

Image Text Enhancer

Algorithm Engineering 2026 Project Paper

Daniel Motz
Friedrich Schiller University Jena
Germany
daniel.motz@uni-jena.de

Leonard Teschner
Friedrich Schiller University Jena
Germany
leonard.teschner@uni-jena.de

Mher Mnatsakanyan
Friedrich Schiller University Jena
Germany
mher.mnatsakanyan@uni-jena.de

Abstract

The five-finger pattern [?]:

- (1) **Topic and background:** What topic does the paper deal with? What is the point of departure for your research? Why are you studying this now?
- (2) **Focus:** What is your research question? What are you studying precisely?
- (3) **Method:** What did you do?
- (4) **Key findings:** What did you discover?
- (5) **Conclusions or implications:** What do these findings mean? What broader issues do they speak to?

Keywords

entity resolution, data cleansing, programming contest

1 Introduction

1.1 Background

1.2 Related Work

1.3 Our Contributions

1.4 Outline

2 The Algorithm

2.1 Internal Representation of Mock Labels

In Figure 1 we convert the mock labels to sorted integer sets.

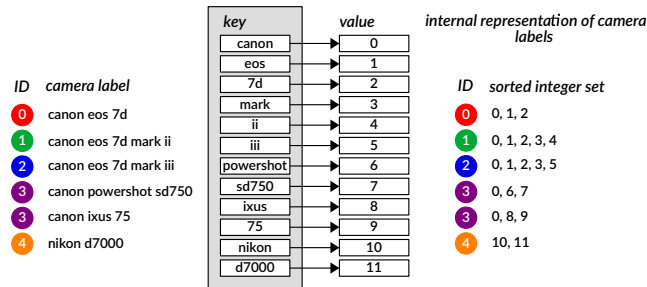


Figure 1: Conversion of mock camera labels to sorted integer sets. We map each unique token (key) in camera labels to a unique value. Based on these key-value-mappings, we convert camera labels to sorted integer sets. A camera can have different names in different countries. Therefore, repeating IDs reference the same cameras (see, for example, ID=3).

2.2 Efficient Preprocessing of Input Data

The following findings are important to speed up preprocessing of the input data:

- Reading many small files concurrently, with multiple threads (compared to a single thread), takes advantage of the internal parallelism of SSDs and thus leads to higher throughput [?].
- C-string manipulation functions are often significantly faster than their C++ pendants. For example, locating substrings with `strstr` is around five times faster than using the C++ `std::string` function `find`.
- Hardcoding regular expressions with `while`, `for`, `switch` or `if-else` statements results in faster execution times than using standard RegEx libraries, where regular expressions are compiled at runtime into state machines.
- Changing strings in place, instead of treating them as immutable objects, eliminates allocation and copying overhead.

3 Experiments

Table 1 shows the running times of the resolution step of the five best placed teams.

Table 1: Comparison of the F-measure and the running times of the resolution step of the five best placed teams. The input data for the resolution step consisted of 29,787 in JSON formatted e-commerce websites. Measurements were taken on a laptop running Ubuntu 19.04 with 16 GB of RAM and two Intel Core i5-4310U CPUs. The underlying SSD was a 500 GB 860 EVO mSATA. We cleared the page cache, dentries, and inodes before each run to avoid reading the input data from RAM instead of the SSD.

Team	Language	F-measure	Running time (s)
PictureMe (this paper)	C++	0.99	0.61
DBGGroup@UniMoRe	Python	0.99	10.65
DBGGroup@SUSTech	C++	0.99	22.13
eats_shoots_and_leaves	Python	0.99	28.66
DBTHU	Python	0.99	63.21

4 Conclusions