Noisy Channel

Noisy Channel

We see an observation x of the misspelled word

Find the correct word w

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

Noisy Channel

We see an observation x of the misspelled word

Find the correct word w

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

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

Noisy Channel

We see an observation x of the misspelled word

Find the correct word w

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

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

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

Creating candidate set

- Words with similar spelling
- Words with similar pronunciation

Some statistics

- 80% of errors are within edit distance 1
- Almost all errors within edit distance 2

Allow deletion of space or hyphen

- thisidea → this idea
- inlaw → in-law

Words within edit distance 1 of *acress*

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

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)

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 are conditioned on previous character

Channel model

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

Channel model for *acress*

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 *acress*

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

- Here we have maximum probability 2.8 therefore correct candidate is **across**
- Without considering context, this method is very good.
- But suppose if we have probabilities values very near like 2.7, 2.71…, than without using context we cannot choose which candidate is correct candidate.

- **Example with context:**
- Consider the sentence **"…….. Versatile acress whose……"**
- So here we have two candidates '**across'** and '**actress'**
- P(actress|versatile) = 0.000021
- P(across|versatile) = 0.000021
- P(whose|actress) = 0.0010, P(whose|across) = 0.000006
- P("versatile actress whose") = 0.000021 * 0.0010 = 210 * 10^{-10}
- P("versatile across whose") = 0.000021 * 0.000006 = 1 * 10^{-10}
- Here **actress** is more probable.

Real-word spelling errors

- The study was conducted mainly **be** John Black
- The design **an** construction of the system ...

25-40% of spelling errors are real words

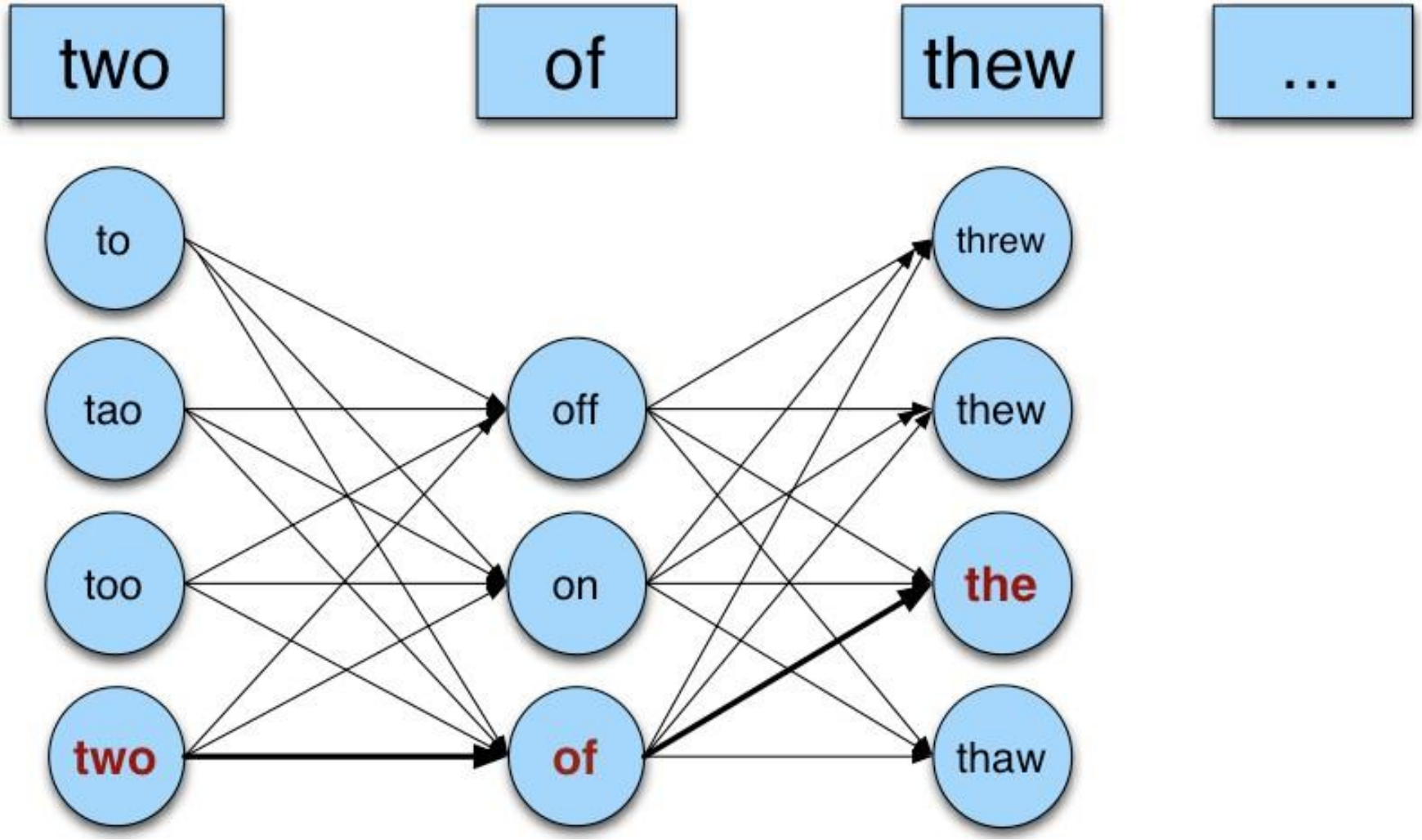
Noisy channel for real-word spell correction

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

- Candidate $(w_1) = \{w_1, w^{j}_1, w^{jj}_1, w^{jjj}_1, \ldots\}$
- Candidate $(w_2) = \{w_2, w^{j}_2, w^{jj}_2, w^{jjj}_2, \ldots\}$
- Candidate $(w_3) = \{w_3, w^{j}_3, w^{jj}_3, w^{jjj}_3, \ldots\}$

Choose the sequence W that maximizes $P(W|X)$

Noisy channel for real-world spell correction



Simplification: One error per sentence

Choose among all possible sentences with one word replaced

two of thew

- w_1, w^{jj}_2, w_3 two **off** thew
- $w_1,\ w_2,\ w^{j}_3$ two of **the**
- w^{jjj}_1, w_2, w_3 **too** of thew

- So here we assume there is error in one word.
- So for assume for w1 there are three possibilities i.e. w1-> w1’ w1’’ & w1(word itself)
- Similarly for w2-> w2’ w2’’ & w2 and w3-> w3’ w3’’ & w3
- So if we assume there is error in one word only the possible combination can be

  w1’ w2 w3
  w1’’ w2 w3
  w1 w2’ w3
  w1 w2’’ w3
  w1 w2 w3’
  w1 w2 w3’’
  w1 w2w3

- So total 6+1 = 7 possibilities.
- Otherwise three words and three combination for each so total 3^3 = 27.

Simplification: One error per sentence

Choose among all possible sentences with one word replaced

two of thew

- w_1, w^{jj}_2, w_3 two **off** thew w_1,
- w_2, w^{j}_3 two of **the** w^{jjj}_1, w_2, w_3
- **too** of thew

Choose the sequence W that maximizes $P(W|X)$

Getting the probability values

Noisy Channel

$$\hat{W} = \underset{W \in S}{\operatorname{argmax}} P(W|X)$$

where X is the observed sentence and S is the set of all the possible sequences from the candidate set

Getting the probability values

Noisy Channel

$$\hat{W} = \underset{W \in S}{\operatorname{argmax}} P(W|X)$$

where X is the observed sentence and S is the set of all the possible sequences from the candidate set

$$= \underset{W \in S}{\operatorname{argmax}} P(X|W)P(W)$$

Getting the probability values

Noisy Channel

$$\hat{W} = \underset{W \in S}{\operatorname{argmax}} P(W|X)$$

where X is the observed sentence and S is the set of all the possible sequences from the candidate set

$$= \underset{W \in S}{\operatorname{argmax}} P(X|W)P(W)$$

P(X|W)

- Same as for non-word spelling correction
- Also require proabability for no error $P(w|w)$

Probability of no error

What is the probability for a correctly typed word? P(“the”|“the”)

It may depend on the source text under consideration

- 1 error in 10 words → 0.9
- 1 error in 100 words → 0.99

Computing $P(W)$

Use Language Model

- Unigram
- Bigram
- ...

Spell corrector

Peter Norvig

http://www.norvig.com/spell-correct.html

1. Spell correction algorithm based on edit distance:
   http://norvig.com/spell-correct.html

2. Python spell correction libraries PyEnchant based on the enchant library (http://pythonhosted.org/pyenchant/), autocorrect, which is available at https://github.com/phatpiglet/autocorrect/

3. DeepSpell: Based on RNN, LSTM and word embedding
   https://github.com/MajorTal/DeepSpell