Using NLTK for educational and scientific purposes

Mykhailo Lobur, Andriy Romanyuk, Mariana Romanyshyn

Abstract – **This paper deals with the importance of Natural Language Toolkit for the course of Computational Linguistics and for scientific research in the field of natural language processing. Peculiarities of Python programming language, used in Natural Language Toolkit, are described. The specific experience of studying Natural Language Toolkit in the course of Computational Linguistics is considered.**

Keywords – **Natural Language Toolkit, Python, Computational Linguistics, natural language processing.**

I. INTRODUCTION

Natural language processing (NLP) is an extremely active field of research, which attracts a lot of students every year. It gives the possibility to study human language from the applied side, not just theoretically, and to try to solve some of the tasks considering human language.

However, the most important reason for choosing NLP as an area of study, is the variety of very interesting problems with no general solutions. For example, the original problem of machine translation still remains one of the hardest to solve, even after twenty years of discussing and active research.

Python and the Natural Language Toolkit (NLTK) allow any programmer, even a beginner, to get acquainted with NLP tasks easily without spending too much time on studying or gathering resources. The aim of this paper is to provide valuable proof and examples, which show how necessary the NLTK is for the course of Computational Linguistics at the university and for researchers in the field of natural language processing.

II. NLTK AND PYTHON AS TOOLS FOR NLP

Computational Linguistics is an interdisciplinary field dealing with the modeling of human language. In a nutshell, it is a scientific study of language from a computational perspective. Computational Linguistics is interested in providing computational models of various kinds of linguistic phenomena. Indeed, the work of computational linguists is incorporated into many working systems today, including speech recognition systems, text-to-speech synthesizers, automated voice response systems, web search engines, and text editors, to name just a few [3].

Thus, taking into account the importance of the natural language processing tasks in Computational Linguistics for our time, students should study the course of Computational Linguistics from its 'applied' side. To help students in this rather difficult task NLTK and Python are used.

Natural Language Toolkit was developed in conjunction with a Computational Linguistics course at the University of Pennsylvania in 2001 [2]. It is a collection of modules and corpora, released under an open-source license, which allows students to learn and conduct research in NLP. NLTK can be used not only as a training complex, but also as a ready analytical tool or basis for the development of applied text processing systems. Nowadays it is widely used in linguistics, artificial intelligence, machine learning projects, etc.

There are a lot of advantages of using NLTK. The most important one is that it is entirely self-contained. Not only does it provide raw and annotated versions of real-world data in the form of 60 corpora, grammar collections, and trained models, it also provides convenient functions that can be used as building blocks for common NLP tasks.

Table 1 includes the list of NLTK modules with the corresponding natural language processing tasks [1].

Among already mentioned advantages of NLTK, most corpora are divided into sections to make it easy and comfortable for users to exploit them. Other important thing is that NLTK is integrated with WordNet, which is a database of semantic relationships for English nouns, verbs, adjectives and adverbs (synonymy, hyperonymy, hyponymy, holonymy and meronymy). NLTK provides frequency and conditional frequency tools, plotting tool, and the most important thing is that everything is accessible after single import. NLTK includes detailed manual, which appears to be extremely useful for beginners. All modules and corpora are provided with distributions for Windows, Mac OSX and Linux on the NLTK site.

NLTK is written in Python, so a short introduction to Python programming is needed. Of course, Python is not the only programming language in the world used to solve natural language processing problems, but it possesses a number of advantages over the other programming languages. These are:

- high readability;
- an easy to use object-oriented paradigm;
- easy extensibility;
- strong Unicode support;
- a powerful standard library.

Python is a simple yet powerful programming language with excellent functionality for processing linguistic data and it can be downloaded for free. Python is heavily used in industry, scientific research, and education around the world.

Python and NLTK are efficient tools, which provide practical introduction to natural language processing and its problems.

Mykhailo Lobur: professor, Head of CAD Department, Lviv Polytechnic National University, 12, S. Bandery Str., Lviv, 79013, Ukraine; e-mail: mlobur@polynet.lviv.ua

Andriy Romanyuk: associate professor, CAD Department, Lviv Polytechnic National University, 12, S. Bandery Str., Lviv, 79013, Ukraine; e-mail: anrom@polynet.lviv.ua

Mariana Romanyshyn: the 1st year postgraduate, CAD Department, Lviv Polytechnic National University, 12, S. Bandery Str., Lviv, 79013, Ukraine; e-mail: mariana.scorp@gmail.com

TABLE 1

LANGUAGE PROCESSING TASKS AND CORRESPONDING NLTK MODULES WITH EXAMPLES OF FUNCTIONALITY

Language processing task	**NLTK modules**	**Functionality**
Accessing corpora	nltk.corpus	standardized interfaces to corpora and lexicons
String processing	nltk.tokenize, nltk.stem	tokenizers, sentence tokenizers, stemmers
Collocation discovery	nltk.collocations	t-test, chi-squared, point-wise mutual information
Part-of-speech tagging	nltk.tag	n-gram, backoff, Brill, HMM, TnT
Classification	nltk.classify, nltk.cluster	decision tree, maximum entropy, naive Bayes, EM, k-means
Chunking	nltk.chunk	regular expression, n-gram, named-entity
Parsing	nltk.parse	chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	nltk.sem, nltk.inference	lambda calculus, first-order logic, model checking
Evaluation metrics	nltk.metrics	precision, recall, agreement coefficients
Probability and estimation	nltk.probability	frequency distributions, smoothed probability distributions
Applications	nltk.app, nltk.chat	graphical concordancer, parsers, WordNet browser, chatbots
Linguistic fieldwork	nltk.toolbox	manipulate data in SIL Toolbox format

III. SPECIFIC EXPERIENCE IN USING NLTK

NLTK can be studied within many different courses: natural language processing, computational linguistics, empirical linguistics, artificial intelligence, information retrieval, machine learning, etc. A lot of universities in the countries all over the world, including USA, Canada, Australia, France, Germany, UK, Spain, and Poland, use NLTK in their courses. The whole list of institutions that use NLTK in their courses can be viewed in [4].

Students of the Department of Applied Linguistics of Lviv Polytechnic National University have the course of Computational Linguistics on the fifth year of their studies (both master students and specialists). During this course they learn to use NLTK and acquire the basics of programming in Python, using *Natural Language Processing with Python* by Steven Bird et al [1] as a guide.

Among other things, students learn to:

- work with different variable types;
- access text corpora and lexical resources;
- process raw text (normalize, tokenize, etc.);
- discover part-of-speech tags;
- use regular expressions;
- use tagging, stemming and chunking;
- work with context-free and feature-based grammars.

Moreover, students acquire skills of writing structured programmes. This book has been chosen to be a textbook for the course of Computational Linguistics because it provides a highly accessible introduction to the field of NLP. The book is intensely practical, containing hundreds of fully-worked examples and graded exercises. This book is unique in providing a comprehensive framework for students to learn about NLP in the context of learning to program.

Table 2 shows the approximate number of lectures necessary to cover the material in the book [1]. As one can see a two-semester course is enough to obtain the main skills of solving natural language processing tasks.

TABLE 2

APPROXIMATE NUMBER OF LECTURES PER CHAPTER

Chapter	**Number of lectures**
1 Language Processing and Python	2-4
2 Accessing Text Corpora and Lexical Resources	2-4

Chapter	Number of lectures
3 Processing Raw Text	2-4
4 Writing Structured Programs	2-4
5 Categorizing and Tagging Words	2-4
6 Learning to Classify Text	0-2
7 Extracting Information from Text	2
8 Analyzing Sentence Structure	2-4
9 Building Feature Based Grammars	2-4
10 Analyzing the Meaning of Sentences	1-2
11 Managing Linguistic Data	1-2
Total	18-36

In this way students learn a lot about natural language processing tasks and ways to solve them. A lot of students choose the topic of their master paper from the field of Computational Linguistics. The knowledge, gained during the course, enables them to develop programmes for their master papers, which makes their work valuable from the applied point.

There are several cutting-edge areas in NLP that currently invoke a large amount of research activity and thus can be chosen by students:

1. Syntax-based machine translation: For the past ten years, most of the research in machine translation has focused on using statistical methods on very large corpora to learn translations of words and phrases. However, more and more researchers are starting to include syntax and semantics into such methods. In this way machine translation becomes more efficient, though, this area is still under development.
2. Automatic multi-document text summarization: There have been a lot of of efforts underway to use computers to automatically generate coherent and informative summaries. This task is considerably more difficult compared to generating a summary for a single document, because there may be redundant information present across multiple documents [5].
3. Automatic syntactic analysis: This phase of NLP determines semantic interpretation in the form of predicate-argument structures, dependency relations or logical form representations. Traditionally, deep, wide-coverage linguistic resources are handcrafted and their creation is very labour- and cost-intensive. Students can work upon the problem of providing the automatically formed syntactic structures.
4. Automatic semantic analysis: Extracting and describing the meaning contained in the linguistic unit is the most difficult to implement and poorly studied phase of NLP. Students can build their own algorithms or develop the existing ones, using various linguistic theories as a guide.
5. Lexicon obtaining: A lot of NLP tasks need a well-formed natural language lexicon. Students can analyze and process different lexical resources, thesauruses and ontologies for this purpose.

IV. Conclusion

Natural Language Toolkit is a convenient tool for solving natural language processing tasks, widely used all over the world for scientific research and studying due to a variety of modules and corpora. It allows students with no previous programming experience to spend more of their time thinking about the logical steps involved in getting the computer to process language data, and less time mastering and getting the computer to do anything at all. It is thoroughly documented, easy to learn, and simple to use.

In this paper the specific experience of using Natural Language Toolkit for the course of Computational Linguistics in Lviv Polytechnic National University has been described. As a result, a lot of students managed to gain good programming skills and chose the area of natural language processing for their master papers. The main areas students can work upon are listed.

NLTK is undergoing continual development as new modules are added and existing ones are improved.

References

[1] Bird S. Natural Language Processing with Python / S. Bird, E. Klein, E. Loper. – Sebastopol: O'Reilly Media, Inc., 2009. – 504 p.

[2] Bird S. Proceedings of the ACL demonstration session / S. Bird, E. Loper // Barcelona, Association for Computational Linguistics. – 2004. – pp 214-217.

[3] Jurafsky D. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition / D. Jurafsky, J.H. Martin. – Upper Saddle River, NJ: Prentice Hall, 2008. – 988 p. – 2nd edition.

[4] List of NLP/CL courses. Available from: http://aclweb.org/aclwiki/index.php?title=List_of_NLP/CL_courses

[5] Madnani N. Getting Started on Natural Language Processing with Python / N. Madnani // Crossroads, The ACM Student Magazine. – 2007. – vol. 13(4).