Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

96

James L. Peterson

Computer Programs for Spelling Correction:

An Experiment in Program Design

Springer-Verlag
Berlin Heidelberg New York 1980

Author

James L. Peterson
The Department of Computer Sciences
The University of Texas
Austin, Texas 78712/USA

AMS Subject Classifications (1979): 68A30
CR Subject Classifications (1974): 5.2, 3.42

ISBN 3-540-10259-0 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-10259-0 Springer-Verlag New York Heidelberg Berlin

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2145/3140-543210

PREFACE

"Anyone who can spell in English can't be very bright."

– George Bernard Shaw

The automatic detection and correction of spelling errors by computers has been a subject of interest for a long time. (Our literature search revealed work as early as 1957.) There have been several papers investigating various algorithms and showing their application to various tasks, generally data entry. Now, however, with the increased interest in computer based text processing (word processing) and the storage of large amounts of textual information in computers (data bases), we suggest that spelling correction will become commonplace. This volume brings together the diverse and scattered work on this topic and shows how it can be applied to create a real general purpose spelling corrector.

The theory behind spelling error detection and correction is only one of the considerations of this volume, however. More generally, we consider the design of the program to implement that theory. This work is an *experiment in program design.* Our objective is to demonstrate the application of modern program design principles on a large real (non-toy) program. The approach taken is:

1. A complete literature search provides the background information upon which our program will be based, introducing the problems and solutions which have been considered before.

2. Using our new familiarity with the subject, a complete, top-down program design will be created.

3. Finally, the design will be implemented by converting it into working code. The complete working program is given in Appendix III.

The purpose of this project is as much to investigate and demonstrate modern programming principles as to produce a spelling corrector. We feel that this project can be used as an example of the design of a large program. As such, it is suitable for study.

Consider using this volume in the classroom. The topic appeals to students; it is new, modern and has the appearance of intelligence. The simplest spelling error detector is an exercise in data structures, but it can be expanded and developed to illustrate many topics (disk I/0, interactive programming, data structures, performance, design, ...). Our experience has shown that it can be a very satisfactory programming project. Combining such a project with the reading of this volume provides both the experience and the example needed to motivate modern programming principles.

The program of Appendix III is available in machine readable form from the author (for a minimal cost to cover the computer time for preparing a copy of the program and shipping charges) for educational purposes.

James L. Peterson
Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712

Acknowledgements – Discussions with William B. Ackerman, Robert Amsler, Michael Conner and Leigh Power helped to understand the problems of spelling programs. Jeanne Peterson assisted with proofreading and defining the style of this volume. Art Rinn, Emmett Griner, Mark McCulloch, Barney McCartney and the staff of Texas Student Publications aided in the preparation of the typeset copy. I greatly appreciate their help and time.

TABLE OF CONTENTS

Part I Background

Part II Design of a Spelling Program