13 – Automatic Language Correction

IA161 Natural Language Processing in Practice

A. Horák, J. Švec

NLP Centre, FI MU, Brno

December 12, 2023

Motivation

> *This tool can be use to find spelling , gramar or stylistic errors in english texts. just paste some text in the the box and click 'Submit to check . Additionally, their are many different dialects you can chose from. Additionally , you can hover your mouse over a error to see it's description and an useful list of posible corrections. You don´t need to worry for your writing skills any more, improving you're text has never be more easier!*

[1]Source: `http://www.onlinecorrection.com/`

Motivation

This tool can be use to find spelling , gramar or stylistic errors in english texts. just paste some text in the the box and click 'Submit to check . Additionally, their are many different dialects you can chose from. Additionally , you can hover your mouse over a error to see it's description and an useful list of posible corrections. You don´t need to worry for your writing skills any more, improving you're text has never be more easier!

Types of errors[1]:

Grammar (6) Spelling (10) Other (2) Spacing (3) Typographical (2) Duplication (1)

[1]Source: `http://www.onlinecorrection.com/`

Automatic language correction

A text with errors...

- is less comprehensible,
- looks less professional,
- poses problems for machine translation

Automatic language correction

A text with errors...

- is less comprehensible,
- looks less professional,
- poses problems for machine translation

People are quite resilient to letter-switching errors:

Example (Cmabrigde Uinervtisy (Cambridge University) effect)

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Example by Davis, M. 2003. Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy
`http://www.mrc-cbu.cam.ac.uk/people/matt.davis/cmabridge/`

Automatic language correction

Automatic language correction:

- spell checking – detect spelling errors in individual words,
- grammar checking – incorrect use of person, number, case or gender, improper verb government, wrong word order, etc. . .
- word completion – suggestion of the word currently being entered.

Spell checking

- detecting which words in a document are misspelled,
- providing spelling suggestions for incorrectly spelled words in a text,
- correction is the task of substituting the well-spelled hypotheses for misspellings,
- usually uses a dictionary of valid words,
- application: word processing and postprocessing optical character recognition [Whitelaw et al., 2009] or speech recognition.

Type of errors

- Non-word errors – the misspelled word is not a valid word in a language,
  - typographic errors – usually keyboard typing error (e.g. "teh" – "the", "speel" – "spell"),
  - cognitive errors – caused by the writer's misconceptions (e.g. "recieve" – "receive", "conspiricy" – "conspiracy"),
  - phonetic errors – substituting a phonetically equivalent sequence of letters (e.g. "seperate" – "separate").
- Real-word errors – sentence contains a valid word, but it is inappropriate in the context [Hladek et al., 2013].

Example

Non-word error: "I'd like a peice of cake."
Real-word error: "I'd like a peace of cake."

Error correction

- Consists of two steps:
  - generation of candidate corrections,
  - ranking of candidate corrections.
- Isolated-word methods:
  - edit distance,
  - similarity keys,
  - character n-gram-based techniques,
  - rule-based techniques,
  - probabilistic techniques,
  - neural networks [Rothe et al., 2021].

Isolated-word methods I

Edit distance

- assumption – person usually makes few errors,
- minimum set of operations to transform a non-word to a dictionary word,
- operations: insertions, deletions and substitutions,
- useful for: correcting errors resulting from keyboard input.

Example

Edit distance between "kitten" and "sitting" is 3:

1. kitten $\rightarrow$ sitten — substitution of "s" for "k"
2. sitten $\rightarrow$ sittin — substitution of "i" for "e"
3. sittin $\rightarrow$ sitting — insertion of "g" at the end

Isolated-word methods II

Similarity keys:

- assign a key to each dictionary word,
- compare with the key computed for the non word,
- most similar key is selected as suggestion.

Soundex – phonetic algorithm (English) [Holmes and McCabe, 2002]

Example

N	Represents letters
1	B, F, P, V
2	C, G, J, K, Q, S, X, Z
3	D, T
4	L
5	M, N
6	R

1. Keep the first letter
2. Drop occurrences of `a`, `e`, `i`, `o`, `u`, `y`, `h`, `w`
3. Replace letters with numbers
4. Merge adjacent identical numbers
5. Add zeroes to the end, or remove right-most numbers

Output: `(letter, number, number, number)`

key("Robert")=R163; key("Robin")=R150 – not similar
key("Smith")=S530; key("Smyth")=S530 – similar

Isolated-word methods III

Character N-gram-based techniques:

- compute similarity coefficient of two strings
- based on the number of shared n-grams (*Jaccard similarity*)

$$\delta_n(a, b) = \frac{|n\text{-}grams(a) \cap n\text{-}grams(b)|}{|n\text{-}grams(a) \cup n\text{-}grams(b)|}$$

Example

fact vs. fract

bigrams("fact")	= {"-f", "fa", "ac", "ct", "t-"}	... 5 bigrams
bigrams("fract")	= {"-f", "fr", "ra", "ac", "ct", "t-"}	... 6 bigrams
... ∩ ...	= {"-f", "ac", "ct", "t-"}	... 4 bigrams
... ∪ ...	= {"-f", "fa", "fr", "ra", "ac", "ct", "t-"}	... 7 bigrams

$$\delta_2(\text{"fact"}, \text{"fract"}) = \tfrac{4}{7} = 0.57$$

Isolated-word methods IV

Rule-based techniques

- a set of rules for common misspellings and typographic errors,
- each rule "fixes" one kind of error
- rules are applied to out-of-vocabulary words

Probabilistic techniques

- based on statistical features of the language (corpus)
  - transition probabilities – probability that a letter is followed by another letter
  - confusion probabilities – how often a letter is mistaken or substituted for another letter

Neural networks

- employs neural language models for context
- word-based – input node = every possible n-gram in every position of a word
- output node for each word in the dictionary
- character-based with recurrent neural networks

Outline

Grammar checking

Example

"That's good to now"
"That's good to know"

Grammar checking starts where spell checking ends

- deals with the most difficult and complex type of language errors
  - wrong word order,
  - verb tense errors,
  - subject/verb agreement,
  - punctuation errors,
  - etc...
- two main approaches
  - rule-based methods – time-consuming, less flexible, more precise better interpretability
  - statistical methods – easier and faster to implement, learn from examples
    need a lot of data [Rothe et al., 2021]

Rule-based grammar checking

Testing the input text against a set of handcrafted rules

Example

rule: I + verb(3rd person, singular form)
→ incorrect verb form usage – "I has a dog"

- ⊕ advantages:
  - rules can be easily added, modified or removed
  - rule can have a corresponding extensive explanation,
  - decisions can be traced to a particular rule,
  - rules can be authored by linguists, no need of programming
- ⊖ disadvantages:
  - large amount of manual work
  - extensive rule set is needed [Mozgovoy, 2011].

Rule-based grammar checker example

LanguageTool[2] – open source grammar checker

1. plain text as input
2. splits text into sentences
3. splits sentences into words
4. finds part-of-speech tags for each word and its base form
   walks – walk
5. matches the analyzed sentences against error patterns and runs rules.

[2] `https://languagetool.org/` [Naber, 2003, Brenneis, 2018]

Rule example in LanguageTool

Example

"I thing that's a good idea."

```
<rule id="YOU_THING" name="Possible typo 'I/you/... thing(think)'">

<pattern mark_from="1">
        <token regexp="yes">I|you</token>
        <token regexp="yes">thing|things</token>
</pattern>

<message>Did you mean <suggestion>think</suggestion> ?</message>
<example type="correct">I <marker>think</marker> that's a good idea.</example>

</rule>
```

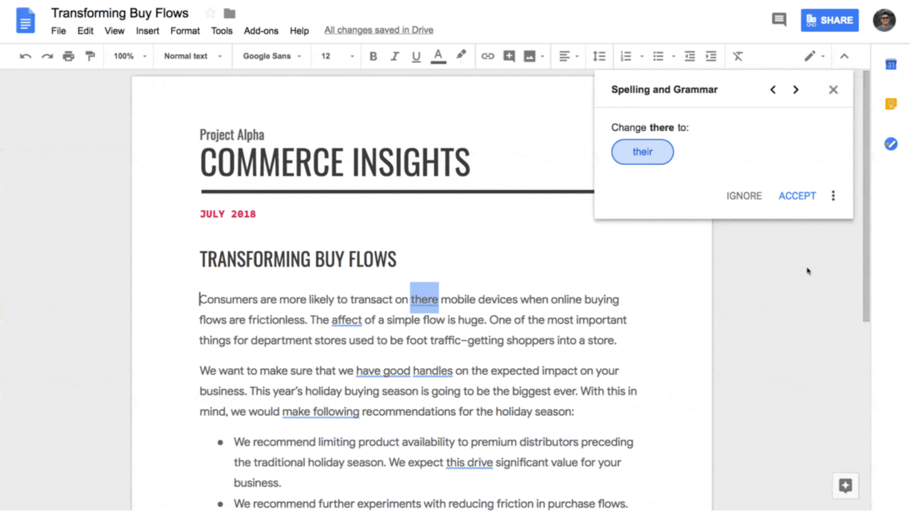
Statistical grammar checking

- based on analysis of grammatically correct POS-annotated corpus,
- build a list of POS tag sequences,
  - some sequences are very common (determiner+adjective+noun as in "the old man")
  - others will probably not occur at all (determiner+determiner+adjective)
- sequences which occur often in the corpus are considered correct,
- uncommon sequences might be errors.

Google Grammar Checker

- available in Google Docs since 2019
- based on neural machine translation architecture
- trains to translate incorrect language → correct language [Rothe et al., 2021]

Google Grammar Checker



Outline

Word completion

- reduce the number of keystrokes
- suggesting the completion of the word
- use context information to predict what block of characters (letters, n-grams, syllables, words, or entire phrases) a person is going to write next
- based on wide-coverage word or language model
- character-based models with transfer from word-based models [Jawahar et al., 2022]

Best results

- Spell checking (first suggestion):
  - English – 97 % [Gupta, 2020]
  - Czech – 95 % [Gupta, 2020]
- Grammar checking (various tests average):
  - English – 78 % [Didenko and Sameliuk, 2023]
  - Czech – 83 % [Rothe et al., 2021, Náplava et al., 2022]

References I

Brenneis, M. (2018).
Development of neural network based rules for confusion set disambiguation in languagetool.
SKILL 2018-Studierendenkonferenz Informatik.

Didenko, B. and Sameliuk, A. (2023).
RedPenNet for grammatical error correction: Outputs to tokens, attentions to spans.
In Romanyshyn, M., editor, *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 121–131, Dubrovnik, Croatia. Association for Computational Linguistics.

Gupta, P. (2020).
A context-sensitive real-time spell checker with language adaptability.
In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pages 116–122. IEEE.

References II

Hladek, D., Stas, J., and Juhar, J. (2013).
Unsupervised spelling correction for Slovak.
Advances in Electrical and Electronic Engineering, 11(5):392–397.

Holmes, D. and McCabe, M. C. (2002).
Improving precision and recall for soundex retrieval.
In *Information Technology: Coding and Computing, 2002. Proceedings. International Conference on*, pages 22–26. IEEE.

Jawahar, G., Mukherjee, S., Dey, D., Abdul-Mageed, M., Lakshmanan, L. V., Mendes, C. C. T., de Rosa, G. H., and Shah, S. (2022).
Small character models match large word models for autocomplete under memory constraints.
arXiv preprint arXiv:2210.03251.

References III

Mozgovoy, M. (2011).
Dependency-based rules for grammar checking with LanguageTool.
In *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, pages 209–212.

Naber, D. (2003).
A rule-based style and grammar checker.

Náplava, J., Straka, M., Straková, J., and Rosen, A. (2022).
Czech grammar error correction with a large and diverse corpus.
Transactions of the Association for Computational Linguistics, 10:452–467.

References IV

Rothe, S., Mallinson, J., Malmi, E., Krause, S., and Severyn, A. (2021).
A simple recipe for multilingual grammatical error correction.
In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Whitelaw, C., Hutchinson, B., Chung, G. Y., and Ellis, G. (2009).
Using the web for language independent spellchecking and autocorrection.
In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 890–899, Stroudsburg, PA, USA. Association for Computational Linguistics.