

Chapter 8

Practical Programming for NLP

Patrick Jeuniaux
Université Laval, Canada

Andrew Olney
The University of Memphis, USA

Sidney D'Mello
The University of Memphis, USA

ABSTRACT

This chapter is aimed at students and researchers who are eager to learn about practical programmatic solutions to natural language processing (NLP) problems. In addition to introducing the readers to programming basics, programming tools, and complete programs, we also hope to pique their interest to actively explore the broad and fascinating field of automatic natural language processing. Part I introduces programming basics and the Python programming language. Part II takes a step by step approach in illustrating the development of a program to solve a NLP problem. Part III provides some hints to help readers initiate their own NLP programming projects.

INTRODUCTION

Natural language processing (NLP) attempts to automatically analyze the languages spoken by humans (i.e., natural languages). For instance, you can program a computer to automatically identify the language of a text, extract the grammatical structure of sentences, categorize texts by genre (e.g., decide whether a text is a scientific or a narrative text), summarize a text, etc. This chapter is aimed at teaching specialized, yet introductory, programming skills that are required to use avail-

able NLP tools. We hope that this chapter serves as a catalyst to launch NLP projects by motivating novice programmers to learn more about programming and encouraging more advanced programmers to develop NLP programs. The chapter is aimed at readers from the interdisciplinary arena that encompasses computer science, cognitive psychology, and linguistics. It is geared for individuals who have a practical NLP problem and for curious readers who are eager to learn about practical solutions for such problems.

Fortifying students with the requisite programming skills to tackle an NLP problem in a single chapter is a daunting task for two primary

DOI: 10.4018/978-1-60960-741-8.ch008

reasons. First, along with advanced statistics, programming is probably the most intimidating task that practitioners in disciplines like linguistics or cognitive psychology can undertake. The typical student or researcher in these fields has little formal training in mathematics, logic, and computer science, hence, their first foray into programming can be a bit challenging. Second, although computer scientists have considerable experience with programming and have mastered many computer technologies, they might not be privy to the libraries or packages that are readily and freely available for NLP projects. In other words, there is a lot to cover if we attempt to address both these audiences, and it seems like an impossible challenge to design a chapter extending from the basics of programming to the specifics of NLP. Fortunately, for the reader and us, the availability of state-of-the-art NLP technologies and the enhanced usability available through easy-to-use interfaces alleviates some of these challenges.

Because of space limitations, we could not achieve the coverage depth we had hoped for. We originally had planned to include programming projects in several languages such as Python, Perl, Java and PHP, along with numerous screen captures of captivating programming demonstrations. The chapter is now more focused on examples in Python. Fortunately, the materials that could not be included in the chapter (e.g., scripts, examples, screen captures), are available for your convenience on the companion website at <http://patrickjeuniaux.info/NLPchapter>. It also provides a series of links to NLP resources, as well as detailed instructions about how to execute the programs that are needed for the exercises. A great advantage of having a website is that it can be updated with current content, so do not hesitate to contact us if you wish to give us feedback.

This chapter has three parts. Part I offers an introduction to programming. Part II gives a concrete example of programming for a specific NLP project. Part III provides general hints about starting your own NLP programming project.

Readers who do not have programming experience or who do not know Python should definitely start with Part I. Individuals who have a working knowledge of Python can skip most of Part I. Among these people, the ones who do not know about NLTK could limit their reading of Part I to the section on functions and onwards. Although Part I covers a lot of material, the topic coverage is far from exhaustive. When you are done with this chapter, we encourage you to read a more complete introduction. We particularly recommend Elkner, Downey, and Meyers (2009). The same can be said of Part II. We also recommend reading Bird, Klein and Loper (2009), who give a thorough treatment of NLP programming with Python's Natural Language Processing Toolkit (NLTK).

PROGRAMMING BASICS

Computers are controlled by sets of instructions called programs. Because they are somewhat simple machines, computers can only follow the most unambiguous instructions. To achieve this ideal of precision, a program is written in a restricted language. Programming languages use a specific vocabulary (i.e. a set of words), a syntax (i.e., a set of rules defining how to use these words), and semantics (i.e., the meaning of the words and rules from a programming point of view). Learning the basic rules of a language is the first step towards writing meaningful and useful programs. But prior to learning a language, it might be good to learn about the history of computer programming. Knowing the historical motivation behind programming will help you grasp what programming is all about.

Programming in Context

Today we usually think of a computer as a general purpose device. However, this was not always the case. Whether you consider Babbage's Difference

33 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/practical-programming-nlp/61046

Related Content

Background Review for Neural Trust and Multi-Agent System

Gehao Lu and Joan Lu (2020). *Natural Language Processing: Concepts, Methodologies, Tools, and Applications* (pp. 1-22).

www.irma-international.org/chapter/background-review-for-neural-trust-and-multi-agent-system/239926

Building Awareness of Language Structures with Visual-Syntactic Text Formatting

Youngmin Park, Mark Warschauer, Penelope Collins, Jin Kyoung Hwang and Charles Vogel (2014). *Computational Linguistics: Concepts, Methodologies, Tools, and Applications* (pp. 640-656).

www.irma-international.org/chapter/building-awareness-of-language-structures-with-visual-syntactic-text-formatting/108743

Leadership and the Function of Discourse on International Political Economy

Pinar Altıok Gürel, Recep Yılmaz, Nevin Özgür Cambaz and Nurten Yılmaz (2014). *Computational Linguistics: Concepts, Methodologies, Tools, and Applications* (pp. 132-145).

www.irma-international.org/chapter/leadership-and-the-function-of-discourse-on-international-political-economy/108718

Neuromorphic Speech Processing: Objectives and Methods

Pedro Gómez-Vilda, José Manuel Ferrández-Vicente, Victoria Rodellar-Biarge, Rafael Martínez-Olalla, Víctor Nieto-Lluis, Luis Mazaira-Fernández and Cristina Miguel Muñoz-Mulas (2011). *Machine Audition: Principles, Algorithms and Systems* (pp. 447-473).

www.irma-international.org/chapter/neuromorphic-speech-processing/45497

The Language of Architecture

(2020). *Grammatical and Syntactical Approaches in Architecture: Emerging Research and Opportunities* (pp. 1-17).

www.irma-international.org/chapter/the-language-of-architecture/245857