



Monsoon 2022

Spelling Correction

- Independent Word Spelling Correction

Dr. Rajendra Prasath

Indian Institute of Information Technology Sri City, Chittoor



DDth September 2022 (rajendra.2power3.com)

> Topics to be covered

➤ Recap:

- Phrase Queries
- Proximity Search
- Permuterm Index
- Bi-gram Indexes

➤ Spelling Correction

- Independent Word Spelling Correction
 - Spelling Detection
 - Specific tasks in Spelling Correction
 - Spelling Suggestion

➤ More topics to come up ... Stay tuned ...!!

Recap: Information Retrieval

- **Information Retrieval (IR)** is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).
- During these days, we frequently think first of web search, but there are many other cases:
 - E-mail search
 - Searching your laptop
 - Corporate knowledge bases
 - Legal information retrieval
 - Images / Patent / Transportation Related Searches
 - and so on . . .

Phrase queries

- Want to be able to answer queries such as “stanford university” – as a phrase
- Thus the sentence “**I went to university at Stanford**” is not a match.
 - The concept of phrase queries has proven easily understood by users; one of the few “advanced search” ideas that works
 - Many more queries are *implicit phrase queries*
- For this, it no longer suffices to store only
 - `<term : docs>` entries

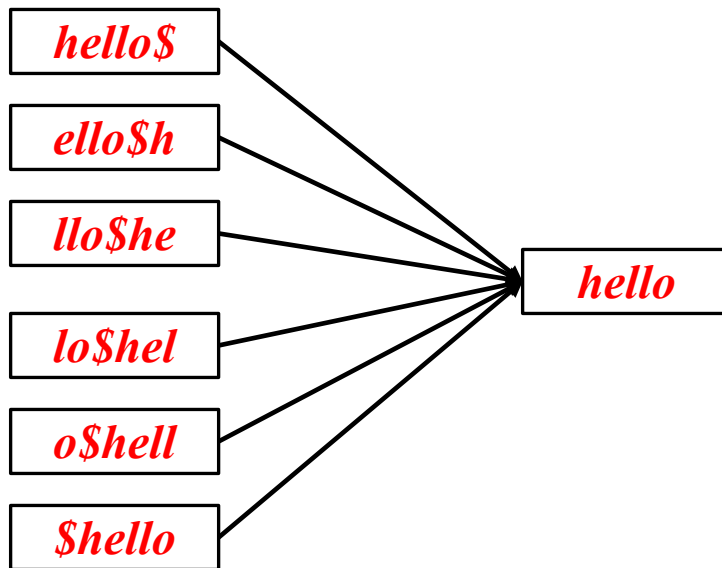
Recap: Wild-card queries: *

- **mon***: find all docs containing any word beginning with “mon”.
- Easy with binary tree (or B-tree) dictionary: retrieve all words in range: **mon** \leq **w** $<$ **moo**
- ***mon**: find words ending in “mon”: harder
 - Maintain an additional B-tree for terms backwards.Can retrieve all words in range: **nom** \leq **w** $<$ **non**.

From this, how can we enumerate all terms meeting the wild-card query **pro*cent** ?

Recap: Permuterm index

- Add a \$ to the end of each term
- Rotate the resulting term and index them in a B-tree
- For term hello, index under:
 - **hello\$, ello\$h, llo\$he, lo\$hel, o\$hell, \$hello**where \$ is a special symbol.



**Empirically, dictionary
quadruples in size**

Spelling Correction

Types of spelling errors

- **Non-word Errors**
 - graffe → giraffe
- **Real-word Errors**
 - Typographical errors
 - three → there
 - Cognitive Errors (homophones)
 - piece → peace,
 - too → two
 - your → you're
- Non-word correction → context insensitive
- Real-word correction almost needs to be context sensitive

Non-word spelling errors

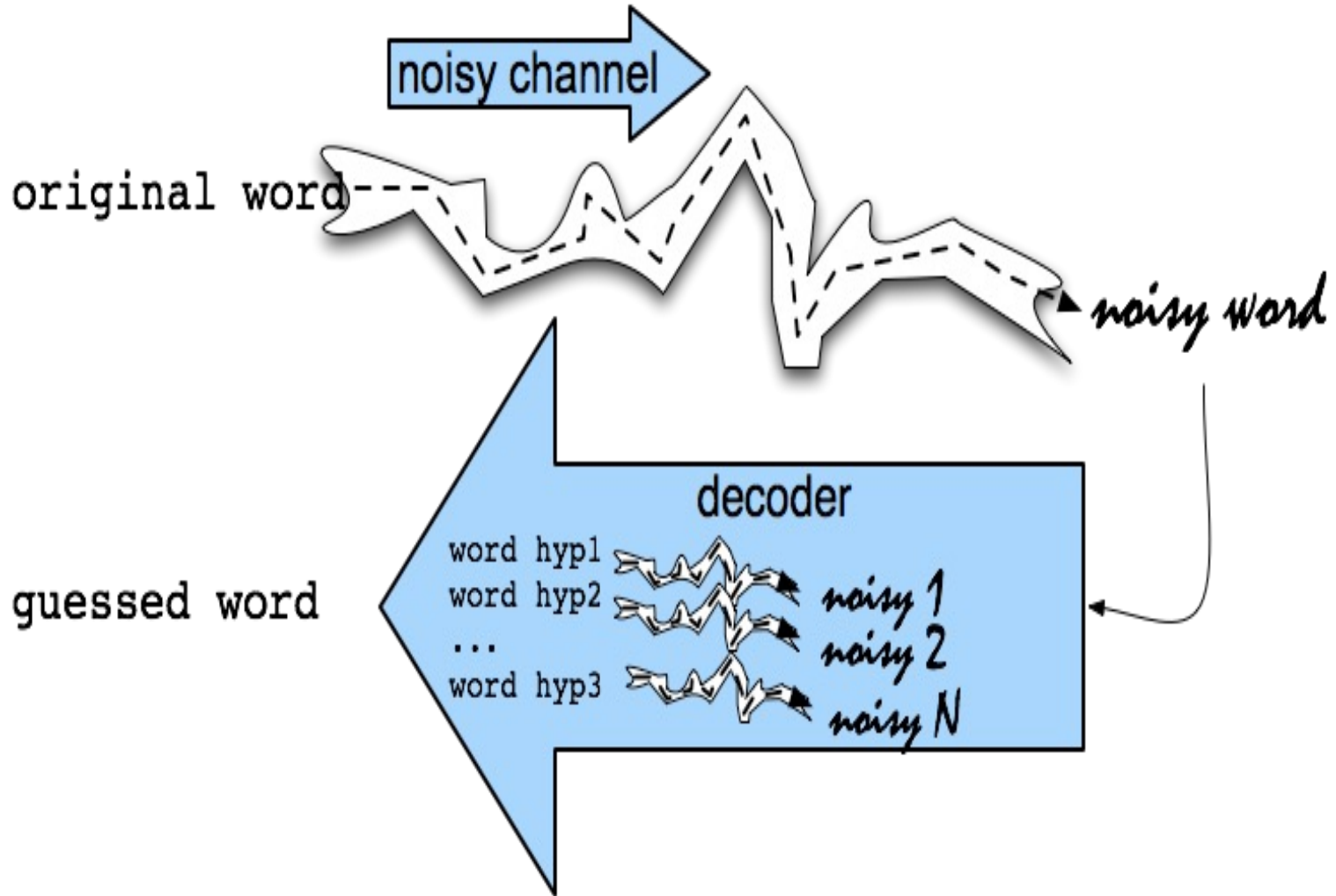
- **Non-word spelling error detection:**
 - Any word not in a dictionary is an error
 - The larger the dictionary the better ... up to a point
 - (The Web is full of mis-spellings, so the Web isn't necessarily a great dictionary ...)
- **Non-word spelling error correction:**
 - Generate candidates: real words that are similar to error
 - Choose the one which is best:
 - Shortest weighted edit distance
 - Highest noisy channel probability

Independent Word Spelling Correction

The Noisy Channel Model of Spelling



Noisy Channel Intuition



Noisy Channel - Bayes' Rule

- We see an observation x of a misspelled word
- Find the correct word \hat{w}

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

← Bayes

↑
Noisy channel model

← Prior

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 - An Example

acress

Candidate Generation

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

Candidate Testing:

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 / deletion

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
- Can also allow merging words
 - data base → database
 - For short texts like a query, can just regard whole string as one item from which to produce edits

How do you generate the candidates?

- Run through dictionary, check edit distance with each word
- Generate all words within edit distance $\leq k$ (e.g., $k = 1$ or 2) and then intersect them with dictionary
- Use a character k -gram index and find dictionary words that share “most” k -grams with word (e.g., by Jaccard coefficient)
- Compute them fast with a Levenshtein finite state transducer
- Have a precomputed map of words to possible corrections

A Paradigm ...

- We want the best spell corrections
- Instead of finding the very best, we
 - Find a subset of pretty good corrections
 - (say, edit distance at most 2)
 - Find the best amongst them
- These may not be the actual best
- This is a recurring paradigm in IR including finding the best docs for a query, best answers, best ads ...
 - Find a good candidate set
 - Find the top K amongst them and return them as the best

With candidates Generated: Now back to Bayes' Rule

- We see an observation x of a misspelled word
- Find the correct word \hat{w}

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

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

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

What's $P(w)$?

Language Model

- Take a big supply of words (your document collection with T tokens); let $C(w) = \#$ occurrences of w

$$P(w) = \frac{C(w)}{T}$$

- In other applications – you can take the supply to be typed queries (suitably filtered) – when a static dictionary is inadequate

Unigram Prior probability

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

word	Frequency of word	$P(w)$
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(y typed as x)
`trans[x,y]` : count(xy typed as yx)

Insertion and deletion conditioned on previous character

Confusion matrix for substitution

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	5	0	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	8	3	0	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	9	0	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

Nearby keys



Generating Confusion Matrix

- [Peter Norvig's list of errors](#)
- [Peter Norvig's list of counts of single-edit errors](#)
- All Peter Norvig's ngrams data links:
<http://norvig.com/ngrams/>

Summary

In this class, we focused on:

(a) Recap: Positional Indexes

- i. Positional Index Size
- ii. Wild card Queries
- iii. Permuterm index

(b) Spelling Correction

- i. Types of Spelling Correction
- ii. Noisy Channel modelling for Spell Correction
- iii. Spelling Suggestions

Acknowledgements

Thanks to ALL RESEARCHERS:

- Modern Information Retrieval Baeza-Yates and Ribeiro-Neto, Addison Wesley, 1999.
- **Introduction to Information Retrieval Manning, Raghavan and Schutze, Cambridge University Press, 2008.**
- Search Engines Information Retrieval in Practice W. Bruce Croft, D. Metzler, T. Strohman, Pearson, 2009.
- Information Retrieval Implementing and Evaluating Search Engines Stefan Büttcher, Charles L. A. Clarke and Gordon V. Cormack, MIT Press, 2010.
- Many Authors who contributed to SIGIR / WWW / KDD / ECIR / CIKM / WSDM and other top tier conferences
- **Prof. Mandar Mitra, Indian Statistical Institute, Kolkatata (<https://www.isical.ac.in/~mandar/>)**

Assistance

- You may post your questions to me at any time
- You may meet me in person on available time or with an appointment
- You may ask for one-to-one meeting

Best Approach

- You may leave me an email any time
(email is the best way to reach me faster)



Questions It's Your Time

