УДК 004.89

СРАВНИТЕЛЬНЫЙ АНАЛИЗ БИБЛИОТЕК ДЛЯ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА (NLP)

© 2024 К. С. Макаров[1], А. А. Халин[2], Д. А. Костенков[3], Э. Э. Муханов[4]

[1]кандидат технических наук, доцент кафедры программного обеспечения и администрирования информационных систем
e-mail: runaway90@mail.ru
[2]кандидат физико-математических наук, доцент кафедры программного обеспечения и администрирования информационных систем
e-mail: khalin_andrey@rambler.ru
[3]студент 2 курса магистратуры направления подготовки «Информатика и вычислительная техника»
e-mail: daniil.koctenkob@gmail.com
[4]студент 2 курса магистратуры направления подготовки «Информатика и вычислительная техника»
e-mail: muhanov00@mail.ru

Курский государственный университет

В данной работе изложены результаты анализа и сравнения библиотек, написанных для языка Python, используемых для обработки естественного языка. Раскрыта важность проблемы выбора технологий для разбора семантики текста. Изложен сравнительный обзор особенностей и специализаций библиотек, рассмотрены недостатки и достоинства каждой из них, а также области их применения.

Ключевые слова: обработка естественного языка, семантика текста, NLP библиотеки.

COMPARATIVE ANALYSIS OF NATURAL LANGUAGE PROCESSING (NLP) LIBRARIES

© 2024 K. S. Makarov[1], A. A. Khalin[2], D. A. Kostenkov[3], E. E. Mukhanov[4]

[1]Candidate of Engineering Sciences, Associate Professor, Department of Software and Administration information systems
e-mail: runaway90@mail.ru
[2]Candidate of Physico-Mathematical Sciences (Ph.D. in Physics and Mathematics), Associate Professor, Department of Software and Administration information systems
email: khalin_andrey@rambler.ru
[3]2nd year master's student in the field of study "Informatics and Computer Science"
e-mail: daniil.koctenkob@gmail.com
[3]2nd year master's student in the field of study "Informatics and Computer Science"
e-mail: muhanov00@mail.ru

Kursk State University

This paper presents the results of the analysis and comparison of libraries written for the Python language used for natural language processing. The importance of the problem of choosing technologies for parsing the semantics of a text is revealed. A comparative overview of the features and specializations of libraries is presented, the disadvantages and advantages of each of them, as well as their areas of application are considered.

Keywords: natural language processing, text semantics, NLP libraries.

Обработка естественного языка (англ. Natural Language Processing, NLP) – это область компьютерных наук и искусственного интеллекта, находящаяся на стыке информатики и лингвистики, которая занимается взаимодействием между ЭВМ и человеческим языком. NLP включает в себя различные методы анализа, понимания и генерации текста на естественных языках [3].

NLP имеет огромное значение в современном мире, так как применяется в следующих областях: машинный перевод, автоматическая обработка текста, автоматическое извлечение информации, анализ настроений в социальных медиа и многое другое [3].

Обработка естественного языка включает в себя решение различных задач, таких как распознавание речи, генерация естественного языка, определение смысла слов, анализ эмоциональной окраски текста, определение перекрестных ссылок, распознавание именованных сущностей [5].

Для решения этих задач применяются различные технологии, включая методы машинного обучения, в частности глубокое обучение, статистический анализ и обработка больших данных [4].

С постоянным развитием компьютерных технологий и искусственного интеллекта, обработка естественного языка будет продолжать играть важную роль в различных сферах, таких как коммуникации, бизнес, здравоохранение, исследования и многое другое [5].

Выбор NLP библиотеки является одним из ключевых этапов решения задач обработки естественного языка наряду с построением архитектуры системы и выбором языка программирования. В первую очередь нужно определиться, какие именно задачи необходимо решить и какие инструменты для это необходимы. Использовать узкоспециализированные библиотеки или подобрать те, которые позволяют решать большой спектр задач. От этого выбора будут зависеть качество обработки данных, скорость разработки и быстродействие всей системы [4].

Целью данной статьи является обзор и сравнительный анализ наиболее популярных библиотек для языка программирования Python, предназначенных для обработки естественного языка.

При работе с большими объемами данных критически важно обеспечить высокую производительность, а именно, система должна выполнять необходимые операции за минимальное время. Тем не менее также важно обеспечить простоту работы с модулями библиотеки, чтобы ускорить и упростить процесс разработки. Разные библиотеки предлагают различные подходы к обработке естественного языка. Помимо общего функционала, они содержат свои, узкоспециализированные модели хранения и обработки данных, что напрямую влияет на время выполнения операций и общую производительность системы.

Выявление этих особенностей, а также корреляция между полученными результатами и возможными областями применения той или иной библиотеки являются основными задачами данной статьи.

Язык программирования Python широко используется для разработки в области обработки естественного языка (NLP) по нескольким причинам [1]:

Простота и удобство использования. Python предлагает лаконичный и понятный синтаксис, что позволяет разработчикам оперативно реализовывать и тестировать алгоритмы [1].

Широкая экосистема библиотек. Для языка Python было создано множество библиотек, специализирующихся на обработке естественного языка, таких как NLTK, Polyglot, SpaCy, Gensim, Stanford CoreNLP, TextBlo и др. Эти библиотеки предоставляют широкий спектр инструментов для анализа текста, обработки данных и применения методов машинного обучения, что упрощает разработку и исследования в этой области.

Машинное обучение и глубокое обучение. Python имеет мощные библиотеки для машинного и глубокого обучения, такие как scikit-learn, TensorFlow, Keras, PyTorch. Это делает его предпочтительным для разработки в области NLP.

Интеграция с другими технологиями. Python легко можно интегрировать с другими технологиями и инструментами. Это позволяет использовать его в различных проектах NLP, включая веб-разработку, базы данных, анализ данных и визуализацию, обеспечивая разнообразные возможности и решения для специалистов.

Для сравнения были взяты три NLP библиотеки: NLTK (рис. 1) как одна из наиболее популярных, поддерживаемая и обновляющаяся разработчиками в настоящее время; SpaCy (рис. 2) как одна из относительно новых библиотек, отличающаяся высокой производительностью; Gensim (рис. 3) как менее популярная, узкоспециализированная библиотека, решающая конкретный спектр задач [7].

NLTK – библиотека Python для решения обширного спектра задач обработки естественного языка, первый релиз которой состоялся в 2001 г. Она предоставляет множество различных функций для обработки текстов, включая токенизацию, выделение корней, теги, синтаксический анализ и многие другие вещи, необходимые для создания самых разных систем или подсистем обработки естественного языка [6].

NLTK находит применение для решения многих задач обработки естественного языка, начиная от преобразования текстовых данных до разработки приложений на основе NLP, таких как системы вопросно-ответной обработки, интеллектуальные помощники и чат-боты, системы анализа тональности, автоматические классификаторы текста и т.д. [6]

Благодаря широкому набору инструментов, отсутствию конкретной специализации и обширной документации NLTK подойдет для решения научно-исследовательских задач в области NLP [6].

К недостаткам можно отнести низкую производительность по сравнению с более современными NLP библиотеки. Следовательно, если необходима скорость обработки или работа с большими объемами данных, NLTK будет уступать в этом [6].

SpaCy написана на языке Cython в 2015 г. и представляет собой продвинутый инструмент в области обработки естественного языка. SpaCy сконцентрирована на предоставлении эффективных инструменты для решения конкретной задачи, хотя и с более скромным набором функциональности, чем у NLTK [2].

В целом SpaCy выделяется своей современной архитектурой, высокой производительностью и готовностью к использованию предобученных моделей и лучше подходит для разработчиков, ориентированных на создание готовых решений с применением технологий передачи обучения [2].

Однако следует отметить, что как для NLTK, так и для SpaCy существует сложность в создании собственных моделей. Этот аспект может стать ограничением для решения ряда задач в области NLP.

Gensim – библиотека, начавшая свое развитие в 2008 г., которая используется для решения задач тематического моделирования и обработки естественного языка. Ее основной фокус направлен на выявления семантического сходства между двумя документами путем применения векторного пространственного моделирования и инструментария тематического моделирования [3].

Данная библиотека способна работать с обширными текстовыми коллекциями, что отличает ее от других программных библиотек машинного обучения, ориентированных на обработку данных в оперативной памяти. Библиотека также предоставляет эффективные реализации различных алгоритмов, способствуя повышению скорости обработки, особенно благодаря поддержке многоядерности и оптимизации использования памяти [3].

Однако Gensim – это узкоспециализированный инструмент. Взамен на высокую производительность при векторном моделировании, разработчик лишает себя возможности решать множество задач с помощью одного инструмента.

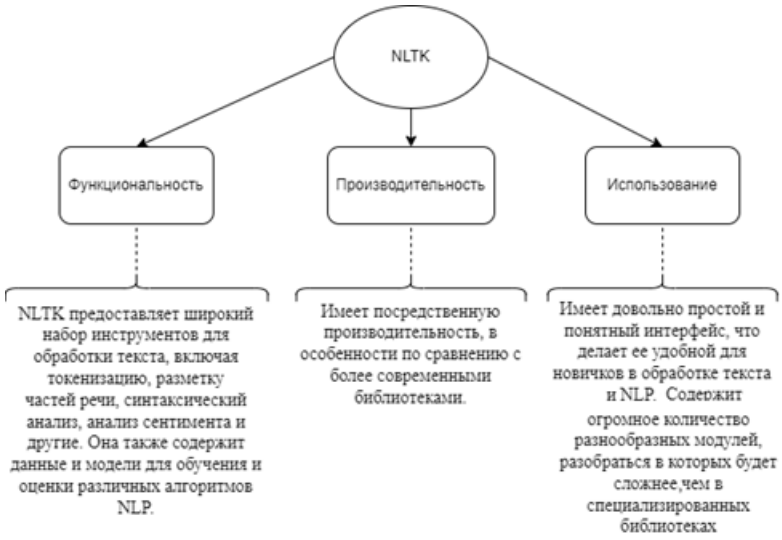

Рис. 1. Особенности библиотеки NLTK

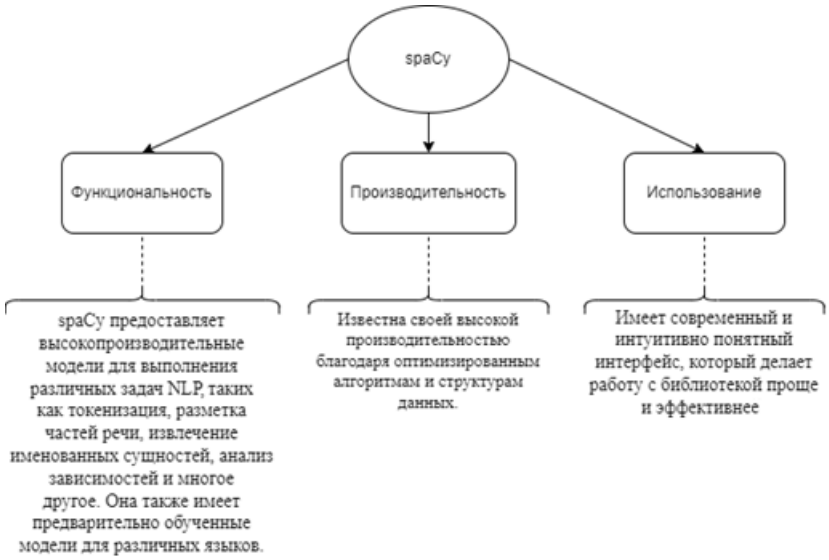

Рис. 2. Особенности библиотеки spaCy

Рис. 3. Особенности библиотеки Gensim

На следующем этапе сравнивалась производительность по времени выполнения, на примере методов векторного представления текста. Для этого были написаны тестовые программы, в которых выполнялись методы соответствующих модулей каждой из библиотек [8]. В таблице 1 приведены результаты выполнения тестов. Для каждого объема данных было сделано 10 проходов, после чего в таблицу занесено среднее значение.

Таблица 1

Время выполнения методов векторного представления текста в рассматриваемых библиотеках

Наименование функции	Объем обрабатываемых данных (лексем)	Среднее время выполнения (с)
SpaCy	100	4,6847E-05
	1000	4,2162E-04
	2500	1,0031E-03
	5000	1,9003E-03
	10000	3,3473E-03
NLTK	100	2,0988E-02
	1000	2,5631E-01
	2500	6,4342E-01
	5000	1,2439E+00
	10000	2,3383E+00
Gensim	100	2,8801E-05
	1000	2,5921E-04
	2500	6,9082E-04
	5000	1,3406E-03
	10000	2,4431E-03

Для определения областей применения библиотек была составлена таблицы 2, в которой указано наличие или отсутствие основных методов NLP в рассматриваемых библиотеках.

Таблица 2

Сравнение функционала NLTK, spaCy и Gensim

Функционал	Библиотека		
	NLTK	spaCy	Gensim
Токенизация	+	+	
Стемминг	+	+	+
Удал. стоп-слов	+	+	+
Частеречная разметка	+	+	
Извлечение именованных сущностей	+	+	
Анализ сентимента	+		
Векторное представление слов	+	+	+
Генерация текста			
Классификация текста	+	+	
Извлечение информации	+	+	
Синтаксический анализ	+	+	
Машинное обучение	+	+	+
Сериализация	+	+	
Создание шаблонов для поиска текста	+		
Создание корпуса и словаря	+	+	+
Индексирование	+	+	+
Моделирование последовательностей	+		+

Из данных, представленных в таблицах 1 и 2, можно сделать вывод о том, что Gensim является наиболее производительной библиотекой для выполнения векторного представления текста, но при этом наиболее ограниченной по своему функционалу, а NLTK – наименее производительная, но с большим функционалом. Результаты по функционалу также подтверждаются информацией, из документации по библиотекам.

Таким образом, были рассмотрены широко используемые NLP библиотеки для языка программирования Python. Проведен сравнительный анализ основных возможностей библиотек и оценена производительность каждой из библиотек на примере методов векторного представления текста.

Библиографический список

1. *Bird, S., Klein, E., & Loper, E.* Обработка естественного языка с помощью Python. O'Reilly Media, 2009.

2. *Bird, S., & Loper, E.* NLTK: Набор инструментов для обработки естественного языка. // Материалы ACL 2004 по интерактивным постерам и демонстрационным сессиям. – 2004. – С. 31–34.

3. *Eisenstein, J.* Введение в обработку естественного языка. Издательство Массачусетского технологического института, 2019.

4. *Honnibal, M., & Montani, I.* spaCy 2: Понимание естественного языка с помощью вложений Блума, сверточных нейронных сетей и инкрементального синтаксического анализа.

5. *Manning, C. D., Raghavan, P., & Schütze, H.* Введение в информационный поиск. Издательство Кембриджского университета, 2008.

6. *Řehůřek, R., & Sojka, P.* Программный каркас для тематического моделирования с большими корпусами // Материалы МКР 2010 Workshop on New Challenges for NLP Frameworks. – 2010. – С. 45–50,

7. Pythonist [Электронный ресурс]. URL: https://pythonist.ru/8-luchshih-bibliotek-obrabotki-estestvennogo-yazyka-dlya-python-nlp/ (дата обращения: 19.12.2023).

8. Репозиторий тестовых программ [Электронный ресурс]. URL: https://github.com/Daniliuk/Test_nlp (дата обращения: 19.12.2023).