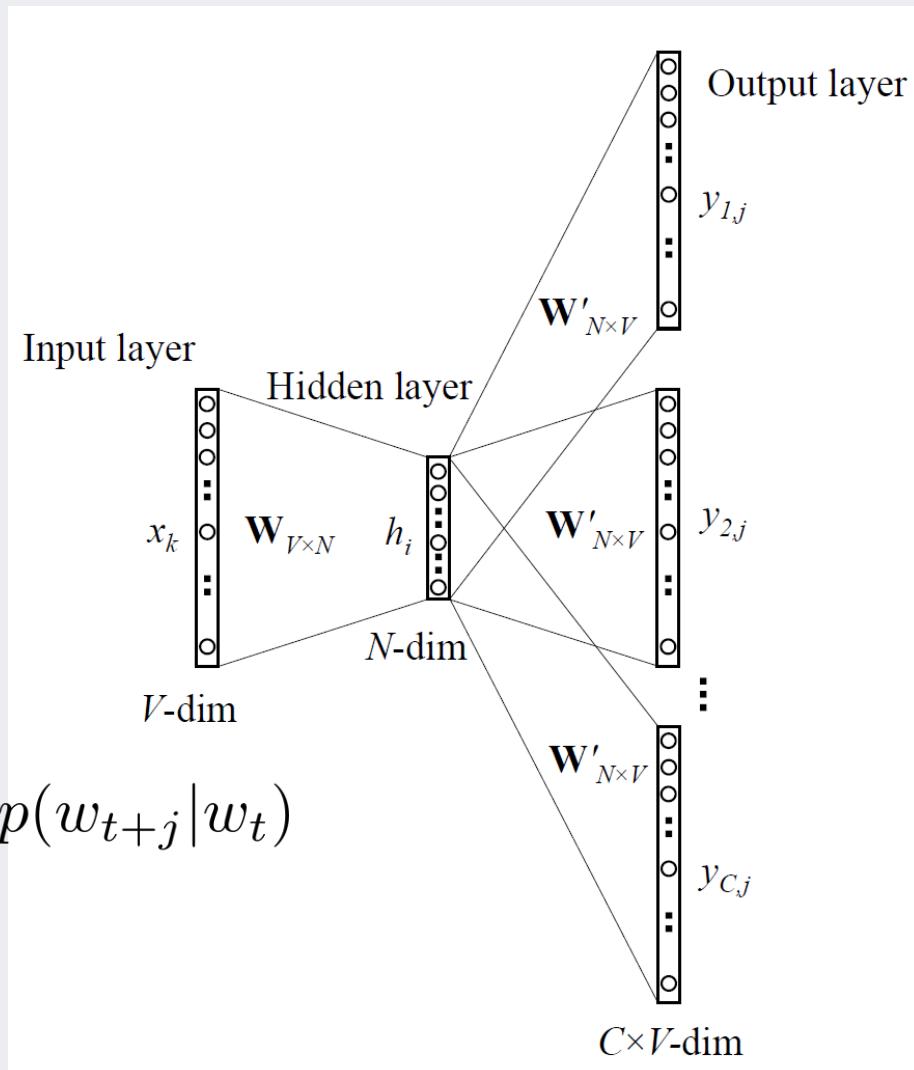
# Word2Vec

Rong (2014)

- Learning representations: Skip-gram approach
  - ✓ Predict surrounding words in a window of length  $m$  of every word
- Objective function
  - ✓ Maximize the log probability of any context word given the current center word

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

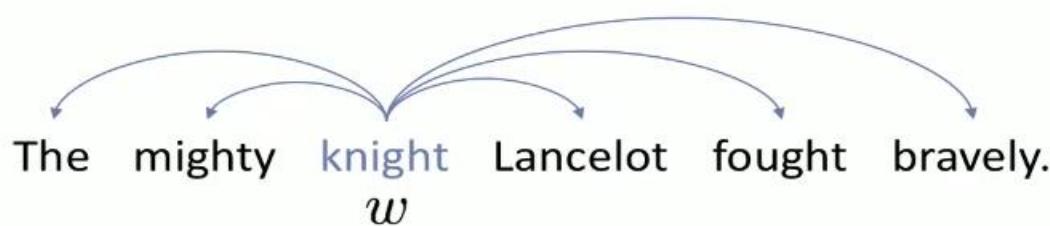
- ✓ where  $\theta$  represents all variables we optimize



# Word2Vec

Bojanowski (2016)

- Skip-gram model



knight → The  
knight → mighty  
knight → Lancelot  
knight → fought  
knight → bravely.

- Model probability of a context word given a word

feature for word  $w$ :  $x_w$

classifier for word  $c$ :  $v_c$

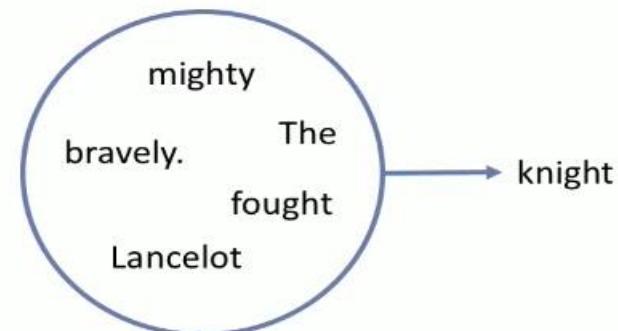
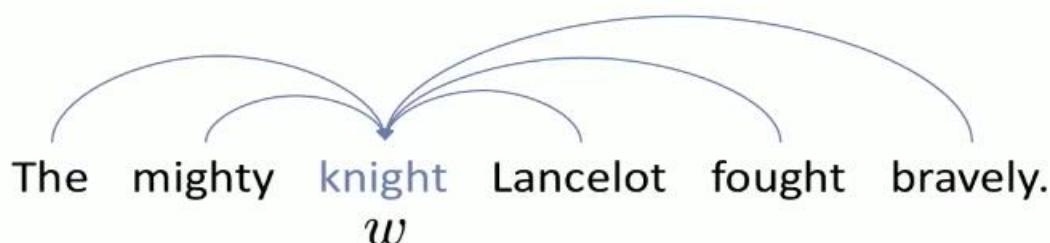
$$p(c|w) = \frac{e^{x_w^\top v_c}}{\sum_{k=1}^K e^{x_w^\top v_k}}$$

- Word vectors  $x_w \in \mathbb{R}^d$

# Word2Vec

Bojanowski (2016)

- CBOW model



- Model probability of a word given a context

feature for context  $\mathcal{C}$ :  $h_{\mathcal{C}}$

classifier for word  $w$ :  $v_w$

$$p(w|\mathcal{C}) = \frac{e^{h_{\mathcal{C}}^\top v_w}}{\sum_{k=1}^K e^{h_{\mathcal{C}}^\top v_k}}$$

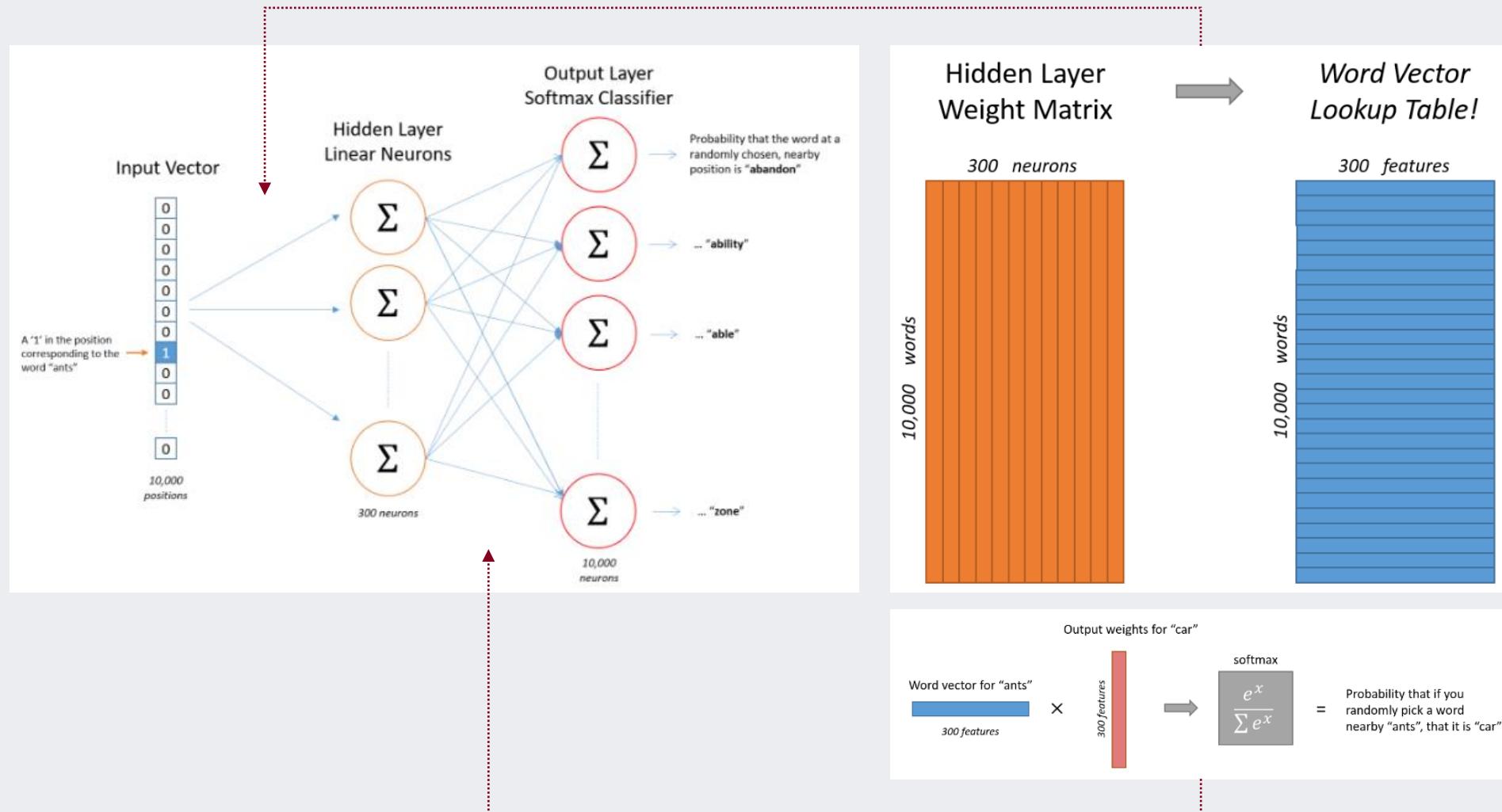
- Continuous Bag Of Words

$$h_{\mathcal{C}} = \sum_{c \in \mathcal{C}} x_c$$

# Word2Vec

McCormick

- Another architecture explanation

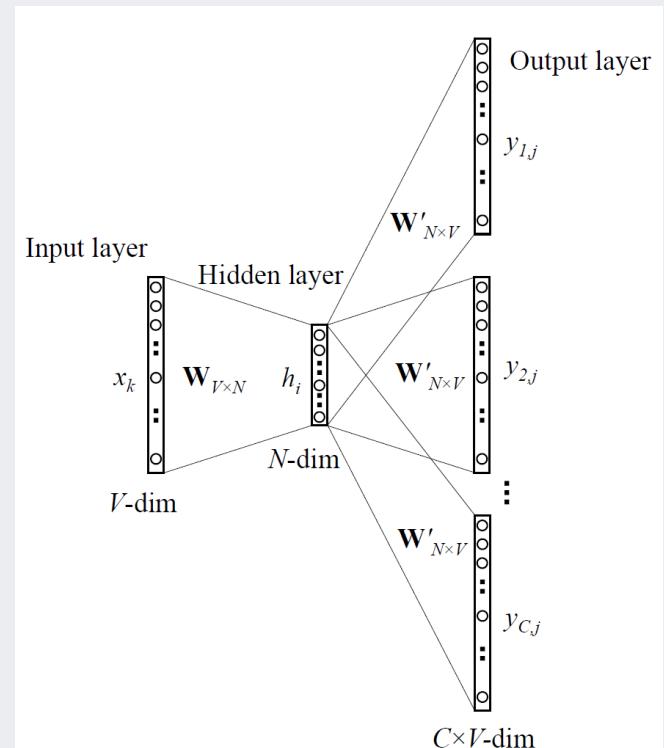


# Word2Vec

- For simplicity, we use the following notation instead of  $p(w_{t+j}|w_t)$

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^W \exp(u_w^T v_c)}$$

- ✓ where  $o$  is the outside (output) word id,  $c$  is the center word id,  $u$  and  $v$  are “outside” and “center” vectors of  $o$  and  $c$
- ✓ Every word has two vectors!
  - $v$  is a **row** of matrix  $\mathbf{W}$
  - $u$  is a **column** of matrix  $\mathbf{W}'$
- ✓ Use  $\mathbf{W}' = \mathbf{W}^T$  in practice for efficient computation



# Word2Vec

- Learning parameters with Gradient Ascent

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

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^W \exp(u_w^T v_c)}$$

✓ Compute the gradient

$$\frac{\partial}{\partial v_c} \log p(o|c) = \frac{\partial}{\partial v_c} \log \frac{\exp(u_o^T v_c)}{\sum_{w=1}^W \exp(u_w^T v_c)}$$

$$= \boxed{\frac{\partial}{\partial v_c} u_o^T v_c} - \boxed{\frac{\partial}{\partial v_c} \log \sum_{w=1}^W \exp(u_w^T v_c)}$$

A

B

# Word2Vec

- Learning parameters with Gradient Ascent

✓ For chunk A

$$\frac{\partial}{\partial v_c} u_o^T v_c = u_o$$

✓ For chunk B

$$\begin{aligned}& -\frac{\partial}{\partial v_c} \log \sum_{w=1}^W \exp(u_w^T v_c) \\&= -\frac{1}{\sum_{w=1}^W \exp(u_w^T v_c)} \cdot \left( \sum_{w=1}^W \exp(u_w^T v_c) \cdot u_w \right) \\&= -\sum_{w=1}^W \frac{\exp(u_w^T v_c)}{\sum_{w=1}^W \exp(u_w^T v_c)} \cdot u_w = -\sum_{w=1}^W P(w|c) \cdot u_w\end{aligned}$$

# Word2Vec

- Learning parameters with Gradient Ascent

$$\frac{\partial}{\partial v_c} \log p(o|c) = u_o - \sum_{w=1}^W P(w|c) \cdot u_w$$

- Update the weight vector

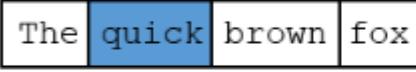
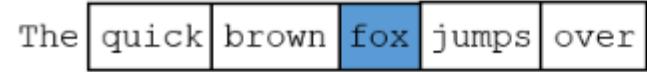
$$v_c(t+1) = v_c(t) + \alpha \left( u_o - \sum_{w=1}^W P(w|c) \cdot u_w \right)$$

# Word2Vec

McCormick

- Learning strategy

- ✓ Do not use all nearby words, but one per each training

Source Text	Training Samples
The quick brown fox jumps over the lazy dog. → 	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog. → 	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog. → 	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown fox jumps over the lazy dog. → 	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

# Word2Vec

- The number of weights to be trained:  $2 \times V \times N$  (Huge network!)

## ✓ Word pairs and phrases

- Treating common word pairs or phrases as single “word”

## ✓ Subsampling frequent words

- To decrease the number of training examples
- The probability of word  $w_i$  being removed

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \quad \begin{array}{l} \xleftarrow{\hspace{1cm}} \text{Threshold } (10^{-5}) \\ \xleftarrow{\hspace{1cm}} \text{Term frequency in the corpus} \end{array}$$

$$\text{if } f(w_i) = 10^{-4}, \ P(w_i) = 1 - \sqrt{\frac{1}{10}} = 0.6838$$

$$\text{if } f(w_i) = 10^{-2}, \ P(w_i) = 1 - \sqrt{\frac{1}{1000}} = 0.9684$$

# Word2Vec

- The number of weights to be trained:  $2 \times V \times N$  (Huge network!)

## ✓ Negative sampling

- Instead of updating the weights associated with all output words, update the weight of a few (5-20) words

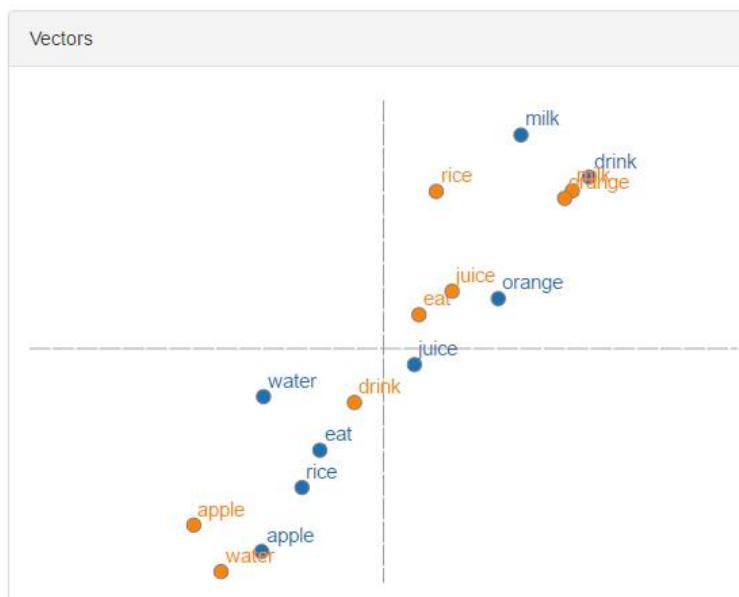
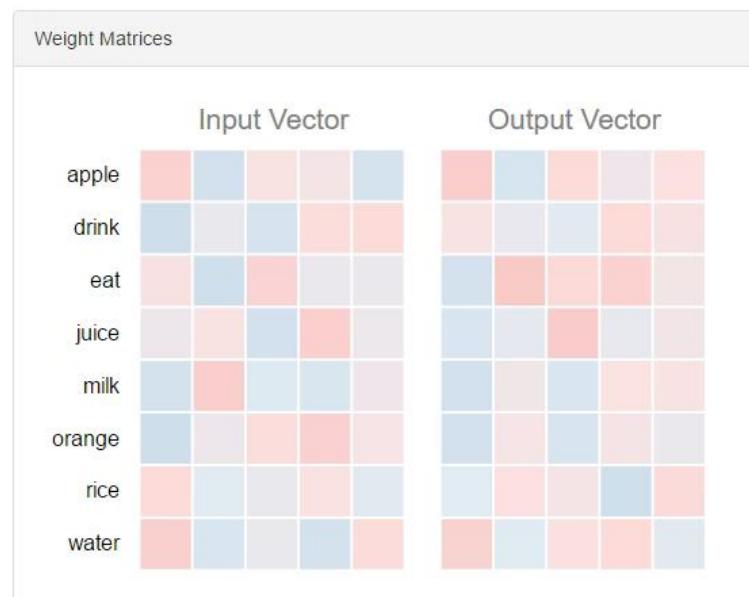
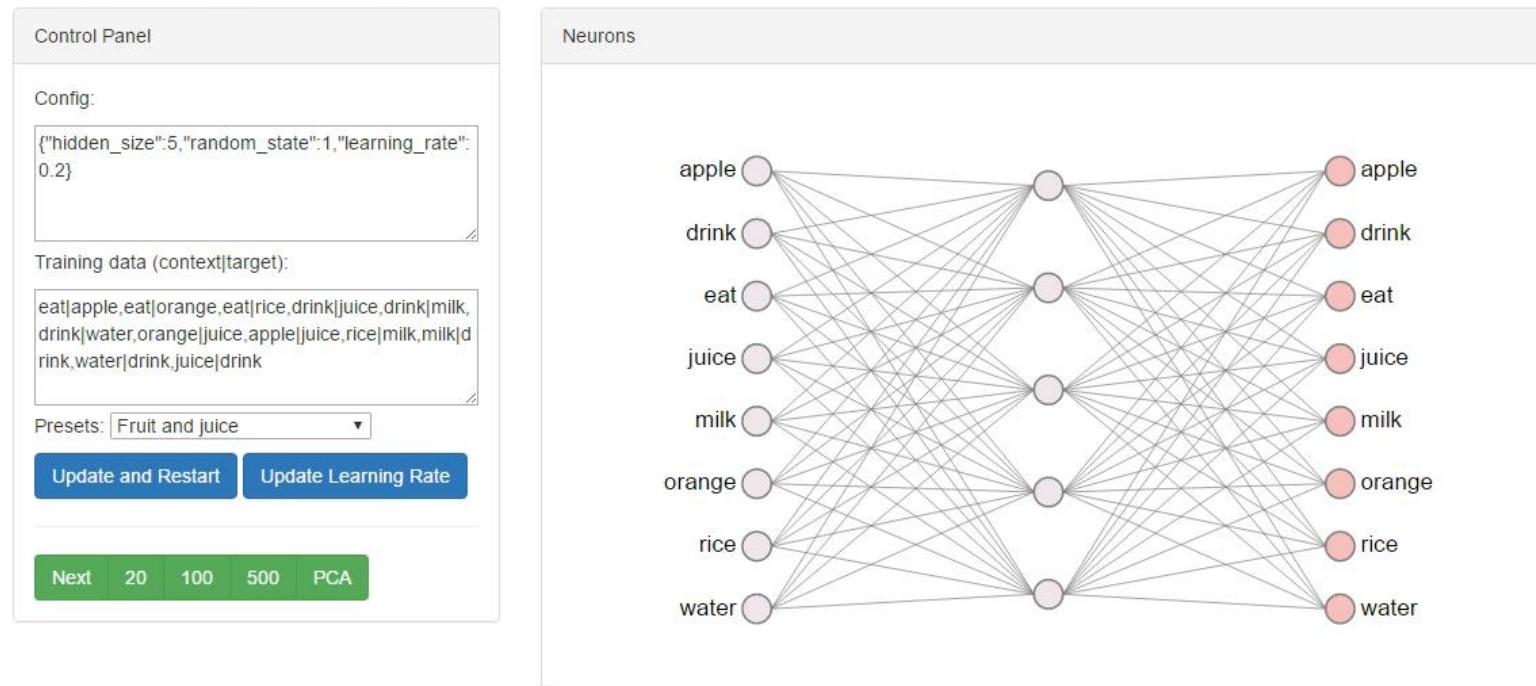
$$J(\theta) = \frac{1}{T} \sum_{t=1}^T \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} | w_t) = \frac{1}{T} \sum_{t=1}^T J_t(\theta)$$

$$J_t(\theta) = \log \sigma(u_o^T v_c) + \sum_{i=1}^k E_{i \sim P(w)} [\log \sigma(-u_i^T v_c)]$$

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^n (f(w_j)^{3/4})}$$

# wevi: word embedding visual inspector

Everything you need to know about this tool - [Source code](#)



# GloVe

Pennington et al. (2014)

- Limitations of Word2Vec

- ✓ The network spends so much time to train some overwhelmingly used words
  - Ex: to learn a distribution for  $P(w|\text{the})$

Theatre or theater is a collaborative form of fine art that uses live performers to present **the experience** of a real or imagined event before a live audience in a specific place. **The performers** may communicate this experience to **the audience** through combinations of gesture, speech, song, music, and dance. Elements of art and stagecraft are used to enhance **the physicality**, presence and immediacy of **the experience**. **The specific** place of **the performance** is also named by **the word** "theatre" as derived from **the Ancient Greek** (*thatron*, "a place for viewing"), itself from (*theomai*, "to see", "to watch", "to observe"). Modern Western theatre comes from large measure from ancient Greek drama, from which it borrows technical terminology, classification into genres, and many of its themes, stock characters, and plot elements. Theatre artist Patrice Pavis denotes theatricality, theatrical language, stage writing, and **the specificity** of theatre as synonymous expressions that differentiate theatre from **the other** performing arts, literature, and **the arts** in general. Theatre today, broadly defined, includes performances of plays and musicals, ballets, operas and various other forms.

# GloVe

Pennington et al. (2014)

- **GloVe**

- ✓ Based on matrix factorization method
- ✓ <http://nlp.stanford.edu/projects/glove/>
- ✓ Notations

- $X \in \mathbb{R}^{V \times V}$  word co-occurrence matrix
- $X_{ij}$  frequency of word  $i$  co-occurring with word  $j$
- $X_i = \sum_k^V X_{ik}$  total number of occurrences of word  $i$  in corpus
- $P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$  a.k.a. probability of word  $j$  occurring within the context of word  $i$
- $w \in \mathbb{R}^d$  a word embedding of dimension  $d$
- $\tilde{w} \in \mathbb{R}^d$  a context word embedding of dimension  $d$

# GloVe

Pennington et al. (2014)

- Motivation

Prob. and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k ice)$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k steam)$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$\frac{P(k ice)}{P(k steam)}$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

- ✓ For words  $k$  related ice but not steam (solid), the ratio  $P_{ik}/P_{jk}$  is large
- ✓ For words  $k$  related steam but not ice (gas) the ratio  $P_{ik}/P_{jk}$  is small
- ✓ For words  $k$  that are either related to both ice and steam, or to neither, the ratio should be close to 1

# GloVe

Pennington et al. (2014)

- Formulation

- ✓ Express the relationship among three words using a function F

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

- ✓ Relationship between  $w_i$  and  $w_j$  is expressed by subtraction

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

- ✓ Inner product is used to link  $\tilde{w}_k$  with  $w_i$  and  $w_j$

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

# GloVe

Pennington et al. (2014)

- Homomorphism

Prob. and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$	
$P(k ice)$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$	
$P(k steam)$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$	
$\frac{P(k ice)}{P(k steam)}$	8.9	$8.5 \times 10^{-2}$	1.36	0.96	

✓ Want to preserve  $\frac{P(k|ice)}{P(k|steam)}$  using  $F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}}$

$$\frac{P(solid|\textcolor{blue}{ice})}{P(solid|\textcolor{red}{steam})} = F\left((\textcolor{blue}{ice} - \textcolor{red}{steam})^T solid\right)$$

$$\frac{P(solid|\textcolor{red}{steam})}{P(solid|\textcolor{blue}{ice})} = F\left((\textcolor{red}{steam} - \textcolor{blue}{ice})^T solid\right)$$

# GloVe

Pennington et al. (2014)

- Homomorphism

$$F\left((\text{ice} - \text{steam})^T \text{solid}\right) = \frac{P(\text{solid}|\text{ice})}{P(\text{solid}|\text{steam})} = \frac{1}{F\left((\text{steam} - \text{ice})^T \text{solid}\right)}$$

$$(\text{ice} - \text{steam})^T \text{solid} = -(\text{steam} - \text{ice})^T \text{solid}$$

inverse element of addition

$$F\left((\text{ice} - \text{steam})^T \text{solid}\right) = \frac{1}{F\left((\text{steam} - \text{ice})^T \text{solid}\right)}$$

inverse element of multiplication

- ✓ Homomorphism preserves an operation, which in turn preserves the inverse element
- ✓ Need a homomorphism from  $(\mathbb{R}, +)$  to  $(\mathbb{R}_{>0}, \times)$

# GloVe

Pennington et al. (2014)

- Homomorphism

- ✓ Function  $F$ : homomorphism that maps  $(\mathbb{R}, +)$  to  $(\mathbb{R}_{>0}, \times)$

$$w_i^T \tilde{w}_k = (w_i - w_j)^T \tilde{w}_k + w_j^T \tilde{w}_k$$

$$\begin{aligned} F(w_i^T \tilde{w}_k) &= F\left((w_i - w_j)^T \tilde{w}_k + w_j^T \tilde{w}_k\right) \\ &= F\left((w_i - w_j)^T \tilde{w}_k\right) \cancel{+} F(w_j^T \tilde{w}_k) \end{aligned}$$

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

- ✓ Finally, we can drive that  $F(x) = \exp(x)$

# GloVe

Pennington et al. (2014)

- Solution

✓ We know that  $F(x) = \exp(x)$  and  $F((w_i - w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)} = \frac{P_{ik}}{P_{jk}}$

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



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



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

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

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

# GloVe

Pennington et al. (2014)

- Objective Function

- ✓ A least squared objective function

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

where  $f$  has the following desiderata:

- ➊  $f(0) = 0$
- ➋  $f(x)$  should be non-decreasing so that rare co-occurrences are not overweighted.
- ➌  $f(x)$  should be relatively small for large values of  $x$ , so that frequent co-occurrences are not overweighted.

# GloVe

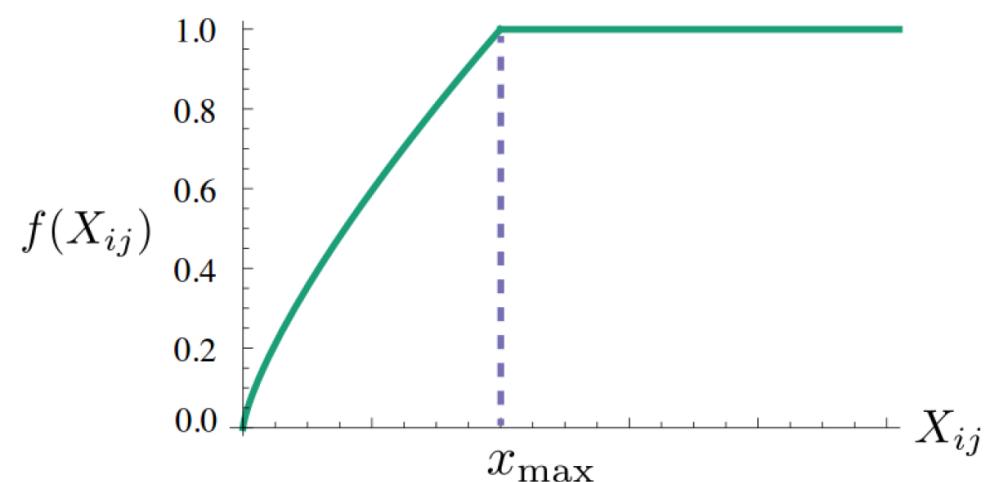
Pennington et al. (2014)

- Objective Function

- ✓ A least squared objective function

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

where  $f(x) = \begin{cases} \left(\frac{x}{x_{\max}}\right)^\alpha & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}$



# GloVe

Pennington et al. (2014)

- Results

- 0. frog
- 1. frogs
- 2. load
- 3. litoria
- 4. leptodactylidae
- 5. rana
- 6. lizard
- 7. eleutherodactylus



3. litoria



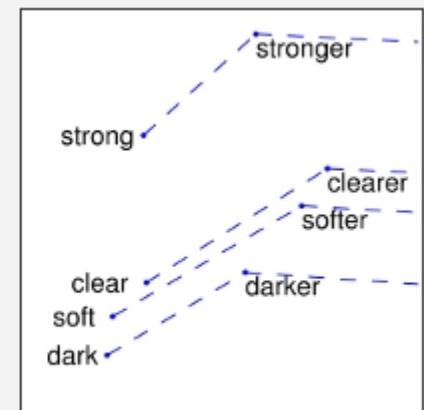
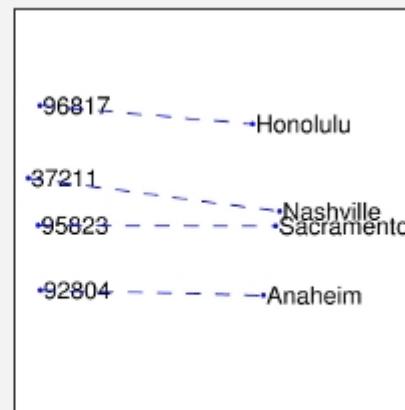
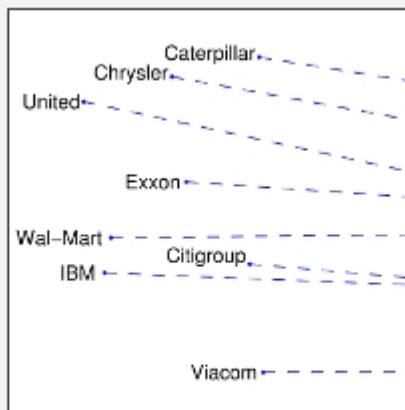
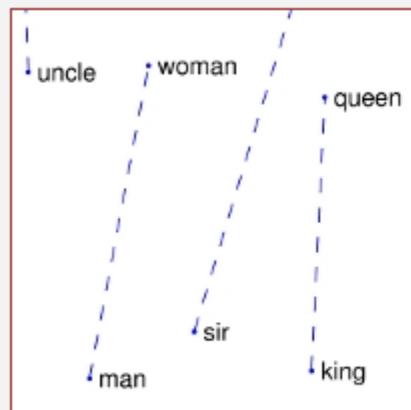
4. leptodactylidae



5. rana



7. eleutherodactylus



# FastText

Bojanowski et al. (2016)

- Limitations of NNLM, Word2Vec, and GloVe
  - ✓ Ignores the morphology or words by assigning a distinct vector to each word
  - ✓ Difficult to apply to morphologically rich languages with large vocabularies and many rare words (Turkish or Finnish)
- Goal
  - ✓ Learn representations for character n-grams
  - ✓ Represent words as the sum of n-gram vectors

# FastText

Bojanowski et al. (2016)

- Revisit Negative Sampling in Word2Vec

$$J_t(\theta) = \log\sigma(u_o^T v_c) + \sum_{i=1}^k E_{i \sim P(w)} [\log\sigma(-u_i^T v_c)]$$

✓ Score is just a dot product between the two embeddings

- Subword model

✓ Define the set of n-grams appearing in w:  $\mathcal{G}_w \subset \{1, \dots, G\}$

$$score(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^T \mathbf{v}_c$$

✓ Represent a word by the sum of the vector representations of its n-grams

# FastText

Bojanowski et al. (2016)

- Subword model

- ✓ n-gram representation

- Include the word  $w$  in the set of its n-grams
- Keep all the n-grams of size 3, 4, 5, and 6
- Different vectors are assigned to a word and a n-gram sharing the same sequence of characters

Word2Vec

parameter	...	...	...	...	...
mang	erai	ange			
man	ang	era	gera		
nge	ger	rai	nger		
Character n-grams			Word itself		



mangerai

Character n-grams

Word itself

FastText

p	...	...	...	...	...	...
...	...	...	...	...	...	...
r	...	...	...	...	...	...
pa	...	...	...	...	...	...
...	...	...	...	...	...	...
er	...	...	...	...	...	...
par	...	...	...	...	...	...
...	...	...	...	...	...	...
ameter	...	...	...	...	...	...
Avg.	...	...	...	...	...	...

# Word Embedding Examples

- Word Embedding examples: English

✓ Word lists that are close to a given word after embedding

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

Collobert et al. (2011)

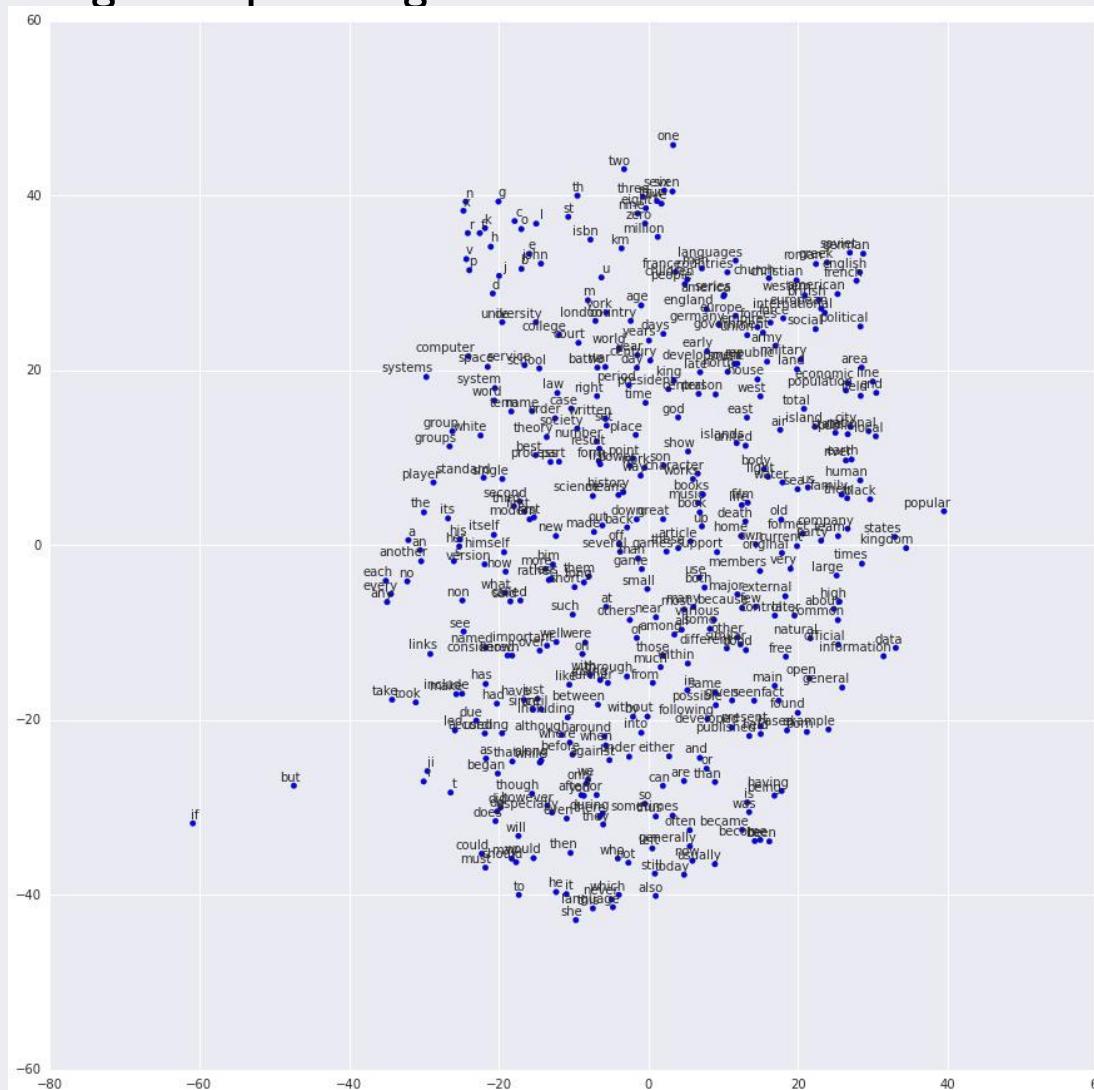
✓ Relationship pairs in a word embedding

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

Mikolov et al. (2013)

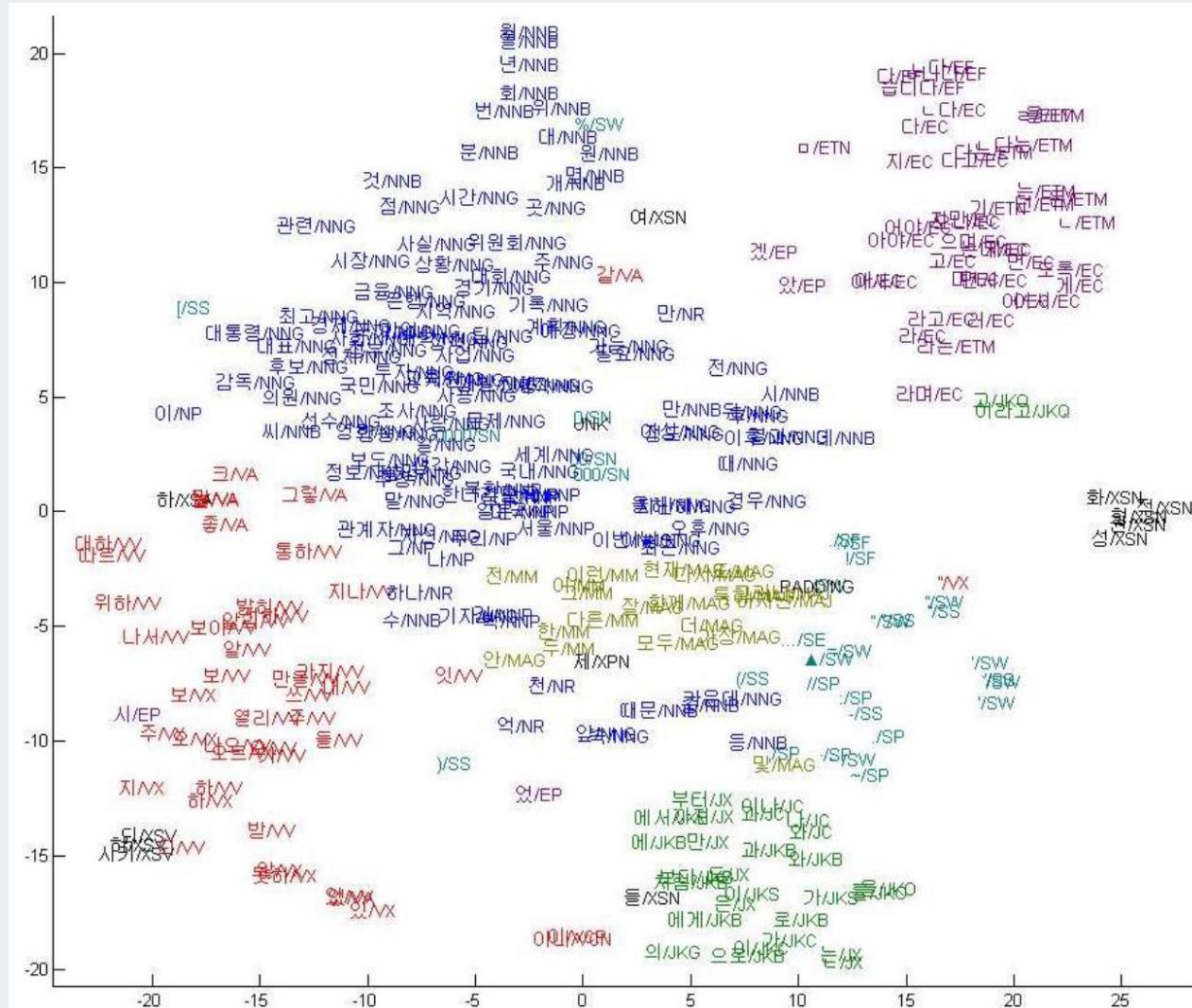
# Word Embedding Examples

- Word Embedding examples: English



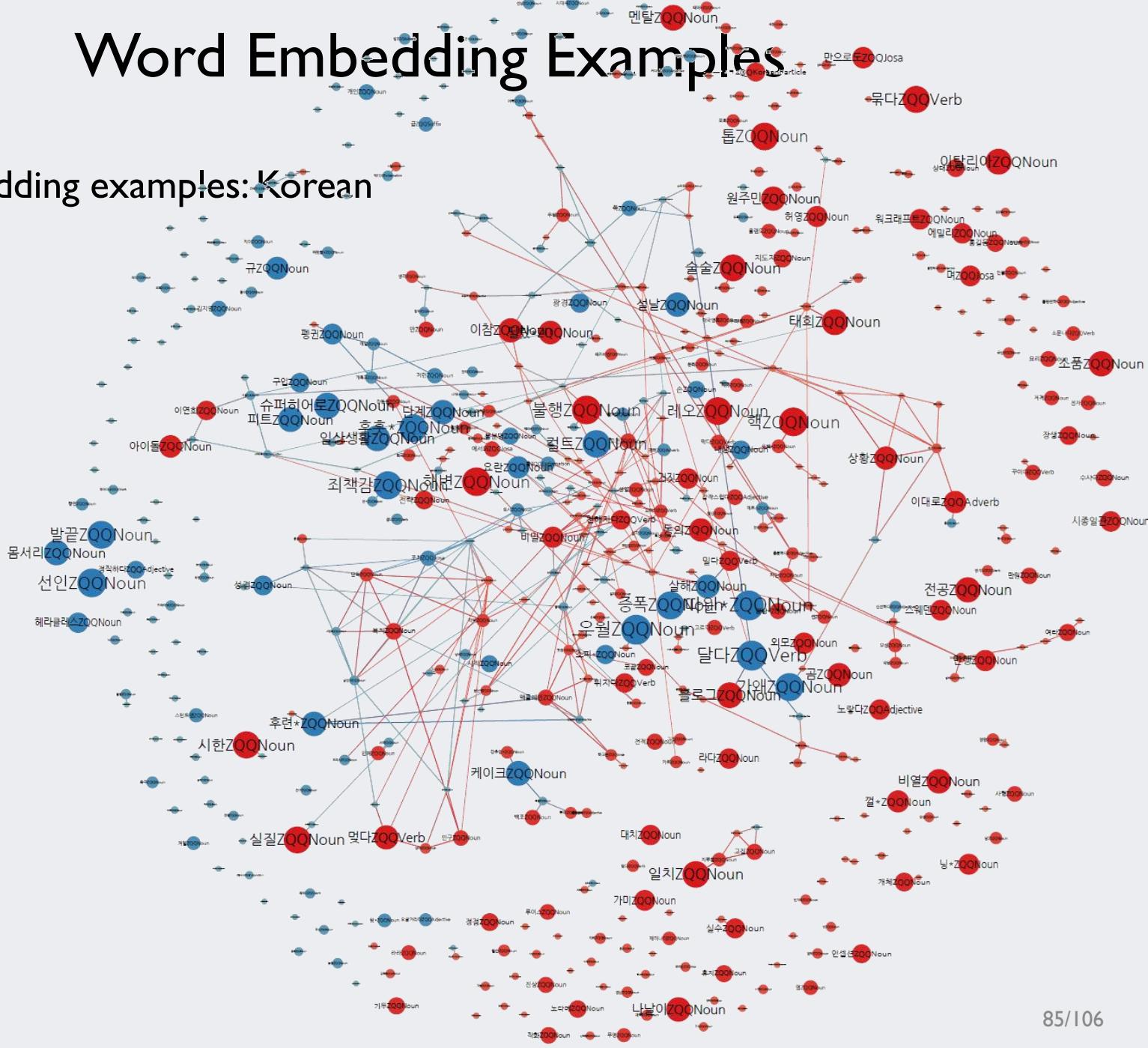
# Word Embedding Examples

- Word Embedding examples: Korean



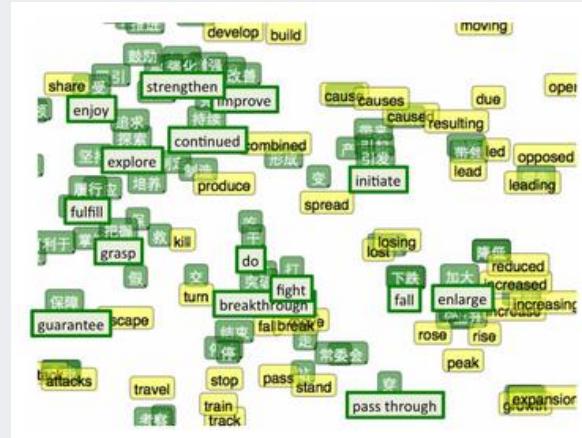
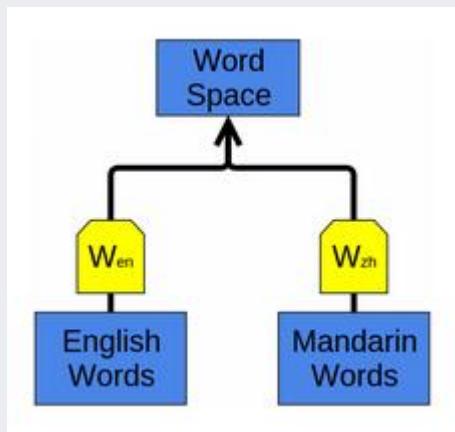
# Word Embedding Examples

- Word Embedding examples: Korean

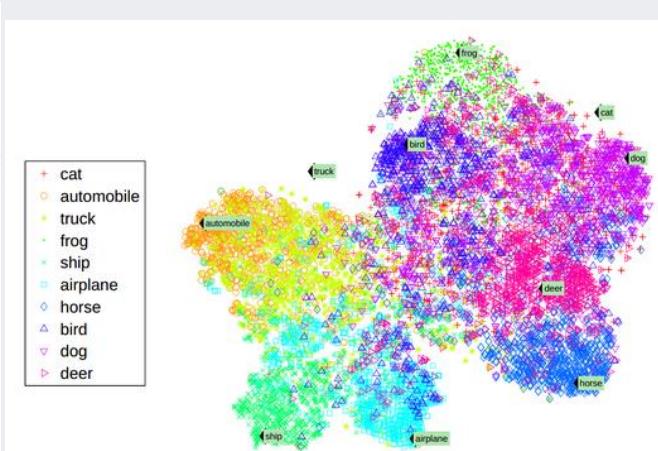
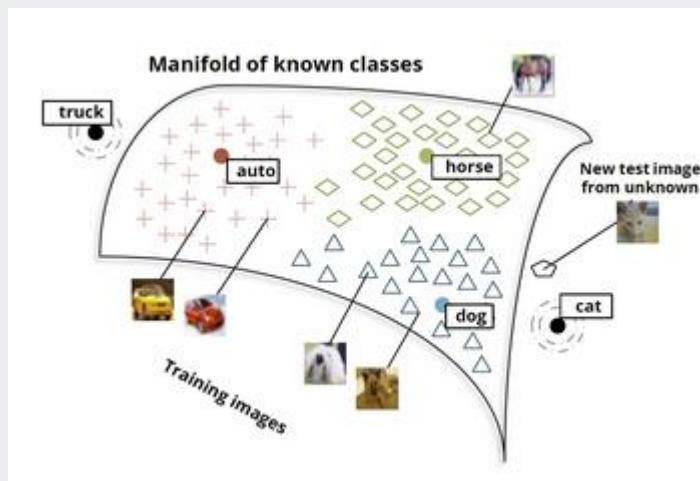
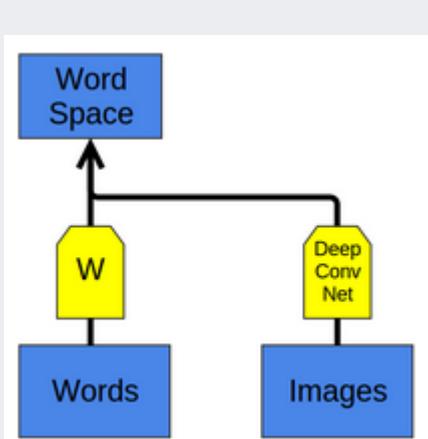


# Word Embedding Examples

- Word Embedding with two different languages



- Word Embedding with Images

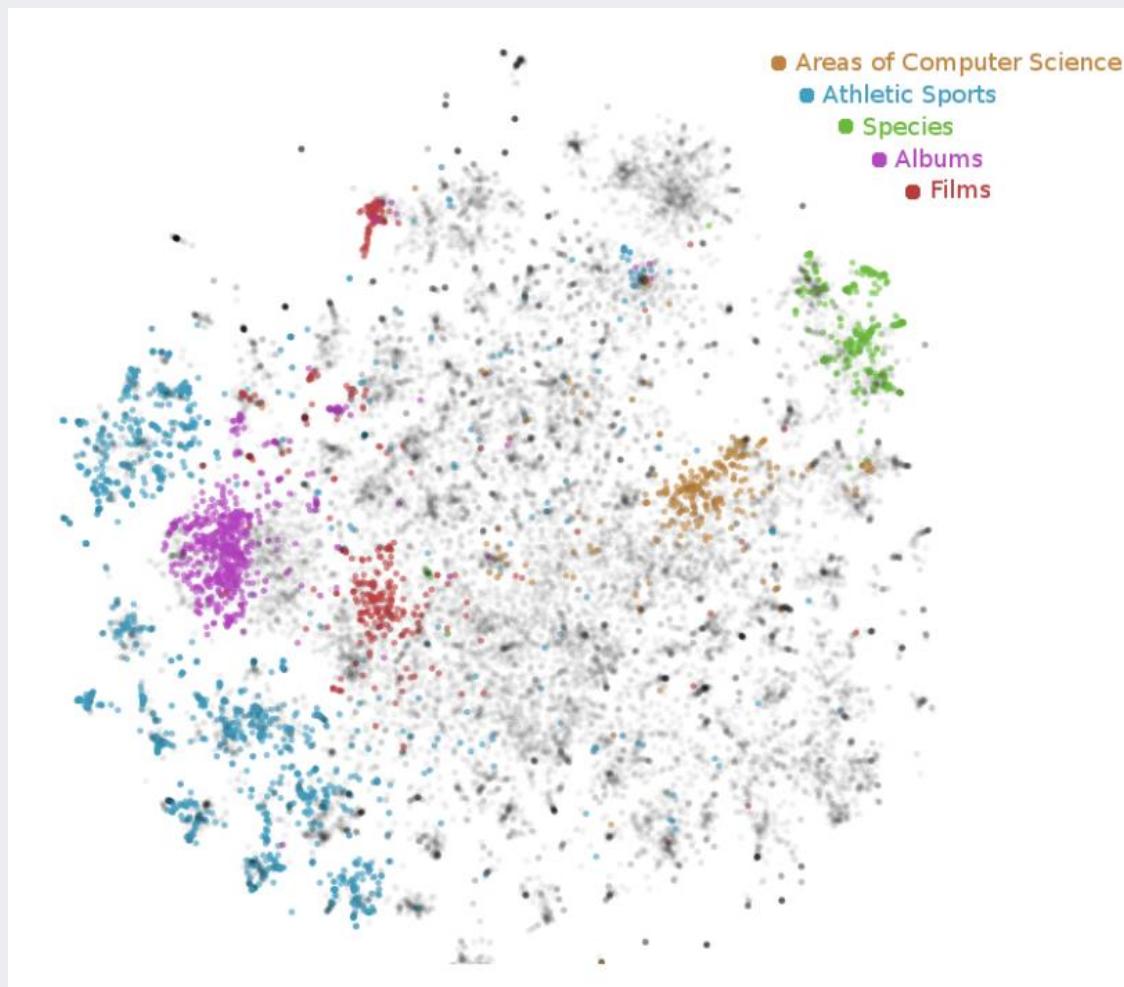


# Document Embedding

Dai et al. (2015)

- If we can embed words, why not sentences, phrases, or documents?

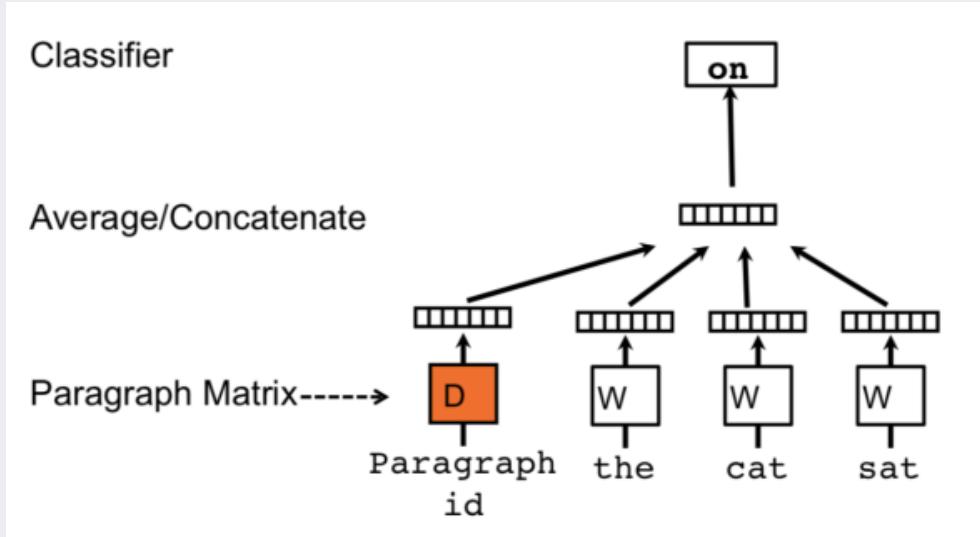
Visualization of Wikipedia paragraph vectors using t-SNE



# Document Embedding

Le and Mikolov (2015)

- Paragraph Vector model: Distributed Memory (PV-DM) model

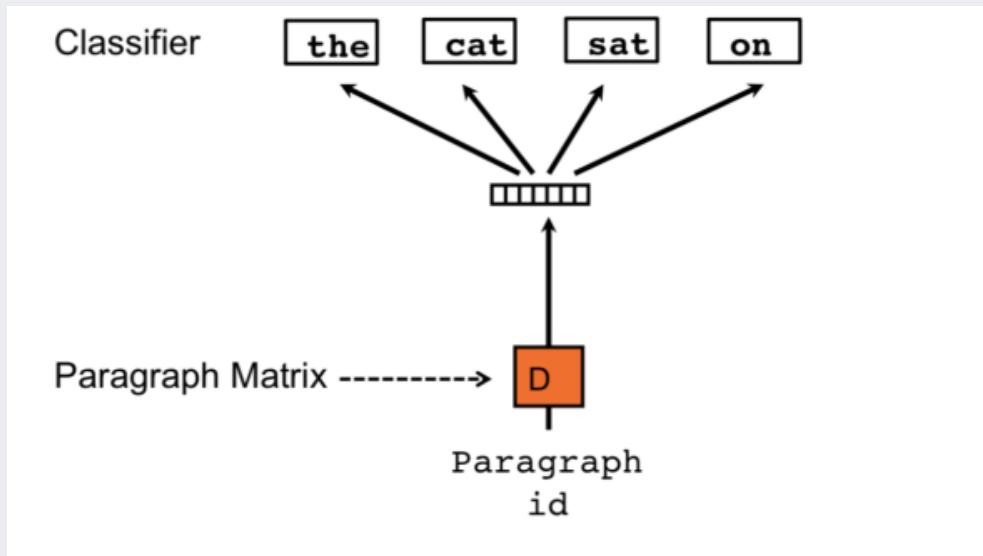


- ✓ The paragraph vectors are also asked to contribute to the prediction task of the next word given many contexts sampled from the paragraph
- ✓ Paragraph vectors are shared for all windows generated from the same paragraph, but not across paragraphs
- ✓ Word vectors are shared across all paragraphs

# Document Embedding

Le and Mikolov (2015)

- Paragraph Vector model: Distributed Bag of Words (PV-DBOW)

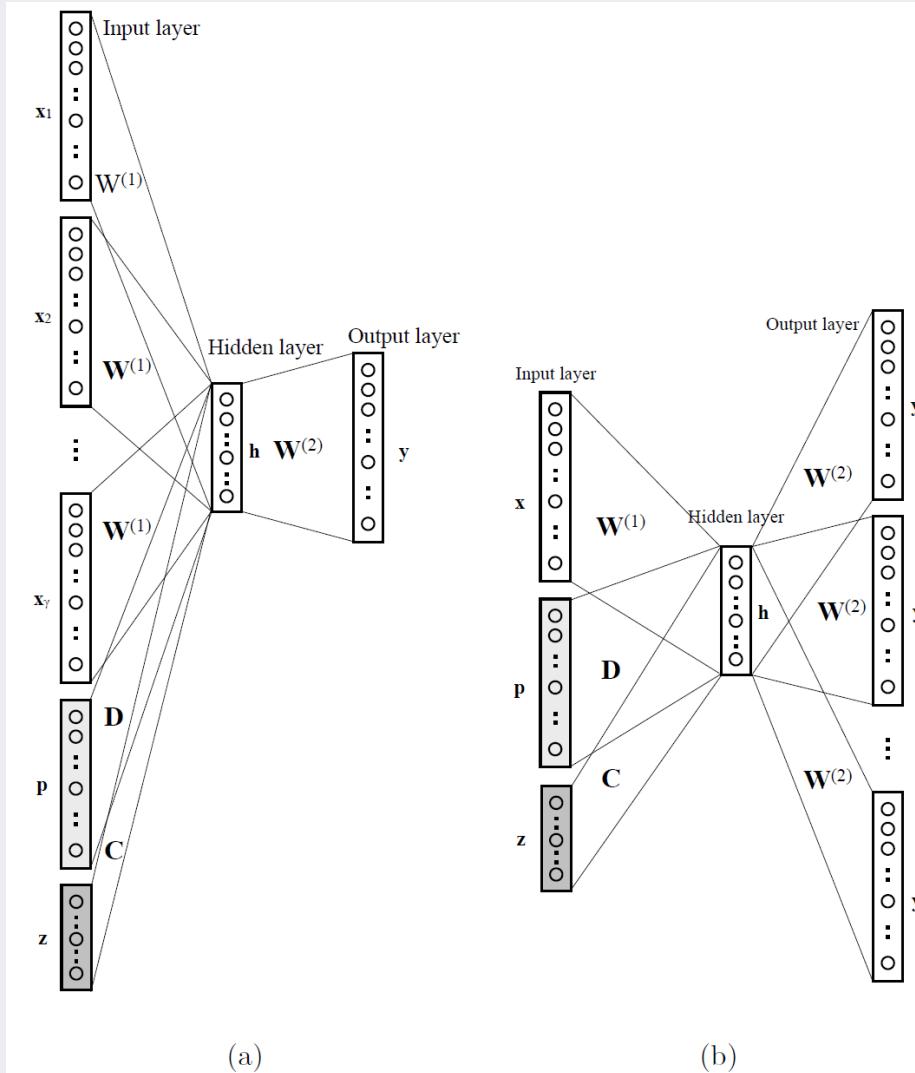


- ✓ Ignore the context words in the input, and force the model to predict words randomly sampled from the paragraph in the output
- ✓ Does not need word vectors
- ✓ PV-DM alone usually works well for most tasks, but the combination of PV-DM and PV-DBOW are recommended

# Let's Embed Everything!

Park et al. (2016+)

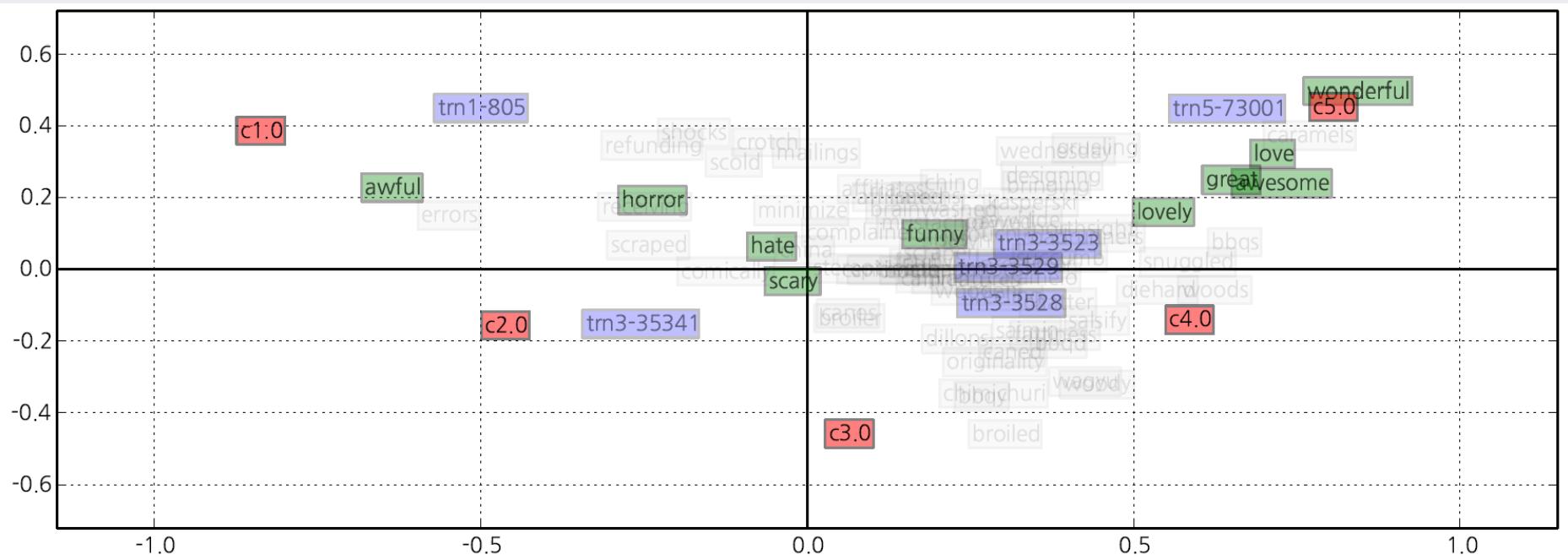
- Supervised Paragraph Vector (SPV) for Class Embedding



# Let's Embed Everything!

Park et al. (2016+)

- Supervised Paragraph Vector (SPV) for Class Embedding



# Let's Embed Everything!

Park et al. (2016+)

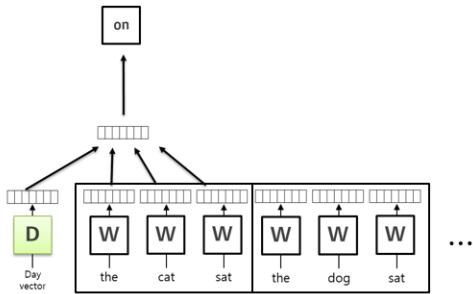
- Supervised Paragraph Vector (SPV) for Class Embedding

	imdb									yelp							
n(epochs)	1	5	10	30	50	70	100	t	n(epochs)	1	5	10	30	50	70	100	t
BOW-TF	<b>85.30</b>	-	-	-	-	-	-	-	BOW-TF	<b>58.42</b>	-	-	-	-	-	-	-
BOW-TFIDF	<b>85.55</b>	-	-	-	-	-	-	-	BOW-TFIDF	<b>58.93</b>	-	-	-	-	-	-	-
PV-DM	77.06	80.78	80.84	79.68	81.22	81.49	<b>82.16</b>	123.14	PV-DM	50.59	51.70	52.67	<b>52.97</b>	51.73	51.81	52.70	546.10
PV-DBOW	85.89	88.19	<b>88.47</b>	88.27	88.22	88.12	88.04	115.22	PV-DBOW	58.53	<b>59.37</b>	58.91	59.13	59.10	59.06	59.21	534.03
SPV-DM	82.57	81.68	82.05	<b>82.66</b>	82.42	82.53	82.61	121.51	SPV-DM	51.48	51.42	52.40	53.14	53.77	53.75	<b>53.84</b>	552.71
SPV-DBOW	87.58	* <b>88.87</b>	88.69	88.53	88.51	88.56	88.49	117.33	SPV-DBOW	60.13	* <b>60.21</b>	60.04	59.93	59.85	59.77	59.93	538.95
	amazon									20news							
n(epochs)	1	5	10	30	50	70	100	t	n(epochs)	1	5	10	30	50	70	100	t
BOW-TF	<b>85.91</b>	-	-	-	-	-	-	-	BOW-TF	<b>58.58</b>	-	-	-	-	-	-	-
BOW-TFIDF	<b>85.97</b>	-	-	-	-	-	-	-	BOW-TFIDF	<b>63.43</b>	-	-	-	-	-	-	-
PV-DM	76.47	77.38	78.86	<b>79.26</b>	75.83	77.00	78.57	361.52	PV-DM	24.45	41.17	45.36	49.04	52.34	53.85	<b>54.27</b>	71.64
PV-DBOW	86.87	87.96	88.30	88.34	88.31	<b>88.42</b>	88.18	339.14	PV-DBOW	53.51	69.16	72.51	74.76	75.22	75.16	<b>75.40</b>	70.37
SPV-DM	79.05	78.35	78.66	80.26	80.36	80.62	<b>80.70</b>	371.95	SPV-DM	44.76	64.18	67.01	67.84	68.32	<b>68.34</b>	67.70	71.88
SPV-DBOW	88.58	89.21	89.16	* <b>89.34</b>	89.08	89.06	89.29	342.40	SPV-DBOW	69.41	76.97	77.95	78.93	78.93	79.47	* <b>79.59</b>	72.18

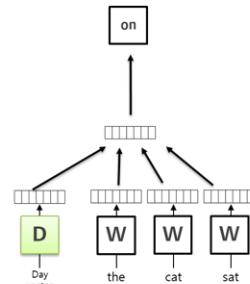
# Let's Embed Everything!

- Day Embedding in News corpus

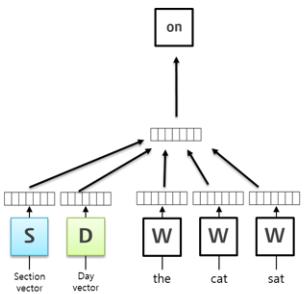
Approach 01: 하루 동안의 기사제목들을 병합 후 Day 태그



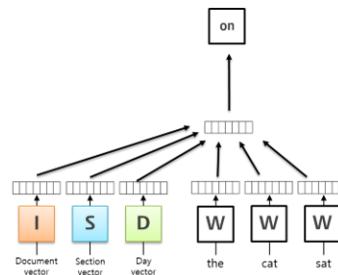
Approach 02: 각 뉴스기사 제목에 Day 태그



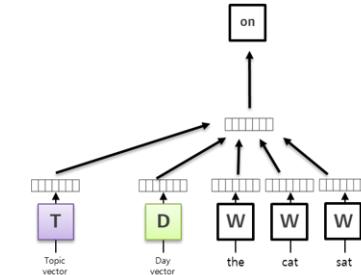
Approach 03: 각 뉴스기사제목에 Day, Section 태그



Approach 04: 각 뉴스기사제목에 Day, Section, IDX 태그



Approach 05: 각 뉴스기사제목에 Day, Topic 태그



# AGENDA

01 Bag of Words

---

02 Word Weighting

---

03 N-Grams

---

04 Distributed Representations

---

05 R Exercise

---

# R Exercise

- 10 speeches delivered by Mr. Barack Obama & Mitt Romney

<http://www.obamaspeeches.com/>

Best Speeches of Barack Obama through his 2009 Inauguration  
Most Recent Speeches are Listed First

- Barack Obama - Inaugural Speech
- Barack Obama - Election Night Victory / Presidential Acceptance Speech - Nov 4 2008
- Barack Obama - Night Before the Election - the Last Rally - Manassas Virginia - Nov 3 2008
- Barack Obama - Democratic Nominee Acceptance Speech 2008 National Democratic Convention
- Barack Obama - "A World that Stands as One" - Berlin Germany - July 2008
- Barack Obama - Final Primary Night Presumptive Nominee Speech
- Barack Obama - North

## Obama Inaugural Address 20th January 2009

My fellow citizens:

I stand here today humbled by the task before us, grateful for the trust you have bestowed, mindful of the sacrifices borne by our ancestors. I thank President Bush for his service to our nation, as well as the generosity and cooperation he has shown throughout this transition.

Forty-four Americans have now taken the presidential oath. The words have been spoken during rising tides of prosperity and the still waters of peace. Yet, every so often the oath is taken amidst gathering clouds and raging storms. At these moments, America has carried on not simply because of the skill or vision of those in high office, but because We the People have remained faithful to the ideals of our forbearers, and true to our founding documents.

So it has been. So it must be with this generation of Americans.

That we are in the midst of crisis is now well understood. Our nation is at war, against a far-reaching network of violence and hatred. Our economy is badly weakened, a consequence of greed and irresponsibility on the part of some, but also our collective failure to make hard choices and prepare the nation for a new age. Homes have been lost; jobs shed; businesses shuttered. Our health care is too costly; our schools fail too many; and each day brings further evidence that the ways we use energy strengthen our adversaries and threaten our planet.

These are the indicators of crisis, subject to data and statistics. Less measurable but no less profound is a sapping of confidence across our land - a nagging fear that America's decline is inevitable, and that the next generation must lower its sights.

Today I say to you that the challenges we face are real. They are serious and they are many. They will not be met easily or in a short span of time. But know this, America - they will be met.

On this day, we gather because we have chosen hope over fear, unity of purpose over conflict and discord.

<http://mittromneycentral.com/speeches/>

## I'm COMMITTED

ARE YOU comMITTED?

Home About Mitt On the Issues Speeches Op-eds Resources About MRC comMITTed

### Mitt Romney Speeches

[Tweet 12](#) [Facebook 133](#)

Here is a list of major speeches given by Governor Romney in recent years. Most of them have both video and transcript.

- Mitt Romney's Concession Speech – 11/7/12
- Real Change From Day One – 11/2/12
- Remarks on the American Economy – 10/26/12
- Foreign Policy Speech 'The Mantel of Leadership' – 10/08/12
- The Clinton Global Initiative – 09/25/12
- U.S. Hispanic Chamber of Commerce – 09/17/12
- The National Guard Association Conference – 09/11/12
- Romney Accepts the GOP Nomination in Tampa – 08/30/12
- Mitt Romney's Speech in Chillicothe, Ohio – 08/14/12
- Mitt Romney & Paul Ryan in Wisconsin – 08/12/12
- Mitt Romney Makes His VP Announcement – 08/11/12
- Mitt Romney's Policy Speech in Jerusalem – 07/29/12
- Remarks At The VFW National Convention – 07/24/12
- Remarks to the NAACP Convention – 07/11/12
- Remarks to NALEO: "Growing Opportunity for All Americans" – 06/21/12
- "A Champion for Free Enterprise" – 06/07/12
- Remarks On Education: "A Chance For Every Child" – 05/23/12

Search OK

Want MRC Sent to Your Inbox?  
Enter your email address:

Like Us on Facebook

**MR 16** [Like](#)  
122,960 people like Mitt Romney Central.

[Facebook social plugin](#)

# R Exercise

- 10 speeches delivered by Mr. Barack Obama & Mitt Romney

Speaker	Speech
1 Obama	<U+FEFF>I stand here today humbled by the task before us, grateful for the trust you have bestowed, mindful of the sacrifices borne by our ancestors. I thank President Bush for his service to
2 Obama	<U+FEFF>If there is anyone out there who still doubts that America is a place where all things are possible; who still wonders if the dream of our founders is alive in our time; who still que
3 Obama	<U+FEFF>what a scene. What a crowd. Thank you for Virginia. (Crowd chants "yes we can.") Let me start by noting, Virginia that this is our last rally. This is the last rally of a campaign tha
4 Obama	<U+FEFF>Well, we are in the middle of a very close race right now in Texas, and we may not even know the final results until morning. We do know that Senator Clinton has won Rhode Island, and
5 Obama	<U+FEFF>To Chairman Dean and my great friend Dick Durbin; and to all my fellow citizens of this great nation; With profound gratitude and great humility, I accept your nomination for the pres
6 Obama	<U+FEFF>Tonight, after fifty-four hard-fought contests, our primary season has finally come to an end. Sixteen months have passed since we first stood together on the steps of the Old State C
7 Obama	<U+FEFF>You know, some were saying that North Carolina would be a game-changer in this election. But today, what North Carolina decided is that the only game that needs changing is the one in
8 Obama	<U+FEFF>I want to start by congratulating Senator Clinton on her victory tonight, and I want to thank the hundreds of thousands of Pennsylvanians who stood with our campaign today. There were
9 Obama	<U+FEFF>Good afternoon. I know I kept a lot of you guys busy this weekend with the comments I made last week. Some of you might even be a little bitter about that. As I said yesterday, I regr
10 Obama	<U+FEFF>"We the people, in order to form a more perfect union." Two hundred and twenty one years ago, in a hall that still stands across the street, a group of men gathered and, with these si
11 Romney	<U+FEFF>Thank you all. It♦♦s great to be back in Iowa. And don♦♦t think that this is the last time you are going to see Paul Ryan and me, because you Iowans may well be the ones who decide
12 Romney	<U+FEFF>Thank you. It♦♦s good to be in the home state of the next Vice President of the United States. Next to Ann Romney, Paul Ryan is the best choice I ever made. Tonight, we enter the fi We are so very grateful to you and to people across the country, for all that you have given of yourselves to this campaign. This is not just about Paul and me<U+2013>it is about America, an
13 Romney	<U+FEFF>I particularly appreciate the introduction from my good friend and tireless campaign companion, Gov. Bob McDonnell. He is showing what conservative leadership can do to build a strong This is the struggle that is now shaking the entire Middle East to its foundation. It is the struggle of millions and millions of people<U+2014>men and women, young and old, Muslims, Christi It is time to change course in the Middle East. That course should be organized around these bedrock principles: America must have confidence in our cause, clarity in our purpose and resolve I know many Americans are asking a different question: ♦♦Why us?♦♦ I know many Americans are asking whether our country today<U+2014>with our ailing economy, and our massive debt, and aft
14 Romney	<U+FEFF>Thank you, Mr. President. I appreciate the kind words and your invitation here today. If there♦♦s one thing we♦♦ve learned this election season, it♦♦s that a few words from Bill I just want to work. Work. That must be at the heart of our effort to help people build economies that can create jobs for people, young and old alike. Work builds self-esteem. It transforms
15 Romney	<U+FEFF>Thank you, Hector, for that warm introduction. And congratulations to Cristina. I♦♦m pleased to be your guest, and to speak as we begin National Hispanic Heritage Month. I♦♦m also I love hearing stories like that. I am convinced that the Republican Party is the rightful home of Hispanic Americans. But my speech today isn♦♦t about my political party. It♦♦s about the
16 Romney	<U+FEFF>Major General Vavala, thank you for your generous introduction. And thank you for your years of service as Chairman of the Board <U+2013> and for your decades of service to our nation Two weeks ago, I saw the Guard in action in Louisiana after it was hit by Hurricane Isaac. For many of the people of the Gulf <U+2013> who had just finished repairing their homes and getting
17 Romney	<U+FEFF>Thank you for that kind introduction, Mayor Barkat, and thank you all for that warm welcome. It♦♦s a pleasure and a privilege to be in Israel again. To step foot into Israel is to s So it is today, as Israel faces enemies who deny past crimes against the Jewish people and seek to commit new ones. When Iran♦♦s leaders deny the Holocaust or speak of wiping this nation o We have seen the horrors of history. We will not stand by. We will not watch them play out again. It would be foolish not to take Iran♦♦s leaders at their word. They are, after all, the pr
18 Romney	<U+FEFF>Thank you. Commander Richard DeNoyer, I appreciate the introduction, and I♦♦m proud to see a combat veteran from Massachusetts serving as National Commander of the VFW. Ladies Auxil This conduct is contemptible. It betrays our national interest. It compromises our men and women in the field. And it demands a full and prompt investigation by a special counsel, with expla Exactly who in the White House betrayed these secrets? Did a superior authorize it? These are things that Americans are entitled to know <U+2013> and they are entitled to know right now. If Why is flexibility with Russian leaders more important than transparency to the American people? President Obama had a moment of candor, however, just the other day. He said that the actions
19 Romney	<U+FEFF>Thank you all very much. It♦♦s good to be back in Iowa. So many friends here hold a special place in my heart. I♦♦ve come here today to talk to you about an issue that affects the Imagine if the federal government was the sole legal supplier of cell phones. First, they♦♦d still be under review, with hearings in Congress. When finally approved, the contract to make t We see the same bureaucracy and overhead in our anti-poverty programs. Last year, the federal government spent more than \$600 billion on more than 100 different programs that aim to help the
20 Romney	<U+FEFF>For the graduates, this moment marks a clear ending and a clear beginning. The task set before you four years ago is now completed in full. To the class of 2012: Well done, and congra Maybe the most confident step Jerry ever took was to open the doors of this school 41 years ago. He believed that Liberty might become one of the most respected Christian universities anywhere In this life, the commitments that come closest to forever are those of family. My Dad, George Romney, was a CEO, a governor, and a member of the President♦♦s Cabinet. My wife Ann asked hi Ann and I feel the same way about our family. I have never once regretted missing a business opportunity so that I could be with my children and grandchildren. Among the things in life that Promotions often mark the high points in a career, and I hope I haven♦♦t seen my last. But sometimes the high points come in unexpected ways. I was asked to help rescue the 2002 Olympics i The Olympics were not a logical choice, but it was one of the best and most fulfilling choices of my life. Opportunities for you to serve in meaningful ways may come at inconvenient times, b

# R Exercise

- Preprocessing

- ✓ Lower case transformation, remove punctuation/numbers/stopwords

- ✓ Stemming

```
1 install.packages("RWeka", dependencies = TRUE)
2 install.packages("tm", dependencies = TRUE)
3
4 library(RWeka)
5 library(tm)
6
7 load("speechdata.RData")
8
9 # Construct corpuses
10 # VectorSource specifies that the source is character vectors.
11 myCorpus <- Corpus(VectorSource(SpeechData$Speech))
12
13 # Preprocessing
14 # 1: to lower case
15 PreCorpus <- tm_map(myCorpus, content_transformer(tolower))
16
17 # 2: remove puntuations
18 PreCorpus <- tm_map(PreCorpus, content_transformer(removePunctuation))
19
20 # 3. remove numbers
21 PreCorpus <- tm_map(PreCorpus, content_transformer(removeNumbers))
22
23 # 4. remove stopwords (SMART stopwords list + obama + romney)
24 myStopwords <- c(stopwords("SMART"), "obama", "romney")
25
26 PreCorpus <- tm_map(PreCorpus, removewords, myStopwords)
27
28 # 5. Stemming
29 stemCorpus <- tm_map(PreCorpus, stemDocument)
```

# R Exercise

- Preprocessing

- ✓ Lower case transformation, remove punctuation/numbers/stopwords

- ✓ Stemming

```
> myCorpus[[3]]  
<<PlainTextDocument (metadata: 7)>>  
<U+FEFF>What a scene. What a crowd. Thank you for Virginia. (Crowd chants "yes we can.") Let me start by noting, Virginia that this is our last rally. This is the last rally of a campaign that began nearly 2 years ago. We've gone to every corner of this country, from here in Northern Virginia to the rocky coasts of Maine, to the open plains of Texas, to the open skies of Montana. I just want to say that whatever happens tomorrow, I have been deeply humbled by this journey. You have welcomed Michelle and me and the girls into your homes. You have shared your stories of struggle, you have spoken of your dreams, along the way, talking with all of you about your own lives. You have enriched my life, you have moved me again and again. You have inspired me. Sometimes when I have been down you have lifted me up. You filled me with new hope for our future and you have reminded me about what makes America so special. In the places I have gone and the people I have met, I have been struck again and again by the fundamental decency and generosity and dignity of men and women who work hard without complaint, to meet their responsibilities every day. I come away with an unyielding belief that if we only had a government as responsible as all of you, as compassionate as the American people, that there is no obstacle that we can't overcome. There is no destiny that we cannot fulfill. Virginia, I have just one word for you, just one word. Tomorrow. Tomorrow. After decades of broken politics in Washington, 8 years of failed policies from George Bush, twenty-one months of campaigning, we are less than one day away from bringing about change in America. Tomorrow you can turn the page on policies that put greed and irresponsibility before hard work and sacrifice. Tomorrow you can choose policies that invest in our middle class, create new jobs and grow this economy so that everybody has a chance to succeed. Not just the CEO but the secretary and the janitor; not just the factory owner but the men and women who work the factory floor. Tomorrow you can pu
```



```
> stemCorpus[[3]]  
<<PlainTextDocument (metadata: 7)>>  
scene crowd virginia crowd chant start note virginia ralli ralli campaign began year ago weve corner countri northern virgi  
nia rocki coast main open plain texa open sky montana tomorrow deepli humbl journey welcom michell girl home share stori stru  
ggl spoken dream talk live enrich life move inspir lift fill hope futur remind make america special place  
peopl met struck fundament decenc generos digniti men women work hard complaint meet respons day unyield belief govern respons  
compassion american peopl obstacl overcom destini fulfil virginia word word tomorrow tomorrow decad broken polit washington year fa  
il polici georg bush twentyon month campaign day bring chang america tomorrow turn page polici put greed irrespons hard work sacrific tomorr  
ow choos polici invest middl class creat job grow economi chanc succeed ceo secretari janitor factori owner men women work factori  
floor tomorrow put end polit divid nation win elect put reason reason citi town republican democrat ask fear time hope tomorrow defi  
n moment histori give countri chang start virginia start manassa chang begin campaign perfect time back iive wouldnt though  
t bit ill campaign weve lot proud tone set argu issu engag person attack weve fierc defend weve make remind suppor  
t black white hispan nativ american asian democrat republican young rich poor gay straight disabl disabl contribut communic year  
afford polit game tactic pit make afraid afford anymor time oppon claim real fake part virginia anymor real fake part america  
citi town proamerica nation proud patriot salut flag men women serv battlefield walk life polit parti fought bled die  
proud flag serv red america blue america serv unit state america campaign call serv unit state america campaign privileq wit am
```

# R Exercise

- Construct a Term-Document Matrix

```
31 # Term-Document Matrix: without Preprocessing  
32 myTDM <- TermDocumentMatrix(myCorpus, control = list(minWordLength = 1))  
33  
34 # Term-Document Matrix: with Preprocessing  
35 stemTDM <- TermDocumentMatrix(stemCorpus, control = list(minWordLength = 1))
```

## ✓ Without preprocessing

- 7,873 terms in 20 documents

```
> myTDM  
<<TermDocumentMatrix (terms: 7873, documents: 20)>>  
Non-/sparse entries: 18734/138726  
Sparsity           : 88%  
Maximal term length: 27  
Weighting          : term frequency (tf)
```

Term reduction ratio:

$$\frac{4382}{7873} = 55.66\%$$

## ✓ With preprocessing

- 3,491 terms in 20 documents

```
> stemTDM  
<<TermDocumentMatrix (terms: 3491, documents: 20)>>  
Non-/sparse entries: 11045/58775  
Sparsity           : 84%  
Maximal term length: 19  
Weighting          : term frequency (tf)
```

# R Exercise

- Construct a Term-Document Matrix

- ✓ Check the term-frequency

Terms	Docs																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
abandon	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
abbottabad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
abid	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0
abil	1	0	0	0	0	0	1	0	0	0	0	0	2	1	0	0	1	2	0	1	
abort	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aboutit	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
aboutmemori	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
abroad	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	
abrupt	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
absenc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
absent	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
absolut	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	1	
absorb	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
abund	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
abus	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
accept	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	
access	0	0	0	0	1	0	0	0	1	1	0	0	2	0	0	0	0	1	0	0	
accompani	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
accomplish	0	0	0	0	0	1	1	0	0	0	2	1	0	0	0	0	0	1	0	1	
account	1	0	0	0	2	0	0	0	1	1	0	1	1	0	0	0	3	1	0		
accus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
achiev	1	4	0	0	1	0	1	0	3	2	5	1	1	3	0	4	1	0	3		
acknowledg	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	
acquir	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
acrosstheboard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
act	1	0	2	0	0	0	1	0	0	1	5	2	0	0	3	0	0	1	0	1	
action	1	0	0	0	0	0	0	0	3	0	0	0	0	0	1	2	0	2	0	1	
actionsnot	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
activeutedi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
activelyto	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	

# R Exercise

- TF Variants

✓ Natural, Logarithm, Augmented

```
40 # TF-IDF Variants
41 # TF 1: Natural
42 natural_tf <- as.matrix(stemTDM)
43
44 # TF 2: Logarithm
45 log_tf <- log(1+natural_tf)
46
47 # TF 3: augmented
48 max_tf <- apply(natural_tf, 2, max)
49 augmented_tf <- 0.5+0.5*t(t(natural_tf)/max_tf)
```

Terms	Docs				
	1	2	3	4	5
abandon	1	0	0	0	1
abbottabad	0	0	0	0	0
abid	0	0	0	0	0
abil	1	0	0	0	0
abort	0	0	0	0	1
aboutit	0	0	0	0	0
aboutmemori	0	0	0	0	0
abroad	0	0	0	0	1
abrupt	0	0	0	0	0
absenc	0	0	0	0	0
absent	0	0	0	0	0
absolut	0	0	0	0	0
absorb	0	0	0	0	0
abund	0	0	0	0	0
abus	0	0	0	0	0
accept	1	0	0	0	1
access	0	0	0	0	1
accompani	0	0	0	0	1
accomplish	0	0	0	0	0
account	1	0	0	0	2

Terms	Docs				
	1	2	3	4	5
abandon	0.6931472	0	0	0	0.6931472
abbottabad	0.0000000	0	0	0	0.0000000
abid	0.0000000	0	0	0	0.0000000
abil	0.6931472	0	0	0	0.0000000
abort	0.0000000	0	0	0	0.6931472
aboutit	0.0000000	0	0	0	0.0000000
aboutmemori	0.0000000	0	0	0	0.0000000
abroad	0.0000000	0	0	0	0.6931472
abrupt	0.0000000	0	0	0	0.0000000
absenc	0.0000000	0	0	0	0.0000000
absent	0.0000000	0	0	0	0.0000000
absolut	0.0000000	0	0	0	0.0000000
absorb	0.0000000	0	0	0	0.0000000
abund	0.0000000	0	0	0	0.0000000
abus	0.0000000	0	0	0	0.0000000
accept	0.6931472	0	0	0	0.6931472
access	0.0000000	0	0	0	0.6931472
accompani	0.0000000	0	0	0	0.6931472
accomplish	0.0000000	0	0	0	0.0000000
account	0.6931472	0	0	1	0.0986123

Terms	Docs				
	1	2	3	4	5
abandon	0.5333333	0.5	0.5	0.5	0.515625
abbottabad	0.5000000	0.5	0.5	0.5	0.500000
abid	0.5000000	0.5	0.5	0.5	0.500000
abil	0.5333333	0.5	0.5	0.5	0.500000
abort	0.5000000	0.5	0.5	0.5	0.515625
aboutit	0.5000000	0.5	0.5	0.5	0.500000
aboutmemori	0.5000000	0.5	0.5	0.5	0.500000
abroad	0.5000000	0.5	0.5	0.5	0.515625
abrupt	0.5000000	0.5	0.5	0.5	0.500000
absenc	0.5000000	0.5	0.5	0.5	0.500000
absent	0.5000000	0.5	0.5	0.5	0.500000
absolut	0.5000000	0.5	0.5	0.5	0.500000
absorb	0.5000000	0.5	0.5	0.5	0.500000
abund	0.5000000	0.5	0.5	0.5	0.500000
abus	0.5000000	0.5	0.5	0.5	0.500000
accept	0.5333333	0.5	0.5	0.5	0.515625
access	0.5000000	0.5	0.5	0.5	0.515625
accompani	0.5000000	0.5	0.5	0.5	0.515625
accomplish	0.5000000	0.5	0.5	0.5	0.500000
account	0.5333333	0.5	0.5	0.5	0.531250

# R Exercise

- IDF Vector

```
> idf_vec[1:20]
   abandon abbottabad    abid      abil      abort    aboutit  aboutmemori    abroad    abrupt    absenc    absent    absolut    absorb
1.6094379 2.9957323 2.3025851 1.0498221 2.9957323 2.9957323 2.9957323 1.8971200 2.9957323 2.9957323 2.9957323 1.3862944 2.9957323
  abund     abus    accept    access accompani accomplish account
2.9957323 2.9957323 1.3862944 1.3862944 2.3025851 1.2039728 0.9162907
```

- TF-IDF Variants

✓ Natural TF \* IDF vs. Log TF \* IDF (cosine normalized)

```
> TFIDF_1[1:20,1:10]
          Docs
Terms      1 2 3 4      5      6      7 8 9      10
abandon  1.6094379 0 0 0 1.609438 0.0000000 0.0000000 0 0 0.0000000
abbottabad 0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
abid      0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
abil      1.0498221 0 0 0 0.0000000 0.0000000 1.049822 0 0 0.0000000
abort      0.0000000 0 0 0 2.995732 0.0000000 0.0000000 0 0 0.0000000
aboutit     0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
aboutmemori 0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 2.9957323
abroad     0.0000000 0 0 0 1.897120 0.0000000 0.0000000 0 0 0.0000000
abrupt     0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
absenc     0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
absent     0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
absolut    0.0000000 0 0 0 0.0000000 1.386294 0.000000 0 0 1.3862944
absorb     0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
abund      0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
abus       0.0000000 0 0 0 0.0000000 0.0000000 0.0000000 0 0 0.0000000
accept     1.3862944 0 0 0 1.386294 0.000000 0.000000 0 0 1.3862944
access     0.0000000 0 0 0 1.386294 0.000000 0.000000 0 0 1.3862944
accompani 0.0000000 0 0 0 2.302585 0.000000 0.000000 0 0 0.0000000
accomplish 0.0000000 0 0 0 0.000000 1.203973 1.203973 0 0 0.0000000
account    0.9162907 0 0 0 1.832581 0.000000 0.000000 0 0 0.9162907
```

```
> TFIDF_2[1:20,1:10]
          Docs
Terms      1 2 3 4      5      6      7 8 9      10
abandon  0.03323211 0 0 0 0.02963354 0.00000000 0.00000000 0 0 0.00000000
abbottabad 0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
abid      0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
abil      0.02167701 0 0 0 0.00000000 0.00000000 0.03004081 0 0 0.00000000
abort      0.00000000 0 0 0 0.05515848 0.00000000 0.00000000 0 0 0.00000000
aboutit     0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
aboutmemori 0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.03873698
abroad     0.00000000 0 0 0 0.03493045 0.00000000 0.00000000 0 0 0.00000000
abrupt     0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
absenc     0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
absent     0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
absolut    0.00000000 0 0 0 0.00000000 0.03851873 0.00000000 0 0 0.01792579
absorb     0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
abund      0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
abus       0.00000000 0 0 0 0.00000000 0.00000000 0.00000000 0 0 0.00000000
accept     0.02862458 0 0 0 0.02552494 0.00000000 0.00000000 0 0 0.01792579
access     0.00000000 0 0 0 0.02552494 0.00000000 0.00000000 0 0 0.01792579
accompani 0.00000000 0 0 0 0.04239601 0.00000000 0.00000000 0 0 0.00000000
accomplish 0.00000000 0 0 0 0.00000000 0.03345285 0.03445186 0 0 0.00000000
account    0.01891982 0 0 0 0.02674001 0.00000000 0.00000000 0 0 0.01184830
```

# R Exercise

- N-Gram TF matrix (N=2,3)

```
69 # Construct N-gram matrix
70 tmpTokenizer <- function(x) NGramTokenizer(x, weka_control(min = 2, max = 3))
71 NgramTDM <- TermDocumentMatrix(stemCorpus, control = list(tokenize = tmpTokenizer))
72 inspect(NgramTDM[30000:30100,1:20])
```

```
> inspect(NgramTDM[30000:30100,1:20])
<<TermDocumentMatrix (terms: 101, documents: 20)>>
Non-/sparse entries: 111/1909
Sparsity           : 95%
Maximal term length: 26
Weighting          : term frequency (tf)

Terms              Docs
shift divers visa   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
shift full          0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
shift full day      0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
shift mom           0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
shift mom rais      0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
shift servic         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
shift servic program 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
shini object         0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
shini object elect   0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
ship china           0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ship china choke     0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ship job              0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ship job oversea      0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ship oversea           0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0
ship oversea pension    0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
ship oversea profit     0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
ship unit              0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
ship unit state        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
ship year              0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
ship year includ       0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
shock flame            0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
shock flame smoke      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
shock ignor             0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
shock ignor struggl    0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
```



ANY  
questions?

# References

## Research Papers

- Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3, 1137-1155.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- Collobert et al. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* 12: 2493-2537.
- Dai, A. M., Olah, C., & Le, Q. V. (2015). Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*.
- Furnkranz, J. (1998). A Study using N-gram Features for Text Categorization. Austrian Research Institute for Artificial Intelligence Technical Report OEFAI-TR-98-30 Schottengasse.
- Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In *International Conference on Machine Learning* (pp. 1188-1196).
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).
- Paltoglou, G. and Thelwall, M. (2010). [A Study of Information Retrieval Weighting Schemes for Sentiment Analysis](#). In *Proceedings of the 48<sup>th</sup> Annual Meeting of the Association for Computational Linguistics*: 1386-1395.
- Park, E., Kang, P., & Cho, S. (2016). Supervised paragraph vector: document classification by jointly embedding words, documents, and class labels.
- Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global Vectors for Word Representation. In *EMNLP* (Vol. 14, pp. 1532-1543).
- Roelleke, T. (2013). *Information Retrieval Models: Foundations and Relationships*. Morgan & Claypool Publishers.
- Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

# References

## Other materials

- Figure in the front page: <http://slideplayer.com/slide/7073400/>
- Bojanowski, P. (2016). fastest: A library for efficient text classification and word representation.  
<https://www.youtube.com/watch?v=CHcExDsDeHU>
- CS224d-L2: Lecture note on “Simple Word Vector representations: word2vec, GloVe”, <http://cs224d.stanford.edu/lectures/CS224d-Lecture2.pdf>
- CS224d-L3: Lecture note on “Advanced word vector representations: language models, softmax, single layer networks”,  
<http://cs224d.stanford.edu/lectures/CS224d-Lecture3.pdf>
- Images in the title page: <http://nlp.stanford.edu/projects/glove/>
- Kauchak, D. (2009). TF-IDF. <http://www.cs.pomona.edu/~dkauchak/classes/f09/cs160-f09/lectures/lecture5-tfidf.pdf>
- Kim, Y., Hernite, Y., Sontag, D., and Rush, A. (2016). Character-Aware Neural Language Models, AAAI 2016 presentation slides,  
[http://www.people.fas.harvard.edu/~yoonkim/data/aaai\\_2016\\_slides.pdf](http://www.people.fas.harvard.edu/~yoonkim/data/aaai_2016_slides.pdf)
- McCormick, C. Word2Vec Tutorial Part I & 2, <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>
- Nayak, P. and Raghavan, P. (2014). Lecture 6: Scoring, Term Weighting and the Vector Space Model.  
<http://web.stanford.edu/class/cs276/handouts/lecture6-tfidf-handout-1-per.pdf>
- Word2Vec online example: <http://bit.ly/wevi-online>
- 박은정, Supervised Feature Representations for Document Classification, PhD Thesis, Seoul National University, 2016.
- 이재천, 김수경, 흥성연. (2015). Data Mining을 이용한 고려대 강의평가 분석: Klue 사이트 강의평가를 기준으로. 2015 캡스톤디자인 II Term Project.