Приложение к рабочей программе дисциплины

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Наименование дисциплины	*Анализ текстовых данных*
Направление подготовки / Специальность	*38.03.01 Экономика*
Направленность (профиль) / Специализация	*Экономика и анализ данных* ОП *ВО*
Форма обучения	*очная*

Разработчик Капелюк С.Д., профессор научно-учебной лаборатории исследований рынка труда

1. Темы дисциплины для самостоятельного освоения обучающимися
Отсутствуют.

2. План самостоятельной работы:

№ п/п	Учебные встречи	Виды самостоятельной работы	Форма отчетности / контроля	Количество баллов	Рекомендуемый бюджет времени на выполнение (ак.ч.)
1.	Введение в анализ текстов, базовые методы предобработки и выделения признаков	1. Подготовка к тестированию	1. Тестирование	-	2
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
2.	Неглубокие векторные представления слов	1. Подготовка к тестированию	1. Тестирование	-	2
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
3.	Классификация текстов	1. Подготовка к тестированию	1. Тестирование	-	2
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
4.	Разметка последовательности	1. Подготовка к тестированию	1. Тестирование	-	2
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
5.	Машинный перевод	1. Подготовка к тестированию	1. Тестирование	-	2
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	4
6.	Предобученные языковые модели-1	1. Подготовка к тестированию	1. Тестирование	-	2
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	4

7.	Предобученные языковые модели-2	1. Подготовка к тестированию	1. Тестирование	-	2
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
8.	Синтаксис в рамках грамматики зависимостей	1. Подготовка к тестированию	1. Тестирование	-	3
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
9.	Тематическое моделирование	1. Подготовка к тестированию	1. Тестирование	-	3
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
10.	Суммаризация и симплификация текстов	1. Подготовка к тестированию	1. Тестирование	-	3
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
11.	QA-системы	1. Подготовка к тестированию	1. Тестирование	-	3
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	6
12.	Графы знаний	1. Подготовка к тестированию	1. Тестирование	-	3
		2. Выполнение задания для самостоятельной работы	2. Выполнение работы	2	8
13.	Подготовка к экзамену	Изучение материалов по дисциплине по вопросам к экзамену	-	-	29
	Итого			24	144

3. Требования и рекомендации по выполнению самостоятельных работ обучающихся, критерии оценивания

Вид: Подготовка к тестированию

Краткая характеристика: вид проверки знаний и умений учащихся, который направлен на выявление степени усвоения изученного материала. Оно содержит обобщенный материал по основным изученным темам, требует от учащихся хорошей ориентировки в явлениях и фактах.

Рекомендации для подготовки:

- просмотр видеолекций;
- чтение основной и дополнительной литературы;
- повтор изученного на лекционных и практических занятиях.

Пример теста:

1. Что из нижеперечисленного можно использовать в качестве признаков при решении задачи классификации предложений? Выберите все правильные варианты
а) Векторное представление на основе синтаксического дерева предложения
б) Количество букв "a" в предложении
в) Мешок слов предложения
г) Усреднение векторов GloVe слов предложения с весами, равными нормированной частоте этих слов в Википедии
д) Значение заданной функции от скалярной метки одного из классов этого предложения
е) Наличие или отсутствие в предложении смайликов
ж) Метка типа погоды, во время которой автор написал это предложение
2. Какая из перечисленных задач не является задачей разметки последовательности?
а) Определение тональности целого предложения
б) Извлечение всех отметок времени из предложения
в) Определение грамматических характеристик каждого слова
г) Определение переключения кодов (перехода с одного языка на другой)
3. Векторное представление предложения (эмбеддинг предложения) можно получить из векторов слов, входящих в предложение. Для этого необходимо…
а) конкатенировать вектора слов
б) усреднить вектора слов
в) взять вектор первого слова
г) взять вектор символа начала предложения
4. Ниже перечислено несколько моделей векторных слов и предложений. Выберите все контекстно-зависимые модели:
а) Word2vec
б) GloVe
в) BERT
г) XLNet
д) Doc2vec
е) Ни одна из вышеперечисленных
5. Для какой задачи машинного чтения сложнее всего собирать обучающие данные?
а) Заполнение пропуска
б) Множественный выбор ответа
в) Определение свободного ответа
г) Извлечение фрагмента текста
6. Выберите все правильные ответы. На вход модели GlossBERT подается:
а) Контекст многозначного слова
б) Определение многозначного слова
в) Два контекста многозначного слова
г) Все возможные определения многозначного слова
7. Какой из фрагментов кода распечатает каждый четвёртый (при нумерации с единицы) символ строки S? Например, для строки '1234567890987654321' в результате должна получиться строка '4884'.
а) print(s[:4:])
б) print(s[::4][1::])
в) print(s[::4])
г) print(s[::4][1::4])
д) print(s[1::4])

е) print(s[1::2][1::2])
ж) print(s[::2][::2])
8. Алгоритм BPE разделяет слова:
а) На морфемы
б) На пересекающиеся n-граммы фиксированной длины
в) На непересекающиеся n-граммы фиксированной длины
г) На частые последовательности символов
9. Выберите все шаги, которые НЕ нужны для обучения мультиязычной системы, использующей принципы переноса обучения:
а) Выравнивание двух пространств языков
б) Обучение мультиязычного классификатора на данных исходного языка
в) Тестирование мультиязычного классификатора на данных целевого языка
г) Перевод данных с исходного языка на целевой
д) Обучение моноязычного классификатора на целевом языке
10. В эксперименте на одной исходной коллекции обучаются тематические модели PLSA и LDA. PLSA обучается на коллекции, в которой оставлены после фильтрации 100K слов, а LDA — на коллекции, в которой остались после фильтрации 50K слов. Количество тем в обоих случаях одинаково и равно 100. В результате обучающая перплексия в PLSA получилась равной 2400, а в LDA — 1200.Какое из утверждений верно?
а) LDA обучилась лучше PLSA с точки зрения перплексии
б) PLSA обучилась лучше LDA с точки зрения перплексии
в) Модели обучились примерно одинаково с точки зрения перплексии
г) Ничего из вышесказанного
11. Приняв гипотезу дистрибутивной семантики, запишите выражение из векторов слов "танк", "гусеницы" и "колёса", которое позволит получить вектор, близкий к слову "бронемашина".
а) танк - гусеницы + колёса
б) танк + гусеницы + колёса
в) танк + гусеницы - колёса
г) танк - гусеницы - колёса
12. Для фраз "красным цветом удобно рисовать красные цветки" и "рисуем красные цветки красиво" составить векторы "мешка слов", полученные до лемматизации. Порядок слов алфавитный, при прочих равных более короткие слова идут раньше.
а) [0,1,1,1,0,1,1,1], [1,1,0,0,1,0,1,0]
б) [0,1,1,1,0,1,1,1], [1,1,0,0,1,0,1,0]
в) [0,1,1,1,0,1,1,1], [1,1,0,0,1,0,1,0]
г) [0,2,1,1,1,1], [1,1,1,0,0,1]
13. Ускорить процесс декодирования в машинном переводе можно, используя...
а) Посимвольный машинный перевод
б) Неавторегресионные модели
в) Ансамбль декодировщиков
г) Несколько механизмов внимания
14. Как формируется словарь мультиязычной модели?
а) Конкатенация словарей для нескольких основных языков
б) Создается общий словарь для всех языков модели
в) Модель поддерживает индивидуальные словарь каждого языка
г) Ни один ответ не подходит
15. Корпус размечен по именованным сущностями с использованием IOB-разметки. В корпусе четыре типа сущностей: LOC, ORG, PER, MISC. Сколько всего различных тегов использовано для разметки корпуса?
а) 4
б) 3

в) 12
г) 9
16. Выберите все верные варианты ответа. Обратный перевод...
а) Это техника генерации синтетических данных для обучения
б) Используется как один из вариантов качества моделей машинного перевода
в) Исправляет синтаксические ошибки в данных на исходном языке
г) Способ получить золотой стандарт для последующей оценки BLEU
17. Каково назначение токена [CLS]?
а) Метка конца предложения
б) Метка начала предложения
в) Этот токен показывает, что мы хотим использовать модель BERT для классификации
г) Этот токен нужен для выравнивания длин последовательностей
18. Какие из перечисленных видов векторных представлений по построению имеют разреженную структуру векторов?
а) FastText
б) One-hot векторы
в) GloVe
г) word2vec
19. В чём преимущества параллельного кодирования реплик в модели для свободного диалога по сравнению с кросс-кодированием?
а) Работает быстрее в режиме применения
б) Работает быстрее в режиме обучения
в) Работает качественнее
г) Требует меньше памяти на хранение векторов реплик
20. Результаты решения каких из указанных задач могут быть использованы как внешние критерии оценки качества векторных представлений при условии использования этих представлений при решении указанных задач? Выберите все правильные варианты
а) Бинарная классификация спама
б) Многоклассовая классификация обращений
в) Машинный перевод
г) Кластеризация новостей
д) Информационный поиск

Вид: Выполнение задания для самостоятельной работы

Задания направлены на развитие у студентов практических навыков анализа текстовых данных: обработка текстов, построение моделей и анализ результатов. Тематика заданий охватывает широкий спектр инструментов обработки естественного языка, которые применяются в практической работе с текстовыми данными.

Рекомендации для подготовки: При выполнении заданий рекомендуется строго следовать формулировке задания и предоставлять результаты в указанном формате. Следует уделить внимание проверке корректности работы программного кода. Для успешного выполнения заданий рекомендуется уделить время изучению официальной документации по инструментам, указанным в задании.

Пример задания для самостоятельной работы к встрече «Введение в анализ текстов, базовые методы предобработки и выделения признаков»:

Воспользуйтесь фрагментом кода для загрузки текстов коллекции 20 News Group с помощью библиотеки scikit-learn: from sklearn.datasets import fetch_20newsgroups data = fetch_20newsgroups().data

1. Замените все переносы строк (символ '\n') на пробелы, токенизируйте текст по пробелам и посчитайте суммарное количество слов в коллекции без учёта пустого слова.

2. Замените все переносы строк на пробелы, токенизируйте текст по пробелам и посчитайте количество уникальных непустых слов в словаре.
3. Замените все переносы строк (символ '\n') на пробелы, удалите все символы пунктуации, содержащиеся в string.punctuation, кроме дефиса (символ '-')

- Токенизируйте текст по пробелам, приведите к нижнему регистру.
- Удалите все слова, содержащиеся в списке стоп-слов библиотеки nltk для английского языка.
- Напишите через запятую топ-5 самых частых слов, содержащих внутри себя хотя бы один дефис и хотя бы один алфавитный символ.

Пример задания для самостоятельной работы к встрече «Неглубокие векторные представления слов»:
Воспользуйтесь фрагментом кода для загрузки текстов коллекции 20 News Group с помощью библиотеки scikit-learn:from sklearn.datasets import fetch_20newsgroups data = fetch_20newsgroups().data

- Замените все переносы строк на пробелы, удалите все символы пунктуации, содержащиеся в string.punctuation.
- Токенизируйте текст по пробелам, приведите к нижнему регистру.
- Удалите все слова, содержащиеся в списке стоп-слов библиотеки nltk для английского языка. Должно получиться примерно 147000 слов в словаре.
- Постройте на этих данных матрицу TF-IDF с помощью векторизатора из scikit-learn с параметрами по умолчанию (токены внутри документа нужно объединить в строку через пробел).
- Постройте усечённое SVD-разложение этой матрицы с 50 компонентами с помощью TruncatedSVD из scikit-learn. Возьмите в качестве векторов слов матрицу из поля components_ получившейся модели и в качестве словаря — значение, возвращаемое методом get_feature_names обученного TF-IDF векторизатора.

Найдите самое близкое по евклидовой метрике на этих векторах слово к слову "car", которое не является самим этим словом или его формой. Выберите это слово из предложенных вариантов.

Пример задания для самостоятельной работы к встрече «Классификация текстов»:
Загрузите данные из файла harry_ech.csv и проверьте распределение классов. Замените все переносы строк и другие служебные символы на пробелы, удалите пунктуацию, токенизируйте текст, удалите стоп-слова, выполните лемматизацию. Замените модель векторизации на модель fasttext, используя предобученные вектора с Wikipedia. Создайте функцию для получения векторного представления текста, усредняя вектора слов.
Подготовьте переменную отклика, заменив классы на числовые значения. Разделите данные на обучающую и тестовую выборки (75% и 25% соответственно). Обучите модель логистической регрессии с кросс-валидацией и параметрами по умолчанию. Оцените качество модели, используя метрики accuracy, precision, recall и f1-measure.

Пример задания для самостоятельной работы к встрече «Разметка последовательности»:
Подготовьте данные и обучите модель для разметки последовательностей в художественной литературе.

1. Используйте корпус LitBank, который собран из популярных художественных произведений на английском языке и содержит разметку по именованным сущностям и событиям. Объем корпуса: 100 текстов по примерно 2000 слов каждый. Данные организованы следующим образом: тексты поделены на предложения, каждое предложение отделено пустой строкой. Токены находятся на отдельных строках, каждому токену соответствует тег в BIO-нотации.
2. Напишите функцию read_bio_dataset(dataset_fpath), которая считывает набор данных и

возвращает два списка: sentences_tokens и sentences_tags. Каждый элемент списка - это список токенов или соответствующих тегов отдельного предложения.
3. Считайте данные для обучения и тестирования, используя функцию read_bio_dataset.
4. Постройте словарь vocab, содержащий все уникальные токены в наборе данных, и набор тегов tagset.
5. Преобразуйте обучающие и тестовые последовательности токенов и тегов в последовательности чисел, используя словари word2ix и tag2ix.
6. Реализуйте наивную модель, которая присваивает каждому токену самый частый встретившийся рядом с ним тег.
7. Оцените качество наивной модели на тестовой выборке, используя метрики mean и joint.
8. Реализуйте нейросетевой бейзлайн для разметки последовательностей.
9. Разбейте данные на батчи с помощью функции generate_batched_dataset.
10. Обучите нейросетевую модель на тренировочных данных и оцените её на валидационной и тестовой выборках.

Пример задания для самостоятельной работы к встрече «Машинный перевод»:
Допустим, модель машинного перевода предложила следующий вариант перевода.
MT1: 'Задача машинного перевода — одна из первых и самых трудных в области обработки текстов'.
Золотой стандарт перевода таков: 'Машинный перевод — одна из первых и самых трудных задач обработки текстов'.
Оцените качество перевода с помощью метрики BLEU.

Пример задания для самостоятельной работы к встрече «Предобученные языковые модели-1»:
Дан фрагмент стихотворения Эдгара Аллана По:
sample_txt = "*Once upon a midnight dreary, while I pondered, weak and weary, \ Over many a quaint and curious volume of forgotten lore — \ While I nodded, nearly napping, suddenly there came a tapping, \ As of some one gently rapping, rapping at my chamber door. \ 'Tis some visitor,' I muttered, 'tapping at my chamber door— \ Only this and nothing more.*'"
Используйте BertTokenizer и посчитайте количество токенов в данном тексте. Не используйте специальные токены. Сколько токенов получилось?

Пример задания для самостоятельной работы к встрече «Предобученные языковые модели-2»:
Выполните классификацию с помощью модели RoBERTa на базе архитектуры BERT:
1. Установите библиотеку transformers и загрузите модуль pipeline, в котором находится нужная модель для классификации с использованием подхода zero-shot.
2. Используйте модель для классификации текстов по заранее заданным классам. Для этого подайте текст и список классов, среди которых необходимо выбрать наиболее подходящий для текста.
3. Протестируйте модель на задаче многоклассовой классификации, используя дополнительные параметры, чтобы получить вероятности для всех классов независимо друг от друга.
4. Исследуйте классификацию тональности текста (например, классификация отзывов как положительный или отрицательный) с использованием модели.
5. Примените модель для классификации текста на разных языках, например, используйте текст на русском языке с английскими или французскими метками классов. Пример текста: "За кого вы голосуете в 2020 году?" (перевод: "Who are you voting for in 2020?").
6. Подберите подходящий шаблон для каждого из языков, чтобы улучшить качество

классификации, учитывая особенности языка.
7. Воспользуйтесь кросс-язычной моделью XLM-RoBERTa для классификации текстов, написанных на разных языках, и протестируйте модель на нескольких языках (например, русский, испанский, французский).

Пример задания для самостоятельной работы к встрече «Синтаксис в рамках грамматики зависимостей»:
1. Используя векторизатор TF-IDF из библиотеки scikit-learn, постройте матрицу TF-IDF для текстов, представленных в виде строк. При этом токены внутри каждого документа должны быть объединены в строку через пробел.
2. Выполните усечённое SVD-разложение этой матрицы с помощью метода TruncatedSVD из scikit-learn, установив количество компонент равным 50. Для этого создайте объект модели TruncatedSVD и примените его к матрице TF-IDF.
3. После выполнения SVD-разложения используйте матрицу компонент, сохранённую в поле components_ модели TruncatedSVD, и извлеките словарь, вызвав метод get_feature_names_out() на обученном векторизаторе TF-IDF.
4. Найдите самое близкое слово по евклидовой метрике к слову "car" среди векторов слов, полученных после разложения. Исключите слово "car" и его формы из возможных вариантов. Выберите слово, которое окажется самым близким к слову "car".
Для выполнения задания используйте векторизацию текста и методы из библиотеки scikit-learn для обработки и анализа данных.

Пример задания для самостоятельной работы к встрече «Тематическое моделирование»:
Задача заключается в поиске T тем, описывающих документы коллекции D со словарём W, где темы представляют собой наборы слов, которые их характеризуют. Большинство тематических моделей оперирует данными в формате "мешка слов", т.е. учитывают только частоты слов в документах, а не их порядок.
1. Скачайте первые 10000 строк дампа Wikipedia в формате vowpal wabbit и сохраните их в файл wiki-enru-10000.txt.
2. Посмотрите, как выглядит набор данных, и переведите vowpal wabbit представление данных в формат "мешка слов" (BoW), читаемый библиотекой scikit-learn.
3. Загрузите данные в BoW-векторизатор sklearn.
4. Проверьте размерности BoW-матрицы и убедитесь, что количество токенов и документов соответствует ожиданиям.
5. Используйте реализацию модели LDA из библиотеки gensim для тематического моделирования.
6. Преобразуйте BoW данные в формат, совместимый с gensim (корпус и словарь).
7. Обучите тематическую модель с различными количествами тем (2, 4, 6, 8, 10, 12, 14, 16, 18).
8. Визуализируйте обученную модель LDA с помощью библиотеки pyLDAvis.
9. Определите распределение указанных слов по темам для различных моделей.
10. Подготовьте tsv-файл, содержащий информацию о распределении слов по темам. Формат файла:
- Колонки: word, topic, model_num_topics.
- Заголовок файла обязателен.
- Слова для анализа: ['банан', 'рыба', 'генерал', 'трек', 'интернет', 'войско', 'веб', 'сингл'].

Пример задания для самостоятельной работы к встрече «Суммаризация и симплификация текстов»:
В этом задании вам предстоит познакомиться с алгоритмом суммаризации текстов *TextRank*, основанным на методах графов, и реализовать его для извлечения краткого содержания из новостных статей.

Для этого необходимо выполнить следующие шаги:

1. Загрузите подвыборку из 300 текстов новостей CNN/DailyMail.
2. Разделите каждый текст на предложения и выполните токенизацию каждого предложения. Используйте библиотеку NLTK для токенизации.
3. Загрузите предобученные векторы слов Glove и создайте векторные представления для предложений, используя взвешенные средние значения векторов слов с весами tf-idf. Для этого примените класс TfidfEmbeddingVectorizer.
4. Постройте граф, где каждое предложение будет представлять вершину. Вес ребра между двумя вершинами будет равен косинусному расстоянию между векторными представлениями соответствующих предложений.
5. Реализуйте алгоритм *PageRank* для оценки важности предложений в тексте. Алгоритм должен вычислять значения для всех предложений, основываясь на их связях в графе.
6. Для каждого текста выберите предложения с наибольшими значениями *PageRank* и составьте из них краткое содержание.
7. Реализуйте функцию, которая будет принимать текст, разбитый на предложения, и возвращать его краткое содержание, состоящее из 5 наиболее важных предложений по версии *TextRank*.

Пример задания для самостоятельной работы к встрече «QA-системы»:

Пусть ответ на вопрос в IR-based QA-системе ищется в параграфе из шести токенов с помощью модели типа DrQA. Для каждого токена вычислены вероятности быть стартовым или завершающим токеном ответа.

Токены и их вероятности:

TOKEN	W_1	W_2	W_3	W_4	W_5	W_6
P_{start}	0.4	0.6	0.4	0.3	0.4	0.2
P_{finish}	0.1	0.3	0.4	0.5	0.7	0.1

Укажите все токены, которые войдут в ответ, при условии, что длина его не больше трёх токенов.

Пример задания для самостоятельной работы к встрече «Графы знаний»:

Используя библиотеку Stanza, создайте программу для анализа текстов на английском и русском языках. Необходимо реализовать следующие шаги:

1. Установить библиотеку Stanza и загрузить языковые модели для английского и русского языков.
2. Создать конвейер обработки текста (pipeline) с использованием процессоров для токенизации, лемматизации и определения частей речи.
3. Написать функцию для лемматизации входного текста, которая будет сохранять результаты в pandas DataFrame с колонками 'word' (исходное слово) и 'lemma' (начальная форма слова).
4. Реализовать функцию для разбора частей речи в тексте, которая будет сохранять результаты в pandas DataFrame с колонками 'word' (слово), 'pos' (часть речи) и 'exp' (расшифровка части речи).
5. Протестировать работу программы на примере английского текста: "The prospects for Britain's orderly withdrawal from the European Union on March 29 have receded further, even as MPs rallied to stop a no-deal scenario."
6. Протестировать работу программы на примере русского текста: "Мы проходим курс по изучению методов обработки естественного языка."
7. Дополнительно реализовать извлечение отношений из текста с помощью модуля OpenIE, создав визуализацию полученных связей в виде направленного графа. Для тестирования использовать простое предложение: "Barack Obama was born in Hawaii".

4. Рекомендации по самоподготовке к промежуточной аттестации по дисциплине

Вопросы для самопроверки к экзамену

1. Задачи анализа текстов: специфика и история.
2. Области применения обработки естественного языка.
3. Лингвистика и NLP: взаимосвязь и отличие.
4. Предобработка текстовых данных: очистка и нормализация.
5. Простейшие текстовые признаки: "мешок слов" и TF-IDF.
6. Выделение признаков из текстов: основные подходы.
7. Предобработка текстовых данных: регулярные выражения.
8. Неглубокие векторные представления слов.
9. Идея векторных представлений: one-hot-векторы и SVD.
10. Модель word2vec: методы и обучение.
11. Оптимизация обучения word2vec: SGNS и иерархический softmax.
12. Модель GloVe: основы и применение.
13. Модель и библиотека FastText, приём Hashing Trick.
14. Оценка качества векторных представлений слов.
15. Классификация текстов: постановка задачи.
16. Логистическая регрессия на счётчиках и TF-IDF.
17. Неглубокие векторные представления документов.
18. Библиотека FastText для классификации текстов.
19. Неглубокие нейросетевые модели классификации.
20. Работа с обучающими данными для классификации текстов.
21. Разметка последовательности: основные методы.
22. Основы построения счетных языковых моделей.
23. Морфологический анализ: принципы и инструменты.
24. Скрытые Марковские модели: теория и использование.
25. Нейросетевые языковые модели: введение и классификация.
26. Генерация текстов: основные подходы.
27. Языковые модели для генерации текстов: вероятностные и нейросетевые методы.
28. Извлечение именованных сущностей (NER).
29. Перекрестное обучение: концепция и примеры.
30. Машинный перевод: основные понятия и задачи.
31. Модели класса кодировщик-декодировщик.
32. Механизм внимания в моделях машинного перевода.
33. Метрики качества в машинном переводе.
34. Предобученные языковые модели.
35. Векторное представление предложений: методы и примеры.
36. Модель ELMo.
37. Модель BERT.
38. Модель GPT2.
39. Оценка качества языковых моделей: метрики и подходы.
40. Синтаксис в рамках грамматики зависимостей.
41. Автоматический синтаксический анализ.
42. Теоретические подходы к синтаксическому анализу.
43. Парсинг на основе грамматики зависимостей.
44. Transition-Based Dependency Parsing (DP).
45. Метрики и соревнования в синтаксическом анализе.
46. Инструменты для синтаксического анализа.
47. Тематическое моделирование: основные понятия и методы.
48. Постановка задачи тематического моделирования.

49. Модель PLSA (Probabilistic Latent Semantic Analysis).
50. EM-алгоритм.
51. Модель ARTM (Additive Regularization of Topic Models).
52. Модель LDA (Latent Dirichlet Allocation).
53. Модель M-ARTM (Multimodal Additive Regularization of Topic Models).
54. Суммаризация текстов: основные методы.
55. Абстрактивная суммаризация.
56. Упрощение текстов.
57. Основы разработки чат-ботов.
58. Инструменты разработки чат-ботов.
59. Виртуальные ассистенты.
60. Вопросно-ответные системы.
61. Машинное чтение.
62. Вопросно-ответные системы в индустрии.
63. Методы извлечения графа знаний.
64. Машинное обучение для извлечения графа знаний.