Robust Error Detection: A Hybrid Approach Combining Unsupervised Error Detection and Linguistic Knowledge

Johnny Bigert and Ola Knutsson
Royal Institute of Technology
Stockholm, Sweden

johnny@nada.kth.se
knutsson@nada.kth.se
Detection of context-sensitive spelling errors

- Identification of less-frequent grammatical constructions in the face of sparse data
- Hybrid method
  - Unsupervised error detection
  - Linguistic knowledge used for phrase transformations
Properties

- Find difficult error types in unrestricted text (spelling errors resulting in an existing word etc.)
- No prior knowledge required, i.e. no classification of errors or confusion sets
A first approach

Algorithm:

for each position $i$ in the stream
  if the frequency of $(t_{i-1} t_i t_{i+1})$ is low
    report error to the user
  report no error
Sparse data

Problems:

- Data sparseness for trigram statistics
- Phrase and clause boundaries may produce almost any trigram
Sparse data

Example:

- "It is every manager's task to..."
- "It is every" is tagged \( \text{(pn.neu.sin.def.sub/obj, vb.prs.akt, dt.utr/neu.sin.ind)} \) and has a frequency of zero

- Probable cause: out of a million words in the corpus, only 709 have been assigned the tag \( \text{(dt.utr/neu.sin.ind)} \)
Sparse data

We try to replace

- "It is every manager's task to..."
with

- "It is a manager's task to..."
Sparse data

- "It is every" is tagged
  (pn.neu.sin.def.sub/obj, vb.prs.akt, dt.utr/neu.sin.ind) and had a frequency of 0

- "It is a" is tagged
  (pn.neu.sin.def.sub/obj, vb.prs.akt, dt.utr.sin.ind) and have a frequency of 231

- (dt.utr/neu.sin.ind) had a frequency of 709
- (dt.utr.sin.ind) has a frequency 19112
Tag replacements

When replacing a tag:

- All tags are not suitable as replacements
- All replacements are not equally appropriate...
- ...and thus, we require a penalty or probability for the replacement
Tag replacements

To be considered:

- Manual work to create the probabilities for each tag set and language
- The probabilities are difficult to estimate manually

- Automatic estimation of the probabilities (other paper)
Tag replacements

Examples of replacement probabilities:

100% vb.prt.akt.kop vb.prt.akt.kop
74% vb.prt.akt.kop vb.prs.akt.kop
50% vb.prt.akt.kop vb.prt.akt___
48% vb.prt.akt.kop vb.prt.sfo

Mannen var glad. (The man was happy.)
Mannen är glad. (The man is happy.)
Tag replacements

Examples of replacement probabilities:

100%  dt.utr/neu.plu.def  dt.utr/neu.plu.def
44%   dt.utr/neu.plu.def  dt.utr/neu.plu.ind/def
42%   dt.utr/neu.plu.def  ps.utr/neu.plu.def
41%   dt.utr/neu.plu.def  jj.pos.utr/neu.plu.ind.nom

Mannen talar med de anställda.
(The man talks to the employees.)

Mannen talar med våra anställda.
(The man talks to our employees.)
Weighted trigrams

Replacing \((t_1 t_2 t_3)\) with \((r_1 r_2 r_3)\):

- \(f = \text{freq}(r_1 r_2 r_3) \cdot \text{penalty}\)
- \(\text{penalty} = \Pr[\text{replace } t_1 \text{ with } r_1] \cdot \Pr[\text{replace } t_2 \text{ with } r_2] \cdot \Pr[\text{replace } t_3 \text{ with } r_3]\)
Weighted trigrams

Replacement of tags:

- Calculate $f$ for all representatives for $t_1$, $t_2$ and $t_3$ (typically $3 \cdot 3 \cdot 3$ of them)
- The weighted frequency is the sum of the penalized frequencies
Algorithm:

for each position $i$ in the stream

if weighted freq for $(t_{i-1} t_i t_{i+1})$ is low
  report error to the user

report no error
An improved algorithm

- Problems with sparse data
- Phrase and clause boundaries may produce almost any trigram
- Use clauses as the unit for error detection to avoid clause boundaries
Phrase transformations

- We identify phrases to transform rare constructions to those more frequent
- Replacing the phrase with its head
- Removing phrases (e.g. AdvP, PP)
Phrase transformations

Example:

Alla hundar som är bruna är lyckliga

(All dogs that are brown are happy)

Hundarna är lyckliga

(The dogs are happy)
Phrase transformations

- Den bruna (jj.sin) hunden (the brown dog)
- De bruna (jj.plu) hundarna (the brown dogs)
Phrase transformations

The same example with a tagging error:

Alla hundar som är bruna (jj.sin) är lyckliga

\[ \text{NP} \]

(All dogs that are brown are happy)

Robust NP detection yield

Hundarna är lyckliga

\[ \text{NP} \]

(The dogs are happy)
Results

Error types found:

- context-sensitive spelling errors
- split compounds
- spelling errors
- verb chain errors
Comparison between probabilistic methods

- The unsupervised method has a good error capacity but also a high rate of false alarms.
- The introduction of linguistic knowledge dramatically reduces the number of false alarms.
Future work

- The error detection method is not only restricted to part-of-speech tags - we consider adopting the method to phrase n-grams
- Error classification
- Generation of correction suggestions
Summing up

- Detection of context-sensitive spelling errors
- Combining an unsupervised error detection method with robust shallow parsing
Internal Evaluation

- POS-tagger: 96.4%
- NP-recognition: $P=83.1\%$ and $R=79.5\%$
- Clause boundary recognition: $P=81.4\%$ and 86.6\%