The Effect of Lexicon Bad Smells on Concept Location in Source Code

Surafel Lemma Abebe[1], Sonia Haiduc[2], Paolo Tonella[1], Andrian Marcus[2]

[1]Software Engineering Research Unit
Fondazione Bruno Kessler
Trento, Italy

[2]Department of Computer Science
Wayne State University
Detroit, MI, USA

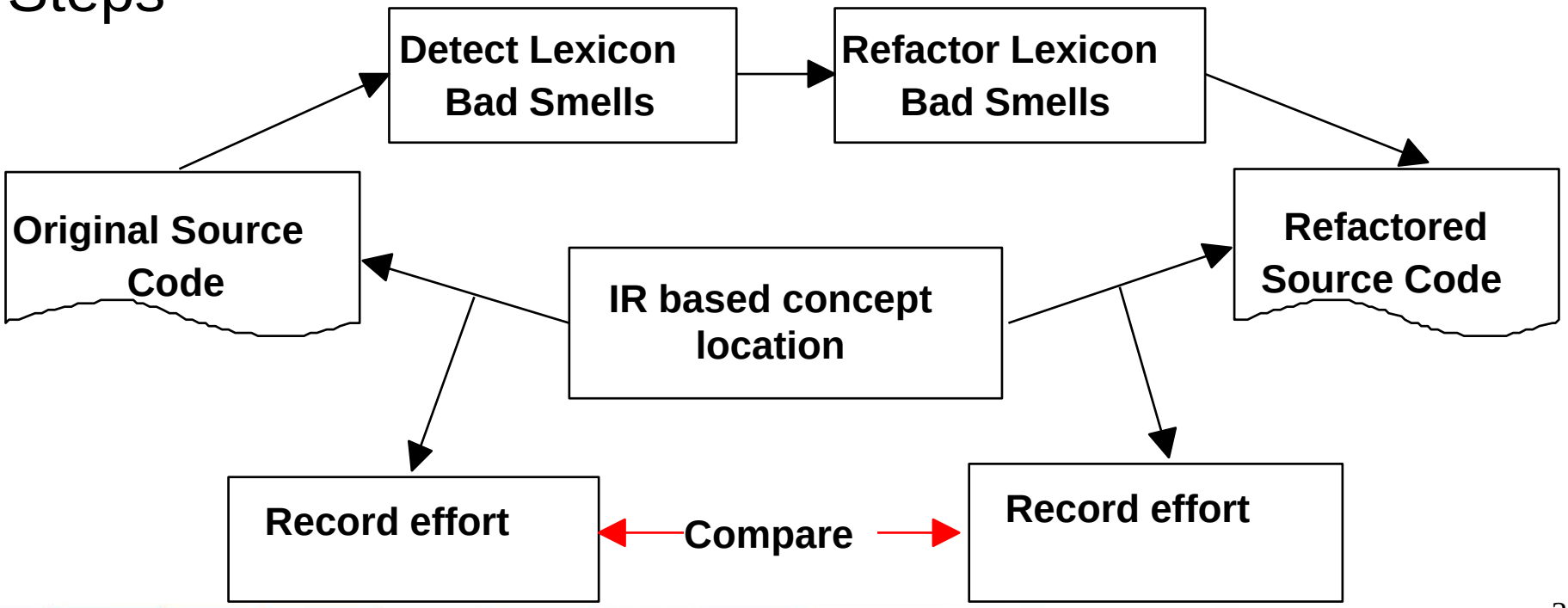
SCAM 2011

Introduction

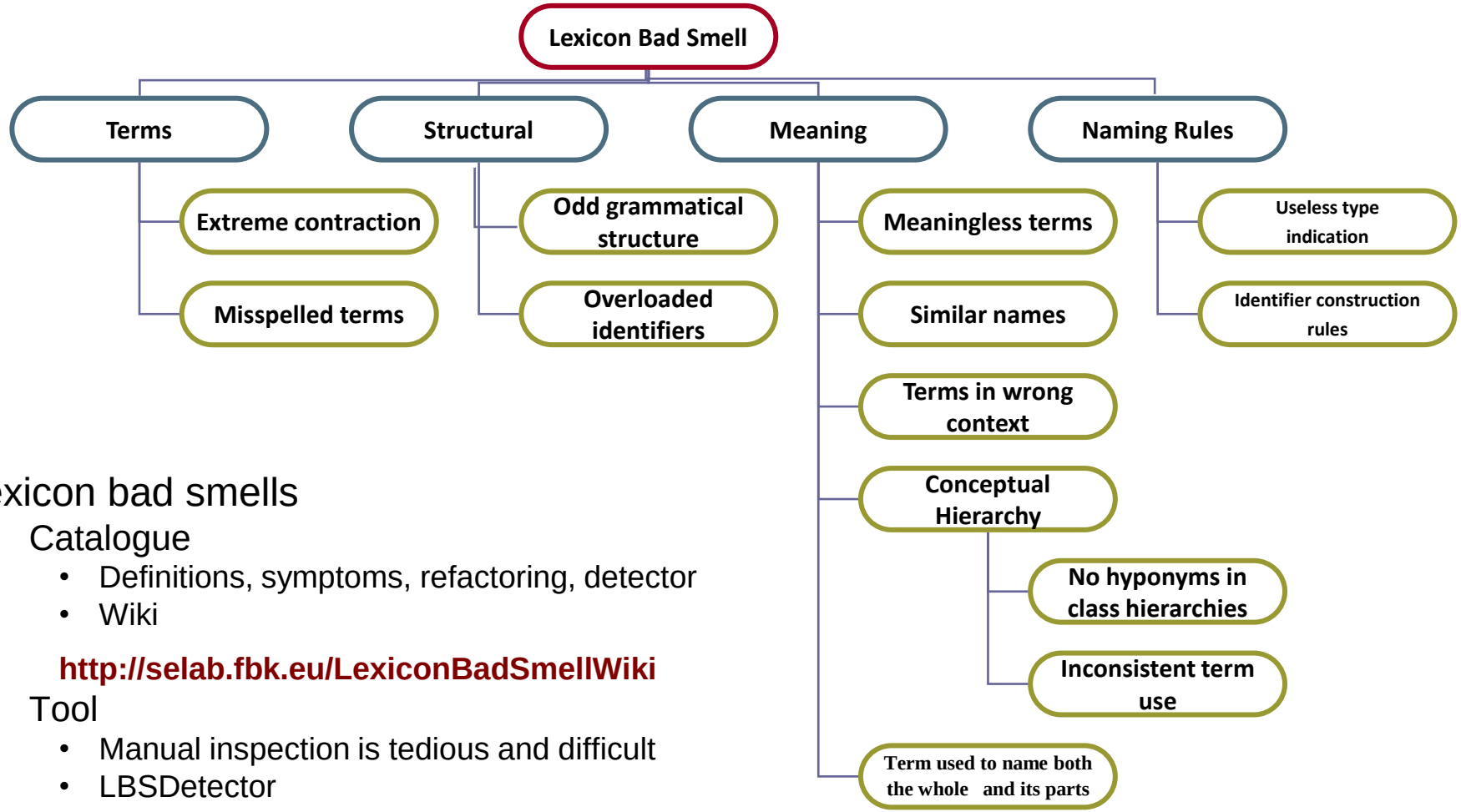
- Reading and understanding code
  - Identifiers and comments play important role
  - Affected by the quality of identifiers
    - Flaws in the naming of identifiers
      - Lexicon bad smells
      - Quality identifier: few to none of lexicon bad smells
  - Importance of high quality identifiers is acknowledged
    - Level of difficulty imposed is unknown
- Effect of lexicon bad smells on concept location

Approach

- Reenactment in a before-and-after study
  - No control group
  - Automated
    - Concept location tools
- Steps



Approach …



- Lexicon bad smells
  - Catalogue
    - Definitions, symptoms, refactoring, detector
    - Wiki

    http://selab.fbk.eu/LexiconBadSmellWiki
  - Tool
    - Manual inspection is tedious and difficult
    - LBSDetector

Approach…

- IR based concept location

- Effort measure
  - Ranks
  - Impact of lexicon bad smells
    - Rank of target classes before and after refactoring

Case study

System	Number of classes	Number of bugs
FileZilla Client 3.0.0	209	29
Open Office 1.0.0	~12,000	19

- Corpus:
  - Identifiers are splitted, originals are kept
  - Common English terms and C++ terms are removed
- Queries:
  - Title + description
- IR techniques
  - Latent semantic Indexing
  - Lucene

Case study...

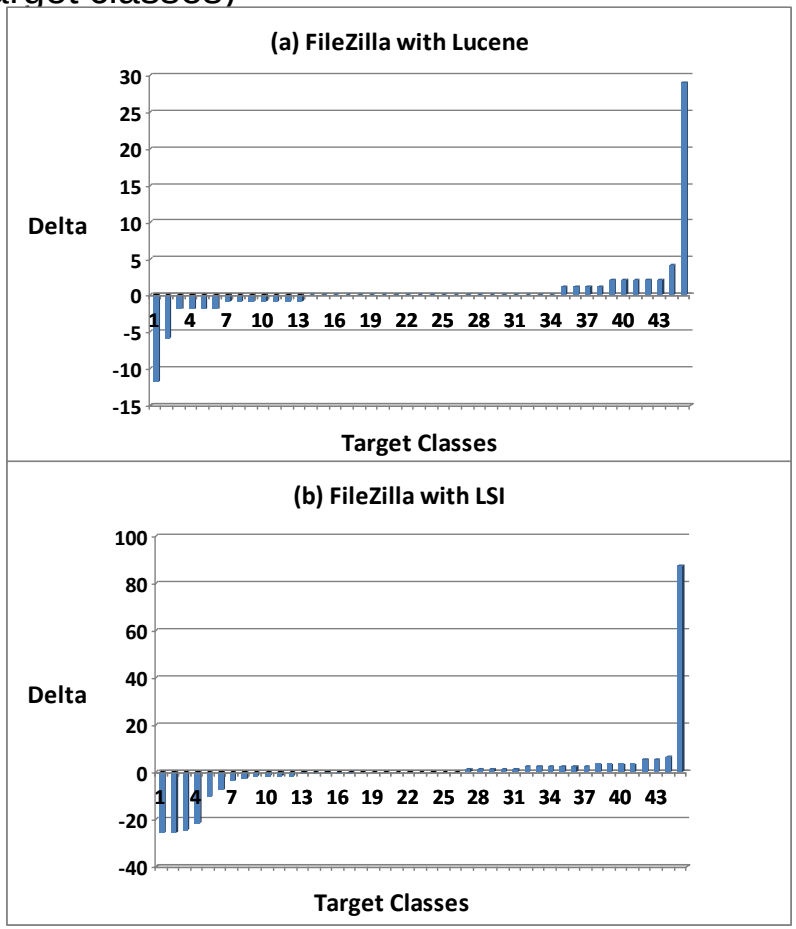
- Lexicon bad smell
  - Extreme contractions
    - Lev → Levenshtein, Exc → Excel
  - Inconsistent identifiers
    - connect, connectToClient → connectToServer
  - Misspelling
    - IsApplyable → isApplicable
  - Odd grammatical structure
    - command → executeCommand
  - Meaningless term
    - Var
- Actions performed
  - Term expansion:
    - nTrot → nTextRotation

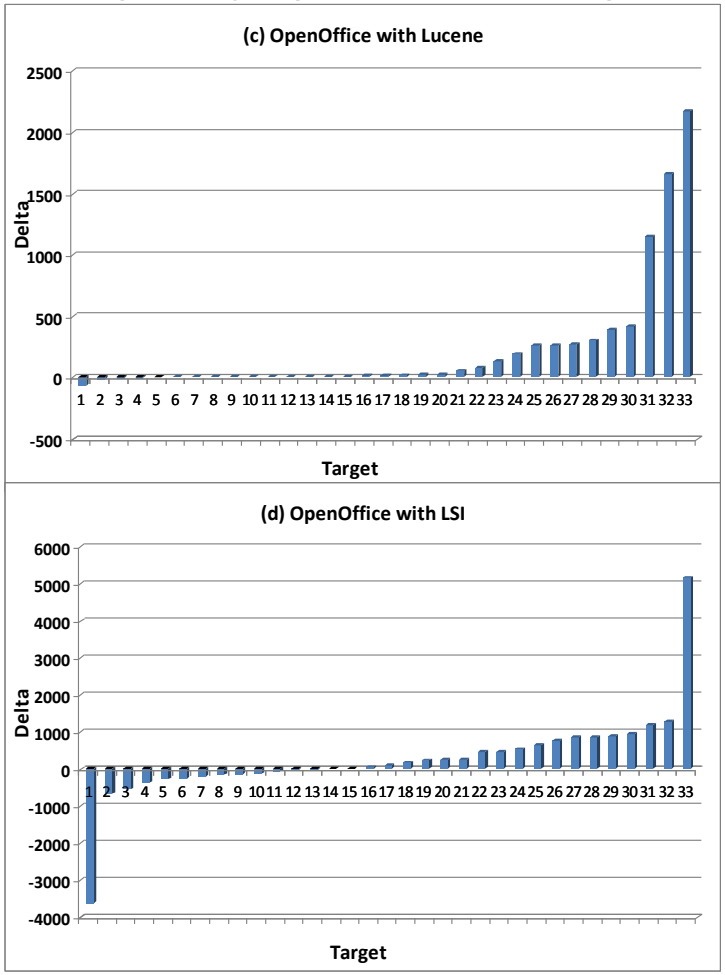
Bad Smell	FileZilla	OpenOffice
Extreme contraction	86	480
Inconsistent identifier use	95	74
Meaningless terms	0	1
Misspelling	64	436
Odd grammatical structure	147	434
Overloaded identifiers	4	12
Useless type indication	2	7
Whole-part	13	25
Number of identifiers containing bad smells in target classes	192	775
Number of identifier occurrences refactored in the system	2,216	90,749
Number of unique target classes	28	26

Type of action while correcting a smell	OpenOffice	FileZilla
Term expansion	484	38
Spelling correction	2	0
Term reordering	35	31
Added term	283	71
Deleted term	139	42
Replaced term	138	37
Language translation	33	0

Case study ...

- Results
  - FileZilla (29 bug reports, 45 non unique target classes)
  - OpenOffice (19 bug reports, 33 non unique target classes)





Statistics	FileZilla 3.0.0		OpenOffice 1.0.0	
	LSI	Lucene	LSI	Lucene
Absolute rank delta	-6	+14	8,315	7,281
Average delta (std dev)	-0.13 (15.4)	+0.31 (4.9)	251.97 (1212.9)	220.64 (495.7)
Delta t-test p-value	0.95	0.68	0.24	0.02

Case study ...

- Example: bug 4378 of OpenOffice
  - Target class: ExcXf8
  - Initial rank:
    - Lucene → 1,174
    - LSI → 691
  - 10 identifiers which contain 22 lexicon bad smells
    - Extreme contraction: 13 bad smells
    - Misspelled terms: 5 bad smells
    - Odd grammatical structure: 4 bad smells
  - Rank after refactoring
    - Lucene → 29 (improvement of 1,145 positions)
    - LSI → 453 (improvement of 242 positions)
  - Improvement
    - Meaningful terms introduced
      - Example: ExcXf8 → ExcelFile8
        nTrot -> nTextRotation
    - Increase in frequency of common terms
      - Example: rotation (1 -> 6)

Bug: 4378 (OpenOffice)			
		Bug description	orientation of cell content gets lost if exporting as excel 97 or html. in my spreadsheet I rotated the writing in one row for 90 degrees to the left. If I export the sheet as excel 97 or html the writing is not rotated anymore.Exporting as excel 95 works fine
	Original	Identifiers with lexicon bad smells	bFMergeCell, bFShrinkToFit, nCIndent, nDgDiag, nGrbitDiag, nIcvDiagSer, nIReadingOrder, nTrot, ExcXf8, GetLen, GetNum
	Original	Terms only in original corpus	xf8, excxf8, trot, ntrot, ncindent, nireadingorder, diag, ngrbitdiag, nicvdiagser, ndgdiag, bfshrinktofit, bfmergecell, num, getnum, len, getlen
	Refactored	Refactored identifiers	bFormatMergeCell, bFormatShrinkToFit, nCharacterIndent, nDiagonalBorderStyle, nGrbitDiagonalBorder, IndexColorValueDiagonalBorderSerial, nIndexReadingOrder, nTextRotation, ExcelFile8 GetLength, GetNumber
	Refactored	Terms only in refactored corpus	excel, file8, excelfile8, ntextrotation, character, ncharacterindent, nindexreadingorder, diagonal, border Ngrbitdiagonalborder, serial, nindexcolorvaluediagonalborderserial, ndiagonalborderstyle, format, bformatshrinktofit, bformatmergecell, number, getnumber, length, getlength

Conclusion and Future works

- Impact of lexicon bad smells on software comprehension task
- Case study: IR-based concept
- Lexicon bad smells can be an important factor
  - Relatively low quality => benefit from the refactoring
- Impact of individual lexicon bad smells on concept location
- Perform empirical studies with developers

Thank you