Text Auto Correction

AI Seminary Presentation by

Yigit Mertol Kayabasi

What is it?

Text replacement, replace-as-you-type or AutoCorrect is an automatic **data validation function.** Its principal purpose is as part of the spell checker to correct common spelling or typing errors, saving time for the user.*

Where is it used?

- “Texting” auto correctors in smartphones , tablets or any device/interface used for text messaging .
- Spellchecking for academic papers, publishings , thesis etc.
- Web searches

What is the main idea behind a text auto correction algorithm?

Given a word, we are trying to choose the most likely spelling correction for that word (the "correction" may be the original word itself).

4 important aspects to finding a correct candidate for “correction” :

- Selection Mechanism
- Candidate Model
- Language Model
- Error Model

A Probabilistic approach is used for the selection,succession and filtering of the candidate

We are trying to find the correction *c*, out of all possible candidate corrections, that maximizes the probability that *c* is the intended correction, given the original word *w*:

$\text{argmax}_{c \in \textit{candidates}} P(c|w)$

By Bayes' Theorem this is equivalent to:

$\text{argmax}_{c \in \textit{candidates}} P(c)\, P(w|c) / P(w)$

Since P(*w*) is the same for every possible candidate *c*, we can factor it out, giving:

$\text{argmax}_{c \in \textit{candidates}} P(c)\, P(w|c)$

Selection Mechanism
Argmax

The candidate with the highest combined probability is chosen with respect to:

$$\mathbf{argmax}_{c\ \in\ candidates}\ \mathrm{P}(c)\ \mathrm{P}(w|c)$$

Candidate Model C

- A list of terms to use as candidates.
- Rather than attempt to build a lexicon of words that are well-spelled, we instead take the most frequent tokens observed on the Web.
- We(Google) used a large (> 1 billion) sample of Web pages, tokenized them, and took the **most frequently occurring ten million tokens,**

  with very simple filters for **non-words** (too much punctuation, too short or long)**.**
- This term list is so large that it should contain most well-spelled words, but also a large number of nonwords or misspellings.

Language Model
P(c)

Simple example:

The probability that c appears as a word of English text. For example, occurrences of "the" make up about 7% of English text, so we should have P(the) = 0.07.

Language Model
P(c)

We can estimate the probability of a word, P(word), by counting the number of times each word appears in a text file of about a million words, big.txt .

P(word) = count(word) / Sum of counts over all words in the term list

Error Model
$P(w|c)$

A simple edit:

- A deletion (remove one letter)
- A transposition (swap two adjacent letters),
- A replacement (change one letter to another) or an insertion (add a letter)

Error Model
P(w|c)

In a simple model,

simple edit operator works as a branching method while searching for candidates within a certain “distance” w.r.t. The estimated amount of misspelling or typo in the word **w.**

P(w|c) is any probability function that is disproportional to the number of simple edits.(Example : 1 / number of edits)

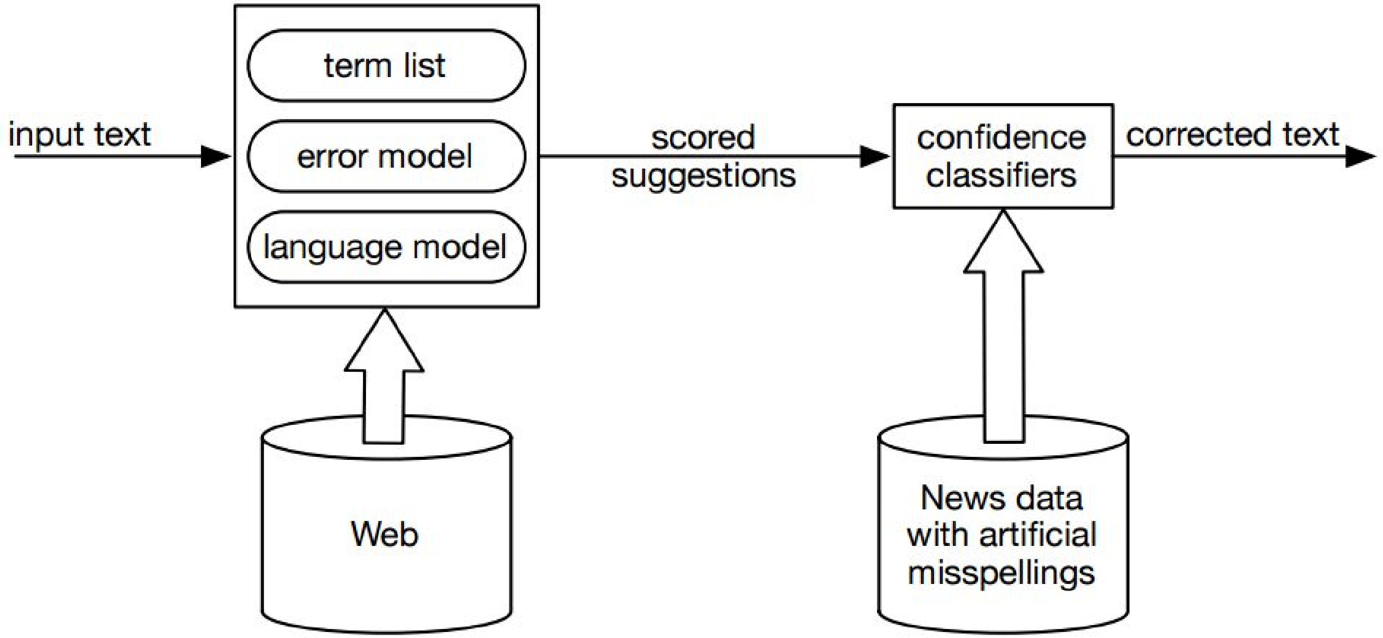
Error Model
P(w|c)

Brill and Moore estimate P(w|s) as follows:

$$P(w|s) \approx \max_{R,\, T \text{ s.t. } |T|=|R|} \prod_{i=1}^{|R|} P(T_i|R_i)$$

Where Ti is a partitioning of w , with empty substrings included as well.

Similarly Ri is a partitioning of s, with empty substrings allowed.


input text
term list
error model
language model
scored suggestions
confidence classifiers
corrected text
Web
News data with artificial misspellings

“To train the error model, we require triples of (intended word, observed word, count), which are described below. We use maximum likelihood estimates of P(Ti |Ri).

To build the error model, we require as training data a set of

(intended word, observed word, count) triples,

which is compiled from the **World Wide Web**. Essentially the triples are built by starting with the term list, and a process that automatically discovers, from that list, putative pairs of spelled and misspelled words, along with their counts.” *****

* (Using the Web for Language Independent Spellchecking and Autocorrection)

Using the Web to Infer Misspellings

- Vast amount of content
- User generated
- Contains both well spelled and misspelled

“For the error model, we use a large corpus (up to 3.7×108 pages) of crawled public Web pages.”*

* (Using the Web for Language Independent Spellchecking and Autocorrection)

Confidence Classifiers

A “simple” classifier

which compares the value of $\log(P(s|w)) - \log(P(w|w))$, for the original word w and the top-ranked suggestion s, with a threshold value. If there are no suggestions other than w, then the $\log(P(s|w))$ term is ignored.

Confidence Classifiers

A logistic regression classifier

that uses five feature sets. The first set is a scores feature that combines the following scoring information:

(i) log(P(s|w)) − log(P(w|w)) for top ranked suggestion s.

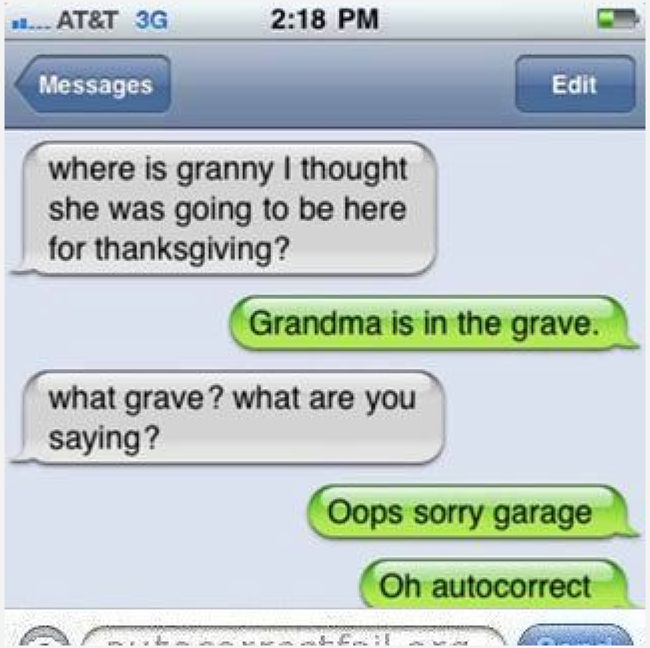
(ii) LM score difference between the original word w and the top suggestion s.

(iii) log(P(s|w)) − log(P(w|w)) for second top ranked suggestion s.

(iv) LM score difference between w and second top-ranked s.

Is it useful enough?

Despite the unending posts about embarrassing autocorrect incidents, auto correction is pretty useful and customizable.(In personal phones, laptops etc. the language model includes person’s typing/texting/searching history)



where is granny I thought she was going to be here for thanksgiving?

Grandma is in the grave.

what grave? what are you saying?

Oops sorry garage

Oh autocorrect

Evaluation

- **Correction Error Rate (CER)** = (E1 + E2 + E3 + E4)/T
- **Flagging Error Rate (FER)** = (E3 + E5)/T
- **Total Error Rate (TER)** = (E1 + E2 + E3 + E4 +E5)/T

Evaluation Metrics

1. E1: A misspelled word is wrongly corrected.

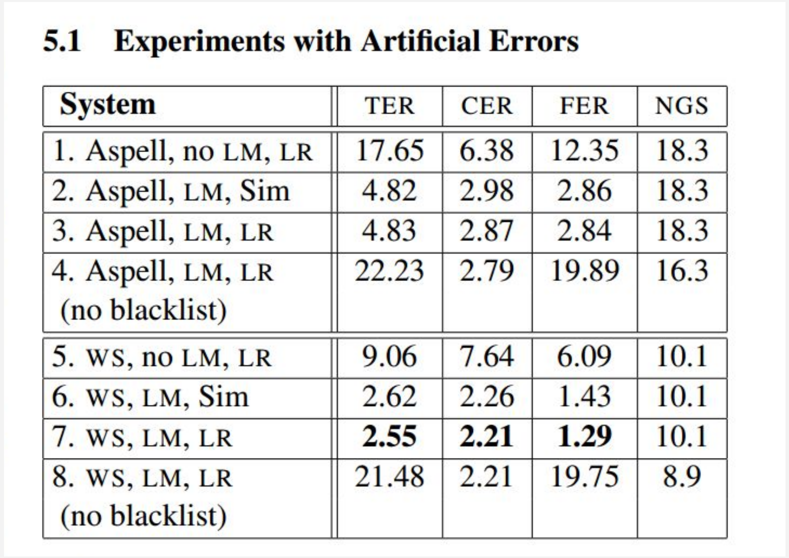
2. E2: A misspelled word is not corrected but is flagged.

3. E3: A misspelled word is not corrected or flagged.

4. E4: A well spelled word is wrongly corrected.

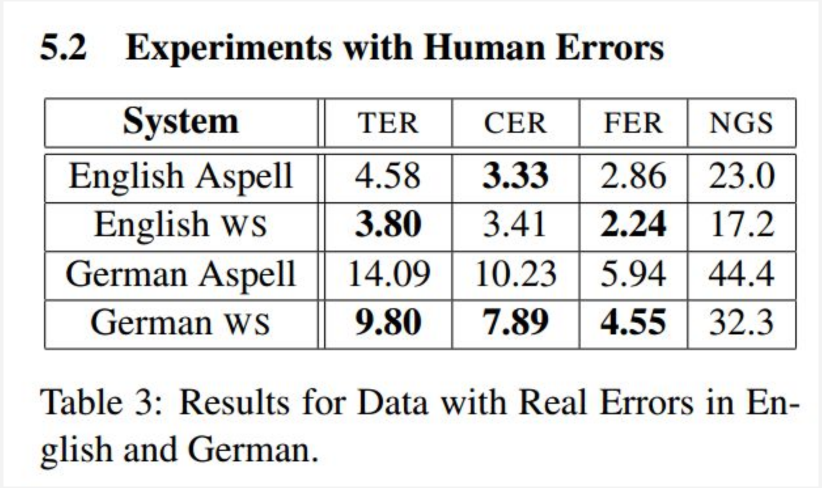
5. E5: A well spelled word is wrongly flagged

Evaluation



5.1 Experiments with Artificial Errors

System	TER	CER	FER	NGS
1. Aspell, no LM, LR	17.65	6.38	12.35	18.3
2. Aspell, LM, Sim	4.82	2.98	2.86	18.3
3. Aspell, LM, LR	4.83	2.87	2.84	18.3
4. Aspell, LM, LR (no blacklist)	22.23	2.79	19.89	16.3
5. WS, no LM, LR	9.06	7.64	6.09	10.1
6. WS, LM, Sim	2.62	2.26	1.43	10.1
7. WS, LM, LR	**2.55**	**2.21**	**1.29**	10.1
8. WS, LM, LR (no blacklist)	21.48	2.21	19.75	8.9



5.2 Experiments with Human Errors

System	TER	CER	FER	NGS
English Aspell	4.58	**3.33**	2.86	23.0
English WS	**3.80**	3.41	**2.24**	17.2
German Aspell	14.09	10.23	5.94	44.4
German WS	**9.80**	**7.89**	**4.55**	32.3

Table 3: Results for Data with Real Errors in English and German.

Aspell: GNU Aspell, an open source spell checker (Atkinson, 2009)
WS : Web-based Suggestions
NGS: No good suggestion rate.

Thank you!
Any questions?

References

- Peter Norvig's post on auto correct, from 2007 http://norvig.com/spell-correct.html
- How Difficult is it to Develop a Perfect Spell-checker? A Cross-linguistic Analysis through Complex Network Approach Monojit Choudhury1 , Markose Thomas2 , Animesh Mukherjee1 , Anupam Basu1 , and Niloy Ganguly1 1Department of Computer Science and Engineering, IIT Kharagpur, India {monojit,animeshm,anupam,niloy}@cse.iitkgp.ernet.in 2Google Inc. Bangalore, India markysays@gmail.com