Spell checking.[1]

Victor Kitov

v.v.kitov@yandex.ru

[1] With materials used from "Speech and Language Processing", D. Jurafsky and J. H. Martin.

Constituents

- Frequency of spelling errors by humans:
  - 1-2% - retyping already printed text
  - 10-15% - web queries
- When spelling checker observes mistake it can
  - underline the word
  - automatically correct the word
- Cause of error:
  - typographical: pressed buttons wrong
    - e.g: there->three
  - cognitive error: didn't write properly the sound
    - e.g: dessert->desert, piece->peace

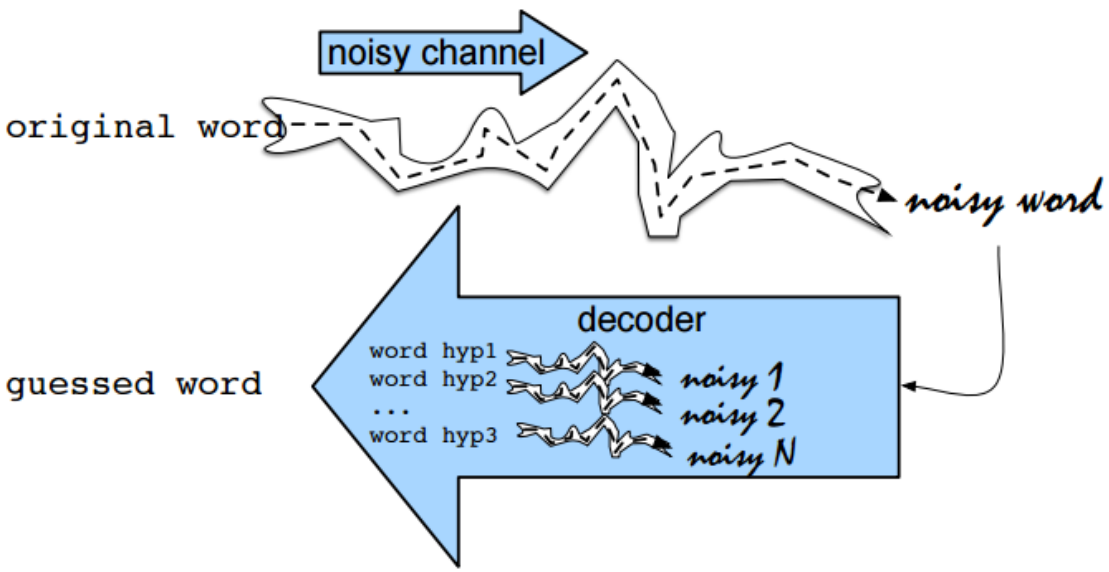
Word / non-word typos

- Spelling errors:
  - **non-word spelling correction:** typo is not a word
    - e.g: giraffe->graffe
  - **real word spelling correction:** typo is another existing word
    - e.g: there->three
- Non-word mistake can easily be detected - word is not in the dictionary!
  - should substitute it with word that is
    - close to typo
    - is a frequent word which matches context
- Real word mistake is harder to detect - any word can be an error!
  - should check possible close corrections of each word, even correct one
    - select a sequence of words that give high probability together

Noisy channel model

In noisy channel model we imagine that:

- original word passes through a «noisy channel» and possibly gets distorted to «noisy word»
- we consider a number of original word hypothesis, pass them through the noisy channel and see which hypothesis matches best the obsered noisy word.



Noisy channel model

- Define:
  - x - observed noisy word
  - w - orginal true word
  - V - vocabulary
- Model:

$$\begin{aligned}\widehat{w} &= \arg\max_{w\in V} p(w|x) = \arg\max_{w\in V} \frac{p(w,x)}{p(x)} \\ &= \arg\max_{w\in V} p(w)p(x|w)\end{aligned}$$

- Interpretation:
  - $p(w)$ language model: how likely is w in given context?
  - $p(x|w)$ - channel model: how likely could w be distorted to x?

Noisy chanel model in practice

- Considering all words is not practical
- Instead only words spelled similarly to x are considered in V
- usually words with edit distance 1 from x
- edit distance accounts for
  - insertion: x->xy
  - deletion: xy->y
  - substitution: x->y
  - **+transposition: xy->yx**
- called Damerau-Levenshtein edit distance

Noisy chanel example

- Example: «acress»
- Possible true words (with edit distance 1):

		Transformation			
Error	Correction	Correct Letter	Error Letter	Position (Letter #)	Type
acress	actress	t	—	2	deletion
acress	cress	—	a	0	insertion
acress	caress	ca	ac	0	transposition
acress	access	c	r	2	substitution
acress	across	o	e	3	substitution
acress	acres	—	s	5	insertion
acress	acres	—	s	4	insertion

Noisy chanel example

- Unigram word prior probabilities:

w	count(w)	p(w)
actress	9,321	.0000231
cress	220	.000000544
caress	686	.00000170
access	37,038	.0000916
across	120,844	.000299
acres	12,874	.0000318

- Channel probabilities $p(x|w)$ ideally should condition on
  - the writer
  - he is left-handed or right-handed
  - etc

Noisy chanel example

- Instead we condition only on the misspleled and true letters.
- We construct 4 confusion matrices:

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)

- Confusion matrices estimation:
  - from frequency of practical errors
    - http://www. dcs.bbk.ac.uk/~ROGER/corpora.html
    - http://norvig. com/ngrams/
  - using EM
    - start from confusion matrices initialized with constant
    - apply spelling corection
    - using typos&corrections reestimate matrices with frequencies
    - apply spelling corection
    - reestimate matrices
    - ...

Prior estimation

- $p(w)$ may be estimated unigram model
  - does not account context!
- bigram-trigram - more accurate
  - need more data
  - use e.g. google bigram statistics: https://catalog.ldc.upenn.edu/LDC2006T13
- we estimate not only the probability of the word itself, but the probability of the whole sentence part given that word:
  - Example: was called a "stellar and versatile **acress** whose combination of sass and glamour has defined her. . . ".
  - P("versatile actress whose")
    =P(actress|versatile)*P(whose|actress)
  - P("versatile across whose")
    =P(across|versatile)*P(whose|across)

Real word spelling errors

- Between 25% and 40% of spelling errors are valid English words
- Examples:
  - This used to belong to thew queen.
  - They are leaving in about fifteen minuets to go to her house.
- Algorithm:
  - consider sentence $x_1, x_2, ...x_N$
  - for each word x_i generate a set $C(x_i)$ of similar valid words
    - e.g. C(thew) = {the, thaw, threw, them, thwe}
  - to omit all possibilities, assume that sentence may contain <=1 mistake
    - e.g. only two of thew apples may be

Example

- Sentence: Only two of thew apples
- Expansion:

```
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
...
```

Example

- For the sentence X and a set of sentence expansions X we get the most likely true word sequence W:

$$\widehat{W} = \arg\max_{W \in C(X)} p(W)p(X|W)$$

- usually $p(W)$ is estimated using trigram model
- $p(X, W)$:
  - assume $\alpha = p(w|w)$ - probability to write word correctly
  - uniform model $p(x|w) = \begin{cases} \alpha & x = w \\ \frac{1-\alpha}{|C(x)|} & x \in C(x) \\ 0 & \text{otherwise} \end{cases}$
  - alternatively - use confusion matrices.

State of the art spell checkers

- Look through all words with 1 word at a time.
  - >1 errors are possible in a sentence
- Spell checkers are prone to overcorrecting
  - they overcorrect rare words!
  - use blacklist of words that are never corrected:
    - numbers, punctuation, single letters
- more careful correction: correct $x \to w$ only if

$$\ln p(w|x) - \ln p(x|x) > \theta, \text{ for some } \theta \geq 0$$

State of the art spell checkers

- may autocorrect or just flag the word and offer suggestions
  - use separate classifier to decide, based on other features
- use very large dictionary, because new words are always appearing
  - may look through web for possible words, but it contains mistakes
  - may use, that mistakes are more rare than correct spellings
  - need to separate probabilistic model to decide the correct spelling!

State of the art spell checkers

- Weighted account of language model and noisy channel models:

$$\widehat{w} = \arg\max_{w} p(x|w)p(w)^{\lambda}$$

  - λ is selected on the validation set
- may train separate classifiers on common mistakes:
  - among/between, peace/piece, affect/effect, weather/whether, ...

Accounting for prononciation

- Consider in $C(x)$ not words with small edit distance from x but words with prononciations, having small edit distance from prononciation of x
  - Example of word->prononciation conversion:
    - 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
- Distance between words-weighted edit distances between spellings and prononciations.