

Context-Sensitive Spelling Correction for Web Queries

Linguistics 445/515

Autumn 2008

1 Spelling Correction for Web Queries

It's hard because it must handle:

- Proper names, new terms, etc. (*blog, shrek, nsync*)
- Frequent and severe spelling errors (10-15%)
- Very short contexts

2 Spelling Correction Algorithm

2.1 Main Idea (Cucerzan and Brill (EMNLP-04))

- Iteratively transform the query into more likely queries
- Use query logs to determine likelihood
 - Despite the fact that many of these are misspelled!
 - Assumptions: the less wrong a misspelling is, the more frequent it is; and correct > incorrect

Example:

anol scwartegger
→ *arnold schwartnegger*
→ *arnold schwarznegger*
→ *arnold schwarzenegger*

2.2 Algorithm

- Compute the set of all *close* alternatives for each word in the query
 - Look at word unigrams and bigrams from the logs; this handles concatenation and splitting of words
 - Use weighted edit distance to determine closeness
- Search sequence of alternatives for best alternative string, using a noisy channel model

Constraint:

- No two adjacent in-vocabulary words can change simultaneously

2.3 The Iterative Algorithm, More Formally:

Given a string s_0 , find a sequence s_1, s_2, \dots, s_n such that:

- $s_n = s_{n-1}$ (stopping criterion)
- $\forall i \in 0 \dots n - 1$,
 - $dist(s_i, s_{i+1}) \leq \delta$ (only a minimal change)
 - $P(s_{i+1}|s_i) = \max_t P(t|s_i)$ (the best change)

3 Examples

Context Sensitivity

- *power crd* → *power cord*
- *video crd* → *video card*
- *platnuin rings* → *platinum rings*

Known Words

- *golf war* → *gulf war*
- *sap opera* → *soap opera*

Tokenization

- *chat inspanich* → *chat in spanish*
- *ditroitigers* → *detroit tigers*
- *britenetspear inconcert* → *britney spears in concert*

Constraints

- *log wood* → *log wood* (not *dog food*)