

Language Processing in Software Engineering

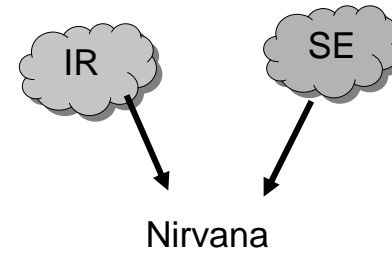
Dave Binkley*

Crest Centre – King's College London

Henry Field, Dawn Lawrie, Steve Maex,
Chris Morrell, Maurizio Pighin

*On Sabbatical Leave from Loyola College in Maryland

IR in SE



Language Processing

1. The *QALP* Score
2. Conciseness and Consistency
3. Natural Language
4. Expansion

1) The *QALP* Score

Goal *rate* modules

- Separate code and comments
- Stop list -- 'an', 'NULL'
- Stemming -- printable -> print
- ***tf-idf*** term weighting – [press any key]
- Cosine similarity

tf-idf Term Weighting

Accounts for **term frequency**

- how important the term is a document

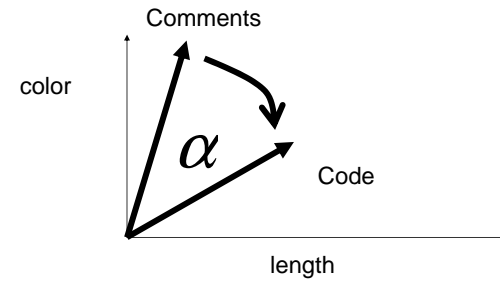
Inverse document frequency

- how common in the entire collection

*High weight --
frequent in document but rare in collection*

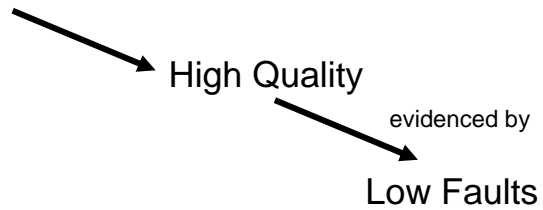
Cosine Similarity

$$= \cos(\alpha)$$



Why QALP ?

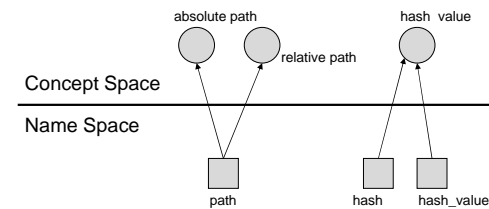
High QALP score



2) Consistency

- **Consistent**

– no homonyms or synonyms

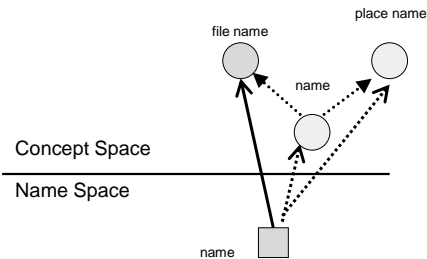


Deissenbock and Pizka

2) and Conciseness

- **concise**

- identifier semantics match concept semantics
- e.g. `output_file_name` for the name of the output file



Example

```
int getopt_long
(int argc, char *argv,
char *short_options,
struct option *long_options,
int *opt_index)
{
...
}
```

Other Examples

status	file_status	
home	home_dir	in_home
prefix	isolate_tilde_prefix	
FILE	copy_file	file_mode log_file
cwd	cwd_len	
home_dir	get_home_dir	
adr	next_adr	

In 50 MLoC

- 2900 consistency failures per program
- 1300 conciseness failures per program
- 72% of consistency and 76% of conciseness failures are “real”

3) Percent Natural Language

- Hard Words

- vertex_zeroinddeg → vertex zeroinddeg

hard word

hard word

- Soft Words

- zeroinddeg → zero-in-deg

soft word

soft word

soft word

3) Percent Natural Language

- Can get interesting

- thenewestone

4) Expansion

- Sources of natural language

- Comments
 - Other identifiers
 - Phrases (from other comments and identifiers)
 - Natural Language Dictionary

- Use wildcard substitution to find possible expansions

- deg → d*e*g* ... finds ... degree

Expansion Examples

	Original	Expanded
The Good	pw_dir	password_dir
	FS_EXISTS	file_status_exists
	stack_dir	stack_direction
The "Bad"	cnt	current
	eol_rn	eol_returns
	da_qsort	dictionary_qsort

Last Example – Future Work

use.c

```
// sort dictionary
```

```
da_qsort(dict);
```

definition.c

```
// dynamic array functions
```

```
da_qsort(Dictionary ...)
```

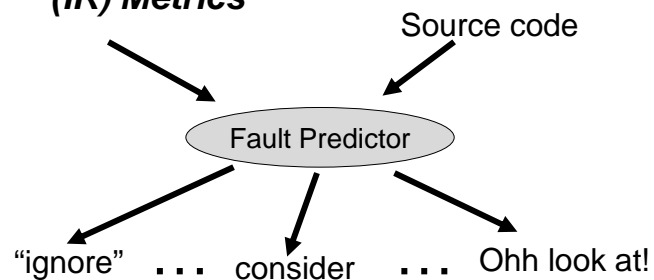
```
...
```

Effectiveness

- 16% of identifiers are modified
- 58% expanded correctly

An Application: Fault Prediction

(IR) Metrics



“Traditional” *Metrics*

- Dozens of **structure** based
 - Lines of code
 - Number of attributes in a class
 - Cyclomatic complexity

Why YAM?

(Yet Another Metric)

1. Many structural metrics bring similar value

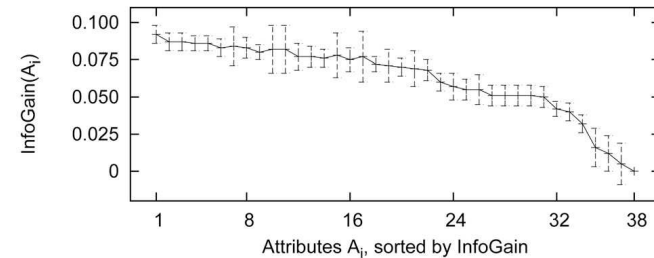
Recent example

Gyimothy et al. "Empirical validation of OO metrics ..." TSE 2007

Why YAM?

2. Menzies et al. "Data mining static code attributes to learn defect predictors."

TSE 2007



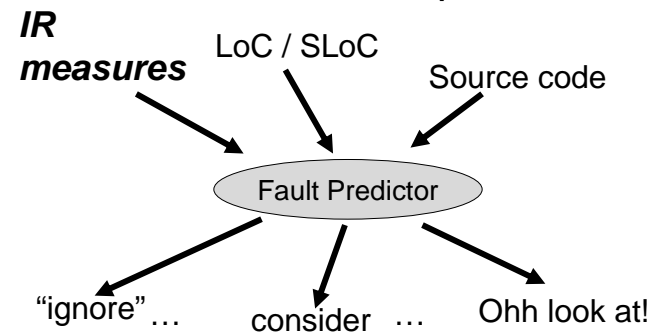
Why YAM? -- Diversity

"...[the] measures used ... [are] less important than having a *sufficient* pool to choose from.

Diversity in this pool is important."

Menzies et al.

Fault Prediction Experiment



Linear Mixed-Effects Regression Models

- Response variable
= f (Explanatory variables)

In the experiment

- Faults = f (IR measures, LoC, SLoC)

Two Test Subjects

- Mozilla – open source
– 3M LoC 2.4M SLoC
- MP – proprietary source
– 454K LoC 282K SLoC

Mozilla Model

- $defects = -0.057$
– 0.001 LoC
+ 0.007 SLoC
+ $QALP(-0.12 - 0.0023 SLoC)$
+ $PNL(0.34 - 0.00061 SLoC)$
+ $v(-1.7 - 0.0075 LoC + 0.024 SLoC)$
- Interactions exist

Mozilla Model Extract for QALP score

- $defects = \dots QALP(-0.12 - 0.0023 SLoC) \dots$
- “Good” when coefficient of QALP < 0

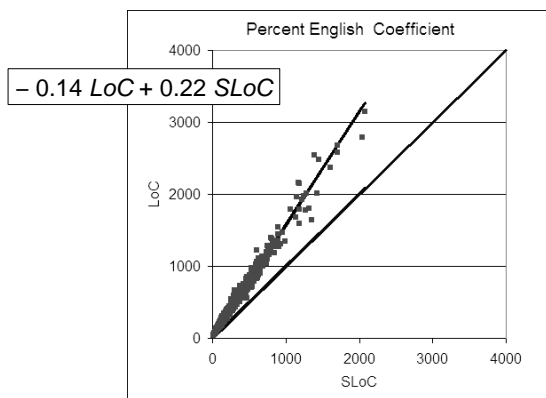
MP Final Model

- $defects = - 2.2$
 - 0.11 LoC
 - + 0.13 $SLoC$
 - + $QALP(-76 + 0.8 LoC - 1.2 SLoC + 255 v)$
 - + $PNL(1 - 0.14 LoC + 0.22 SLoC)$
 - + $v(-0.22 - 0.038 LoC)$
- Again interactions exist

MP Model Extract for Percent Natural Language

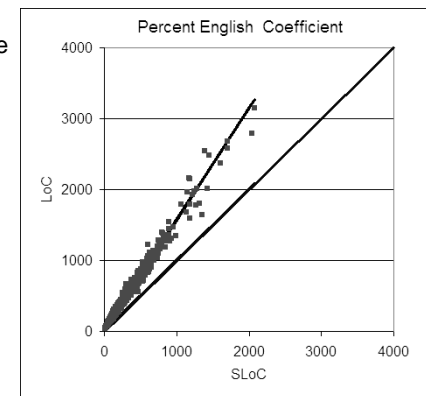
- $defects = \dots PNL(1 - 0.14 LoC + 0.22 SLoC) \dots$
- “Good” when coefficient of $PNL < 0$

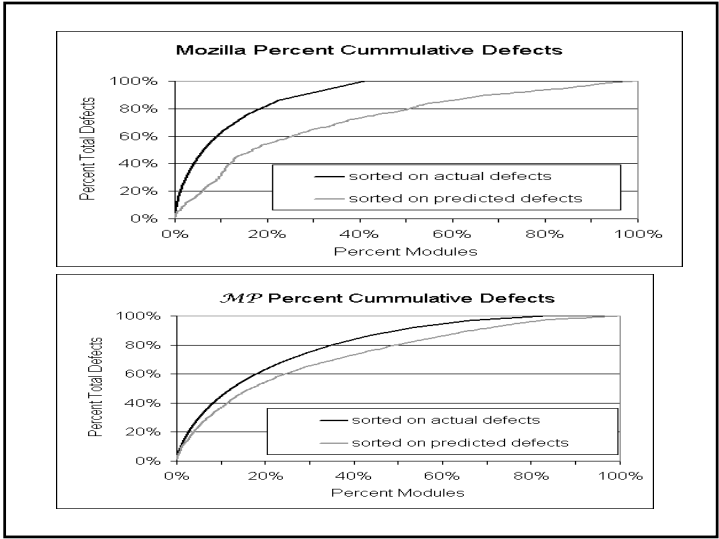
Percent Natural Language Visual Analysis



Percent Natural Language Visual Analysis

- Good == points are above the line
- 63% do so

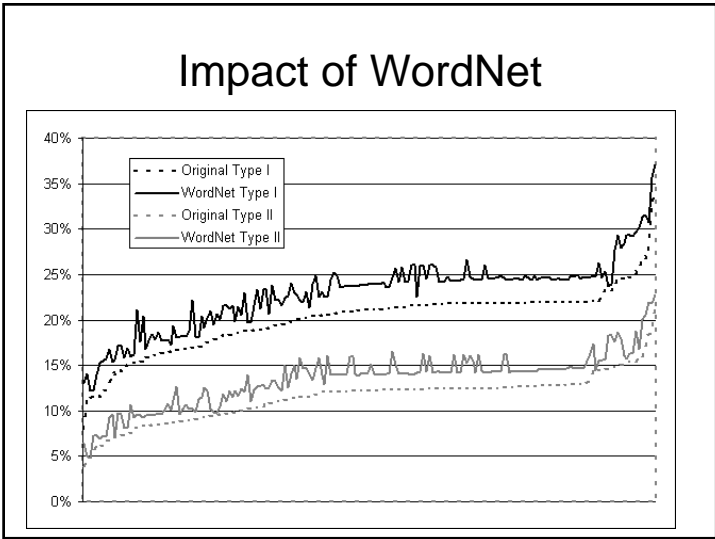




(pre summary)
Measure Correlations

Correlation Matrix	LoC (SLoC similar)	QALP	Percent Natural Language	C&C Violations
QALP	0.17	-	0.12	0.02
Percent Natural Language	0.24	0.12	-	0.07
C&C Violations	0.27	0.02	0.07	-

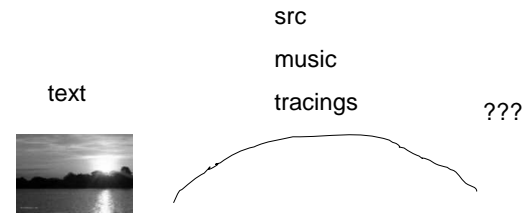
- ### Summary
- Little Picture
 - IR metric's bring Diversity to Fault Prediction
 - Big Picture
 - Solutions to SE problems can exploit IR techniques



Object Lifetime

- low $tf \cdot idf$ == long lived?

Meta Clustering Question



Question?