

Spelling Correction and Edit Distance

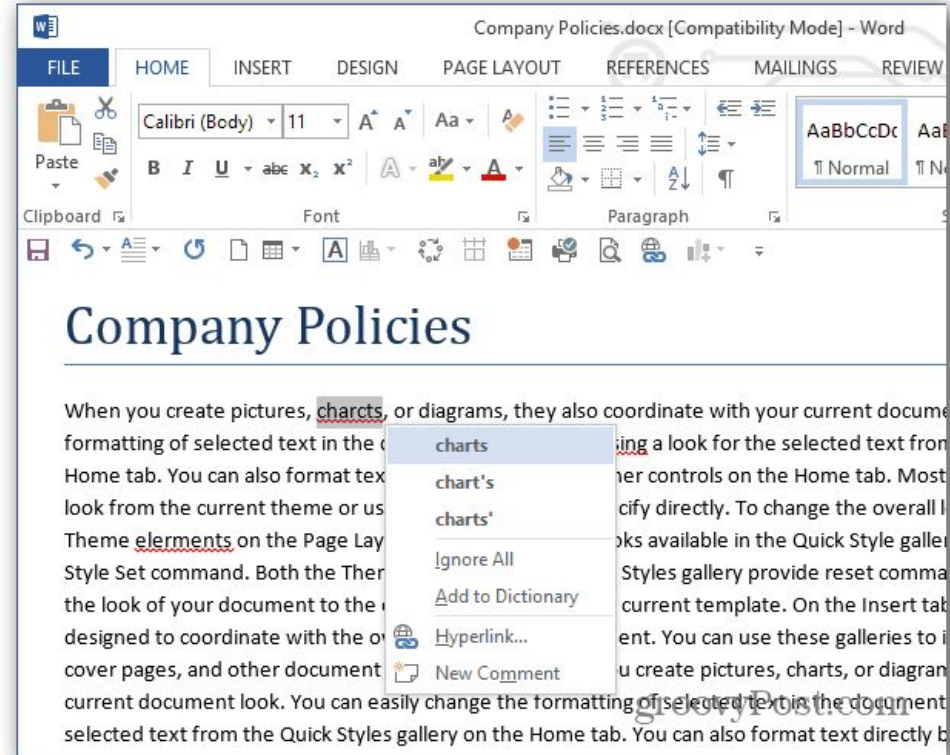
Topic-5

Agenda

- Spelling correction
- Edit distance algorithm
- Variants of edit distance
- Noisy channel model for spelling correction (probabilistic model)

Spelling Correction

- Words not in dictionary--- error
- Suggest some closest words
- Considering words in isolation, without considering the context
- Closest words are picked
- **How to define the closeness?**



Edit Distance

- Given two strings, what is the minimum number of edit operations to be performed to reach from one word to another
- Edit operations
 - Insert
 - Delete
 - Substitute
- Edit distance is number of edit operations
- Example: Intention and Execution

I N T E * N T I O N

* E X E C U T I O N

- Here 1 delete, 3 substitutions and 1 insert, so cost is 5

Edit Distance

- We can assume each operation have equal cost of 1 (levenshtein)
- We can also consider substitution have cost 2, other operations have cost 1
- **Initial State:** the word we are transforming
- **Operations:** edit operations
- **Goal state:** the final word
- **Path cost:** minimum number of edit distance
- How to find the Minimum edit distance?
 - Do all possible operations
 - Find the cost
 - Not efficient

Edit Distance Algorithm

For two string X of length n and
Y of length m

- Define $D(i,j)$ = minimum edit distance between $X[1..i]$ and $Y[1..j]$
- Edit distance of X and Y will be $D(n,m)$
- Dynamic programming approach

Dynamic Programming Approach

Initialization

$$D(i, 0) = i$$

$$D(0, j) = j$$

Recurrence Relation:

For each $i = 1 \dots M$

For each $j = 1 \dots N$

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \end{cases}$$

$$\begin{cases} 2, & \text{if } X(i) \neq Y(j) \\ 0, & \text{if } X(i) = Y(j) \end{cases}$$

Termination:

$D(N, M)$ is distance

Example

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

N	9	8	9	10	11	12	11	10	9	8
O	8	7	8	9	10	11	10	9	8	9
I	7	6	7	8	9	10	9	8	9	10
T	6	5	6	7	8	9	8	9	10	11
N	5	4	5	6	7	8	9	10	11	10
E	4	3	4	5	6	7	8	9	10	9
T	3	4	5	6	7	8	7	8	9	8
N	2	3	4	5	6	7	8	7	8	7
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

How to find the alignment?

Keep a pointer to trace back.

Every cell, add a pointer to indicate where the value came from

n	9	↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙←↓ 12	↓ 11	↓ 10	↓ 9	↙ 8	
o	8	↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↓ 10	↓ 9	↙ 8	← 9	
i	7	↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	↙ 8	← 9	← 10	
t	6	↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙ 8	← 9	← 10	←↓ 11	
n	5	↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙↓ 10	
e	4	↙ 3	← 4	↙← 5	← 6	← 7	←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	
t	3	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙ 7	←↓ 8	↙←↓ 9	↓ 8	
n	2	↙←↓ 3	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↓ 7	↙←↓ 8	↙ 7	
i	1	↙←↓ 2	↙←↓ 3	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙ 6	← 7	← 8	
#	0	1	2	3	4	5	6	7	8	9	
	#	e	x	e	c	u	t	i	o	n	

Base conditions:

$$D(i, 0) = i$$

$$D(0, j) = j$$

Termination:

$D(N, M)$ is distance

Recurrence Relation:

For each $i = 1 \dots M$

For each $j = 1 \dots N$

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 & \text{deletion} \\ D(i, j-1) + 1 & \text{insertion} \\ D(i-1, j-1) + \begin{cases} 2; & \text{if } X(i) \neq Y(j) \\ 0; & \text{if } X(i) = Y(j) \end{cases} & \text{substitution} \end{cases}$$
$$\text{ptr}(i, j) = \begin{cases} \text{LEFT} & \text{insertion} \\ \text{DOWN} & \text{deletion} \\ \text{DIAG} & \text{substitution} \end{cases}$$

Weighted Edit Distance

- Some errors are more likely than some other errors
- The common sources are:
 - Vowels (they may sound similar)
 - Keys are very close in keyboard
- Apply a low cost to common errors
- Use a variable costs according to the statistics

Some letters are more likely to be mistyped

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

Initialization:

$$D(0,0) = 0$$

$$D(i,0) = D(i-1,0) + \text{del}[x(i)]; \quad 1 < i \leq N$$

$$D(0,j) = D(0,j-1) + \text{ins}[y(j)]; \quad 1 < j \leq M$$

Recurrence Relation:

$$D(i,j) = \min \begin{cases} D(i-1,j) + \text{del}[x(i)] \\ D(i,j-1) + \text{ins}[y(j)] \\ D(i-1,j-1) + \text{sub}[x(i),y(j)] \end{cases}$$

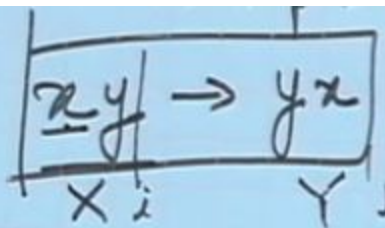
Termination:

$D(N,M)$ is distance

Transpose

- Another common edit operation is transpose
- $xy \rightarrow yx$
- Example
 - Teh \rightarrow the
- How to incorporate transpose operation in our dynamic programming algorithm?
- Just think!

$$D(i, j) = \min \left(\begin{array}{l} D(i-2, j-2) + 1 \quad \text{if } X[i-1] = Y[j] \\ \text{and } X[i] = Y[j-1] \end{array} \right)$$

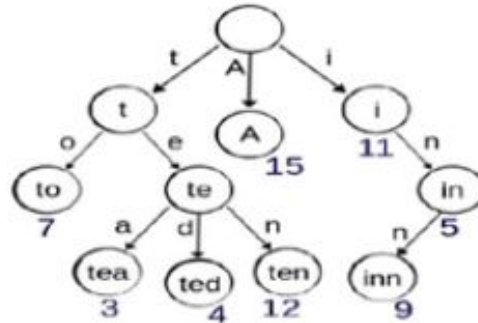

 → 1 for transposition
 (transpose)

How to list possible candidates for spelling correction?

Naïve Method

Compute edit distance from the query term to each dictionary term – an exhaustive search

Can be made efficient if we do it over a trie structure



How to list possible candidates for spelling correction?

- Generate all possible terms with an edit distance ≤ 2 (deletion + transpose + substitution + insertion) from the query term and search them in the dictionary.
- For a word of length 9, alphabet of size 36, this will lead to 114,324 terms to search for
- For Chinese alphabet size is 70,000 (Unicode Han Characters)

Edit Distance- Variant

Symmetric Delete Spelling Correction

- Generate terms with an edit distance ≤ 2 (deletes) from each dictionary term (offline)
- Generate terms with an edit distance ≤ 2 (deletes) from the input terms and search in dictionary

Number of deletes within edit distance ≤ 2 for a word of length 9 will be 45

A further check is required to remove the false positives

Spelling Correction

- Non-word error: word not in dictionary
 - Behalf → Behalf
 - Generate candidates with minimum edit distance
 - Choose the one with shortest distance
- Read word error
 - Peace → piease
 - There → three
 - Use probabilistic models
 - **Noisy channel model for spelling correction**

Noisy Channel Model

The intuition of the noisy channel model is to treat the misspelled noisy channel word as if a correctly spelled word had been “distorted” by being passed through a noisy communication channel.

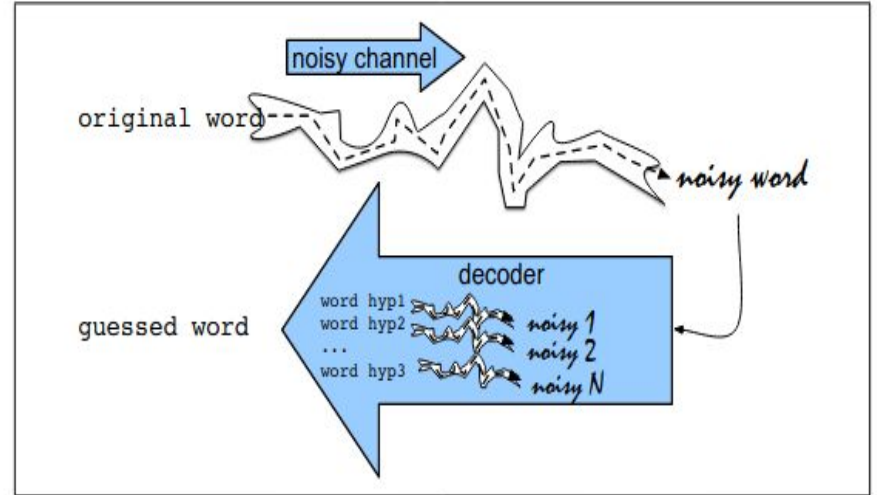


Figure B.1 In the noisy channel model, we imagine that the surface form we see is actually a “distorted” form of an original word passed through a noisy channel. The decoder passes each hypothesis through a model of this channel and picks the word that best matches the surface noisy word.

Noisy Channel Model

This channel introduces “noise” in the form of substitutions or other changes to the letters, making it hard to recognize the “true” word. Our goal, then, is to build a model of the channel.

Given this model, we then find the true word by passing every word of the language through our model of the noisy channel and seeing which one comes the closest to the misspelled word.

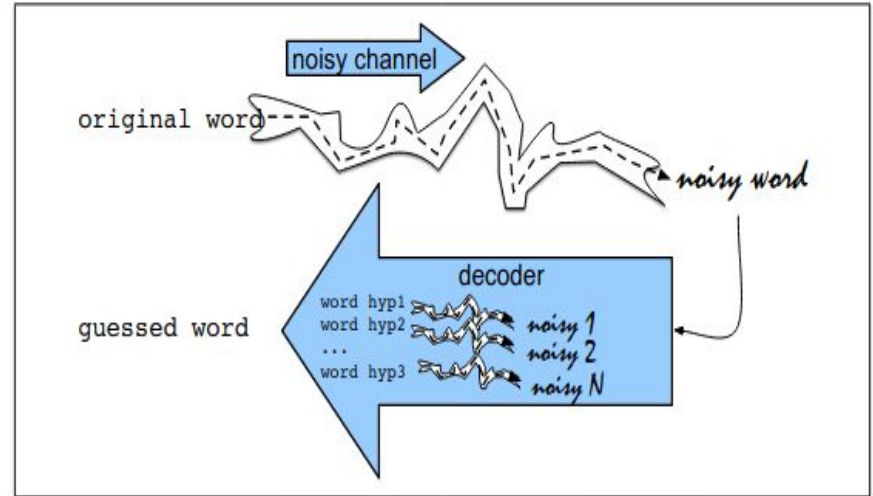


Figure B.1 In the noisy channel model, we imagine that the surface form we see is actually a “distorted” form of an original word passed through a noisy channel. The decoder passes each hypothesis through a model of this channel and picks the word that best matches the surface noisy word.

Noisy Channel Model for Spelling correction

We see an observation x of the misspelled word

Find the correct word w

$$\begin{aligned}\hat{w} &= \arg \max_{w \in V} P(w|x) \\ &= \arg \max_{w \in V} \frac{P(x|w)P(w)}{P(x)} \\ &= \arg \max_{w \in V} P(x|w)P(w)\end{aligned}$$

Example: non-word error

. . . was called a “stellar and versatile **acress** whose combination of sass and glamour has defined her. . .”.

Example: words within 1 edit distance of across

The version of edit distance with transposition is called **Damerau-Levenshtein edit distance**.

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
acress	acres	-	s	insertion

Prior probability $P(w)$

The prior probability of each correction $P(w)$ is the language model probability of the word w in context, which can be computed using any language model

For example: the language model from the 404,253,213 words in the Corpus of Contemporary English (COCA).

Likelihood Probability $P(w|x)$

A perfect model of the probability that a word will be mistyped would condition on all sorts of factors: who the typist was, whether the typist was left-handed or right-handed, and so on

A simple model might estimate, for example, $p(\text{acress}|\text{across})$ just using the number of times that the letter e was substituted for the letter o in some large corpus of errors.

Confusion matrix

- $\text{del}[x,y]$: count (xy typed as x)
- $\text{ins}[x,y]$: count (x typed as xy)
- $\text{sub}[x,y]$: count (x typed as y)
- $\text{trans}[x,y]$: count(xy typed as yx)

Channel Model

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

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
acres	-	s	es e	.0000321	.0000318	1.0
acres	-	s	ss s	.0000342	.0000318	1.0

implementation of the noisy channel model chooses **across** as the best correction, and **actress** as the second most likely word.

Unfortunately, the algorithm was wrong here; the writer's intention becomes clear from the context: . . . was called a “stellar and versatile **actress** whose combination of sass and glamour has defined her. . .”.

We need to use context (bigram language models)

- “ ... versatile actress whose ...”
- Counts from the Corpus of Contemporary American English with add-1 smoothing
- $P(\text{actress}|\text{versatile}) = 0.000021$, $P(\text{actress}|\text{actress}) = 0.000021$
- $P(\text{whose}|\text{actress}) = 0.0010$, $P(\text{whose}|\text{actress}) = 0.000006$
- $P(\text{“versatile actress whose”}) = 0.000021 * 0.0010 = 210 \times 10^{-10}$

Real word spelling errors

- This used to belong to thew queen. They are leaving in about fifteen minuets to go to her house.
- The design an construction of the system will take more than a year.
- Can they lave him my messages?
- The study was conducted mainly be John Black.

Noisy channel model

Given a sentence $X = w_1, w_2, w_3, \dots, w_n$

- Candidate (w_1) = $\{w_1, w'_1, w''_1, w'''_1, \dots\}$
- Candidate (w_2) = $\{w_2, w'_2, w''_2, w'''_2, \dots\}$
- Candidate (w_3) = $\{w_3, w'_3, w''_3, w'''_3, \dots\}$

With edit distance 1, a common choice the candidate set for the real word error thew (a rare word meaning 'muscular strength') might be $C(\text{thew}) = \{\text{the, thaw, threw, them, thwe}\}$.

We then make the simplifying assumption that every sentence has only one error.

Thus the set of candidate sentences $C(X)$ for a sentence $X = \text{Only two of thew apples}$ would be:

only two of thew apples
oily two of thew apples
only too of thew apples
only to of thew apples
only tao of the apples
only two on thew apples
only two off thew apples
only two of the apples
only two of threw apples
only two of thew applies
only two of thew dapples
...

Each sentence is scored by the noisy channel:

$$\hat{W} = \operatorname{argmax}_{W \in C(X)} P(X|W) P(W)$$

Compute $P(W)$

We need language models

Unigram

Bigram

Tri-gram

Generally, n-gram models

Next Topic: **n-gram language models**