Spelling Correction and the Noisy Channel

Spelling Tasks

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

Types of Spelling Errors

- Non-word Errors: **Non-word spelling correction** is the detection and correction of spelling errors that result in non-words
  - *graffe* → *giraffe*

- Real-word Errors: **Real word spelling correction** is the task of detecting and correcting spelling errors even if they accidentally result in an actual word.
  - Typographical errors
    - *three* → *there*
  - Cognitive Errors (homophones)
    - *piece* → *peace,*
    - *too* → *two*

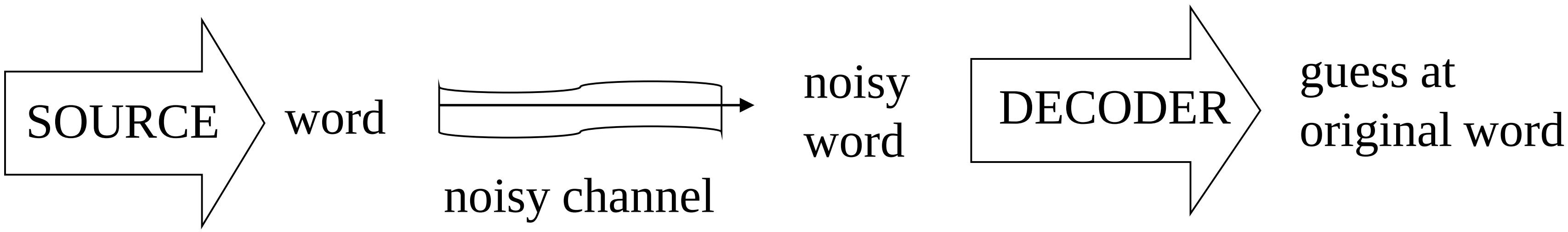
Non-word Spelling Errors

- 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

Noisy Channel Model of Spelling



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

Applying Bayes to a Noisy Channel

- In applying probability theory to a noisy channel, what we are looking for is the most probable *source* given the observed *signal*. We can denote this:

$$\textbf{mostprobable-source} = \textbf{argmax}_{\textbf{Source}}\ \textbf{P(Source|Signal)}$$

- Unfortunately, we don't usually know how to compute this.
  - We cannot directly know : what is the probability of a source given an observed signal?
  - We will apply Bayes' rule

Applying Bayes to a Noisy Channel

- From Bayes rule, we know that:

$$P(Source \mid Signal) = \frac{P(Signal \mid Source) P(Source)}{P(Signal)}$$

- So, we will have:

$$\arg\max_{Source} \frac{P(Signal \mid Source) P(Source)}{P(Signal)}$$

- For each *Source*, *P(Signal)* will be same. So we will have:

$$\mathbf{argmax_{Source}\ P(Signal|Source)\ P(Source)}$$

Applying Bayes to a Noisy Channel to Spelling

- We have some word that has been misspelled and we want to know the real word.
- In this problem, the real word is the source and the misspelled word is the signal.
- We are trying to estimate the real word.
- Assume that

V	is the space of all the words we know
s	denotes the misspelling (signal)
ϖ	denotes the correct word (estimate)

- So, we will have the following equation:

$$\varpi = \operatorname{argmax}_{w \in V} P(s|w)\, P(w)$$

Noisy Channel to Spelling

- We can use a **candidate list C** instead vocubalary **V**

$$\varpi = \text{argmax}_{w \in V} P(s|w)\, P(w) \quad \Rightarrow \quad \varpi = \text{argmax}_{w \in C} P(s|w)\, P(w)$$

channel model **prior (a language model)**

function NOISY CHANNEL SPELLING(*word x*, *dict D*, lm, *editprob*) **returns** *correction*

  if $x \notin D$
    candidates, edits ← All strings at edit distance 1 from x that are $\in D$, and their edit
    for each c, e in candidates, edits
      channel ← editprob(e)
      prior ← lm(x)
      $score[c] = \log channel + \log prior$
    return $\text{argmax}_c\ score[c]$

Getting Numbers

- We need a corpus to compute: **P(w)** and **P(s|w)**
- Computing P(w) : a unigram language model
  - We will count how often the word w occurs in the corpus.
  - So, P(w) = C(w)/N where C(w) is the number of w occurs in the corpus, and N is the total number of words in the corpus.
  - What happens if P(w) is zero.
    - We need a ***smoothing*** technique (getting rid of zeroes).
    - A smoothing technique: P(w) = (C(w)+0.5) / (N+0.5*VN) where VN is the number of words in V (our dictionary).

- Computing P(s|w)
  - It is fruitless to collect statistics about the misspellings of individual words for a given dictionary. We will likely never get enough data.
  - We need a way to compute P(s|w) without using direct information.
  - We can use spelling error pattern statistics to compute P(s|w).

Spelling Error Patterns

- There are four patterns:

Deletion	-- ther for there
Insertion	-- ther for the
Substitution	-- noq for now
Transposition	-- hte for the

- For each pattern we need a **confusion matrix**.
  - **del[x,y]** contains the number of times in the training set that characters xy in the correct word were typed as x.
  - **ins[x,y]** contains the number of times in the training set that character x in the correct word were typed as xy.
  - **sub[x,y]** contains the number of times that x was typed as y.
  - **trans[x,y]** contains the number of times that xy was typed as yx.

Estimating P(s|w)
Noisy Channel Model for Spelling Correction

- Assuming a single spelling error, P(s|w) will be computed as follows.

$P(s|w) = del[w_{i-1},w_i] / count[w_{i-1}w_i]$ if deletion

$P(s|w) = ins[w_{i-1},s_i] / count[w_{i-1}]$ if insertion

$P(s|w) = sub[w_i,s_i] / count[w_i]$ if substitution

$P(s|w) = trans[w_i,w_{i+1}] / count[w_iw_{i+1}]$ if transposition

Words within 1 edit distance of misspelled word 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

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

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

Noisy 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

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

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(actress|versatile)=.000021   P(whose|actress) = .0010

P(across|versatile) =.000021   P(whose|across) = .000006

P(“versatile actress whose”) = .000021*.0010 = 210×10^{-10}

P(“versatile across whose”) = .000021*.000006 = 1×10^{-10}

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.

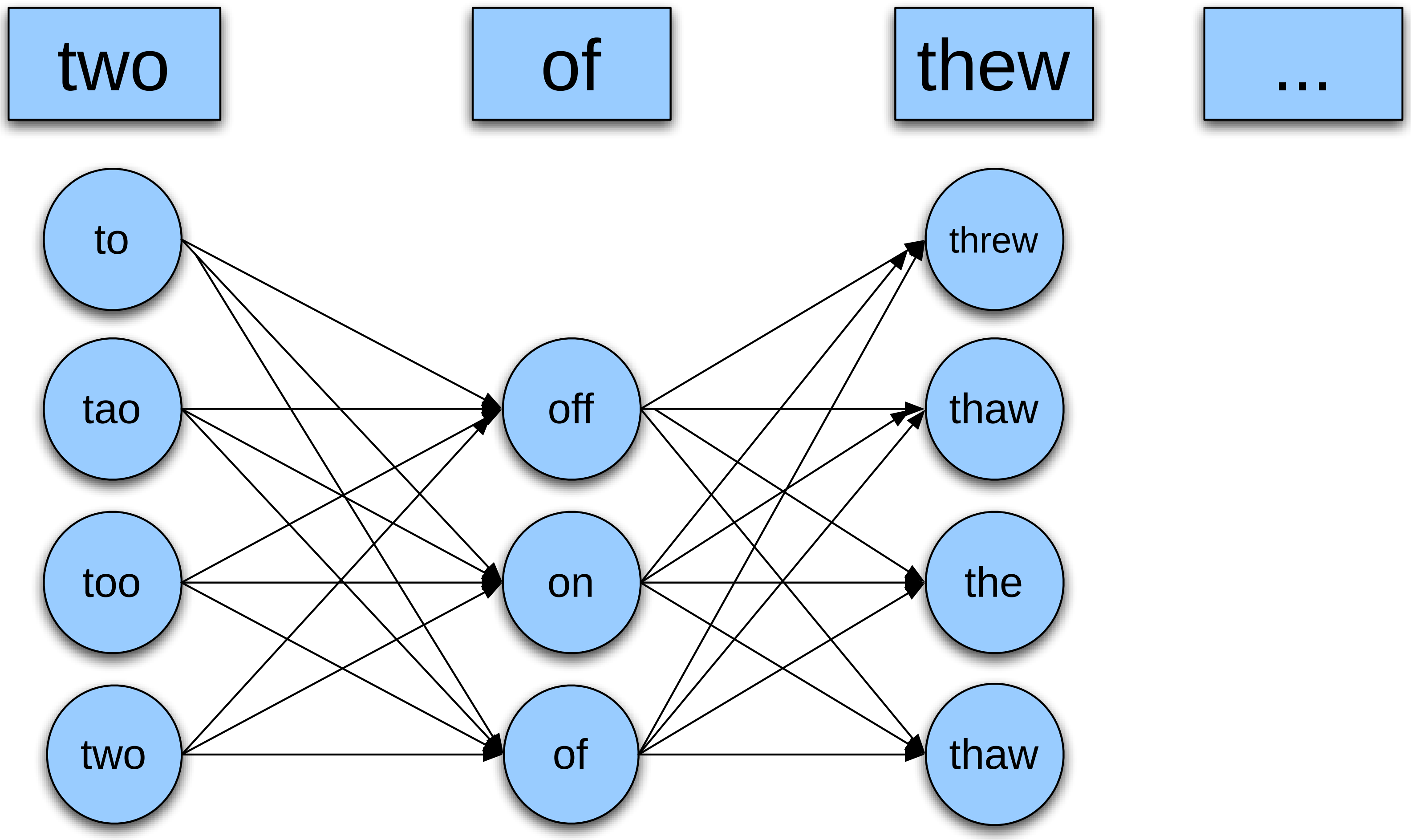
Solving Real-world Spelling Errors

- 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

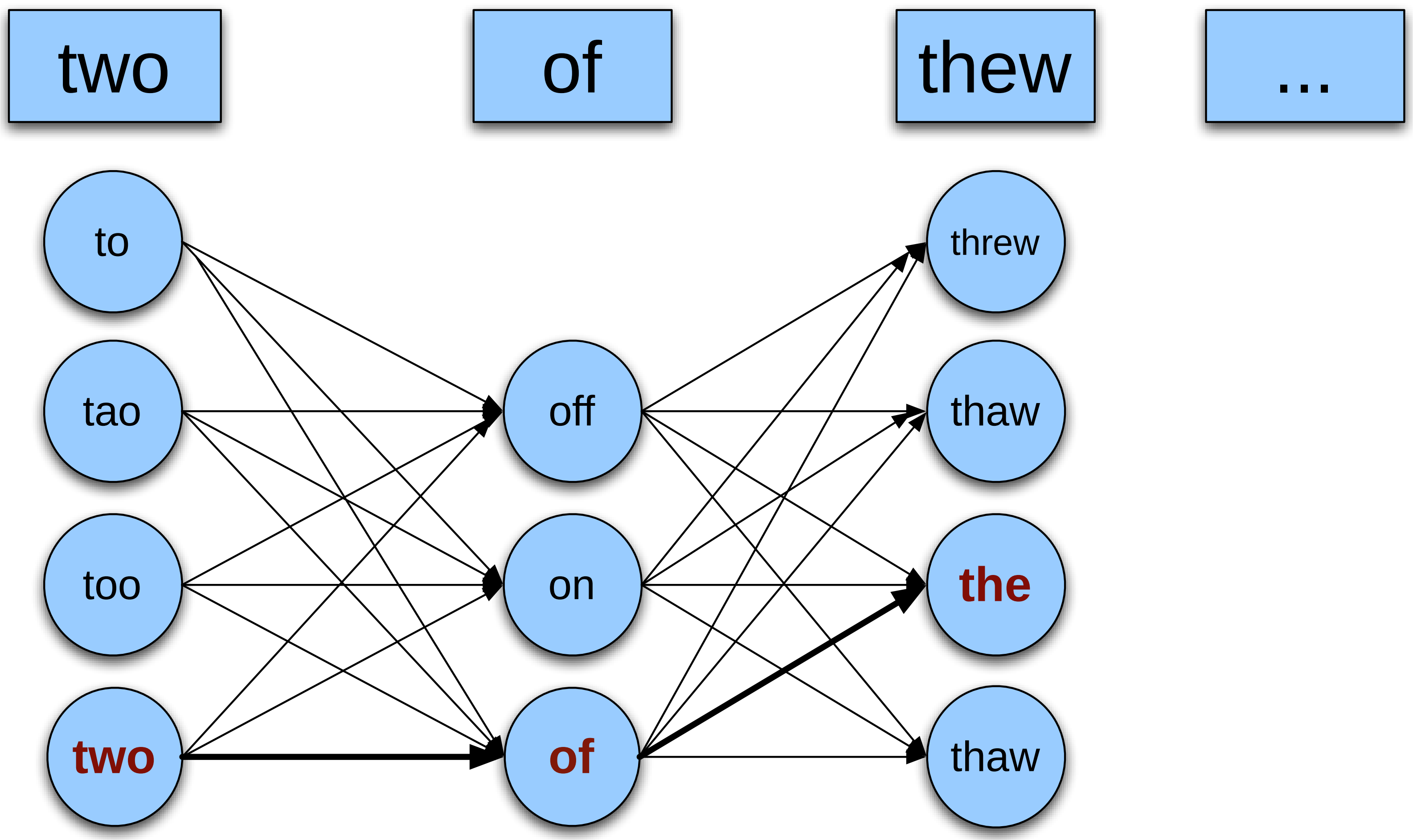
Noisy Channel for Real-word Spell Correction

- Given a sentence $w_1, w_2, w_3, \ldots, w_n$
- Generate a set of candidates for each word w_i
  - Candidate(w_1) = $\{w_1, w'_1, w''_1, w'''_1, \ldots\}$
  - Candidate(w_2) = $\{w_2, w'_2, w''_2, w'''_2, \ldots\}$
  - Candidate(w_n) = $\{w_n, w'_n, w''_n, w'''_n, \ldots\}$
- 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, $\mathbf{w''_2}$, w_3, w_4 two **off** thew
  - w_1, w_2, $\mathbf{w'_3}$, w_4 two of **the**
  - $\mathbf{w'''_1}$, w_2, w_3, w_4 **too** of thew
  - …
- Choose the sequence W that maximizes P(W)

Where to Get Probabilities

- Language model: Unigram, Bigram, …
- 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("the"|"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)