

УДК 81'32

ББК 81.1

DOI: <https://doi.org/10.17308/lic/1680-5755/2022/4/128-143>**ВЫЯВЛЕНИЕ МЕТАФОРИЧЕСКОЙ СОЧЕТАЕМОСТИ МЕТОДАМИ
МАШИННОГО ОБУЧЕНИЯ****О. В. Дони́на***Воронежский государственный университет***IDENTIFICATION OF METAPHORS WITH THE HELP
OF MACHINE LEARNING****O. V. Donina***Voronezh State University*

Аннотация: в рамках данной статьи рассмотрены возможности создания классификатора по автоматическому определению метафор методами машинного обучения. Нами был собран представительный датасет из 389 857 примеров, размеченных вручную, на основе которого и происходило обучение модели. В статье описана серия экспериментов, возникшие сложности, а также способы их решения. Так, для решения поставленной задачи были применены: наивный байесовский классификатор, логистическая регрессия и искусственные нейронные сети. Эксперименты происходили при изменении следующих параметров: наличие стоп-слов, лемматизация, стемминг, количество N-gram; для нейронных сетей также корректировались параметры: количество эпох, размер партии, количество примеров для обучения и валидации и пр. Лучшие результаты (Accuracy = 0,88, F1-score = 0,87) были достигнуты при помощи сверточной нейронной сети со следующими параметрами: эпохи = 10, слои = 6 (в том числе 2 слоя dropout), batch_size = 500, обучение – на 70 % данных, валидация – на 30 % данных, векторизация = 2 и 3 символа, функция активации = relu и sigmoid, оптимизатор = Adamax, loss_func = binary_crossentropy. В результате проделанной работы удалось разработать средства автоматизации классификации корпусных примеров метафорической сочетаемости, что в перспективе должно содействовать интенсификации и популяризации исследований в области изучения метафор в связи с уменьшением трудо- и времязатрат исследователей по обработке корпусных примеров.

Ключевые слова: машинное обучение, Text Mining, Natural Language Processing, автоматическое выявление метафор, криптоклассный анализ, нейронные сети, обучение с учителем.

Abstract: the article discusses the possibilities of creating a classifier for the automatic metaphor identification with the help of machine learning. The model was trained on the basis of a representative dataset of 389 857 examples which was marked up by us manually. The article describes a series of experiments, the difficulties encountered, as well as ways we used to solve them. The following machine learning methods were used: naive Bayes, logistic regression and artificial neural networks. The following parameters were changed: stop words, lemmatization, stemming, the number of N-grams; for neural networks, the parameters were also adjusted: the number of epochs, batch size, the number of examples for training and validation, etc. The best results (Accuracy = 0.88, F1-score = 0.87) were achieved using a convolutional neural network with the following parameters: epochs = 10, layers = 6 (including 2 dropout layers), batch_size = 500, training – 70 % of data, validation – 30 % of data, vectorization = 2 and 3 characters, activation function = relu and sigmoid, optimizer = Adamax, loss_func = binary_crossentropy. As a result we developed automation tools for the classification of corpus examples of metaphorical compatibility, which in the future should contribute to the intensification and popularization of research in this area, due to the reduction of labor and time spent by researchers on processing corpus queries and their classification.

© Дони́на О. В., 2022

Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.

Key words: machine learning, Text Mining, Natural Language Processing, metaphor identification, cryptotype analysis, CNN, supervised learning.

Введение

Данное исследование проводилось в рамках одного из направлений работы Воронежской лексико-типологической группы, занимающейся изучением лексико-грамматических классов имен, которые в одних языках оформлены грамматически (например, в виде именных классов, как в некоторых языках Африки), а в других выражены лексически и могут быть выявлены только при помощи анализа сочетаемости. Такие лексико-грамматические классы называются *криптоклассами* (скрытыми классами) [1].

Суть методики криптоклассного анализа состоит в анализе большого объема метафорических слово-сочетаний заданного абстрактного имени типа *shame, fear, passion* с целью дальнейшего определения вхождения имен в криптоклассы английского языка. Восемь описанных на настоящий момент криптоклассов английского языка представляют собой скрытые лексико-грамматические категории, которые типологически выделены и закреплены в явной грамматике других языков мира.

Целью нашей работы является разработка средств автоматизации выявления скрытой категоризации лексики естественного языка. Новизна работы заключается в новаторском подходе к реализации криптоклассного анализа, так как здесь впервые обсуждается возможность автоматизации выявления метафорической сочетаемости на базе криптоклассов.

Материалы и методы исследования

Изначально мы рассматривали несколько возможных вариантов создания автоматической классификации метафорической сочетаемости:

- 1) поиск по шаблонам;
- 2) использование грамматик;
- 3) машинное обучение (и возможное последующее использование ХАИ – explainable Artificial Intelligence);
- 4) кластеризация размеченных предложений и выявление отличительных характеристик каждого класса для последующего использования этих данных при классификации;
- 5) синтаксический парсинг и классификация с учетом результатов парсинга.

В связи с тем, что к настоящему моменту мы имеем довольно внушительный корпус размеченных вручную примеров метафорической сочетаемости (389 857 предложений), наиболее оптимальным решением выступает создание классификатора методами машинного обучения. Ручная разметка проводилась в основном студентами факультета романо-германской филологии

Воронежского государственного университета, обучающихся на направлении «Фундаментальная и прикладная лингвистика», в рамках учебной практики по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности в период 2016–2020 гг. (подробнее в статьях [2; 3]). В указанном корпусе наличие метафоры размечалось как «1», а отсутствие метафорического употребления – как «0», таким образом, стоящая перед нами задача автоматического выявления метафорической сочетаемости может быть представлена как задача бинарной классификации [4–6].

Перед началом создания классификатора были проанализированы похожие реализованные проекты, такие как, например, «Компьютерная программа для диагностирования пола и возраста участника интернет-коммуникации на основе количественных параметров его текстов (с учетом возможного искажения признаков письменной речи) с оценкой их эффективности» [7], где применялись следующие библиотеки Python: numpy, pandas, scikit-learn, tensorflow, keras, tqdm, ufal.udpipe, conllu. В рамках этого проекта при определении пола использовалась бинарная классификация (0 – женский пол, 1 – мужской пол), а при выявлении возраста определялась одна из трех возрастных групп: 20–29 лет, 30–39 лет, 40–49 лет.

Результаты исследования

Изначально мы апробировали некоторые подходы на небольшой выборке. Первым шагом стало определение BaseLine, т.е. случайное определение ответа на основе вероятностных классов. В табл. 1 отражены полученные результаты. В качестве оценки использовалась метрика F1-Score, которая лучше, чем Accuracy, отражает результаты при сильном перевесе в классах. Таким образом, случайное определение класса показало F1-Score = 73 %, что стало минимальной отправной точкой при дальнейшем сопоставлении наших результатов.

Т а б л и ц а 1

BaseLine – случайный стратифицированный (с учетом представительности классов) выбор метки

	Precision	Recall	f1-score	Support
0	0,81	0,99	0,89	260
1	0,50	0,03	0,06	63
avg / total	0,75	0,80	0,73	323
Accuracy	0,80			

Далее была использована нейронная сеть на основе LSTM (Long short-term memory), словоформы кодировались через word2vec, без приведения к начальной форме (у каждого слова был флаг, показывающий, является ли оно анализируемым или нет).

В табл. 2 представлен результат для 15 запусков. При этом каждый корпус делился на 3 части: 72 % – на обучение, 8 % – на валидацию, 20 % – на тестирование. Деление на тренировочное и тестировочное множества проводилось по группам пар – пары из тренировочного множества не встречались в тестировочном. В таблице также дан статистический разброс точности (std – среднеквадратичное отклонение), значения которого отражают неустойчивость полученных результатов. Несмотря на то, что показатели нейросетевой модели выше случайного выбора (см. табл. 1), эта разница лежит в границе статистической погрешности, что говорит о том, что либо нужно увеличить объем корпуса, либо использовать другую модель.

Т а б л и ц а 2

Нейросетевая модель и среднеквадратичное отклонение

Модель	F1 score		Accuracy		Support
	mean	std	mean	std	
Neural network	0,82	0,10	0,85	0,06	2962
Baseline	0,78	0,08	0,77	0,05	2962
Neural network	0,79	0,09	0,84	0,07	10013
Baseline	0,76	0,06	0,76	0,03	10013

Проведенные эксперименты привели нас к выводу о необходимости использовать весь набор данных (389 857 примеров), а также попробовать применить другие модели, воспользовавшись библиотеками scikit-learn и NLTK. Последняя библиотека часто используется для анализа текста; она содержит разнообразные алгоритмы обработки естественного языка, в том числе токенизацию, частеречную разметку (POS tagging), удаление стоп-слов, lexicon normalization (стемминг, лемматизацию), сентимент-анализ, topic modelling, извлечение сущностей (NER), классификацию текстов и многое другое [8].

Вначале был использован наивный Байес по отдельным словам; полученная точность составила: Accuracy (MultinomialNB) = 0,79.

Далее мы попробовали применить логистическую регрессию, хорошо работающую при решении задач выявления бинарной классификации. В табл. 3 даны результаты применения логистической регрессии к корпусу без стоп-слов, где разбиение шло по слову, при этом ни лемматизация, ни стемминг применены не были.

Т а б л и ц а 3

Результаты применения логистической регрессии

	Precision	Recall	f1-score	Support
0	0,85	0,91	0,88	84539
1	0,71	0,56	0,63	32419
Accuracy			0,81	116958
Macro avg	0,78	0,74	0,75	116958
Weighted avg	0,81	0,81	0,81	116958

С целью проверки влияния качества размеченного корпуса на результаты работы классификатора, половина нашего датасета (167 765 предложений) была перепроверена еще раз вручную тремя разметчиками. Дальнейшие эксперименты с линейной регрессией проходили на материале обоих имеющихся корпусов; для удобства в дальнейшем будем обозначать изначальный датасет как dataset_volume, а перепроверенные данные – как dataset_quality.

Вначале мы разбили мешок слов (bag of words) на униграммы (liblinear-unigram), стоп-слова не были удалены, стемминг и лемматизация не использовались. Результаты применения такого подхода к двум описанным выше датасетам можно найти в табл. 4.

Далее для этого же подхода (логистическая регрессия + униграммы) были удалены стоп-слова (но стемминг и лемматизация все так же не использовались). Результаты этого эксперимента представлены в табл. 5.

Т а б л и ц а 4

Сравнение результатов применения логистической регрессии (униграммы) для двух наборов данных

		Precision	Recall	f1-score	Support
1	2	3	4	5	6
0	dataset_volume	0,85	0,93	0,89	85410
	dataset_quality	0,84	0,90	0,87	30821
1	dataset_volume	0,75	0,57	0,65	32418
	dataset_quality	0,82	0,73	0,78	19509

Окончание табл. 4

1	2	3	4	5	6
Accuracy			dataset_volume	0,83	116958
			dataset_quality	0,83	50330
Macro avg	dataset_volume	0,80	0,75	0,77	116958
	dataset_quality	0,83	0,82	0,82	50330
Weighted avg	dataset_volume	0,82	0,83	0,82	116958
	dataset_quality	0,83	0,83	0,83	50330

Таблица 5

Сравнение результатов применения логистической регрессии (униграммы – стоп-слова) для двух наборов данных

		Precision	Recall	f1-score	Support
0	dataset_volume	0,84	0,93	0,88	84540
	dataset_quality	0,83	0,88	0,85	30821
1	dataset_volume	0,73	0,54	0,62	32418
	dataset_quality	0,79	0,71	0,75	19509
Accuracy			dataset_volume	0,82	116958
			dataset_quality	0,82	50330
Macro avg	dataset_volume	0,79	0,73	0,75	116958
	dataset_quality	0,81	0,80	0,80	50330
Weighted avg	dataset_volume	0,81	0,82	0,81	116958
	dataset_quality	0,82	0,82	0,81	50330

После этого к описанной выше модели (логистическая регрессия: униграммы – стоп-слова) был добавлен стемминг (табл. 6).

Еще в одном эксперименте мы использовали лемматизацию вместо стемминга (табл. 7).

Таблица 6

Сравнение результатов применения логистической регрессии (униграммы – стоп-слова + стемминг) для двух наборов данных

		Precision	Recall	f1-score	Support
0	dataset_volume	0,84	0,93	0,89	84540
	dataset_quality	0,84	0,90	0,87	30821
1	dataset_volume	0,75	0,55	0,63	32418
	dataset_quality	0,82	0,73	0,77	19509
Accuracy			dataset_volume	0,83	116958
			dataset_quality	0,83	50330
Macro avg	dataset_volume	0,80	0,74	0,76	116958
	dataset_quality	0,83	0,81	0,82	50330
Weighted avg	dataset_volume	0,82	0,83	0,82	116958
	dataset_quality	0,83	0,83	0,83	50330

Т а б л и ц а 7

Сравнение результатов применения логистической регрессии (униграммы – стоп-слова + лемматизация) для двух наборов данных

		Precision	Recall	f1-score	Support
0	dataset_volume	0,84	0,93	0,88	84540
	dataset_quality	0,83	0,88	0,86	30821
1	dataset_volume	0,74	0,53	0,61	32418
	dataset_quality	0,79	0,71	0,75	19509
Accuracy			dataset_volume	0,82	116958
			dataset_quality	0,82	50330
Macro avg	dataset_volume	0,79	0,73	0,75	116958
	dataset_quality	0,81	0,80	0,80	50330
Weighted avg	dataset_volume	0,81	0,82	0,81	116958
	dataset_quality	0,82	0,82	0,81	50330

Как видно из приведенных в таблицах данных, лучшие результаты из этой серии экспериментов были получены для униграмм (с учетом стоп-слов и без лемматизации/стемминга) на материале второго, перепроверенного корпуса (см. табл. 4).

После этого на материале нового набора данных была апробирована логистическая регрессия с делением на биграммы (табл. 8).

Т а б л и ц а 8

Результаты применения логистической регрессии (биграммы)

	Precision	Recall	f1-score	Support
0	0,82	0,94	0,88	30821
1	0,88	0,67	0,76	19509
Accuracy			0,84	50330
Macro avg	0,85	0,81	0,82	50330
Weighted avg	0,84	0,84	0,83	50330

Далее к аналогичной модели (логистическая регрессия: биграммы) мы применили стемминг, что, к сожалению, привело к ухудшению полученных результатов (табл. 9). В дальнейших экспериментах стемминг больше не применялся.

Т а б л и ц а 9

Результаты применения логистической регрессии (биграммы + стемминг)

	Precision	Recall	f1-score	Support
0	0,84	0,90	0,87	30821
1	0,82	0,73	0,77	19509
Accuracy			0,83	50330
Macro avg	0,83	0,81	0,82	50330
Weighted avg	0,83	0,83	0,83	50330

В следующей серии экспериментов мешок слов был разбит на символы. Сначала на обоих наборах данных использовалось 2-символьное разбиение (2-char-gram) (табл. 10).

Т а б л и ц а 10

Сравнение результатов применения логистической регрессии (2-char-gram) для двух наборов данных

		Precision	Recall	f1-score	Support
0	dataset_volume	0,78	0,94	0,85	84540
	dataset_quality	0,81	0,87	0,84	30821
1	dataset_volume	0,67	0,33	0,44	32418
	dataset_quality	0,76	0,68	0,72	19509
Accuracy			dataset_volume	0,77	116958
			dataset_quality	0,79	50330
Macro avg	dataset_volume	0,73	0,63	0,65	116958
	dataset_quality	0,79	0,77	0,78	50330
Weighted avg	dataset_volume	0,75	0,77	0,74	116958
	dataset_quality	0,79	0,79	0,79	50330

Также была рассмотрена возможность разбиения мешка слов по 3 символа (3-char-gram) (табл. 11).

В рамках последнего эксперимента с логистической регрессией были рассмотрены как 2-символьные, так и 3-символьные разбиения (табл. 12).

Т а б л и ц а 11

Сравнение результатов применения логистической регрессии (3-char-gram) для двух наборов данных

		Precision	Recall	f1-score	Support
0	dataset_volume	0,85	0,93	0,89	84540
	dataset_quality	0,87	0,90	0,89	30821
1	dataset_volume	0,75	0,58	0,65	32418
	dataset_quality	0,84	0,79	0,81	19509
Accuracy			dataset_volume	0,83	116958
			dataset_quality	0,86	50330
Macro avg	dataset_volume	0,80	0,75	0,77	116958
	dataset_quality	0,85	0,84	0,85	50330
Weighted avg	dataset_volume	0,82	0,83	0,82	116958
	dataset_quality	0,86	0,86	0,86	50330

Т а б л и ц а 12

Сравнение результатов применения логистической регрессии (2,3-char-gram) для двух наборов данных

		Precision	Recall	f1-score	Support
0	dataset_volume	0,85	0,93	0,89	84540
	dataset_quality	0,87	0,90	0,88	30821
1	dataset_volume	0,75	0,58	0,65	32418
	dataset_quality	0,83	0,78	0,81	19509
Accuracy			dataset_volume	0,83	116958
			dataset_quality	0,86	50330
Macro avg	dataset_volume	0,80	0,75	0,77	116958
	dataset_quality	0,85	0,84	0,85	50330
Weighted avg	dataset_volume	0,82	0,83	0,82	116958
	dataset_quality	0,86	0,86	0,86	50330

Таким образом, эксперименты с логистической регрессией показали, что лучшие результаты в рамках этой модели на наших данных можно получить при 3-символьном разбиении мешка слов на материале нового корпуса контекстов (dataset_quality), меньшего размера, но перепроверенного несколькими разметчиками (табл. 11). В связи с полученными результатами в дальнейших экспериментах мы будем использовать только обновленный корпус примеров.

Следующим шагом стало использование нейронных сетей. Для обучения искусственной нейронной сети использовалась библиотека Keras, к которой в сети представлена подробная документация. Помимо официальной документации, мы также ознакомились с примерами использования нейронных сетей для решения задач Text Mining, анализируя реализованные модели с целью их последующей модификации и использования в разрабатываемом нами классифи-

каторе; так, например, в статье [9] приводится схема нейронной сети в Keras для анализа тональностей, где 0 – это отрицательные отзывы, а 1 – положительные.

Представим этапы создания нейронной сети в Keras.

1. Подготовка данных: векторизация примеров, приведение примеров к одинаковому размеру, преобразование переменных в тип float; разделение набора данных на обучающий и тестировочный (стандартное соотношение: 80 к 20 %).

2. Создание модели: определение типа модели (последовательная или с функциональным API); добавление входных, скрытых и выходных слоев; предотвращение переобучения, которое может произойти, если модель научится выявлять шаблоны, характерные только для обучающей выборки, а не обобщенные паттерны (регуляризация весов; dropout (исключение некоторых нейронов в слоях для пре-

дотвращения их переобучения), коэффициент исключения = от 20 до 50 % нейронов); использование агрегирующей функции (dense) и функции активации (ReLU – Rectified Linear Activation, линейный фильтр, tanh, сигмоида, линейная, ступенчатая); компиляция модели (оптимизатор ‘adam’ (во время обучения изменяет веса и смещение), функции потерь (бинарная кросс-энтропия), метрика оценки).

3. Обучение модели: размер партии (т. е. количество распространяемых по сети элементов, стандарт = 32); количество эпох (т. е. проходов всех элементов, рекомендуется выбирать 2 для исключения возможности переобучения модели).

Как отмечалось ранее, мы использовали фреймворк Keras с Tensorflow в качестве бэкенда. На вход нейронной сети поступала матрица подсчета токенов, в которую была преобразована коллекция текстовых документов при помощи CountVectorizer (sklearn.feature_extraction.text.CountVectorizer). Хотя для логистической регрессии мы пробовали разные варианты мешков слов, для искусственных нейронных сетей мы решили остановиться на 2-символьном разбиении.

Используемые нами последовательные сверточные нейронные сети дают возможность выстраивать модель послойно; слои добавляются при помощи команды add(). Опишем строение сверточной нейронной сети, которое мы взяли за базовое: она состоит из шести слоев, два из которых – dropout, позволяющие избежать переобучения модели; в качестве функции активации используются ReLU и Sigmoid; в качестве оптимизатора используется Adam, а в качестве loss (потери) – binary_crossentropy.

В первом эксперименте обучение проводилось за одну эпоху (итерацию) с размером партии (batch_size) = 500; обучение проходило на 50 330 примерах, валидация – на 117 435 (табл. 13).

Т а б л и ц а 13

Результаты применения сверточной нейронной сети (nb_epoch = 1; batch_size = 500; teach 50330, validate 117435)

	Precision	Recall	f1-score	Support
0	0,83	0,87	0,85	71778
1	0,77	0,71	0,74	45657
Accuracy			0,81	117435
Macro avg	0,80	0,79	0,79	117435
Weighted avg	0,81	0,81	0,81	117435

На следующем шаге мы оставили неизменными все характеристики, кроме количества эпох, заменив его на два (табл. 14).

Т а б л и ц а 14

Результаты применения сверточной нейронной сети (nb_epoch = 2; batch_size = 500; teach 50330, validate 117435)

	Precision	Recall	f1-score	Support
0	0,81	0,89	0,85	71778
1	0,79	0,68	0,73	45657
Accuracy			0,81	117435
Macro avg	0,80	0,78	0,79	117435
Weighted avg	0,80	0,81	0,80	117435

После этого мы рассмотрели модель, обучение которой проходило на 117 435 примерах, а валидация – на 50 330 примерах, в 1 эпоху (при этом все остальные показатели остались прежними) (табл. 15).

Т а б л и ц а 15

Результаты применения сверточной нейронной сети (nb_epoch = 1; batch_size = 500; teach 117435, validate 50330)

	Precision	Recall	f1-score	Support
0	0,85	0,88	0,87	30821
1	0,80	0,75	0,78	19509
Accuracy			0,83	50330
Macro avg	0,83	0,82	0,82	50330
Weighted avg	0,83	0,83	0,83	50330

Последний эксперимент в этой серии включал архитектуру, рассмотренную выше, но рассчитанную на две эпохи (табл. 16).

Из проведенной серии экспериментов видно, что полученные результаты классификатора, построенного на искусственных нейронных сетях, пока оказываются ниже, чем при логистической регрессии.

Т а б л и ц а 16

Результаты применения сверточной нейронной сети (nb_epoch = 2; batch_size = 500; teach 117435, validate 50330)

	Precision	Recall	f1-score	Support
0	0,85	0,87	0,89	30821
1	0,79	0,76	0,77	19509
Accuracy			0,83	50330
Macro avg	0,82	0,82	0,82	50330
Weighted avg	0,83	0,83	0,83	50330

Прежде чем продолжать эксперименты с нейронными сетями, мы решили сравнить модели с различным количеством нейронов в скрытых слоях, а также влияние dropout на получаемый результат. Для этого

сначала были созданы 3 модели без dropout с 16, 100 и 512 узлами.

Визуализируем полученные Keras-модели при помощи graphviz (рис. 1–3).

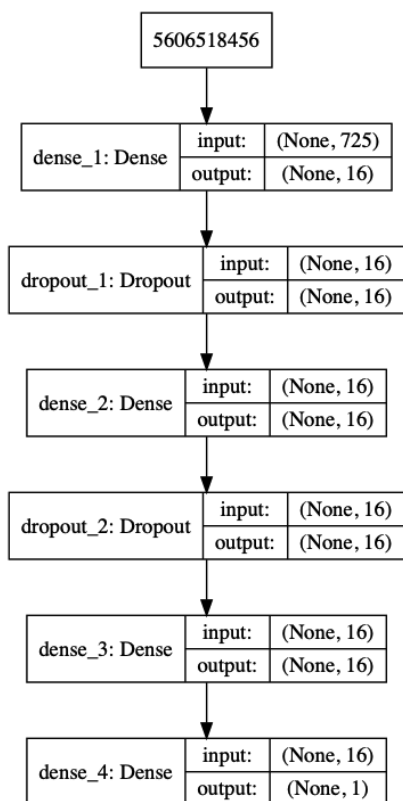


Рис. 1. Keras-модель с 16 нейронами в скрытых слоях (без dropout)

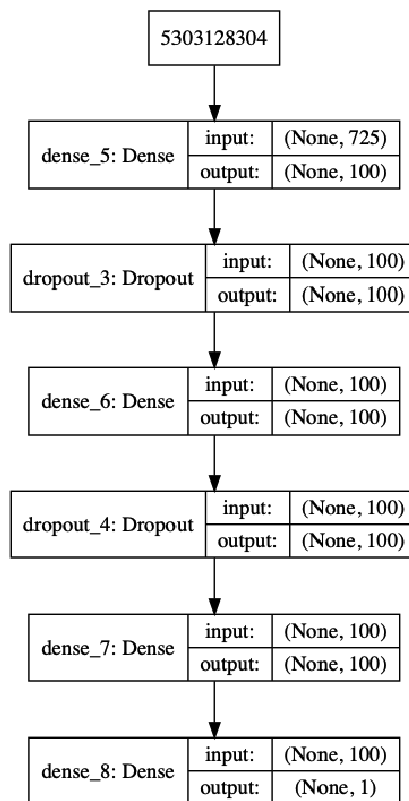


Рис. 2. Keras-модель со 100 нейронами в скрытых слоях (без dropout)

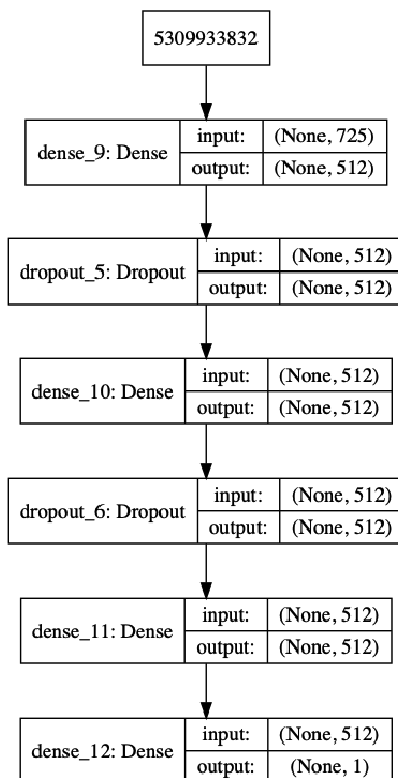


Рис. 3. Keras-модель с 512 нейронами в скрытых слоях (без dropout)

Рассмотрим полученные для этих моделей показатели (табл. 17).

После этого мы построили график потерь (рис. 4): непрерывная линия на графике отражает

потери во время обучения, прерывистая – во время валидации; стоит отметить, что модель является наиболее точной, когда потери на проверочных данных минимальны.

Таблица 17

Сравнение моделей с разным количеством нейронов в скрытых слоях (без dropout)

		Precision	Recall	f1-score	Support
0	small (16)	0,85	0,87	0,86	30821
	medium (100)	0,84	0,90	0,87	30821
	big (512)	0,86	0,87	0,87	30821
1	small (16)	0,78	0,75	0,77	19509
	medium (100)	0,82	0,73	0,77	19509
	big (512)	0,79	0,78	0,78	19509
Accuracy			small (16)	0,82	50330
			medium (100)	0,83	50330
			big (512)	0,83	50330
Macro avg	small (16)	0,82	0,81	0,81	50330
	medium (100)	0,83	0,81	0,82	50330
	big (512)	0,83	0,82	0,82	50330
Weighted avg	small (16)	0,82	0,82	0,82	50330
	medium (100)	0,83	0,83	0,83	50330
	big (512)	0,83	0,83	0,83	50330

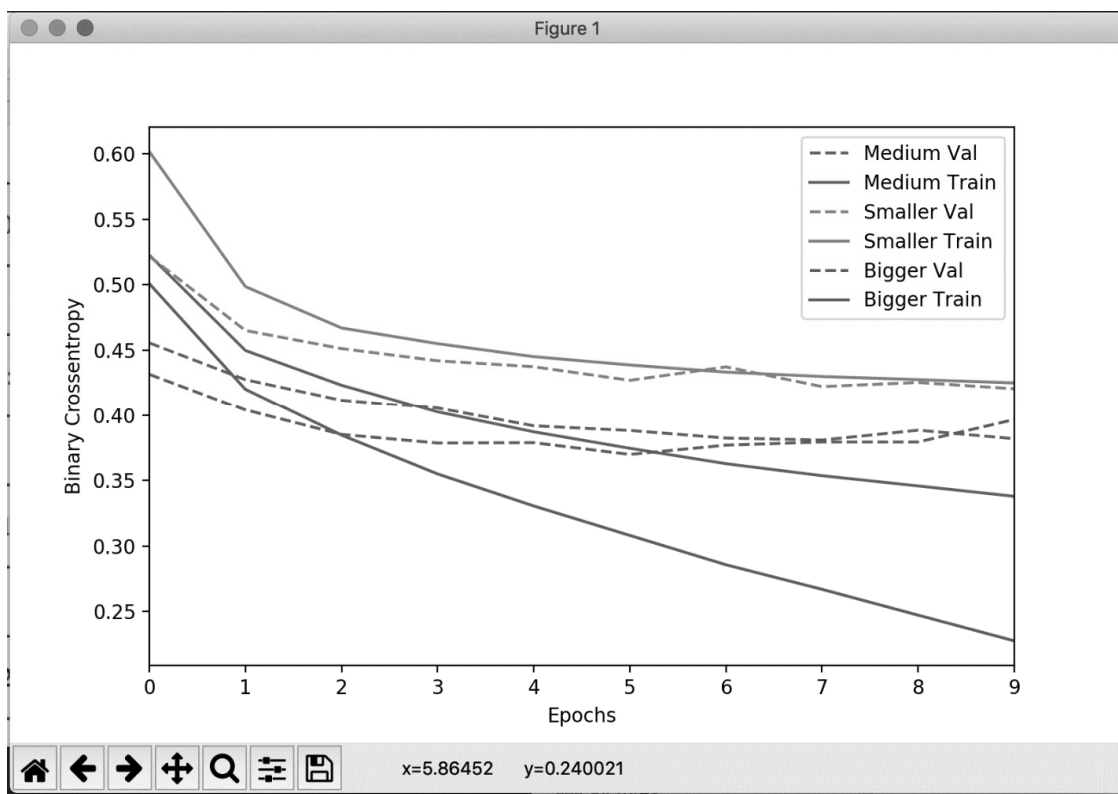


Рис. 4. График потерь (без dropout)

Аналогичные шаги были предприняты для модели с dropout.

Ниже представлена визуализация полученных моделей (рис. 5–7).

Сравнение результатов рассматриваемых моделей представлено в табл. 18.

Также рассмотрим график потерь для полученных моделей с dropout (рис. 8).

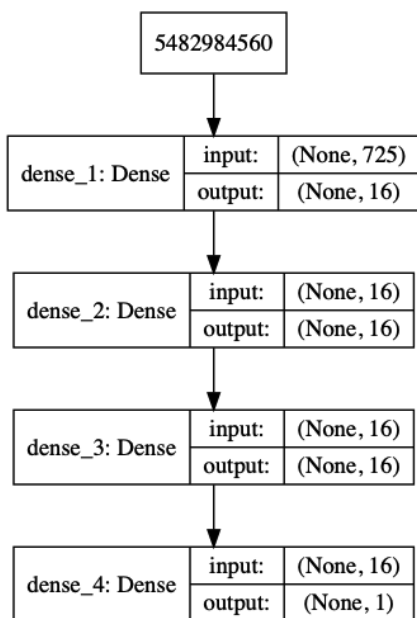


Рис. 5. Keras-модель с 16 нейронами в скрытых слоях (с dropout)

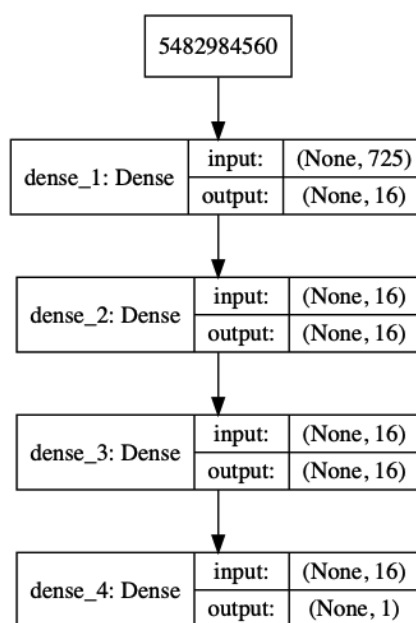


Рис. 6. Keras-модель со 100 нейронами в скрытых слоях (с dropout)

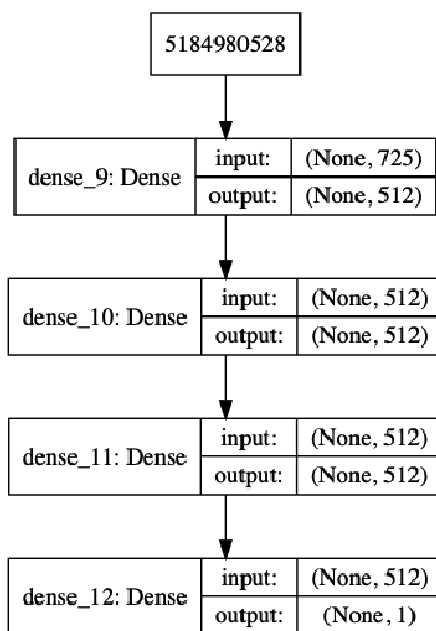


Рис. 7. Keras-модель с 512 нейронами в скрытых слоях (с dropout)

Таблица 18

Сравнение моделей с разным количеством нейронов в скрытых слоях (с dropout)

		Precision	Recall	f1-score	Support
0	small (16)	0,85	0,87	0,86	30821
	medium (100)	0,84	0,90	0,87	30821
	big (512)	0,86	0,87	0,87	30821
1	small (16)	0,78	0,75	0,77	19509
	medium (100)	0,82	0,73	0,77	19509
	big (512)	0,79	0,78	0,78	19509
Accuracy			small (16)	0,82	50330
			medium (100)	0,83	50330
			big (512)	0,83	50330
Macro avg	small (16)	0,82	0,81	0,81	50330
	medium (100)	0,83	0,81	0,82	50330
	big (512)	0,83	0,82	0,82	50330
Weighted avg	small (16)	0,82	0,82	0,82	50330
	medium (100)	0,83	0,83	0,83	50330
	big (512)	0,83	0,83	0,83	50330

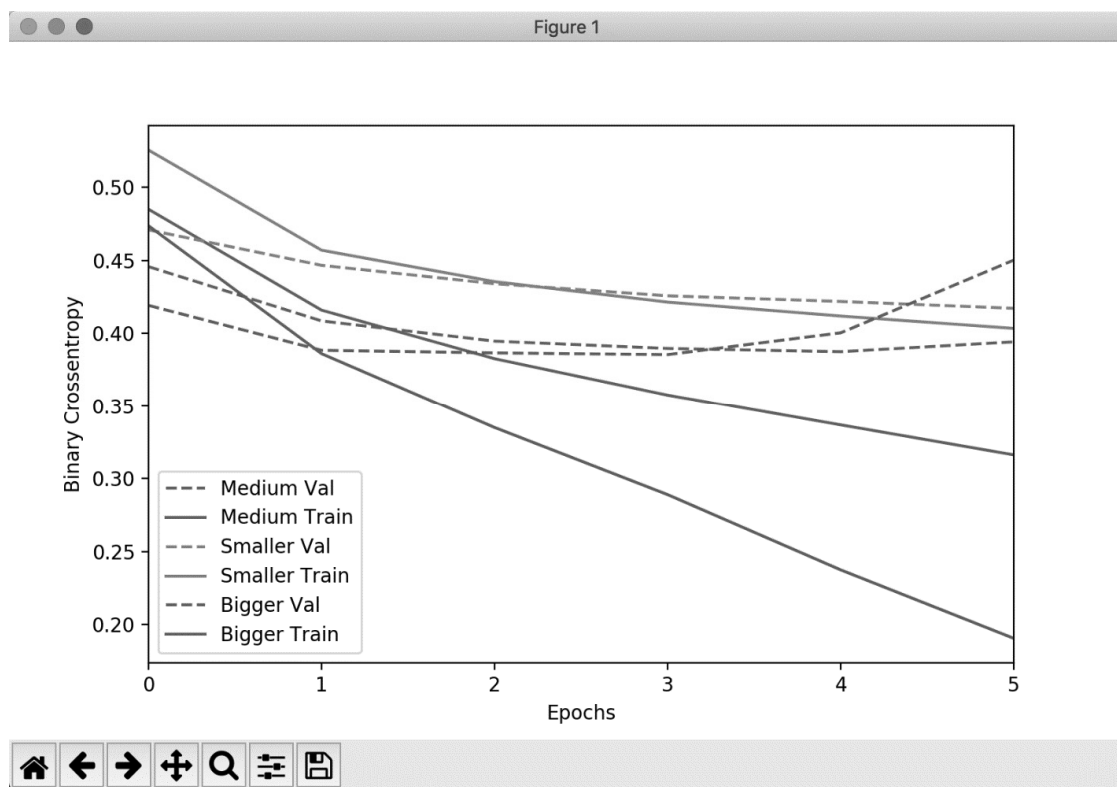


Рис. 8. График потерь (с dropout)

Проведенный анализ показал, что dropout позволяет уменьшить риск переобучения сети.

После ряда экспериментов с различными признаками последовательной сверточной нейронной сети, с учетом всех сделанных ранее выводов, удалось разработать модель, показавшую лучшие результаты для бинарной классификации при выявлении метафор: F1-score (avg) = 0,87 (табл. 19). Опишем параметры полученной модели сверточной нейронной сети: количество эпох = 10, слои = 6 (в том числе 2 слоя dropout), batch_size = 500, обучение – на 70 % данных, валидация – на 30 % данных, векторизация = 2 и 3 символа, функция активации = relu и sigmoid, оптимизатор = Adamax, loss_func = binary_crossentropy.

Т а б л и ц а 19

Результаты применения сверточной нейронной сети
(nb_epoch = 10; batch_size = 500; teach 117435,
validate 50330)

	Precision	Recall	f1-score	Support
0	0,88	0,92	0,90	30821
1	0,87	0,81	0,84	19509
Accuracy			0,88	50330
Macro avg	0,88	0,87	0,87	50330
Weighted avg	0,88	0,88	0,88	50330

Обсуждение результатов

Сравним полученные нами результаты по выявлению криптоклассной сочетаемости с другими попытками автоматизации поиска метафор в текстах.

Так, в работе [10] при сопоставлении различных способов автоматического выявления метафор было показано, что наиболее надежной процедурой является использование лексических связей (мера точности (precision) достигла 100 %), а наименее надежной – кластеризация (мера точности – 40 %); что касается инструментов, то лучшие результаты показала семантическая связь (мера полноты – 85 % и мера точности – 86 %), а наихудшие – ключевые слова (мера полноты – 11 %, мера точности – 9 %).

Разрабатываются многочисленные алгоритмы для выявления метафоры на материале больших текстовых корпусов; так, в статье [11] предлагается алгоритм ССО (Concrete Category Overlap), основанный на правилах и анализирующий три различных типа синтаксических конструкций: 1) структуры «a subject is an object»; 2) структуры «глагол + прямое дополнение»; 3) структуры «прилагательное + существительное». В работе по выявлению концептуальных метафор [12] также описаны результаты выявления метафор в аналогичных синтаксических конструкциях. Позднее к этим же трем типам был применен метод MII (Metaphor Identification by Learning) (под-

робнее в [13]). В табл. 20 представлено сопоставление результатов этих исследовательских групп.

Т а б л и ц а 20

Сопоставление эффективности классификации трех типов синтаксических конструкций метафоризации

		Precision	Recall, %	F-мера, %
Тип 1	Результаты по [11]	84,1	85,9	85
	Результаты по [12]	83,9	97,5	90,1
	Результаты по [13]	86	92,5	89,2
Тип 2	Результаты по [11]	62	83,8	71,3
	Результаты по [12]	76,1	82	78,9
	Результаты по [13]	65,2	77	70,6
Тип 3	Результаты по [11]	69,8	88,1	77,9
	Результаты по [12]	54,4	43,5	48,3
	Результаты по [13]	46,8	39,7	42,9

В работе [14] описана методология для выявления метафорического употребления на материале греческого языка. Используя корпус газетных текстов, исследователи получили следующие показатели для выявления метафорического употребления: мера точности = 41 %, мера полноты = 49,4 %, F-мера = 0,448.

Была описана система MetaNet для глубокого семантического автоматического анализа метафор [15]. Оценка данной системы происходила на материале трех языков: английского, испанского и русского; при этом, как и в большинстве других примеров в компьютерной лингвистике, результаты работы классификатора сравнивались с так называемым «золотым стандартом» (gold standard data), т. е. с размеченными вручную аннотаторами данными. Полученные результаты представлены в табл. 21.

Т а б л и ц а 21

Оценка системы MetaNet (по [15])

Язык	Recall	Precision
Английский	0,86	0,85
Испанский	0,88	0,86
Русский	0,41	0,9

В монографии [16] описываются примеры применения различных компьютерных технологий для выявления метафор. Так, на примере 164 вручную размеченных глагольных метафор были получены следующие показатели [17]: мера точности = 0,68, мера полноты = 0,66. В исследовании [18] при аннотировании метафор на уровне предложения на материале 1298 английских предложений и 140 русских

предложений была получена F-мера = 0,78 (для английского языка) и F-мера = 0,76 (для русского языка). В работе [19] была показана важность учета синтаксической информации при идентификации метафор, так как включение деревьев зависимостей способствовало получению результата F-меры = 0,75. Исследования [20; 21] предлагают идентифицировать идиомы при помощи измерения семантической близости внутри и между буквальными и переносными значениями частей предложений, что было проверено на материале набора данных из 3964 предложений, содержащих 17 идиом, и были получены результаты F-мера = 0,75. При использовании логистической регрессии к выявлению метафор [22] были получены следующие результаты: 1) для корпуса школьных эссе: F-мера = 0,64, мера точности = 0,79, мера полноты = 0,54; 2) для корпуса новостных статей из British National Corpus (BNC): F-мера = 0,51, мера точности = 0,61, мера полноты = 0,43. При обучении без учителя (unsupervised method) [23] удалось достигнуть меры полноты = 0,61.

В исследовании [24] на примере автоматического выявления метафоры в русском языке лучшие из полученных результатов составили: ассигасу = 68 %, а F-мера = 0,71. В работе [25] приводятся результаты применения метода опорных векторов (SVM) к выявлению метафоры: ассигасу = 11,34, precision = 72,5, recall = 82,86, F-мера = 77,34.

Методы глубокого обучения применялись и для выявления метафоричности в словосочетаниях типа «прилагательное + существительное» [26], где была достигнута ассигасу = 0,91.

В статье [27] применительно к выявлению метафоры сопоставляются различные алгоритмы классификации, а именно: логистическая регрессия – Logistic Regression (LR), «случайный лес» – Random Forest (RF), метод опорных векторов – Linear Support Vector Machine (LSVM), Radial kernel SVM (RSVM), искусственные нейронные сети – Neural Networks (NN). Согласно полученным результатам (табл. 22), лучший показатель точности показал алгоритм логистической регрессии, в то время как метод RSVM оказался лучшим по результату полноты и F-меры.

Т а б л и ц а 22

Результаты сопоставления алгоритмов классификации (по [27])

	Precision	Recall	F-score
LR	0,823	0,805	0,813
RF	0,741	0,821	0,778
NN	0,799	0,814	0,806
LSVM	0,814	0,796	0,8
RSVM	0,784	0,852	0,815

Та же группа авторов рассмотрела влияние контекста на качество выявления метафоры [28]; наиболее удачные результаты работы модели отражены в табл. 23.

Т а б л и ц а 23

Результаты сопоставления алгоритмов классификации с учетом контекста (по [28])

	Accuracy	Precision	Recall	F-score
LR	69,93	76	70,81	73,31
RF	73,19	79,47	73,62	76,43
LSVM	71,74	77,34	72,5	74,84
RSVM	72,83	78	73,58	75,73

Проведенный выше анализ работ аналогичной проблематики показал, что достигнутые нами результаты (см. табл. 19) не уступают другим исследованиям в данной области.

Заключение

В рамках настоящей работы решалась прикладная задача компьютерной лингвистики, обусловленная потребностью оптимизировать и интенсифицировать исследовательскую деятельность в сфере выявления метафорической сочетаемости средствами криптоклассного анализа. Криптоклассный анализ именной классификативности является, на наш взгляд, удобным инструментом, с помощью которого можно посмотреть на, казалось бы, достаточно хорошо изученное явление с другого ракурса, что в свою очередь обеспечит более глубокое проникновение в его (явления) сущность и эволюцию. Данная методика позволяет учитывать когнитивную и психолингвистическую природу категоризационных процессов для понимания и объяснения классификационных процессов, в частности механизмов именной классификативности в языках мира на основе выявления и сравнения скрытых языковых категорий (криптоклассов), что открывает новые горизонты для дальнейших исследований в данной области.

Наша работа имеет как теоретическую, так и практическую значимость. С одной стороны, мы надеемся, что уменьшение трудоемкой работы по ручному выявлению и классификации примеров криптоклассной сочетаемости приведет к популяризации и активному развитию этой теории. С другой стороны, наша работа может быть использована в Text Mining (интеллектуальном анализе текстовых данных), где в качестве одной из основных задач выступает семантическая обработка исследуемых данных. Для естественного языка характерно наличие различных типов лексической и семантической неоднозначности, в том числе метафор, омонимии, кореференции, пр. В связи с этим исследование кате-

горизации лексики естественного языка является важным для практической разработки автоматических естественно-языковых систем по работе с семантикой.

Новизна исследования заключалась в том, что в нем впервые была предпринята попытка автоматизации выявления криптоклассной специфики имен с применением методов Machine Learning для анализа метафоры. В процессе работы удалось разработать классифицирующую модель, достигшую точности 87 % при выявлении метафор.

В перспективах исследования мы планируем на материале используемого размеченного набора данных усовершенствовать разработанные нами ранее классификаторы для автоматического определения варианта английского языка и источника материала (новостной текст или интернет-коммуникация) [29; 30].

ЛИТЕРАТУРА

1. *Boriskina O. O., Marchenko T.* An algorithm for analysis of distribution of abstract nouns in cryptotypes // Proceedings of the 2010 International Conference on Artificial Intelligence, ICAI 2010. 2010. P. 907–913.
2. *Донина О. В.* Реализация концепции корпусного исследования лексики в ходе учебной практики бакалавров лингвистики // Территория науки. 2017. № 4. С. 173–177.
3. *Борискина О. О., Донина О. В.* Корпусные исследования в контексте современных технологий обучения языку // Лингвориторическая парадигма : теоретические и прикладные аспекты. 2017. № 22-2. С. 154–158.
4. *Donina O. V.* How To Use Machine Learning To Automatically Detect Dead Metaphors // RaAM14. Conference Book of Abstracts. 2021. Pp. 247–248.
5. *Дмитриев Д. С., Донина О. В.* Возможность использования методов машинного обучения для автоматического выявления стертых метафор // Лингвистический форум 2020 : язык и искусственный интеллект. М. : Институт языкознания РАН, 2020. С. 83–84.
6. *Донина О. В.* Автоматизация лингвистических исследований. Воронеж : Издательский дом ВГУ, 2022. 125 с.
7. *Sag A.* Программа для диагностирования пола и возраста автора текста с учетом возможного искажения признаков письменной речи с оценкой их эффективности. 2018. URL: https://github.com/sag111/author_gender_and_age_profiling_with_style_imitation_detection
8. *Navlani A.* Text Analytics for Beginners using NLTK. 2018. URL: <https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>
9. *Глек П.* Тьюториал : создание нейросети для анализа настроений в комментариях с Keras. 2018. URL: <https://neurohive.io/ru/tutorial/nejronnaya-set-keras-python/>
10. *MacArthur F. et al.* Metaphor in Use : Context, Culture, and Communication. John Benjamins Publishing. 2012. 379 p.
11. *Neuman Y. et al.* Metaphor Identification in Large Texts Corpora // PLoS One. 2013. № 8 (4). Pp. 36–39.
12. *Gandy L. et al.* Automatic Identification of Conceptual Metaphors with Limited Knowledge // Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence. 2013. Pp. 328–334.
13. *Shlomo Y. B., Last M.* MIL: Automatic Metaphor Identification by Statistical Learning // Proceedings of DMNLP, Workshop at ECML/PKDD. 2014. Pp. 18–29.
14. *Pechlivanis K., Konstantopoulos S.* Corpus Based Methods for Learning Models of Metaphor in Modern Greek // Statistical Language and Speech Processing : Third International Conference, SLSP. 2015. Pp. 219–229.
15. *Dodge E., Hong J., Stickles E.* MetaNet : Deep semantic automatic metaphor analysis // Proceedings of the Third Workshop on Metaphor in NLP. 2015. Pp. 40–49.
16. *Veale T., Shutova E., Klebanov B. B.* Metaphor : A Computational Perspective. Morgan & Claypool Publishers, 2016. 160 p.
17. *Shutova E., Teufel S.* Metaphor corpus annotated for source-target domain mappings // Proceedings of LREC 2010 : The 7th international conference on Language Resources and Evaluation. 2010. Pp. 3255–3261.
18. *Birke J., Sarkar A.* A Clustering Approach for the Nearly Unsupervised Recognition of Nonliteral Language // 11th Conference of the European Chapter of the Association for Computational Linguistics. 2006. Pp. 329–336.
19. *Hovy D. et al.* Identifying metaphorical word use with tree kernels // Proceedings of the First Workshop on Metaphor in NLP. 2013. Pp. 52–59.
20. *Sporleder C., Li L.* Unsupervised Recognition of Literal and Non-Literal Use of Idiomatic Expressions // Conference : EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference. 2009. Pp. 754–762.
21. *Li L., Sporleder C.* Linguistic Cues for Distinguishing Literal and Non-Literal Usages // Conference : COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume. 2010. Pp. 683–691.
22. *Klebanov B. B. et al.* Different texts, same metaphors : Unigrams and beyond // Proceedings of the Second Workshop on Metaphor in NLP. 2014. Pp. 11–17.
23. *Shutova E., Sun L.* Unsupervised metaphor identification using hierarchical graph factorization clustering // Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies. 2013. Pp. 978–988.
24. *Panicheva P. V., Badryzlova Yu. G.* Distributional Semantic Features in Russian Verbal Metaphor Identification // Computational Linguistics and Intellectual Technologies : Proceedings of the Annual International Conference “Dialogue” (2017). Moscow : RSUH, 2017. Vol. 16. Pp. 179–190.
25. *Kaushik S. et al.* Information, Communication and Computing Technology. Springer, 2017. 388 p.
26. *Bizzoni Y., Chatzikyriakidis S., Ghanimifard M.* “Deep” Learning : Detecting Metaphoricity in Adjec-

tive-Noun Pairs // Proceedings of the Workshop on Stylistic Variation. 2017. Pp. 43–52.

27. Rai S., Chakraverty S., Garg A. Effect of Classifiers on Type-III Metaphor Detection // Towards Extensible and Adaptable Methods in Computing. Springer, 2018. Pp. 241–250.

28. Rai S. et al. A Study on Impact of Context on Metaphor Detection // The Computer Journal. 2018. Vol. 61, Iss. 11. Pp. 1667–1682.

29. Сидоров К. А. и др. Возможности использования искусственных нейронных сетей для классификации текстов по варианту языка и жанру // Математика и междисциплинарные исследования – 2020 : материалы Всерос. науч.-практ. конф. молодых ученых с междунар. участием. Пермь, 2020. С. 189–193.

30. Сидоров К. А., Дони́на О. В., Коротких А. Д. Автоматизация бинарной классификации текстов английского языка по варианту языка и жанру с применением технологии искусственных нейронных сетей // Информатика : проблемы, методы, технологии : материалы XXI Междунар. науч.-метод. конф. Воронеж, 2021. С. 1508–1514.

REFERENCES

- Boriskina O. O., Marchenko T. An algorithm for analysis of distribution of abstract nouns in cryptotypes. In: *Proceedings of the 2010 International Conference on Artificial Intelligence, ICAI 2010*. 2010. Pp. 907–913.
- Donina O. V. *Realizaciya koncepcii korpusnogo issledovaniya leksiki v xode uchebnoj praktiki bakalavrov lingvistiki* [Implementation of the concept of corpus research of vocabulary during the educational practice of bachelors of Linguistics]. In: *Territoriya nauki*. 2017. No. 4. Pp. 173–177.
- Boriskina O. O., Donina O. V. *Korpusnye issledovaniya v kontekste sovremennyh tekhnologij obucheniya yazyku* [Corpus research in the context of modern language teaching technologies]. In: *Lingvoritoricheskaya paradigma: teoreticheskie i prikladnye aspekty*. 2017. No. 22-2. Pp. 154–158.
- Donina O. V. *How To Use Machine Learning To Automatically Detect Dead Metaphors*. In: *RaAMI4. Conference Book of Abstracts*. 2021. Pp. 247–248.
- Dmitriev D. S., Donina O. V. *Vozmozhnost' ispol'zovaniya metodov mashinnogo obucheniya dlya avtomaticheskogo vy'yavleniya sterty'x metafor* [using machine learning methods to automatically identify erased metaphors]. In: *Lingvisticheskij forum 2020: Yazy'k i iskusstvenny'j intellekt. Institut yazy'koznaniya RAN*, 2020. Pp. 83–84.
- Donina O. V. *Avtomatizaciya lingvisticheskix issledovaniy* [Automation of linguistic research]. Voronezh : Izdatel'skij dom VGU, 2022. 125 p.
- Sag A. *Programma dlya diagnostirovaniya pola i vozrasta avtora teksta s uchetom vozmozhnogo iskazheniya priznakov pis'mennoj rechi s ocenкой ix e'ffektivnosti* [A program for diagnosing the gender and age of the author of the text, taking into account the possible distortion of the signs of written speech with an assessment of their effectiveness]. 2018. Available at: https://github.com/sag111/author_gender_and_age_profiling_with_style_imitation_detection
- Navlani A. *Text Analytics for Beginners using NLTK*. 2018. Available at: <https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>
- Glek P. *Tutorial: sozдание nejroseti dlya analiza nastroyenij v kommentariyax c Keras* [Tutorial: creating a neural network for sentiment analysis in comments with Keras]. 2018. Available at: <https://neurohive.io/ru/tutorial/nejronnaya-set-keras-python/>
- MacArthur F., Oncins-Martinez J. L., Sánchez-García M., Piquer-Piriz A. M. *Metaphor in Use: Context, Culture, and Communication*. John Benjamins Publishing, 2012. 379 p.
- Neuman Y., Assaf D., Cohen Y., Last M., Argamon S., Howard N., Frieder O. *Metaphor Identification in Large Texts Corpora*. In: *PLoS One*. 2013. No. 8 (4). Pp. 36–39.
- Gandy L., Allan N., Atallah M., Frieder O., Howard N., Kanareykin S., Koppel M., Last M., Neuman Y., Argamon S. *Automatic Identification of Conceptual Metaphors with Limited Knowledge*. In: *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. 2013. Pp. 328–334.
- Shlomo Y. B., Last M. *MIL: Automatic Metaphor Identification by Statistical Learning*. In: *Proceedings of DMNLP, Workshop at ECML/PKDD*. 2014. Pp. 18–29.
- Pechlivanis K., Konstantopoulos S. *Corpus Based Methods for Learning Models of Metaphor in Modern Greek*. In: *Statistical Language and Speech Processing: Third International Conference, SLSP*. 2015. Pp. 219–229.
- Dodge E., Hong J., Stickles E. *MetaNet: Deep semantic automatic metaphor analysis*. In: *Proceedings of the Third Workshop on Metaphor in NLP*. 2015. Pp. 40–49.
- Veale T., Shutova E., Klebanov B. B. *Metaphor: A Computational Perspective*. Morgan & Claypool Publishers, 2016. 160 p.
- Shutova E., Teufel S. *Metaphor corpus annotated for source-target domain mappings*. In: *Proceedings of LREC 2010: The 7th international conference on Language Resources and Evaluation*. 2010. Pp. 3255–3261.
- Birke J., Sarkar A. *A Clustering Approach for the Nearly Unsupervised Recognition of Nonliteral Language*. In: *11th Conference of the European Chapter of the Association for Computational Linguistics*. 2006. Pp. 329–336.
- Hovy D., Srivastava S., Jauhar S. K., Sachan M., Goyal K., Li H., Sanders W., Hovy E. *Identifying metaphorical word use with tree kernels*. In: *Proceedings of the First Workshop on Metaphor in NLP*. 2013. Pp. 52–59.
- Sporleder C., Li L. *Unsupervised Recognition of Literal and Non-Literal Use of Idiomatic Expressions*. In: *Conference: EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*. 2009. Pp. 754–762.
- Li L., Sporleder C. *Linguistic Cues for Distinguishing Literal and Non-Literal Usages*. In: *Conference:*

COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume. 2010. Pp. 683–691.

22. Klebanov B. B., Leong B., Heilman M., Flor M. *Different texts, same metaphors: Unigrams and beyond*. In: *Proceedings of the Second Workshop on Metaphor in NLP*. 2014. Pp. 11–17.

23. Shutova E., Sun L. *Unsupervised metaphor identification using hierarchical graph factorization clustering*. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013. Pp. 978–988.

24. Panicheva P. V., Badryzlova Yu. G. *Distributional Semantic Features in Russian Verbal Metaphor Identification*. In: *Computational Linguistics and Intellectual Technologies: Proceedings of the Annual International Conference "Dialogue" (2017)*. Moscow: RSUH, 2017. Vol. 16. Pp. 179–190.

25. Kaushik S., Gupta D., Kharb L., Chahal D. *Information, Communication and Computing Technology*. Springer, 2017. 388 p.

26. Bizzoni Y., Chatzikiyiakidis S., Ghanimifard M. *"Deep" Learning: Detecting Metaphoricity in Adjective-Noun Pairs*. In: *Proceedings of the Workshop on Stylistic Variation*. 2017. Pp. 43–52.

27. Rai S., Chakraverty S., Garg A. *Effect of Classifiers on Type-III Metaphor Detection*. In: *Towards Extensible and Adaptable Methods in Computing*. Springer, 2018. Pp. 241–250.

28. Rai S., Chakraverty S., Tayal D. K., Kukreti Y. *A Study on Impact of Context on Metaphor Detection*. In: *The Computer Journal*. 2018. Vol. 61, Iss. 11. Pp. 1667–1682.

29. Sidorov K. A., Donina O. V., Korotkix A. D., Pandyurina A. A. *Vozmozhnosti ispol'zovaniya iskusstvenny'x neyronny'x setej dlya klassifikacii tekstov po variantu yazy'ka i zhanru* [using artificial neural networks to classify texts by language variant and genre]. In: *Matematika i mezhdisciplinarny'e issledovaniya – 2020. Materialy`Vserossijskoj nauchno-prakticheskoy konferencii molody'x ucheny'x s mezhdunarodny'm uchastiem*. Perm', 2020. Pp. 189–193.

30. Sidorov K. A., Donina O. V., Korotkix A. D. *Avtomatizaciya binarnoj klassifikacii tekstov anglijskogo yazy'ka po variantu yazy'ka i zhanru s primeneniem texnologii iskusstvenny'x neyronny'x setej* [Automation of binary classification of English language texts by language variant and genre using artificial neural network technology]. In: *Informatika: Problemy`, Metody`, Texnologii. Materialy` XXI Mezhdunarodnoj nauchno-metodicheskoy konferencii*. Voronezh, 2021. Pp. 1508–1514.

Воронежский государственный университет

Донина О. В., кандидат филологических наук, доцент
кафедры теоретической и прикладной лингвистики

E-mail: olga-donina@mail.ru

Поступила в редакцию 15 апреля 2022 г.

Принята к публикации 26 сентября 2022 г.

Для цитирования:

Донина О. В. Выявление метафорической сочетаемости методами машинного обучения // Вестник Воронежского государственного университета. Серия: Лингвистика и межкультурная коммуникация. 2022. № 4. С. 128–143. DOI: <https://doi.org/10.17308/lic/1680-5755/2022/4/128-143>

Voronezh State University

Donina O. V., Candidate of Philology, Associate Professor of the Theoretical and Applied Linguistics Department

E-mail: olga-donina@mail.ru

Received: 15 April 2022

Accepted: 26 September 2022

For citation:

Donina O. V. Identification of metaphors with the help of machine learning. *Proceedings of Voronezh State University. Series: Linguistics and Intercultural Communication*. 2022. No. 4. Pp. 128–143. DOI: <https://doi.org/10.17308/lic/1680-5755/2022/4/128-143>