A generalized method for iterative error mining in parsing results

Daniel de Kok, Gertjan van Noord

January 22, 2009

What is error mining?

Two common types of parsing problems:

- Incorrect parse, e.g. by incorrect disambiguation.
- Incomplete parse: no analysis spanning the full sentence could be found, usually due to missing dictionary items or an incomplete grammar.

It would be useful if we can automatically find out what word or n-gram causes the incomplete parse.

What is error mining? (2)

The basic idea:

- Parse a (unannotated) corpus.
- Extract parsable and unparsable sentences.
- Look what n-grams occur in the list of unparsable sentences, but do not in the list of parsable sentences.
- Assign some score to n-grams.

Van Noord (2004)

Suspicion of n-grams:

$$S(w_i..w_j) = \frac{C(w_i...w_j|error)}{C(w_i...w_j)} \tag{1}$$

Problem: although one or just a few forms are responsible for most parsing failures, all the forms occuring in an unparsable sentence take blame.

Sagot and de la Clergerie (2006)

Iterative error mining method wherein:

- Observations of forms within sentences start with uniform suspicions ($S_{i,j}^{(0)} = \frac{error(s_i)}{|S_i|}$)
- The suspicion of a form is the mean of the suspicions of all observations of that form.
- The suspicion of an observation is the suspicion of its form, normalized by the sum of the suspicions of all forms that occur in the sentence, multiplied by the sentence error rate (normally 0.0 or 1.0).
- Mining on unigrams and bigrams.

Usefulness of n-grams

However, often the parsability of a word depends on the context. For instance consider experiments with Alpino:

- The word *via* had a suspicion of less than 0.1.
- The parser was unable to parse the expression *via via*.

Evaluation methodology

- We want to improve the coverage of a grammar: we are interested in seeing forms with many unparsable sentences first (recall).
- Additionally, we are interested in forms that primarily occur in unparsable sentences (precision).
- Combined: f-score, or in our case f0.5-score

Precision/recall/f-score

- Precision: $P = \frac{|\{unparsable\ \ sentences\} \cap \{retrieved\ \ sentences\}|}{|\{retrieved\ \ sentences\}|}$
- Recall: $R = \frac{|\{unparsable\ \ sentences\} \cap \{retrieved\ \ sentences\}|}{|\{unparsable\ \ sentences\}|}$
- F-score: $\frac{(1+\beta^2) \cdot (precision \cdot recall)}{(\beta^2 \cdot precision + recall)}$

Evaluation material

Testing was performed on the Dutch Wikipedia corpus, that was parsed with the wide-coverage Alpino parser.

- 109 million words
- 7 million sentences
- 8.4% of the sentences were unparsable

Scoring methods

Sagot and De la Clergerie proposed the following scoring methods:

- Concentrating on suspicions:

$$M_f = S_f \tag{2}$$

- Concentrating on most frequent potential errors:

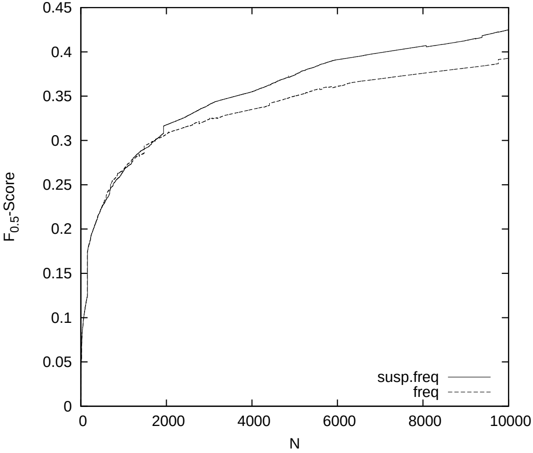
$$M_f = S_f|O_f| \tag{3}$$

- Balancing between these possibilities:

$$M_f = S_f \cdot ln|O_f| \tag{4}$$

We replaced $|O_f|$ in the last two cases by $|\{O_{f,unparsable}\}|$.

Scoring methods (results)



Purifying suspicions

- After mining some forms have a very low suspicion, high frequency. If we use a frequency-based scoring method these forms still get a high score.
- Solution: during mining exclude forms that are practically negiable (e.g. with a suspicion below 0.001). Added bonus: very suspicious forms get a suspicion of 1.

Mining of n-grams (2)

However, blindly adding n-grams as forms distorts mining.
Consider the sequence

A B C

where *B* only occurs in unparsable sentences. In this case, the bigrams *A B* and *B C*, and the trigram *A B C* will also be highly suspicious, but if we want to be specific we should only return *B* as a suspicious form.
Additionally, adding all n-grams for a larger n is expensive (time and memory-wise).

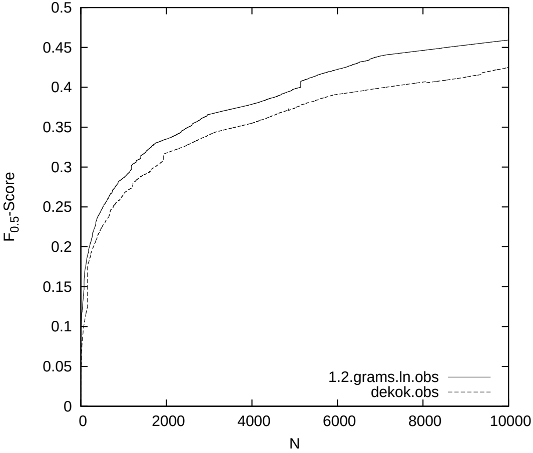
Mining of n-grams (3)

Preprocessing method:

- Iterate through a sentence by unigram.
- Try to extend each unigram stepwise, where an extension is allowed if the *unparsable/all* ratio of an n+1-gram is higher than both of its n-grams.
- The sentence is represented by the n-grams $n_0..n_x$, $n_1..n_y$, ... $n_{|s_i|-1}..n_{|s_i|-1}$.

We then start mining with the resulting forms and observations.

Mining of n-grams (results)



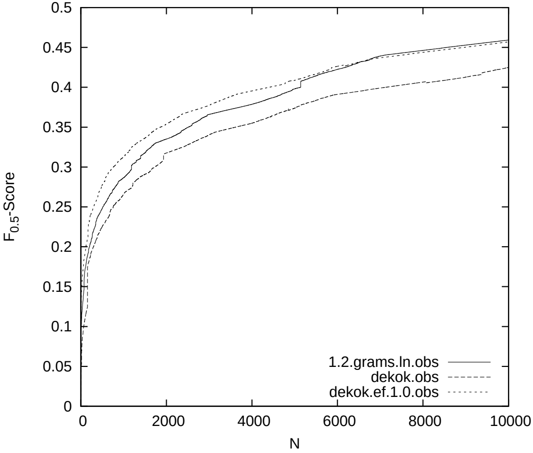
Mining of n-grams (data sparseness)

To negate the problem of data sparseness, we added a factor that is dependant on the for frequency, such that extension only happens when $R(AB) > extFactor \cdot R(A)$ and $R(AB) > extFactor \cdot R(B)$.

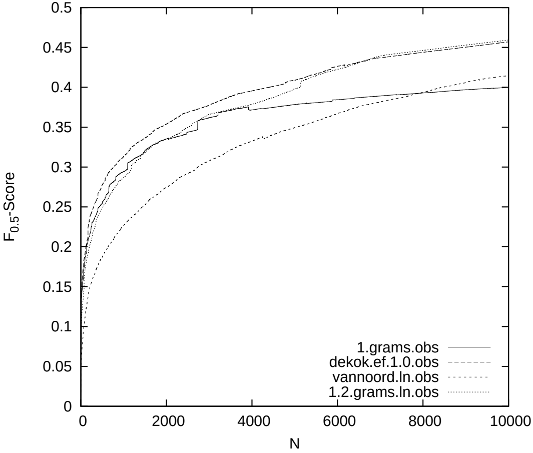
This multiplication factor is defined as:

$$extFactor = 1 + e^{-\alpha |O_{f,unparsable}|} \tag{5}$$

Mining of n-grams (results)



Overall results



Mining a Flemish corpus

Flemish Mediargus corpus, parsed with the Dutch Alpino parser.

- 1.1 billion words
- 67 million sentences
- 9.2% unparsable.

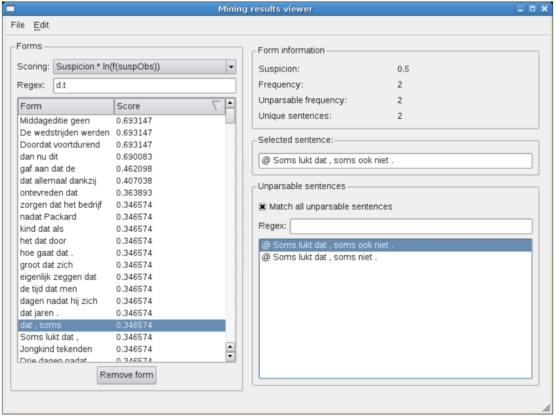
Flemish expressions

- Telkens hij [Everytime he]
- (had er AMOUNT) voor veil [(had AMOUNT) for sale]
- (om de muren) van op te lopen [to get terribly annoyed by]
- Ik durf zeggen dat [I dare to say that]
- op punt stellen [to fix/correct something]
- de daver (op het lijf) [shocked]
- (op) de tippen (van zijn tenen) [being very careful]
- ben fier dat [am proud of]
- Nog voor halfweg [still before halfway]
- (om duimen en vingers) van af te likken [delicious]

Long n-grams

- Het stond in de sterren geschreven dat NAME
- zowat de helft van de [...]
- er zo goed als zeker van dat
- laat ons hopen dat het dit/lukt

Mining viewer



Forms

Scoring: Suspicion * ln(f(suspObs))

Regex: d.t

Form	Score
Middageditie geen	0.693147
De wedstrijden werden	0.693147
Doordat voortdurend	0.693147
dan nu dit	0.690083
gaf aan dat de	0.462098
dat allemaal dankzij	0.407038
ontevreden dat	0.363893
zorgen dat het bedrijf	0.346574
nadat Packard	0.346574
kind dat als	0.346574
het dat door	0.346574
hoe gaat dat .	0.346574
groot dat zich	0.346574
eigenlijk zeggen dat	0.346574
de tijd dat men	0.346574
dagen nadat hij zich	0.346574
dat jaren .	0.346574
dat , soms	0.346574
Soms lukt dat ,	0.346574
Jongkind tekenden	0.346574
Drie dagen nadat	0.346574

Form information

Suspicion:	0.5
Frequency:	2
Unparsable frequency:	2
Unique sentences:	2

Selected sentence:

@ Soms lukt dat , soms ook niet .

Unparsable sentences

☒ Match all unparsable sentences

Regex:

@ Soms lukt dat , soms ook niet .
@ Soms lukt dat , soms niet .

Software

- Van Noord (2006): http://www.let.rug.nl/vannoord/software.html
- Iterative method, extensions, mining viewer: http://www.let.rug.nl/dekok/errormining/