DOI 10.37882/2223–2966.2022.11–2.02

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА АНАЛИЗА ТЕКСТА НА ОСНОВЕ NLP

NLP-BASED INTELLIGENT TEXT ANALYSIS SYSTEM

E. Akbasheva
G. Akbasheva
I. Tlupov

Summary. The problem of representing the results of language models and evaluating their quality is understudied. This paper examines the application of language models and NLP techniques to develop an intelligent text analysis system. The idea is to use the capabilities of natural language processing algorithms to analyze semantics in order to demonstrate the possibilities of artificial intelligence for fast analysis of texts of any volume and subject matter. A graph representation of text is used to represent the results of the system.

Keywords: natural language processing, language models, graph, machine learning, BERT model, SpaCy.

Акбашева Евгения Амировна
Старший преподаватель
Кабардино-Балкарский государственный университет
Нальчик
akbash_e@mail.ru
Акбашева Галина Амировна
Старший преподаватель
Кабардино-Балкарский государственный университет
Нальчик
galina_akbash@mail.ru
Тлупов Ислам Заурбекович
Студент
Кабардино-Балкарский государственный университет
Нальчик
tlupovislam@gmail.com

Аннотация. Проблема представления результатов работы языковых моделей и оценка их качества является недостаточно изученной. В данной статье рассматривается применение языковых моделей и методов NLP для разработки интеллектуальной системы анализа текстов. Идея заключается в использовании возможностей алгоритмов естественной обработки языка для анализа семантики, чтобы продемонстрировать возможности искусственного интеллекта для быстрого анализа текстов любых объемов и тематики. Для представления результатов работы системы используется графовое представление текста.

Ключевые слова: обработка естественного языка, языковые модели, граф, машинное обучение, модель BERT, SpaCy.

Введение

Задачи обработки естественного языка (NLP) в настоящее время являются одним из самых востребованных направлений в области искусственного интеллекта и компьютерной лингвистики. Большое количество областей применения алгоритмов, позволяющих обрабатывать и интерпретировать человеческую речь в ее естественном виде, имеет тенденцию увеличиваться в современном мире. Это такие области, как автоматический перевод, оптимизация поисковых запросов, таргетированный маркетинг, распознавание речи и речевых команд, синтез речи, семантический анализ и классификация текста, разработка диалоговых систем и др.

Если задачи определения формальных языков достаточно просты и изучены, то задачи понимания, обработки, генерации и интерпретации естественного языка являются чрезвычайно сложными для реализа-

ции на ЭВМ. Одним из подходов к изучению естественного языка является языковое моделирование. Оно позволяет назначить вероятности словам, фразам или предложениям, а также позволяет предсказать вероятность появления того или иного слова после какого-то заданного. Языковое моделирование является важнейшим компонентом практических приложений, например машинного перевода и автоматического распознавания речи, когда система порождает несколько гипотез о переводе или транскрипции, которые затем оцениваются языковой моделью. Поэтому языковое моделирование играет центральную роль в обработке естественного языка, искусственном интеллекте и исследованиях по машинному обучению.

Однако, проблема представления результатов работы языковых моделей и оценка их качества остается достаточно неизученной. В данной статье рассматривается применение языковых моделей и методов NLP для разработки интеллектуальной системы анализа текстов. Идея заключается в использовании возможностей алгоритмов естественной обработки языка для анализа семантики, чтобы продемонстрировать возможности искусственного интеллекта для быстрого анализа текстов любых объемов и тематики. Разработанная система позволит оценить качество используемой языковой модели для анализа смыслового содержания текста любого объема.

Для представления результатов работы системы используется графовое представление текста. В статье рассматриваются различные типы структур графов, применяемых для представления, приведен их сравнительный анализ.

1. Языковое моделирование

Формально задача языкового моделирования ставится так: назначить вероятность произвольной последовательности слов $w_{1:n}$, т.е. оценить величину $P(w_{1:n})$. Применив цепное правило исчисления вероятностей, мы сможем записать эту вероятность в виде:

$$P(w_{1:n}) = P(w_1)\ P(w_2|w_1)\ P(w3|w_{1:2})\ P(w_4|w_{1:3})\ \ldots\ P(w_n|w_{1:n-1}), \quad (1)$$

т. е. в виде последовательности задач предсказания слова, в которой каждое предсказание обусловлено предшествующими словами. Хотя задача моделирования одного слова на основе его левого контекста кажется проще, чем назначение оценки вероятности всему предложению, последний член в этом равенстве все равно требует обусловливания n — 1 словами, а это такая же трудная задача, как моделирование всего предложения. Поэтому в языковых моделях применяется марковское предположение, утверждающее, что будущее не зависит от прошлого при условии настоящего [1]. Языковые модели, построенные на марковских цепях, стали родоначальниками существующих современных моделей на основе нейронных сетей и носят название статистических моделей языка.

Нейросети могут учитывать большее количество закономерностей в тексте в отличие от марковской цепи, а также гораздо более длинный контекст [2].

Также нейросеть позволяет обрабатывать смысловые отношения, что полностью отсутствуют в статистических моделях.

Одной из первых нейросетевых языковых моделей стала рекуррентная нейросетевая языковая модель (RNNLM), базирующаяся на сжатом векторном представлении слов или эмбеддинге. Данная модель обучается на известном алгоритме представления текста «мешок слов» [3].

Еще одна языковая модель, основывающаяся на векторных представлениях слов, это Word2vec. Word2vec получает на вход большой корпус текста и создает векторное пространство (пространство признаков), как правило, из нескольких сотен измерений, причем каждому уникальному слову в корпусе назначается соответствующий вектор в пространстве. Это делается путем создания пар контекста и целевых слов, что также зависит от размера выбранного контекстного окна. В качестве алгоритма обучения используется Skip-gram [4]. Контекстное окно образуется из нескольких слов, следующих друг за другом. Суть алгоритма в том, что одно из слов намеренно пропускается, и нейросеть должна его предсказать. То есть, слова, встречающиеся часто в идентичном контексте, будут иметь сходные векторы [5].

Развитием Word2vec стала модель GloVe. Алгоритм GloVe [6] строит явную матрицу слово-контекст и обучает вектор слов и контекстов, учитывая совместную встречаемость слов.

В последнее время огромный прогресс в области обработки естественного языка привел к появлению инновационных архитектур языковых моделей, таких как GPT-3 и BERT [7].

BERT, он же Bidirectional Encoder Representations from Transformers,— это предварительно обученная языковая модель, разработанная Google. Модель, предварительно обученная на 2500 миллионах интернет-слов и 800 миллионах слов книжного корпуса, ис-

Таблица 1. Сравнительный анализ некоторых языковых моделей

Языковая модель	Преимущества	Недостатки
Статистическая модель	Простая, не ресурсоемкая	Не учитывает смысловые отношения, короткий контекст
Рекуррентная нейросетевая модель	Простота, быстрое обучение, есть предварительно обученные модели	Не учитывает долгосрочные зависимости
Word2vec	Простая архитектура, быстрое обучение, универсальность	Обучение на уровне слов, не учитывается совместная встречаемость слов, плохо обрабатывает неизвестные слова
GloVe	Простая архитектура без нейронной сети, быстрота, учитывает совместную встречаемость слов, эффективнее Word2Vec	Обучение на уровне слов, плохо обрабатывает неизвестные слова
BERT	С открытым исходным кодом,	Необходима тонкая настройка, требует большого количества данных для обучения
GPT-3	Проста для использования, высокая скорость работы, не требует огромного количества данных для обучения, самая эффективная	Большой размер, коммерческая модель с закрытым исходным кодом

пользует архитектуру на основе трансформеров (один из типов архитектуры нейронных сетей).

Подобно BERT, GPT-3 также является крупномасштабной языковой моделью на основе трансформеров, которая обучена на 175 миллиардах параметров, что в 10 раз больше, чем у предыдущих моделей. Эта модель предсказания языка третьего поколения является авторегрессионной и работает подобно традиционным моделям, где она принимает на вход вектор слов и предсказывает выходные данные на основе своего обучения. Благодаря обучению с учителем и few-shot learning (модель обучения, которая предполагает предварительную настройку модели на тренировочном наборе данных таким образом, чтобы в дальнейшем она успешно обучалась на каком-то определенном количестве новых размеченных данных [8]), эта модель работает в контексте.

С точки зрения архитектуры, в то время как BERT обучается на латентных вызовах отношений между текстами различных контекстов, подход к обучению GPT-3 относительно прост по сравнению с BERT. Поэтому GPT-3 является более предпочтительным выбором в задачах, где нет достаточного количества данных, с более широким диапазоном применения. Хотя трансформер включает два отдельных механизма — кодировщик и декодировщик, модель BERT работает только на механизмах кодирования для создания языковой модели, а GPT-3 объединяет процесс кодирования и декодирования, чтобы получить трансформер-декодер для создания текста.

Также GPT-3 генерирует вывод по одному токену за раз, BERT же не является авторегрессионным, поэтому использует глубокий двунаправленный контекст для прогнозирования.

В таблице 1 приведено сравнение самых распространенных языковых моделей для обработки естественного языка.

2. Представление на основе графов в обработке естественного языка

Некоторые задачи обработки естественного языка зависят от различных типов структур графов, например графы совместной встречаемости (коокурренции) слов, графы слов-документов, предложения как графы и графы знаний.

Граф совместной встречаемости (коокурренции) слов может быть также идентифицирован как граф совпадений слов на основе локального контекста. В этом типе предполагается, что слова встречаются друг с другом в пределах контекстного окна слов — последовательность из нескольких слов, следующих друг за другом. Основная информация из графа используется несколькими моделями для эмбеддинга, например, в известной модели SkipGram [4], используемой

Таблица 2. Анализ типов представления на основе графов в обработке естественного языка

Графовое представление	Описание	Область применения
Граф совместной встречаемости	Предполагается, что слова встречаются друг с другом в пределах контекстного окна. Основная информация используется несколькими моделями для обучения эмбеддингу слов.	Извлечение ключевых слов и ключевых фраз Биомедицинская область Машинный перевод
Граф слов-документов	Информация о встречаемости слова может быть закодирована на уровне документа. Важная информация используется для изучения представлений слов и документов. Основную информацию предоставляют статистические модели.	Латентное распределение Дирихле
Граф знаний	Граф в этом типе представлен кодирующим различные сущностные отношения.	Ответы на вопросы и поиск информации
Граф фраз	Граф, представленный в виде закодированного с помощью минимального автомата большого набора фраз. Граф фраз состоит из узла в любом обновлении статуса для каждой появляющейся фразы и ребра между каждым набором двух фраз, используемых рядом в любом обновлении статуса.	Обнаружение плагиата Суммирование текста Классификация текста Кластеризация текста
Предложения как граф	Граф представлен как закодированное отношение синтаксической и семантической зависимости между словами.	Машинный перевод Семантическая маркировка ролей Классификация предложений Потоковая передача данных в социальных сетях Суммирование текста

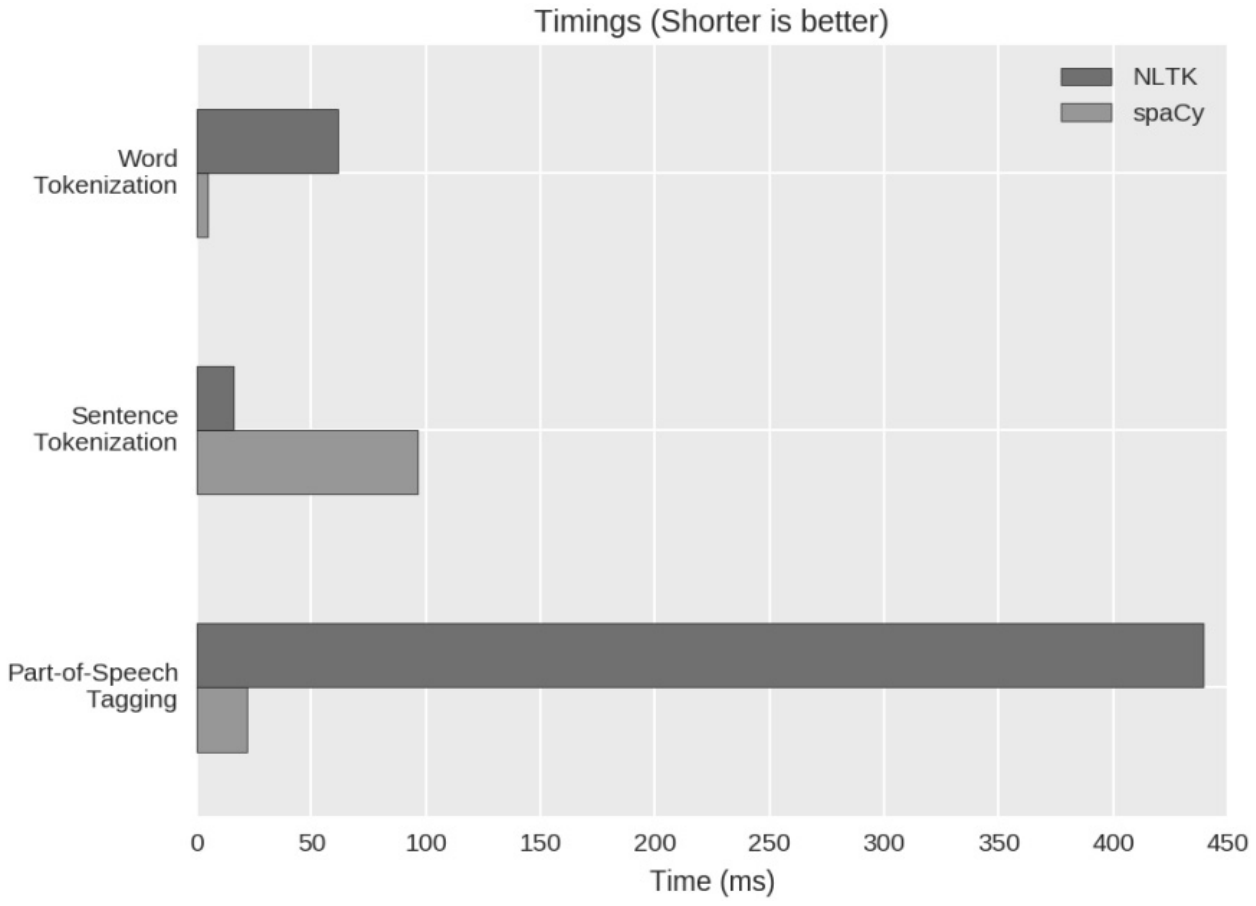

Рис. 1. Сравнение производительности библиотек SpaCy и NLTK

Рис. 2. Фрагмент графового представления текста статьи

для обучения векторных представлений слов в модели Word2vec, и глобальных векторов для представления слов в модели GloVe [9], которая пытается эффективно использовать статистику совпадений.

В графе слов-документов может быть закодирована информация о появлении слова на уровне документа. Важная информация используется для изучения представлений слов и документов. Такие модели, как статистические тематические модели и параграфы, предоставляют основную информацию, например, латентное распределение Дирихле [10].

Третий тип представления на основе графов называется предложения как графы. В этом типе граф представлен как результат кодирования отношений синтаксической и семантической зависимости между словами. Этот тип ценен для различных задач, таких как машинный перевод и маркировка семантических ролей для классификации предложений [11].

Четвертый тип называется графом знаний. Этот тип графа представлен путем кодирования отношений между различными сущностями и подходит для задач поиска информации [12].

Шестой тип представляет граф фраз. Фраза текста представлена двумя или более терминами в предложениях. При определении типа фразы наблюдается дублирование между типами представления на основе слов

и на основе предложений. Граф фраз состоит из узла в любом обновлении состояния для каждой появляющейся фразы и ребра между каждым набором двух фраз, используемых рядом в любом обновлении состояния.

В таблице 2 представлен анализ типов представления на основе графов в обработке естественного языка. Он сосредоточен на описании идеи для каждого типа, а также некоторых областях исследований, в которых были реализованы эти типы.

Текущий прогресс в области представления и обучения на основе графов обеспечивает понимание возможностей его применения в обработке текста на естественном языке и представлении графами различных элементов в обработке естественного языка.

3. Интеллектуальная система анализа текста

Учитывая актуальность темы визуализации текста, была разработана интеллектуальная система графового представления текста.

В качестве фреймворка NLP выбрана библиотека Python SpaCy [13], которая выделяется на фоне всех других фреймворков, в частности самого известного NLTK, впечатляющей скоростью работы с большими текстами, благодаря основной части, которая реализована на языке C++. На рисунке 1 представлен сравнительный анализ производительности библиотеки SpaCy с NLTK [14].

Кроме того, SpaCy позволяет быстро импортировать модели из NLTK, что позволяет разработчикам, использующим данную систему, быстро переключаться между собственными языковыми моделями.

Разработанная интеллектуальная система предлагает по умолчанию модель на базе простого и эффективного токенизатор tok2vec [15], однако при желании можно переключить модель на более мощную, но значительно более ресурсоемкую реализацию BERT — RoBERTa [16].

В результате работы программа генерирует HTML-файл с динамическим графовым представлением, что позволяет мгновенно оценить содержимое текста, а после поделиться файлом с окружающими. Полученный HTML-файл универсален и запускается под браузерами: Google Chrome, Opera, Mozilla Firefox, IE8 и выше. На рисунке 2 представлен результат обработки текста статьи «Astra to sell electric thrusters to Airbus OneWeb Satellites» с сайта SpaceNews.

Заключение

В связи с широким развитием технологий анализа текста и появлением всё новых применений алгоритмов NLP, задачи быстрого визуального представления результатов работы языковых моделей становятся всё более актуальными. Всё более важным становится извлечение смысла из большого объема текстовых данных.

Рассматриваемые в данной статье вопросы являются весьма актуальными и перспективными, а разработанная система интеллектуального анализа текста позволит исследователям и разработчикам в сфере NLP оптимизировать свою работу.

ЛИТЕРАТУРА

1. Гольдберг Й. Нейросетевые методы в обработке естественного языка. — М.: ДМК Пресс, 2019. — 282 с.
2. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin. A Neural Probabilistic Language Model. Journal of Machine Learning Research 3 (2003), p. 1137–1155.
3. Мишунин О.Б., Савинов А.П., Фирстов Д.И. Проблемы, возникающие в интеллектуальных обучающих системах при оценке ответов на естественном языке // Современные проблемы науки и образования. — 2015. — № 2–2.
4. Y. Song, S. Shi, J. Li, and H. Zhang, Directional skip-gram: Explicitly distinguishing left and right context for word embeddings, in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., 2018, pp. 175–180.
5. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space. International Conference on Learning Representations, 2013.
6. Jeffrey Pennington, Richard Socher and Christopher Manning. GloVe: global vectors for word representation. In Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 1532–1543, Doha, Qatar. Association for Computational Linguistics, October 2014.
7. Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.
8. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language Models are Few-Shot Learners. ArXiv, abs/2005.14165.
9. J. Pennington, R. Socher, and C. Manning, Glove: Global vectors for word representation, in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1543.

10. D.M. Blei, A.Y. Ng, and M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res., vol. 3, pp. 993–1022, Mar. 2003.
11. D. Marcheggiani and I. Titov, Encoding sentences with graph convolutional networks for semantic role labeling, 2017, arXiv:1703.04826.
12. U. Sawant, S. Garg, S. Chakrabarti, and G. Ramakrishnan, Neural architecture for question answering using a knowledge graph and Web corpus, Inf. Retr. J., vol. 22, nos. 3–4, pp. 324–349, Aug. 2019.
13. Бенгфорт Б., Билбро Р., Охеда Т. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка.— СПб.: Питер, 2019 г.
14. Погорельцев С.А. Краткий обзор NLP библиотеки SpaCy. 2020. URL: https://habr.com/ru/post/504680. (Дата обращения 22.10.2022).
15. Официальная документация SpaCy. URL: https://spacy.io/api/tok2vec. (Дата обращения 22.10.2022).
16. Фостер Д. Генеративное глубокое обучение. Творческий потенциал нейронных сетей.— СПб.: Питер, 2020.— 336 с.

Акбашева Галина Амировна (galina_akbash@mail.ru), Тлупов Ислам Заурбекович (tlupovislam@gmail.com).
Журнал «Современная наука: актуальные проблемы теории и практики»

Кабардино-Балкарский государственный университет им. Х.М. Бербекова