INTT: Identifier Name Tokenisation Tool
v0.2.0

Simon Butler
Centre for Research in Computing, Department of Computing
The Open University, Milton Keynes, United Kingdom

Abstract

Identifier names are the main vehicle for semantic information during program comprehension. For tool-supported program comprehension tasks, including concept location and requirements traceability, identifier names need to be tokenised into their semantic constituents.

We present INTT, a Java library that implements an approach to the automated tokenisation of identifier names which improves on existing techniques in two ways. First, it improves the tokenisation accuracy for single-case identifier names and for identifier names containing digits, which existing techniques largely ignore. Second, performance gains over existing techniques are achieved using smaller oracles, making the approach easier to deploy.

Our tokenisation library and the datasets used for its evaluation are made available in this package. Also included is a database of unique identifier names extracted from the 60 Java projects, as a resource for further research on program comprehension.

1 Introduction

INTT is a Java library for tokenising identifier names. INTT improves on existing methods for tokenising identifier names composed of characters of a single case or containing digits, in terms of accuracy and the size of the oracles used. The API supports a wide range of configuration options including the use of custom oracles.

INTT improves on existing techniques for tokenising single case identifier names through a reduction in *oversplitting* – the tendency to divide words and compounds further than required – and a simple approach to the resolution of ambiguous tokenisations. For example, there are a number of possible tokenisations of `thenewestone` that consist of dictionary words.

Previous approaches to tokenising identifier names containing digits have, at best, treated digits as separate tokens. While such an approach can lead to accurate tokenisation in some cases, semantic information is lost when known abbreviations – e.g. `3D` – are divided. INTT employs an oracle of abbreviations containing digits and a set of heuristics to support the tokenisation of identifier names containing digits.

A detailed explanation of INTT and the algorithms used can be found in 'Improving Identifier Name Tokenisation' by S. Butler, M. Wermelinger, Y. Yu and H. Sharp, in Proc. ECOOP 2011, available at `http://oro.open.ac.uk/25656`.

This release of INTT is version 0.2.0.

2 Copyright and Licence

INTT is Copyright ©2011 The Open University and is licensed for use under the following conditions:

1. INTT may be incorporated in binary form into non-commercial software.
2. INTT may be redistributed only as part of another software package and must be accompanied by a copy of this licence.
3. The original authors should be attributed and The Open University's copyright must be acknowledged in any software and accompanying documentation.
4. Disassembly and reverse-engineering are prohibited.
5. Disclaimer: The INTT package is provided 'as is' and no warranty is expressed or implied concerning its fitness or merchantability. The copyright holders and contributors shall not be liable for any losses incurred through use of the software.

For other licences, please contact the author.

3 Installation and Use

The file intt.jar was compiled with Java v1.6 and has no external dependencies. The file should be placed in the classpath of the application into which it is being incorporated. The Javadocs describing the INTT API are provided in a separate directory in the distribution ZIP file and can be copied to a convenient location for viewing.

3.1 Features

INTT is designed to tokenise identifier names composed of English language words and abbreviations. It incorporates dictionaries to support the tokenisation where abbreviations containing digits are found, and to support the tokenisation of single case identifier names, i.e. identifier names without word boundaries marked by separator characters or internal capitalisation. Each dictionary can be replaced using the API and thus INTT may be configured to tokenise identifier names in languages other than English. See the Javadocs for full details.

3.2 Additional Materials

Also distributed with INTT are the test sets of identifier names used to evaluate the tool, an example program, and a database of identifier names extracted from 60 open source Java projects.

Test sets Seven test sets of 4,000 identifier names each, extracted at random from the database, are provided in the folder 'tests'. The test sets are available in two forms: a 'plain' file consisting of a list of identifier names, and a 'reference' form containing the same identifier names with a reference tokenisation. Each line of the reference file consists of an identifier name followed by a tab character and the reference tokenisation with lower case tokens separated by hyphens, e.g. `AClassName`⟨*tab*⟩`a-class-name`.

Example program The example program is a simple demonstration of instantiating and using the tokeniser.

To compile the example program:

```
javac -cp intt.jar Example.java
```

To run the example program:

```
java -cp intt.jar:.  Example inputfile.name
```

The input file for the example program should be one of the plain format test sets, or any other list of identifier names.

Database A database of identifier names is provided as a set of files, one for each project analysed, with one pair of the form $\langle species\text{-}name \rangle \langle space \rangle \langle identifier\text{-}name \rangle$ per line. The term *species* is used to identify the role an identifier name plays in the programming language, such as a class or method name. A list of the projects analysed can be found in the file java-projects.csv.

4 Future Development

INTT is part of a larger research project and is under active development. The most recent version of INTT is always available at `http://oro.open.ac.uk/28352`

5 Contact

Simon Butler (email: `simon@facetus.org.uk`) is the contact for all enquiries concerning INTT, including defect reports.