



THE UNIVERSITY OF CHICAGO
GRAHAM SCHOOL
CONTINUING LIBERAL AND PROFESSIONAL STUDIES



Natural Language Processing

Session 3

Nick Kadochnikov



Session 3 Agenda

- Minimum Edit Distance
- N-Grams
- Spelling Correction and the Noisy Channel





How similar are two strings?

- Spell correction

- The user typed “graffe”

Which is closest?

- graf
 - graft
 - grail
 - giraffe

- Computational Biology

- Align two sequences of nucleotides

```

AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTGCCCCGAC
    
```

- Resulting alignment:

```

-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC--GGTCGATTGCCCCGAC
    
```

- Also for Machine Translation, Information Extraction, Speech Recognition



Edit Distance

- The minimum edit distance between two strings
- Is the minimum number of editing operations
 - Insertion
 - Deletion
 - Substitution
- Needed to transform one into the other



Minimum Edit Distance

- Two strings and their **alignment**:

I	N	T	E	*	N	T	I	O	N
*	E	X	E	C	U	T	I	O	N



Minimum Edit Distance

I	N	T	E	*	N	T	I	O	N
*	E	X	E	C	U	T	I	O	N
d	s	s		i	s				

- If each operation has cost of 1
 - Distance between these is 5
- If substitutions cost 2 (Levenshtein)
 - Distance between them is 8



Other uses of Edit Distance in NLP

- Evaluating Machine Translation and speech recognition

R Spokesman confirms senior government adviser was shot

H Spokesman said the senior adviser was shot dead

S

I

D

I

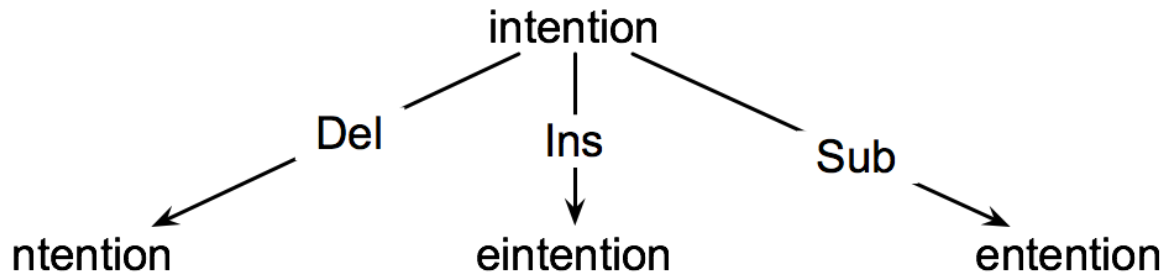
- Named Entity Extraction and Entity Coreference

- IBM Inc. announced today
- IBM profits
- Stanford President John Hennessy announced yesterday
- for Stanford University President John Hennessy



How to find the Min Edit Distance?

- Searching for a path (sequence of edits) from the start string to the final string:
 - **Initial state:** the word we're transforming
 - **Operators:** insert, delete, substitute
 - **Goal state:** the word we're trying to get to
 - **Path cost:** what we want to minimize: the number of edits





Minimum Edit as Search

- But the space of all edit sequences is huge!
 - We can't afford to navigate naïvely
 - Lots of distinct paths wind up at the same state.
 - We don't have to keep track of all of them
 - Just the shortest path to each of those revisited states.



Defining Min Edit Distance

- For two strings
 - X of length n
 - Y of length m
- We define $D(i,j)$
 - the edit distance between $X[1..i]$ and $Y[1..j]$
 - i.e., the first i characters of X and the first j characters of Y
 - The edit distance between X and Y is thus $D(n,m)$

Minimum Edit Distance

Weighted Minimum Edit Distance



Weighted Edit Distance

- Why would we add weights to the computation?
 - Spell Correction: some letters are more likely to be mistyped than others
 - Biology: certain kinds of deletions or insertions are more likely than others



Confusion matrix for spelling errors

sub[X, Y] = Substitution of X (incorrect) for Y (correct)

X	Y (correct)																									
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0



Introduction to N-grams



Probabilistic Language Models

- Today's goal: assign a probability to a sentence

- Machine Translation:

- $P(\text{high winds tonite}) > P(\text{large winds tonite})$

- Spell Correction

- The office is about fifteen **minuets** from my house

- $P(\text{about fifteen minutes from}) > P(\text{about fifteen minuets from})$

- Speech Recognition

- $P(\text{I saw a van}) \gg P(\text{eyes awe of an})$

- + Summarization, question-answering, etc., etc.!!

Why?



Probabilistic Language Modeling

- Goal: compute the probability of a sentence or sequence of words:

$$P(W) = P(w_1, w_2, w_3, w_4, w_5 \dots w_n)$$

- Related task: probability of an upcoming word:

$$P(w_5 | w_1, w_2, w_3, w_4)$$

- A model that computes either of these:

$P(W)$ or $P(w_n | w_1, w_2 \dots w_{n-1})$ is called a **language model**.

- Better: **the grammar** But **language model** or **LM** is standard



How to compute $P(W)$

- How to compute this joint probability:
 - $P(\text{its, water, is, so, transparent, that})$
- Intuition: let's rely on the Chain Rule of Probability



Reminder: The Chain Rule

- Recall the definition of conditional probabilities

Rewriting:

- More variables:

$$P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$$

- The Chain Rule in General

$$P(x_1, x_2, x_3, \dots, x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \dots P(x_n|x_1, \dots, x_{n-1})$$



The Chain Rule applied to compute joint probability of words in sentence

$$P(w_1 w_2 \square \dots w_n) = \prod_i P(w_i \mid w_1 w_2 \square \dots w_{i-1})$$

$P(\text{"its water is so transparent"}) =$

$P(\text{its}) \times P(\text{water} \mid \text{its}) \times P(\text{is} \mid \text{its water})$

$\times P(\text{so} \mid \text{its water is}) \times P(\text{transparent} \mid \text{its water is so})$



How to estimate these probabilities

- Could we just count and divide?

$$P(\text{the} \mid \text{its water is so transparent that}) = \frac{\textit{Count}(\text{its water is so transparent that the})}{\textit{Count}(\text{its water is so transparent that})}$$

- No! Too many possible sentences!
- We'll never see enough data for estimating these



Markov Assumption



Andrei Markov

- Simplifying assumption:

$P(\text{the} \mid \text{its water is so transparent that}) \gg P(\text{the} \mid \text{that})$

- Or maybe

$P(\text{the} \mid \text{its water is so transparent that}) \gg P(\text{the} \mid \text{transparent that})$



Markov Assumption

$$P(w_1 w_2 \square \dots w_n) \approx \prod_i P(w_i | w_{i-k} \square \dots w_{i-1})$$

- In other words, we approximate each component in the product

$$P(w_i | w_1 w_2 \square \dots w_{i-1}) \approx P(w_i | w_{i-k} \square \dots w_{i-1})$$



Simplest case: Unigram model

$$P(w_1 w_2 \square w_n) \gg \prod_i P(w_i)$$

Some automatically generated sentences from a unigram model

fifth, an, of, futures, the, an, incorporated, a,
a, the, inflation, most, dollars, quarter, in, is,
mass

thrift, did, eighty, said, hard, 'm, july, bullish

that, or, limited, the



Bigram model

- Condition on the previous word:

$$P(w_i | w_1 w_2 \square \dots w_{i-1}) \gg P(w_i | w_{i-1})$$

texaco, rose, one, in, this, issue, is, pursuing, growth, in,
a, boiler, house, said, mr., gurria, mexico, 's, motion,
control, proposal, without, permission, from, five, hundred,
fifty, five, yen

outside, new, car, parking, lot, of, the, agreement, reached
this, would, be, a, record, november



N-gram models

- We can extend to trigrams, 4-grams, 5-grams
- In general this is an insufficient model of language
 - because language has **long-distance dependencies**:

“The computer which I had just put into the machine room on the fifth floor crashed.”
- But we can often get away with N-gram models



Google N-Gram Release, August 2006

AUG

3

All Our N-gram are Belong to You

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word [n-gram models](#) for a variety of R&D projects,

...

That's why we decided to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.



Google N-Gram Release

- serve as the incoming 92
- serve as the incubator 99
- serve as the independent 794
- serve as the index 223
- serve as the indication 72
- serve as the indicator 120
- serve as the indicators 45
- serve as the indispensable 111
- serve as the indispensable 40
- serve as the individual 234

<http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html>



Google Book N-grams

- <http://ngrams.googlelabs.com/>

AVOID SHOWING NEAR-DUPLICATE NEWS ARTICLES

Top Stories



Congress returns to work Saturday as lawmakers press to keep shutdown short-lived

Washington Post · 1h ago

Senate rejects funding bill, partial shutdown begins

The Hill · 7h ago

Government Shuts Down as Trump Feuds With Democrats

U.S. News & World Report · 47m ago

Federal government shuts down after Senate talks fail

Featured · NBCNews.com · 9h ago

Shutdown? It Could Be Forgotten in a Trumpian Flash

Opinion · New York Times · 8h ago

[View full coverage](#) →



MORE ABOUT

Donald Trump

Republican Party

Government
shutdown in the
United States

Democratic Party

AVOID SHOWING NEAR-DUPLICATE NEWS ARTICLES



CNN

See realtime coverage

Syria: Obama authorizes boots on ground to fight ISIS

CNN - 2 hours ago



Washington (CNN) The United States is set to deploy troops on the ground in Syria for the first time to advise and assist rebel forces combating ISIS, the White House said Friday.

[Obama to send small Special Operations force to Syria](#) Washington Post

[Dems Decry Obama Decision to Deploy Special Ops to Syria](#) ABC News

From Syria: Kurds, Syrian rebels to extend anti-ISIS military campaign in Raqqa
ARA News

Live Updating: [Watch Live: White House Announces US Troops to Go to Syria](#) TIME

Related

[Syria »](#)

[United States of America »](#)

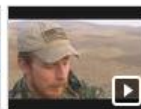
[Islamic State of Iraq and the Levant »](#)



USA TODAY



CNN



CNN



NBCNews....



Washingto...



Reuters



ABC News



USA TODAY



Washing



New York Times

See realtime coverage

Meg Whitman Seeks Reinvention for HP as It Prepares for Split

New York Times - 5 hours ago



PALO ALTO, Calif. - When Meg Whitman, Hewlett-Packard's chief executive, was preparing to cleave her company in two, she feared reliving a shoe disaster from early in her career.

[Bye-bye HP, it's the end of an era](#) Fortune

[As HP splits, the two companies came up with a plan to share custody of the ...](#) Business Insider

Related

[Hewlett-Packard »](#)



Bloomberg



Washingto...



Huffington ...



Voice of A...



Fortune



Business In...



NewsFacto...



FierceCIO



CTV News

HP, a Silicon Valley icon, is ready for its break-up



FILE - In this Aug. 21, 2014, file photo, Meg Whitman, CEO of Hewlett-Packard, is interviewed on the floor of the New York Stock Exchange in New York. Hewlett-Packard, one of the nation's most storied tech companies will split in ... [more >](#)

By BRANDON BAILEY - Associated Press - Friday, October 30, 2015

SAN FRANCISCO (AP) - One of the nation's most storied tech companies will split in two this weekend, another casualty of seismic shifts in the way people use technology - and big-company sluggishness in responding.

By BRANDON BAILEY - Associated Press - Friday, October 30, 2015

HP, a Silicon Valley icon, is ready for its breakup



Hewlett-Packard, one of the nation's most storied tech companies will split in two this weekend of Oct. 31, 2015, another casualty of seismic shifts in the way people use technology and big-company sluggishness in responding. (ECKEHARD SCHULZ, AP)

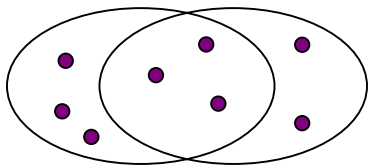
By **Brandon Bailey**
AP

OCTOBER 30, 2015, 10:57 AM | SAN FRANCISCO

One of the nation's most storied tech companies will split in two this weekend, another casualty of seismic shifts in the way people use technology — and big-company sluggishness in responding.

DISTANCE MEASURES

- **Goal: Find near-neighbors in high-dim. space**
 - We formally define “near neighbors” as points that are a “small distance” apart
- For each application, we first need to define what “**distance**” means
- **Today: Jaccard distance/similarity**
 - The **Jaccard similarity** of two **sets** is the size of their intersection divided by the size of their union:
$$\text{sim}(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$$
 - **Jaccard distance:** $d(C_1, C_2) = 1 - |C_1 \cap C_2| / |C_1 \cup C_2|$



3 in intersection

8 in union

Jaccard similarity = $3/8$

Jaccard distance = $5/8$

DOCUMENTS AS HIGH-DIM DATA

- Step 1: *N-Gramming*: Convert documents to sets
- Simple approaches:
 - Document = set of words appearing in document
 - Document = set of “important” words
 - Don’t work well for this application. Why?
- Need to account for ordering of words!
- A different way: **N-Grams!**

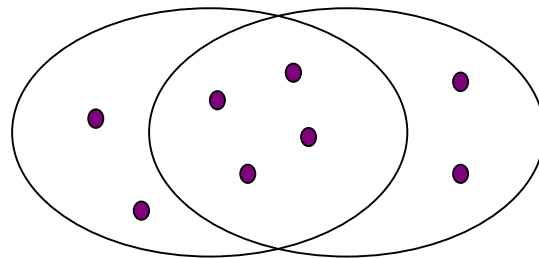
DEFINE: SHINGLES

- An n -gram (or k -shingle) for a document is a sequence of n tokens that appears in the doc
 - Tokens can be **characters**, **words** or something else, depending on the application
 - Assume tokens = characters for examples
- **Example:** $n=2$; document $\mathbf{D}_1 = \text{abcab}$
Set of 2-grams: $\mathbf{S}(\mathbf{D}_1) = \{\text{ab}, \text{bc}, \text{ca}\}$
 - **Option:** n -grams as a bag (multiset), count ab twice: $\mathbf{S}'(\mathbf{D}_1) = \{\text{ab}, \text{bc}, \text{ca}, \text{ab}\}$

SIMILARITY METRIC FOR N-GRAMS

- Document D_1 is a set of its n -grams $C_1 = S(D_1)$
- Equivalently, each document is a 0/1 vector in the space of n -grams
 - Each unique shingle is a dimension
 - Vectors are very sparse
- A natural similarity measure is the **Jaccard similarity**:

$$\text{sim}(D_1, D_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$$



WORKING ASSUMPTION

- Documents that have lots of n -grams in common have similar text, even if the text appears in different order
- **Caveat:** You must pick n large enough, or most documents will have most n -grams
 - $n = 5$ is OK for short documents
 - $n = 10$ is better for long documents



THE UNIVERSITY OF CHICAGO
GRAHAM SCHOOL
CONTINUING LIBERAL AND PROFESSIONAL STUDIES

N-Grams in Python



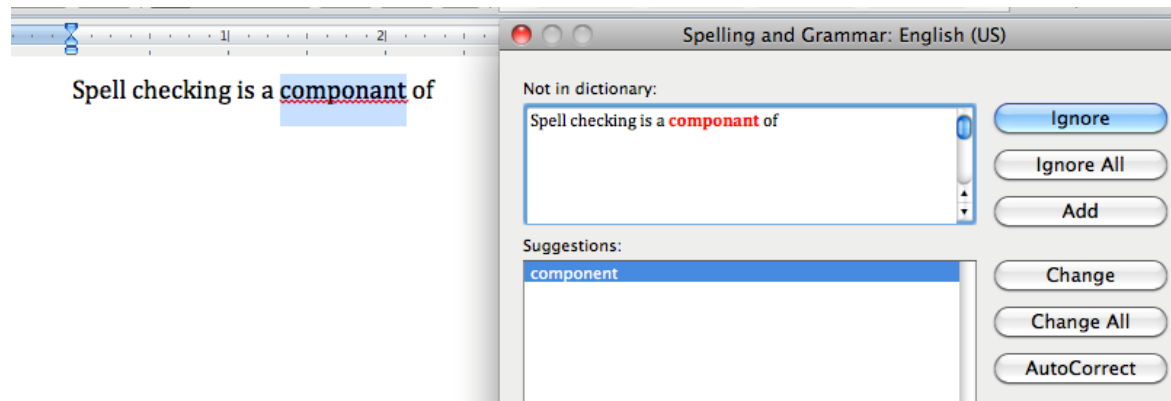
Spelling Correction and the Noisy Channel

The Spelling Correction Task



Applications for spelling correction

Word processing



Phones



Web search



Showing results for natural language processing
 Search instead for natural language processing



Spelling Tasks

- Spelling Error Detection
- Spelling Error Correction:
 - Autocorrect
 - hte → the
 - Suggest a correction
 - Suggestion lists



Types of spelling errors

- Non-word Errors
 - *graffe* → *giraffe*
- Real-word Errors
 - Typographical errors
 - *three* → *there*
 - Cognitive Errors (homophones)
 - *piece* → *peace*,
 - *too* → *two*



Rates of spelling errors

26%: Web queries [Wang et al. 2003](#)

13%: Typing, no delete/ backspace: [Whitelaw et al.](#)
[English&German](#)

7%: Words corrected retyping on phone-sized organizer

2%: Words uncorrected on organizer [Soukoreff & MacKenzie 2003](#)

1-2%: Retyping: [Kane and Wobbrock 2007](#), [Gruden et al. 1983](#)



Non-word spelling errors

- Non-word spelling error detection:
 - Any word not in a ***dictionary*** is an error
 - The larger the dictionary the better
- Non-word spelling error correction:
 - Generate ***candidates***: real words that are similar to error
 - Choose the one which is best:
 - Shortest weighted edit distance
 - Highest noisy channel probability



Real word spelling errors

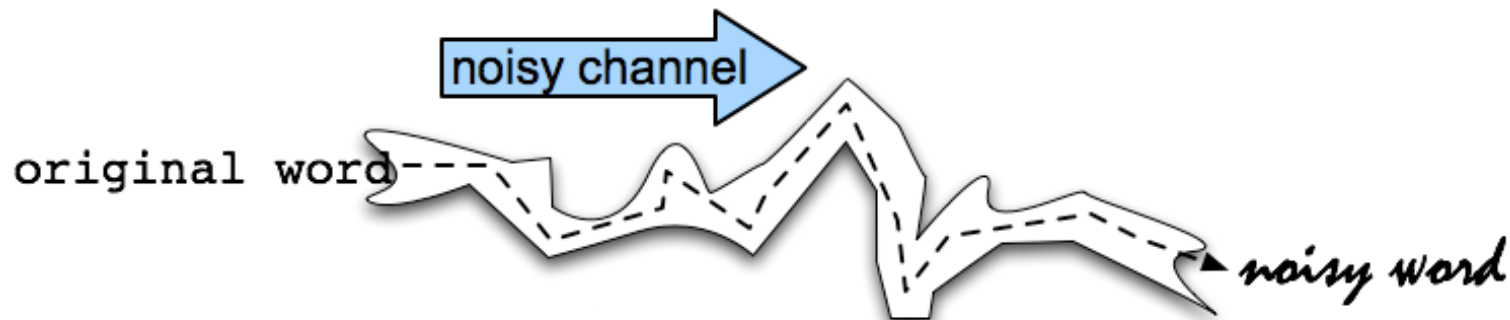
- For each word w , generate candidate set:
 - Find candidate words with similar ***pronunciations***
 - Find candidate words with similar ***spelling***
 - Include w in candidate set
- Choose best candidate
 - Noisy Channel
 - Classifier

Spelling Correction and the Noisy Channel

The Noisy Channel Model of Spelling



Noisy Channel Intuition





Noisy Channel

- We see an observation x of a misspelled word
- Find the correct word w

$$\begin{aligned}\hat{w} &= \operatorname{argmax}_{w \in V} P(w | x) \\ &= \operatorname{argmax}_{w \in V} \frac{P(x | w)P(w)}{P(x)} \\ &= \operatorname{argmax}_{w \in V} P(x | w)P(w)\end{aligned}$$



History: Noisy channel for spelling proposed around 1990

- **IBM**

- Mays, Eric, Fred J. Damerau and Robert L. Mercer. 1991. Context based spelling correction. *Information Processing and Management*, 23(5), 517–522

- **AT&T Bell Labs**

- Kernighan, Mark D., Kenneth W. Church, and William A. Gale. 1990. A spelling correction program based on a noisy channel model. *Proceedings of COLING 1990*, 205-210



Non-word spelling error example

acress



Candidate generation

- Words with similar spelling
 - Small edit distance to error
- Words with similar pronunciation
 - Small edit distance of pronunciation to error



Damerau-Levenshtein edit distance

- Minimal edit distance between two strings, where edits are:
 - Insertion
 - Deletion
 - Substitution
 - Transposition of two adjacent letters



Words within 1 of across

Error	Candidate Correction	Correct Letter	Error Letter	Type
acress	actress	t	-	deletion
acress	cress	-	a	insertion
acress	caress	ca	ac	transposition
acress	access	c	r	substitution
acress	across	o	e	substitution
acress	acres	-	s	insertion



Candidate generation

- 80% of errors are within edit distance 1
- Almost all errors within edit distance 2
- Also allow insertion of **space** or **hyphen**
 - `thisidea` → `this idea`
 - `inlaw` → `in-law`



Language Model

- Use any of the language modeling algorithms we've learned
- Unigram, bigram, trigram
- Web-scale spelling correction
 - Stupid backoff



Unigram Prior probability

Counts from 404,253,213 words in Corpus of Contemporary English (COCA)

word	Frequency of word	P(word)
actress	9,321	.0000230573
cress	220	.0000005442
caress	686	.0000016969
access	37,038	.0000916207
across	120,844	.0002989314
acres	12,874	.0000318463



Channel model probability

- **Error model probability, Edit probability**
- *Kernighan, Church, Gale 1990*
- *Misspelled word $x = x_1, x_2, x_3 \dots x_m$*
- *Correct word $w = w_1, w_2, w_3, \dots, w_n$*
- $P(x|w)$ = probability of the edit
 - (deletion/insertion/substitution/transposition)



Computing error probability: confusion matrix

```
del[x,y]:      count(xy typed as x)
ins[x,y]:      count(x typed as xy)
sub[x,y]:      count(x typed as y)
trans[x,y]:    count(xy typed as yx)
```

Insertion and deletion conditioned on previous character



Confusion matrix for spelling errors

sub[X, Y] = Substitution of X (incorrect) for Y (correct)

X	Y (correct)																									
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0



Generating the confusion matrix

- Peter Norvig's list of errors
- Peter Norvig's list of counts of single-edit errors



Channel model

Kernighan, Church, Gale 1990

$$P(x|w) = \begin{cases} \frac{\text{del}[w_{i-1}, w_i]}{\text{count}[w_{i-1} w_i]}, & \text{if deletion} \\ \frac{\text{ins}[w_{i-1}, x_i]}{\text{count}[w_{i-1}]}, & \text{if insertion} \\ \frac{\text{sub}[x_i, w_i]}{\text{count}[w_i]}, & \text{if substitution} \\ \frac{\text{trans}[w_i, w_{i+1}]}{\text{count}[w_i w_{i+1}]}, & \text{if transposition} \end{cases}$$



Channel model for across

Candidate Correction	Correct Letter	Error Letter	$x w$	$P(x word)$
actress	t	-	c ct	.000117
cress	-	a	a #	.00000144
caress	ca	ac	ac ca	.00000164
access	c	r	r c	.000000209
across	o	e	e o	.0000093
acres	-	s	es e	.0000321
acres	-	s	ss s	.0000342



Noisy channel probability for across

Candidate Correction	Correct Letter	Error Letter	$x w$	$P(x word)$	$P(word)$	$10^9 * P(x w)P(w)$
actress	t	-	c ct	.000117	.0000231	2.7
cress	-	a	a #	.00000144	.000000544	.00078
caress	ca	ac	ac ca	.00000164	.00000170	.0028
access	c	r	r c	.000000209	.0000916	.019
across	o	e	e o	.00000093	.000299	2.8
66 acres	-	s	es e	.0000321	.0000318	1.0



Noisy channel probability for across

Candidate Correction	Correct Letter	Error Letter	$x w$	$P(x word)$	$P(word)$	$10^9 * P(x w)P(w)$
actress	t	-	c ct	.000117	.0000231	2.7
cress	-	a	a #	.00000144	.000000544	.00078
caress	ca	ac	ac ca	.00000164	.00000170	.0028
access	c	r	r c	.000000209	.0000916	.019
across	o	e	e o	.0000093	.000299	2.8
67 acres	-	s	es e	.0000321	.0000318	1.0



Using a bigram language model

- "a stellar and versatile **acress** whose combination of sass and glamour..."
- Counts from the Corpus of Contemporary American English with add-1 smoothing
- $P(\text{actress}|\text{versatile}) = .000021$ $P(\text{whose}|\text{actress}) = .0010$
- $P(\text{across}|\text{versatile}) = .000021$ $P(\text{whose}|\text{across}) = .000006$
- $P(\text{"versatile actress whose"}) = .000021 * .0010 = 210 \times 10^{-10}$
- $P(\text{"versatile across whose"}) = .000021 * .000006 = 1 \times 10^{-10}$



Using a bigram language model

- "a stellar and versatile **acress** whose combination of sass and glamour..."
- Counts from the Corpus of Contemporary American English with add-1 smoothing
- $P(\text{actress}|\text{versatile}) = .000021$ $P(\text{whose}|\text{actress}) = .0010$
- $P(\text{across}|\text{versatile}) = .000021$ $P(\text{whose}|\text{across}) = .000006$
- $P(\text{"versatile actress whose"}) = .000021 * .0010 = 210 \times 10^{-10}$
- $P(\text{"versatile across whose"}) = .000021 * .000006 = 1 \times 10^{-10}$



Evaluation

- Some spelling error test sets
 - [Wikipedia's list of common English misspelling](#)
 - [Aspell filtered version of that list](#)
 - [Birkbeck spelling error corpus](#)
 - [Peter Norvig's list of errors \(includes Wikipedia and Birkbeck, for training or testing\)](#)

Spelling Correction and the Noisy Channel

Real-Word Spelling Correction



Real-word spelling errors

- ...leaving in about fifteen ***minuets*** to go to her house.
- The design ***an*** construction of the system...
- Can they ***lave*** him my messages?
- The study was conducted mainly ***be*** John Black.
- 25-40% of spelling errors are real words [Kukich 1992](#)



Solving real-world spelling errors

- For each word in sentence
 - Generate *candidate set*
 - the word itself
 - all single-letter edits that are English words
 - words that are homophones
- Choose best candidates
 - Noisy channel model
 - Task-specific classifier

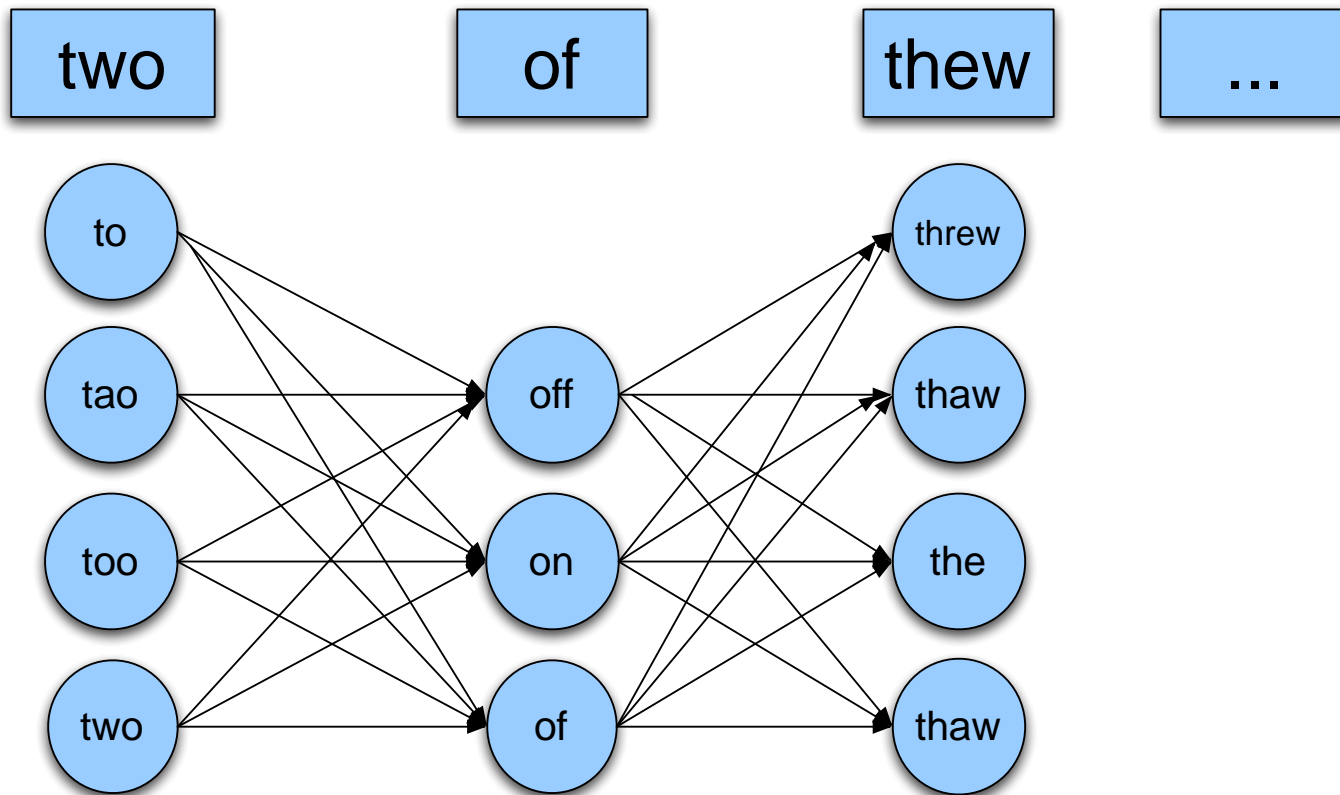


Noisy channel for real-word spell correction

- Given a sentence $w_1, w_2, w_3, \dots, w_n$
- Generate a set of candidates for each word w_i
 - $\text{Candidate}(w_1) = \{w_1, w'_1, w''_1, w'''_1, \dots\}$
 - $\text{Candidate}(w_2) = \{w_2, w'_2, w''_2, w'''_2, \dots\}$
 - $\text{Candidate}(w_n) = \{w_n, w'_n, w''_n, w'''_n, \dots\}$
- Choose the sequence W that maximizes $P(W)$

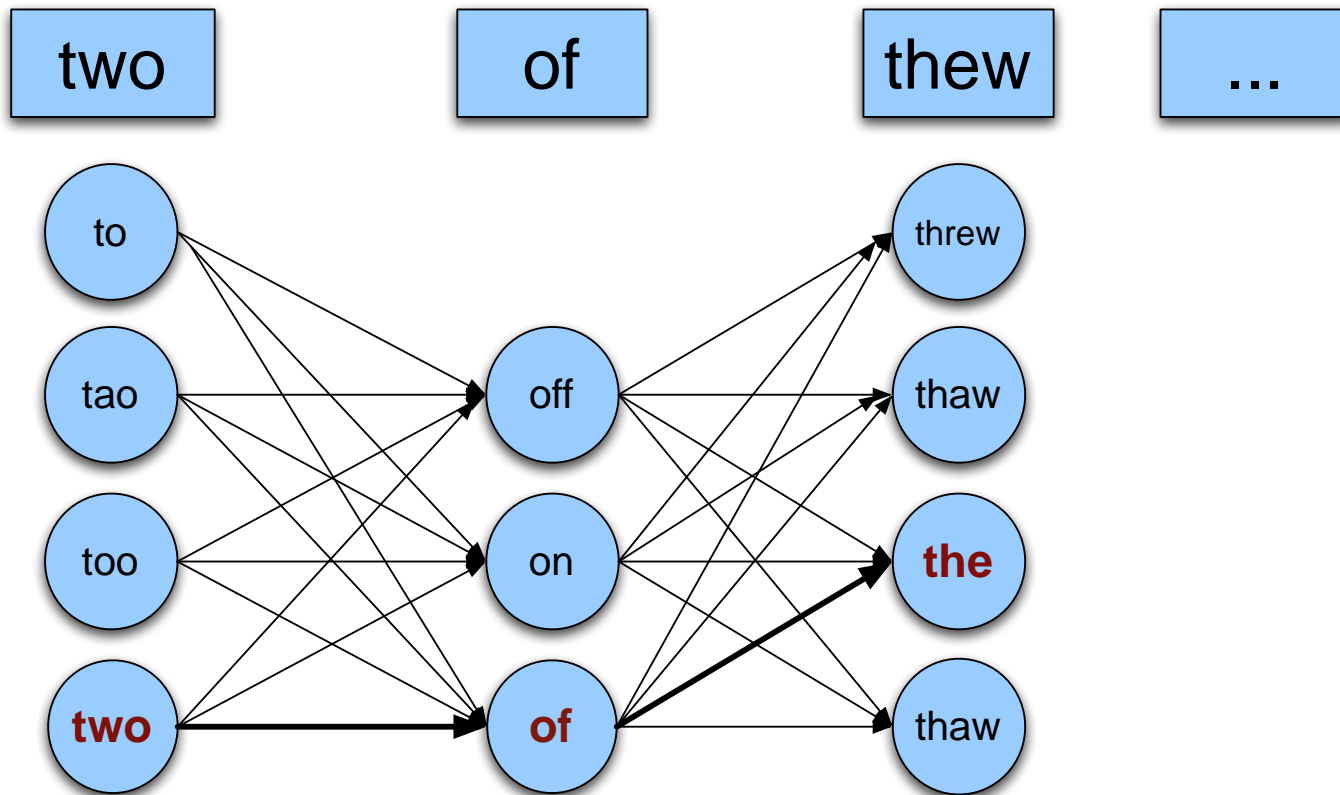


Noisy channel for real-word spell correction





Noisy channel for real-word spell correction





Simplification: One error per sentence

- Out of all possible sentences with one word replaced
 - w_1, w''_2, w_3, w_4 two off thew
 - w_1, w_2, w'_3, w_4 two of the
 - w'''_1, w_2, w_3, w_4 too of thew
 - ...
- Choose the sequence W that maximizes $P(W)$



Where to get the probabilities

- Language model
 - Unigram
 - Bigram
 - Etc
- Channel model
 - Same as for non-word spelling correction
 - Plus need probability for no error, $P(w|w)$



Probability of no error

- What is the channel probability for a correctly typed word?
- $P(\text{"the"} | \text{"the"})$
- Obviously this depends on the application
 - .90 (1 error in 10 words)
 - .95 (1 error in 20 words)
 - .99 (1 error in 100 words)
 - .995 (1 error in 200 words)



Peter Norvig's "thew" example

x	w	x w	$P(x w)$	$P(w)$	$10^9 P(x w)P(w)$
thew	the	ew e	0.000007	0.02	144
thew	thew		0.95	0.000000009	90
thew	thaw	e a	0.001	0.00000007	0.7
thew	threw	h hr	0.000008	0.0000004	0.03
thew	thwe	ew we	0.000003	0.000000004	0.0001

Spelling Correction and the Noisy Channel

State-of-the-art Systems



HCI issues in spelling

- If very confident in correction
 - Autocorrect
- Less confident
 - Give the best correction
- Less confident
 - Give a correction list
- Unconfident
 - Just flag as an error

HCI = Human Computer Interaction



State of the art noisy channel

- We never just multiply the prior and the error model
- Independence assumptions \rightarrow probabilities not commensurate
- Instead: Weigh them

$$\hat{w} = \operatorname{argmax}_{w \in V} P(x | w) P(w)^\lambda$$

- Learn λ from a development test set



Phonetic error model

- Metaphone, used in GNU aspell
 - Convert misspelling to metaphone pronunciation
 - “Drop duplicate adjacent letters, except for C.”
 - “If the word begins with 'KN', 'GN', 'PN', 'AE', 'WR', drop the first letter.”
 - “Drop 'B' if after 'M' and if it is at the end of the word”
 - ...
 - Find words whose pronunciation is 1-2 edit distance from misspelling’s
 - Score result list
 - Weighted edit distance of candidate to misspelling
 - Edit distance of candidate pronunciation to misspelling pronunciation



Improvements to channel model

- Allow richer edits (Brill and Moore 2000)
 - $\text{ent} \rightarrow \text{ant}$
 - $\text{ph} \rightarrow \text{f}$
 - $\text{le} \rightarrow \text{al}$
- Incorporate pronunciation into channel (Toutanova and Moore 2002)



Channel model

- Factors that could influence $p(\text{misspelling} | \text{word})$
 - The source letter
 - The target letter
 - Surrounding letters
 - The position in the word
 - Nearby keys on the keyboard
 - Homology on the keyboard
 - Pronunciations
 - Likely morpheme transformations



Nearby keys





Classifier-based methods for real-word spelling correction

- Instead of just channel model and language model
- Use many features in a classifier (next lecture).
- Build a classifier for a specific pair like:

whether/weather

- “cloudy” within +/- 10 words
- ____ to VERB
- ____ or not



THE UNIVERSITY OF CHICAGO
GRAHAM SCHOOL
CONTINUING LIBERAL AND PROFESSIONAL STUDIES

Thank You!

