Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Филологический факультет

УТВЕРЖДЕНО:
Декан
И. В.Тубалова

Рабочая программа дисциплины

Технологии автоматической обработки текста

по направлению подготовки

45.03.03 Фундаментальная и прикладная лингвистика

Направленность (профиль) подготовки:
Фундаментальная и прикладная лингвистика

Форма обучения
Очная

Квалификация
Бакалавр

Год приема
2024

СОГЛАСОВАНО:
Руководитель ОП
А.В. Васильева

Председатель УМК
Ю.А. Тихомирова

Томск – 2024

1. Цель и планируемые результаты освоения дисциплины (модуля)

Целью дисциплины является изучение основных принципов и методов автоматической обработки текстов на естественном языке (ЕЯ)

Целью освоения дисциплины является формирование следующих компетенций:

ИПК-4.1 Применяет способы формализации и алгоритмизации поставленных задач в сфере автоматической обработки текстов

2. Задачи освоения дисциплины

– Изучение методов обработки естественного языка, применение междисциплинарных методов в обработке исследовательских данных.

– Научиться применять понятийный математический аппарат в области лингвистики для решения практических задач профессиональной деятельности.

– Приобрести навыки хранения, структуризации, анализа и визуализации текстового массива данных

3. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина относится к обязательной части образовательной программы.

4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине

Семестр 7, экзамен.

5. Входные требования для освоения дисциплины

Для успешного освоения дисциплины требуются компетенции, сформированные в ходе освоения образовательных программ предшествующего уровня образования.

Для успешного освоения дисциплины требуются результаты обучения по следующим дисциплинам: «Введение в языкознание», «Общая фонетика», «Общая морфология», «Общий синтаксис», «Общая семантика», «Информационные технологии и основы информационной культуры в лингвистике», «Информатика и основы программирования», «Квантитативные методы лингвистики», «Вероятностные модели», «Лингвистические базы данных».

6. Язык реализации

Русский

7. Объем дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 з.е., 144 часов, из которых:
– лекции: 20 ч.;
– семинарские занятия: 0 ч.
– практические занятия: 30 ч.;
– лабораторные работы: 0 ч.
в том числе практическая подготовка: 30 ч.

Объем самостоятельной работы студента определен учебным планом.

8. Содержание дисциплины (модуля), структурированное по темам

Тема 1. Автоматическая обработка устной речи

Цель и задачи автоматической обработки устной речи. Метод Байеса. Архитектура систем автоматической обработки текстов

Тема 2. Автоматы, формальные грамматики и языки

Лингвистический автомат. Уровневое построение систем АОТ и ЛА. Подблок опознания формата текста и его частей, а также определение их жанровой и тематической принадлежности

Тема 3. Морфологический анализ в системах автоматической обработки текста

Использование и запуск морфологического анализатора mystem в языке программирования R. Квантитативный анализ частей речи, их визуализация и анализ

Тема 4. Синтаксический анализ в системах автоматической обработки текста

Использование и запуск морфологического анализатора udpipe в языке программирования R. Квантитативный анализ частей речи, их визуализация и анализ

Тема 5. Семантический анализ в системах автоматического анализа текста

Формальные грамматики, извлечение сущностей из текста (нейронные сети и/или формальныеграмматики). Tomita parser, Spacy

Тема 6. Словарная поддержка. Типы словарей. Компьютерные (электронные) словари.

Создание тематических словарей для классификации текстов (sentiment analysis)

Тема 7. Синтез текстов на естественном языке

Понятие нейронной сети, принципы работы. Искусственные нейронные сети: keras, ruGPT-3, seq2seq

Тема 8. Морфологическая разметка корпусов текстов. Корпус русского языка

Принципы разметки, виды и типы морфологических теггеров.

Тема 9. Автоматическая обработка данных в корпусах русского языка и BNC

Поиск и сравнение лексем в корпусах, метрики сравнения: IPM, TF-IDF, LL-score, коэффициент Жуйана.

Тема 10. Итоговая презентация проекта

9. Текущий контроль по дисциплине

Текущий контроль успеваемости обучающихся направлен на определение соответствия результатов обучения после освоения элемента по дисциплине проводится путем контроля посещаемости, проведения контрольных работ, тестов по лекционному материалу, разработки кода, выполнения домашних заданий и фиксируется в форме контрольной точки не менее одного раза в семестр. Примерные задания текущего контроля:

Примерные тестовые задания по 1 модулю

1. Разработать парсер новостного сайта, включающий категории текста
2. Разработать парсер сайта «Отзовик», включающий категории текста
3. Построить частотную матрицу относительных величин

10. Порядок проведения и критерии оценивания промежуточной аттестации

Экзамен состоит из двух частей и предполагает устную и письменную формы взаимдоействия, направленных на оценку сформированности компетенции ИПК-4.1.

Первая часть состоит из теоретических вопросов, предпологающих устный и письменный ответы

Вторая часть предусматривает защиту проекта, програмного кода и презентации

Примерный перечень теоретических вопросов

1. Опишите типологизацию направления «Обработка естественного языка»
2. Опишите методы векторизации слов? Какие методы существуют их преимущества и недостатки
3. Опишите принцип работы метода векторизации word2vec, опишите его преимущества и недостатки
4. Опишите принцип работы метода векторизации FastText, опишите его преимущества и недостатки
5. Дайте определение расстояния. В каких методах обработки естественного языка применяется данный подход?

6. Автоматическая обработка естественного языка, ее цели и задачи. Предмет и объект данной области знаний.

7. Принцип работы контекстно свободных грамматик

8. Опишите принцип работы Word Embeddings: независимые от контекста представления слов

9. Опишите принцип работы Word Embeddings: зависимые от контекста представления слов

10. Обучение интеллектуальных систем. Виды обучающихся интеллектуальных систем

11. Предобработка данных, типы предобработки, ее особенности.

12. Задачи и методы генерации текстов. Файнтьюн обученной модели.

13. Принципы работы нейронной сети GPT-3.

14. Опишите методы извлечения фактов из текста. Опишите их преимущества и недостатки.

Примерный перечень практических вопросов:

Измените структуру кода для своих данных:

```
shapiro.test(w_dict_dfm_mycorp_promouns_full$you)
cor(w_dict_dfm_mycorp_promouns_full[,2:5], method = "spearman")#pearson
cor.test(w_dict_dfm_mycorp_int[,2:5], method = "spearman")
boxplot(w_dict_dfm_mycorp_promouns_full$you ~
        w_dict_dfm_mycorp_promouns_full$ComType)
summary(w_dict_dfm_mycorp_promouns_full$you)
"pearson" - парам
"sperman/kendall" - непарам
help(cor.test)
cor.test(w_dict_dfm_mycorp_promouns_full$personal,
w_dict_dfm_mycorp_promouns_full$we,
         method = "spearman")
w_dict_dfm_mycorp_dlgs2 <- w_dict_dfm_mycorp_dlgs
w_dict_dfm_mycorp_dlgs2  <-  as.data.frame(w_dict_dfm_mycorp_dlgs2 )
w_dict_dfm_mycorp_dlgs2 <- as.factor(w_dict_dfm_mycorp_dlgs2$ComType)
w_dict_dfm_mycorp_dlgs2$personal <-  as.integer(w_dict_dfm_mycorp_dlgs2$)
boxplot(w_dict_dfm_mycorp_promouns_full$personal ~
w_dict_dfm_mycorp_promouns_full$ComType)
colnames(w_dict_dfm_mycorp_promouns_full) <- c("doc_id",  "я", "ты" ,  "мы" ,
"сам" ,  "ComType",  "gnd")
t.test(w_dict_dfm_mycorp_promouns_full$я ~ w_dict_dfm_mycorp_promouns_full$gnd)
w_dict_dfm_full2 <- w_dict_dfm_full
w_dict_dfm_mycorp_vk2 <- w_dict_dfm_mycorp_vk
dim(w_dict_dfm_mycorp_dlgs2)
dim(w_dict_dfm_full2)
dim(w_dict_dfm_mycorp_vk2)
w_dict_dfm_mycorp_dlgs2$ComType <-"dlgs"
w_dict_dfm_full2$ComType <- "intv"
w_dict_dfm_mycorp_vk2$ComType <- "wllVK"
w_dict_dfm_full <-
rbind(w_dict_dfm_mycorp_dlgs2,w_dict_dfm_full2,w_dict_dfm_mycorp_vk2)

summary(w_dict_dfm_full)
tapply(w_dict_dfm_mycorp_promouns_full$personal,
     w_dict_dfm_mycorp_promouns_full$gnd, summary)
tapply(w_dict_dfm_full$you, w_dict_dfm_full$ComType, summary)
```

```
tapply(w_dict_dfm_full$we, w_dict_dfm_full$ComType, summary)
tapply(w_dict_dfm_full$self, w_dict_dfm_full$ComType, summary)

boxplot(w_dict_dfm_full$personal ~ w_dict_dfm_full$ComType)
boxplot(w_dict_dfm_full$you ~ w_dict_dfm_full$ComType)
boxplot(w_dict_dfm_full$self ~ w_dict_dfm_full$ComType)
boxplot(w_dict_dfm_full$we ~ w_dict_dfm_full$ComType)

myglm <- glm(w_dict_dfm_full$ComType ~ w_dict_dfm_full$personal + w_dict_dfm_full$you+w_dict_dfm_full$we+
               w_dict_dfm_full$self)

kruskal.test(w_dict_dfm_full$personal ~ w_dict_dfm_full$ComType)
kruskal.test(w_dict_dfm_full$you ~ w_dict_dfm_full$ComType)
kruskal.test(w_dict_dfm_full$self ~ w_dict_dfm_full$ComType)
kruskal.test(w_dict_dfm_full$we ~ w_dict_dfm_full$ComType)

shapiro.test(x_m2$neg)


library(outliers)
grubbs.test(w_dict_dfm_full$personal, type = 10)
grubbs.test(w_dict_dfm_full$you, type = 10)
grubbs.test(w_dict_dfm_full$self, type = 10)
grubbs.test(w_dict_dfm_full$we, type = 10)
library(ggplot2)
p = ggplot(w_dict_dfm_mycorp_promouns_full[,-1], aes(x=personal))
(p <- p+geom_density(aes(fill=ComType), alpha=1/2))
w_dict_dfm_full <- w_dict_dfm_mycorp_promouns_full
# Sample data
data <- w_dict_dfm_mycorp_promouns_full[, 2:5] # Numerical variables
groups <- as.factor(w_dict_dfm_mycorp_promouns_full[, 7]) # Factor variable (groups)
# Plot correlation matrix
pairs(data)

# Equivalent with a formula
pairs(~ personal + you + self + we, data = w_dict_dfm_mycorp_vk)


pairs(data,                  # Data frame of variables
      labels = colnames(data),  # Variable names
      pch = 1,               # Pch symbol
      bg = rainbow(2)[groups],  # Background color of the symbol (pch 21 to 25)
      col = rainbow(2)[groups], # Border color of the symbol
      main = "",    # Title of the plot
      row1attop = TRUE,        # If FALSE, changes the direction of the diagonal
      gap = 1,                 # Distance between subplots
      cex.labels = NULL,       # Size of the diagonal text
      font.labels = 1)         # Font style of the diagonal text

panel.hist <- function(x, ...) {
  usr <- par("usr")
```

```
  on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5))
  his <- hist(x, plot = FALSE)
  breaks <- his$breaks
  nB <- length(breaks)
  y <- his$counts
  y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = rgb(0, 1, 1, alpha = 0.5), ...)
  # lines(density(x), col = 2, lwd = 2) # Uncomment to add density lines
}

# Creating the scatter plot matrix
pairs(data,
      upper.panel = NULL,         # Disabling the upper panel
      diag.panel = panel.hist)    # Adding the histograms
# Function to add correlation coefficients
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {
  usr <- par("usr")
  on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  Cor <- abs(cor(x, y)) # Remove abs function if desired
  txt <- paste0(prefix, format(c(Cor, 0.123456789), digits = digits)[1])
  if(missing(cex.cor)) {
    cex.cor <- 0.4 / strwidth(txt)
  }
  text(0.5, 0.5, txt,
       cex = 1 + cex.cor * Cor) # Resize the text by level of correlation
}

# Plotting the correlation matrix
pairs(data,
      upper.panel = panel.cor,    # Correlation panel
      lower.panel = panel.smooth) # Smoothed regression lines

# install.packages("gclus")
library(gclus)

# Correlation in absolute terms
corr <- abs(cor(data))

colors <- dmat.color(corr)
order <- order.single(corr)

cpairs(data,                 # Data frame of variables
       order,                # Order of the variables
       panel.colors = colors,   # Matrix of panel colors
       border.color = "grey70", # Borders color
       gap = 0.45,              # Distance between subplots
       main = "Ordered variables colored by correlation", # Main title
       show.points = TRUE,      # If FALSE, removes all the points
       pch = 21,                # pch symbol
       bg = rainbow(2)[groups]) # Colors by group
```

```
#install.packages("psych")
library(psych)

pairs.panels(data,
             smooth = TRUE,      # If TRUE, draws loess smooths
             scale = FALSE,      # If TRUE, scales the correlation text font
             density = TRUE,     # If TRUE, adds density plots and histograms
             ellipses = TRUE,    # If TRUE, draws ellipses
             method = "spearman", # Correlation method (also "spearman" or "kendall")
             pch = 21,           # pch symbol
             lm = FALSE,         # If TRUE, plots linear fit rather than the LOESS (smoothed)
fit
             cor = TRUE,         # If TRUE, reports correlations
             jiggle = FALSE,     # If TRUE, data points are jittered
             factor = 2,         # Jittering factor
             hist.col = 4,       # Histograms color
             stars = TRUE,       # If TRUE, adds significance level with stars
             ci = TRUE)          # If TRUE, adds confidence intervals

library(psych)

corPlot(data, cex = 1.2)

library(ggplot2)
# install.packages("ggExtra")
library(ggExtra)
p_base <- ggplot(w_dict_dfm_full,aes(x=we,y=self,color=gnd)) + geom_point()
ggExtra::ggMarginal(p_base, groupColour = TRUE, groupFill = TRUE)
library(otrimle)
clus <- otrimleg(w_dict_dfm_full[,c(1,2)], G=2:5, monitor=1) # параметр monitor
позволяет видеть ход выполнения



# Equivalent but using the plot function
plot(data)
library(tidyverse)
cor(w_dict_dfm_full[,2:4] %>% w_dict_dfm_mycorp_vk$gnd=="m")
w_dict_dfm_mycorp_vk  %>%
  group_by(gnd) %>%
  plot(w_dict_dfm_mycorp_vk$self)




w_dict_dfm_full$gnd <- as.factor(w_dict_dfm_full$gnd)
w_dict_dfm_full_01 <- w_dict_dfm_full
w_dict_dfm_full_01$gnd01[w_dict_dfm_full$gnd=="m"] <- 1
w_dict_dfm_full_01$gnd01[w_dict_dfm_full$gnd=="f"] <- 0
```

```
fitglm <- glm(w_dict_dfm_full_01$gnd01 ~ w_dict_dfm_full_01$personal - w_dict_dfm_full_01$you +
                w_dict_dfm_full_01$we +w_dict_dfm_full_01$self)

summary(fitglm)
anova(fitglm, test = "Chisq")

summary(w_dict_dfm_full)
tapply(w_dict_dfm_full$personal, w_dict_dfm_full$gnd, summary)
tapply(w_dict_dfm_full$you, w_dict_dfm_full$gnd, summary)
tapply(w_dict_dfm_full$we, w_dict_dfm_full$gnd, summary)
tapply(w_dict_dfm_full$self, w_dict_dfm_full$gnd, summary)
shapiro.test(w_dict_dfm_full$personal)
shapiro.test(w_dict_dfm_full$you)
shapiro.test(w_dict_dfm_full$we)
shapiro.test(w_dict_dfm_full$self)
wilcox.test(w_dict_dfm_full$personal~w_dict_dfm_full$gnd)
wilcox.test(w_dict_dfm_full$you~w_dict_dfm_full$gnd)
wilcox.test(w_dict_dfm_full$we~w_dict_dfm_full$gnd)
wilcox.test(w_dict_dfm_full$self~w_dict_dfm_full$gnd)
wilcox.test(w_dict_dfm_full$self+w_dict_dfm_full$personal+w_dict_dfm_full$you+w_dict_dfm_full$we~w_dict_dfm_full$gnd)
shapiro.test(c(w_dict_dfm_full$self+w_dict_dfm_full$personal+w_dict_dfm_full$you+w_dict_dfm_full$we))

kruskal.test(w_dict_dfm_full$personal~w_dict_dfm_full$ComType)
kruskal.test(w_dict_dfm_full$you~w_dict_dfm_full$ComType)
kruskal.test(w_dict_dfm_full$we~w_dict_dfm_full$ComType)
kruskal.test(w_dict_dfm_full$self~w_dict_dfm_full$ComType)
shapiro.test(c(w_dict_dfm_full$self+w_dict_dfm_full$personal+w_dict_dfm_full$you+w_dict_dfm_full$we))
boxplot(stacked_df$values ~ stacked_df$ind,
      col = rainbow(ncol(trees)))

boxplot(w_dict_dfm_full$personal ~ w_dict_dfm_full$gnd)
boxplot(w_dict_dfm_full$we ~ w_dict_dfm_full$gnd)
boxplot(w_dict_dfm_full$self ~ w_dict_dfm_full$gnd)
boxplot(w_df_dict_dfm_mycorp_int$you ~ norm_df_dict_dfm_mycorp_int$gnd)
boxplot(w_df_dict_dfm_mycorp_int$self ~ norm_df_dict_dfm_mycorp_int$gnd)
boxplot(w_df_dict_dfm_mycorp_int$we ~ norm_df_dict_dfm_mycorp_int$gnd)

boxplot(w_df_dict_dfm_mycorp_dlgs$personal ~ norm_df_dict_dfm_mycorp_dlgs$gnd)
boxplot(w_df_dict_dfm_mycorp_dlgs$you ~ norm_df_dict_dfm_mycorp_dlgs$gnd)
boxplot(w_df_dict_dfm_mycorp_dlgs$self ~ norm_df_dict_dfm_mycorp_dlgs$gnd)
boxplot(w_df_dict_dfm_mycorp_dlgs$we ~ norm_df_dict_dfm_mycorp_dlgs$gnd)

boxplot(w_df_dict_dfm_mycorp_vk$personal ~ norm_df_dict_dfm_mycorp_vk$gnd)
boxplot(w_df_dict_dfm_mycorp_vk$you ~ norm_df_dict_dfm_mycorp_vk$gnd)
boxplot(w_df_dict_dfm_mycorp_vk$self ~ norm_df_dict_dfm_mycorp_vk$gnd)
boxplot(w_df_dict_dfm_mycorp_vk$we ~ norm_df_dict_dfm_mycorp_vk$gnd)
#Statistica
shapiro.test(norm_df_dict_dfm_mycorp_int$personal)
```

```
shapiro.test(norm_df_dict_dfm_mycorp_int$you)
shapiro.test(norm_df_dict_dfm_mycorp_int$self)
shapiro.test(norm_df_dict_dfm_mycorp_int$we)
wilcox.test(norm_df_dict_dfm_mycorp_int$personal ~ norm_df_dict_dfm_mycorp_int$gnd) #W = 3190, p-value = 0.0001323
wilcox.test(norm_df_dict_dfm_mycorp_int$you ~ norm_df_dict_dfm_mycorp_int$gnd) #W = 2748, p-value = 0.0548
wilcox.test(norm_df_dict_dfm_mycorp_int$self ~ norm_df_dict_dfm_mycorp_int$gnd) #W = 2709, p-value = 0.08177
wilcox.test(norm_df_dict_dfm_mycorp_int$we ~ norm_df_dict_dfm_mycorp_int$gnd) #W = 2911, p-value = 0.009087

shapiro.test(norm_df_dict_dfm_mycorp_vk$personal)
shapiro.test(norm_df_dict_dfm_mycorp_vk$you)
shapiro.test(norm_df_dict_dfm_mycorp_vk$self)
shapiro.test(norm_df_dict_dfm_mycorp_vk$we)
wilcox.test(norm_df_dict_dfm_mycorp_vk$personal ~ norm_df_dict_dfm_mycorp_vk$gnd) #W = 3190, p-value = 0.0001323
wilcox.test(norm_df_dict_dfm_mycorp_vk$you ~ norm_df_dict_dfm_mycorp_vk$gnd) #W = 2748, p-value = 0.0548
wilcox.test(norm_df_dict_dfm_mycorp_vk$self ~ norm_df_dict_dfm_mycorp_vk$gnd) #W = 2709, p-value = 0.08177
wilcox.test(norm_df_dict_dfm_mycorp_vk$we ~ norm_df_dict_dfm_mycorp_vk$gnd) #W = 2911, p-value = 0.009087
norm_df_dict_dfm_mycorp_vk$gnd <- as.factor(norm_df_dict_dfm_mycorp_vk$gnd)

library(caret)
validation_index <- createDataPartition(w_dict_dfm_mycorp_promouns_full$ComType, p=0.80, list=FALSE)
# select 20% of the data for validation
test <- w_dict_dfm_mycorp_promouns_full[-validation_index,]
# use the remaining 80% of data to training and testing the models
train <- w_dict_dfm_mycorp_promouns_full[validation_index,]
test <- test[,-c(1,7)]
train <- train[,-c(1,7)]

library(rpart)
##Trees
author.tree<-rpart(ComType~.,data=train)
plot(author.tree,margin=0.1,uniform=T)
text(author.tree,use.n=T)
printcp(author.tree)
plotcp(author.tree)

author.predict.train<-predict(author.tree,train,type = "class")
ttreetrain<-table(train$ComType,author.predict.train)
error(ttreetrain)

author.predict.test<-predict(author.tree,test,type = "class")
ttreetest<-table(test$ComType,author.predict.test)
error(ttreetest)
```

```
error <- function(err) {
  acc = sum(diag(ttreetest))/sum(err)
  print(paste("Accuracy=", acc))
}

sum(ttreetest[,1])
confusionMatrix(data = data.frame(author.predict.test), test$ComType)

caretsum(ttreetest[1,])
F1(ttreetest)
table(test$ComType)
LS.train<-subset(train,ComType%in%c("intv","wllVK"))
LS.train$ComType<-factor(LS.train$ComType)
LS.test<-subset(test,ComType%in%c("intv","wllVK"))
LS.test$ComType<-factor(LS.test$ComType)

library(randomForest)
LSRandomForest<-
randomForest(LS.train$ComType~.,data=LS.train,ntree=5000,importance=TRUE)
print(LSRandomForest)

hist(treesize(LSRandomForest))
importance(LSRandomForest)
varImpPlot(LSRandomForest)

LS.predictrf<-predict(LSRandomForest,LS.train,type="class")
LSrftrain<-table(LS.train$ComType,LS.predictrf)
error(LSrftrain)

LS.predictrftest<-predict(LSRandomForest,LS.test,type="class")
LSrftest<-table(LS.test$ComType,LS.predictrftest)
error(LSrftest)

# Loading package
library(e1071)
library(caTools)
library(caret)
# Fitting Naive Bayes Model
# to training dataset
classifier_auth <- naiveBayes(ComType ~ ., data = train)

classifier_auth

# Predicting on test data'
y_pred <- predict(classifier_auth, newdata = test)

# Confusion Matrix
cm <- table(test$ComType, y_pred)
cm

library(MASS)
##LDA
```

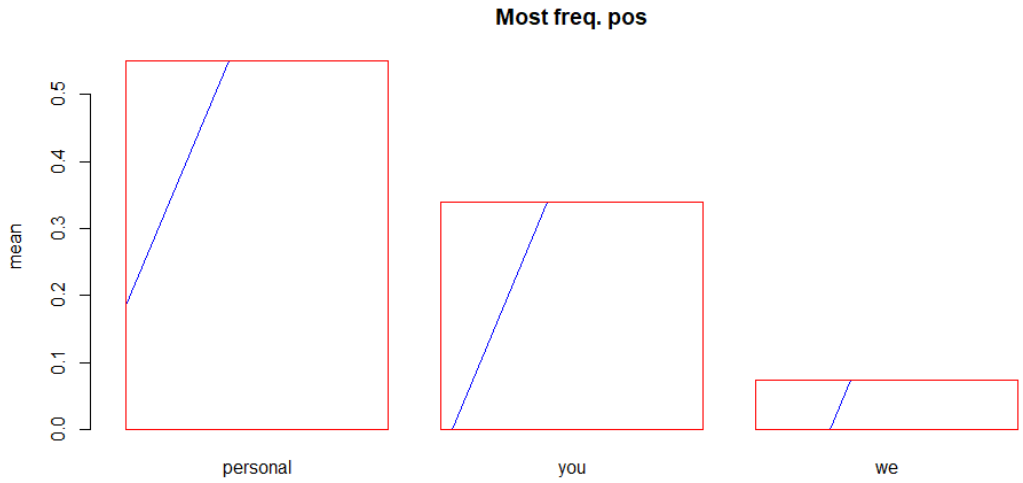
```
author.lda<-lda(ComType~.,data=train)
print(author.lda$scaling)
summary(author.lda)
author.ldapredicttrain<-predict(author.lda,train)
tldatrain<-table(train$ComType,author.ldapredicttrain$class)
error(tldatrain)
f1(tldatrain)
author.ldapredicttest<-predict(author.lda,test)
summary(author.ldapredicttest)
tldatest<-table(test$ComType,author.ldapredicttest$class)
error(tldatest)
print(paste0(c("text: ", 55)))
x1 <- "string1"
x2 <- "sring2"
paste0(x1, x2)
f1(tldatest)
plot(author.lda,col=author.train$colour)
plot(author.lda,dimen=1,type="both")
plot(author.lda,dimen=1,type="density")


require(nnet)
##Logistic Regression
lr<-multinom(ComType~.,data=train)
help(multinom)
lr.train<-predict(lr,train,type = "class")
error(table(train$ComType,lr.train))

lr.test<-predict(lr,test,type = "class")
error(table(test$ComType,lr.test))
##SVM
# author.numtr$colour = NA
# author.numtr$colour[author.numtr$Author == "Austen"] = 0
# author.numtr$colour[author.numtr$Author == "London"] = 1
# author.numtr$colour[author.numtr$Author == "Milton"] = 2
# author.numtr$colour[author.numtr$Author == "Shakespeare"] = 3
# author.numtr <- author.numtr[,-15]
author.numtr$Author <- as.factor(author.numtr$Author)
author.svm<-svm(Author~.,data=author.numtr,
            kernel="sigmoid")
help("svm")
summary(author.svm)

author.pred<-predict(author.svm,
                 author.numtr,decision.values=T)
ttrain<-table(author.numtr$Author,author.pred)
error(ttrain)

author.predtestsvm<-predict(author.svm,author.numte,decision.values=T)
ttest<-table(author.numte$Author,author.predtestsvm)
error(ttest)
```

```
row.names(w_dict_dfm_mycorp_promouns_full) <- w_dict_dfm_mycorp_promouns_full[,1]

help(dist)
tstat_dist <- dist(w_dict_dfm_mycorp_promouns_full[,2:5], method = "manhattan")
help("hclust")

clust <- hclust(tstat_dist, method = "ward.D2")

plot(clust, xlab = "Distance", ylab = NULL,hang = -5, cex = 0.2)

plot(as.phylo(clust), type = "unrooted", cex = 0.6,
    no.margin = TRUE)


HC = hclust(dist(w_dict_dfm_mycorp_promouns_full[,2:4], method = "euclidean"))
plot(w_dict_dfm_mycorp_promouns_full[,2:4], pch=20, col=cutree(HC,6))


library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization
df <- w_dict_dfm_mycorp_promouns_full[,2:5]
df <- na.omit(df)
df <- scale(df)
distance <- get_dist(df, method = "spearman")
help("get_dist")
fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
k2 <- kmeans(df, centers = 2, nstart = 25)
str(k2)
k2$cluster
fviz_cluster(k2, data = df)

# df %>%
#   as_tibble() %>%
#   mutate(cluster = k2$cluster,
#          state = row.names(USArrests)) %>%
#   ggplot(aes(UrbanPop, Murder, color = factor(cluster), label = state)) +
#   geom_text()
#
 k3 <- kmeans(df, centers = 3, nstart = 25)
 k4 <- kmeans(df, centers = 4, nstart = 25)
 k5 <- kmeans(df, centers = 5, nstart = 25)

# plots to compare
p1 <- fviz_cluster(k2, geom = "point", data = df) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = df) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = df) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = df) + ggtitle("k = 5")

library(gridExtra)
```

```
grid.arrange(p1, p2, p3, p4, nrow = 2)
# Compute k-means clustering with k = 4
set.seed(123)
final <- kmeans(df, 4, nstart = 25)
print(final)
fviz_cluster(final, data = df)

set.seed(123)
gap_stat <- clusGap(df, FUN = kmeans, nstart = 25,
                    K.max = 10, B = 50)
# Print the result
print(gap_stat, method = "Tibs2001SEmax")
fviz_gap_stat(gap_stat)


library("ggplot2")
library("ggdendro")
ggdendrogram(clust)
ggdendrogram(clust, rotate = TRUE, theme_dendro = FALSE)
# Build dendrogram object from hclust results
dend <- as.dendrogram(clust)
# Extract the data (for rectangular lines)
# Type can be "rectangle" or "triangle"
dend_data <- dendro_data(dend, type = "rectangle")
# What contains dend_data
names(dend_data)
head(dend_data$segments)
head(dend_data$labels)
p <- ggplot(dend_data$segments) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend))+
  geom_text(data = dend_data$labels, aes(x, y, label = label),
            hjust = 1, angle = 90, size = 1)+
  ylim(-1, 2)
print(p)



data <- scale(w_dict_dfm_mycorp_promouns_full[,2:4])
dist.res <- dist(data)
hc <- hclust(dist.res, method = "ward.D2")
dend <- as.dendrogram(hc)
plot(dend,  cex.sub = 0.2,  # Subtitle size
     cex.lab = 0.3,    # X-axis and Y-axis labels size
     cex.axis = 0.5, cex = 0.1)

plot(dend, which.plots=2, cex=0.1)


library(cluster)
data(votes.repub)
agn1 <- agnes(w_dict_dfm_mycorp_promouns_full[,2:5], metric = "manhattan", stand =
TRUE)
```

```
plot(hc, which.plots=2)
plot(hc, which.plots=2, cex=0.1)
saveRDS(w_dict_dfm_mycorp_promouns_full, "w_dict_full")

w_dict_dfm_mycorp_dlgs_m <- subset(w_dict_dfm_mycorp_dlgs[,2:7], gnd=="m")
w_dict_dfm_mycorp_dlgs_f <- subset(w_dict_dfm_mycorp_dlgs[,2:7], gnd=="f")
pos_table <- sort(colMeans(w_dict_dfm_mycorp_dlgs_m[,1:3], na.rm = TRUE),decreasing = T)
barplot(pos_table,
    main="Most freq. pos",
    xlab="POS",
    ylab="mean",
    border="red",
    col="blue",
    density=1
)
```

Визуализируйте данные:



Результат сформированности компетенций ИПК-4.1 определяются оценками на основе экзамена в седьмом семестре, который проводится в письменной форме по билетам. Экзаменационный билет состоит из двух частей. Продолжительность экзамена 1,5 часа.

Первая часть представляет собой тест из 2 вопросов. Ответы на вопросы первой части даются путем выбора из списка предложенных.

Вторая часть содержит один практический вопрос и предполагает решение задач и краткую развернутую интерпретацию полученных результатов.

Примерный перечень теоретических вопросов

1. Опишите типологизацию направления «Обработка естественного языка»

2. Опишите методы векторизации слов? Какие методы существуют их преимущества и недостатки

3. Опишите принцип работы метода векторизации word2vec, опишите его преимущества и недостатки

4. Опишите принцип работы метода векторизации FastText, опишите его преимущества и недостатки

5. Дайте определение расстояния. В каких методах обработки естественного языка применяется данный подход?

6. Автоматическая обработка естественного языка, ее цели и задачи. Предмет и объект данной области знаний.

7. Принцип работы контекстно свободных грамматик

8. Опишите принцип работы Word Embeddings: независимые от контекста представления слов

9. Опишите принцип работы Word Embeddings: зависимые от контекста представления слов

10. Обучение интеллектуальных систем. Виды обучающихся интеллектуальных систем

11. Предобработка данных, типы предобработки, ее особенности.

12. Задачи и методы генерации текстов. Файнтьюн обученной модели.

13. Принципы работы нейронной сети GPT-3.

14. Опишите методы извлечения фактов из текста. Опишите их преимущества и недостатки.

15. Опишите понятие энтропии Шеннона, дайте примеры использования. В каком методе машинного обучения используется энтропия?

16. Чем отличаются цели классификации и регрессии в машинном обучении?

17. Какие методы векторизации используются в репрезентации текстового массива данных? Опишите преимущества и недостатки методов

18. В каких случаях применяется индекс прироста информации? Опишите алгоритм его работы.

19. Опишите формальные метрики точности работы классификаторов. В чем преимущества и недостатки формальных метрик?

Примеры задач:

1. Задача 1. Построение линейных классификаторов

Дано: Матрица для обучения с признаками, исходный код двух классификаторов:

##LDA

author.lda<-lda(Author~.,data=author.numtr)

print(author.lda$scaling)

summary(author.lda)

author.ldapredicttrain<-predict(author.lda,author.numtr)

```
tldatrain<-table(author.numtr$Author,author.ldapredicttrain$class)

error(tldatrain)

f1(tldatrain)

author.ldapredicttest<-predict(author.lda,author.numte)

summary(author.ldapredicttest)

tldatest<-table(author.numte$Author,author.ldapredicttest$class)

error(tldatest)

print(paste0(c("text: ", 55)))

x1 <- "string1"

x2 <- "sring2"

paste0(x1, x2)

f1(tldatest)

plot(author.lda,col=author.train$colour)

plot(author.lda,dimen=1,type="both")

plot(author.lda,dimen=1,type="density")
```

Результаты экзамена определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Критерии экзамена обусловлены логической демонстрацией приобретенных компетенций в соответствии с текущей программой. Демонстрация предусматривает уверенное использование терминологии, понимание и корректное использование математического аппарата, предусматривает корректность написания кода, его понимание и корректное использование в нем математических методов. Отметка «хорошо» выставляется за счет демонстрации полученных компетенций, владение и понимание кода, теоретических аспектов его применения в практике работы с текстовыми массивами данных допускаются недочеты в понятийном аппарате математики. Отметка «удовлетворительно» позволяет допустить ошибки в разработке кода, но учитывает последовательную логику изложения структуры кода, его интерпретацию, связь теоретических аспектов лингвистики и математики, демонстрация понимания хода обработки текста. Минимальный порог оценки «отлично» составляет 90-100 баллов, хорошо 75-89, удовлетворительно «55-74» ниже 55 – «неудовлетворительно»

11. Учебно-методическое обеспечение

а) Электронный учебный курс по дисциплине в электронном университете «Moodle» - https://moodle.tsu.ru/course/view.php?id=12998

б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине.

в) План семинарских / практических занятий по дисциплине.

Семинар №1

1. Основы языка программирования R
2. Переменные, структуры данных
3. Циклы, проверка условий

Семинар №2

1. Сбор и структуризация текстовых данных
2. Работа с файлами
3. Лемматизация текстов при помощи mystem

Семинар №3

1. Библиотека quanteda для векторизации и анализа текстовых массивов данных
2. Векторизация текстов. Принципы, методы
3. Составление словарей, n-грамы

Семинар №4

1. Статистический анализ матрицы слов текста
2. Описательная статистика
3. Корреляционный анализ
4. Проверка статистических гипотез
5. Кластерный анализ

Семинар №6

1. Сокращение пространства признаков
2. Визуализация и анализ текстовых данных

Подготовка к проведению лабораторных работ начинается в начале теоретического изложения изучаемой темы и продолжается по ходу её изучения при освоении материала на занятиях в рамках практических заданий и работе над ним в ходе самостоятельной подготовки дома и в библиотеках. Для качественного выполнения лабораторных работ студентам необходимо:

1) повторить теоретический материал по конспекту и учебникам;

2) ознакомиться с описанием лабораторной работы;

3) в специальной тетради для лабораторных работ записать название и номер работы, перечень необходимого программного обеспечения, подготовить алгоритм или код;

4) выяснить цель работы, четко представить себе поставленную задачу и способы её достижения, продумать ожидаемые результаты опытов;

5) ответить устно или письменно на контрольные вопросы по изучаемой теме или решить ряд задач;

6) изучить порядок выполнения лабораторной работы. Подготовить среду выполнения кода к работе. После проверки правильности алгоритма работы программы преподавателем можно начинать выполнение лабораторной работы.

д) Методические указания по организации самостоятельной работы студентов.

Формы самостоятельной работы студентов разнообразны. Они включают в себя:

– изучение и систематизацию практических и теоретических примеров в рамках выполнения текущих заданий по предмету;

– изучение учебной, научной и методической литературы, материалов периодических изданий с привлечением электронных средств официальной, статистической, периодической и научной информации;

– подготовку докладов и презентаций, написание программного кода и его отладка;

– участие в работе студенческих конференций, комплексных научных исследованиях.

Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.

Примеры самостоятельной работы студентов:

Создание словаря и иго частотного распределения в текстах:

```
library(readxl)
#library(quanteda.sentiment)
library(quanteda)
```

```
# install.packages("remotes")
# remotes::install_github("quanteda/quanteda.sentiment")
tmp <- read_excel("full_word_rating_after_coding.xlsx", col_names = TRUE)

df #head body stem class

mycorp <- corpus(df, text_field = "stem", )



dict <- dictionary(list(negative = c(tmp$word[tmp$value==-1]),
                        neg_negative = c(tmp$word[tmp$value==-2]),
                        pos_positive = c(tmp$word[tmp$value==2]),
                        neutral = c(tmp$word[tmp$value==0]),
                        positive = c(tmp$word[tmp$value==1])))
valence(dict) <- list(negative = -1, neg_negative = -2, pos_positive = 2, neutral = 0, positive
=1)

dict2 <- dictionary(list(neg = c(tmp$word[tmp$value < 0]),
                        neut = c(tmp$word[tmp$value==0]),
                        pos = c(tmp$word[tmp$value>0])))
valence(dict2) <- list(neg = -1, pos = 1, neut = 0)
polarity(data_dictionary_LSD2015) <- dict
 # list(pos = c("positive", "neg_negative"), neg = c("negative", "neg_positive"))
sent_pres <- mycorp_vk %>%
  corpus_subset(gnd == "f")
sent_pres2 <- mycorp_vk %>%
  corpus_subset(gnd == "m")
summary(mycorp_vk)
x_m <- tokens_lookup(tokens(sent_pres), dictionary = dict2) %>%
  dfm()
x_f <- tokens_lookup(tokens(sent_pres2), dictionary = dict2) %>%
  dfm()
x_m <- dfm_weight(x_m, scheme = "prop")
x_f <- dfm_weight(x_f, scheme = "prop")
x_full <- tokens_lookup(tokens(mycorp_vk), dictionary = dict2) %>%
  dfm()

x_f <-convert(x_f, to = "data.frame")
x_m <-convert(x_m, to = "data.frame")
x_m$gnd <- "f"
x_f$gnd <- "m"
x_full_abs_vkwall <- rbind(x_f, x_m)
x_full_abs_vkwall$ComType <- "vkw"
write.csv(x_full_abs_vkwall, "sentiment_vkwall_gnd.csv")

ggplot(x_full_abs_vkwall, aes(doc_id, neut, fill = gnd, group = gnd)) +
  geom_bar(stat='identity', position = position_dodge(), size = 1) +
scale_fill_brewer(palette = "Set1") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  ggtitle("Sentiment scores in twelve Sherlock Holmes novels") + xlab("")
x_m2 <-  convert(x_m, to = "data.frame")
```

```
x_m2 <- as.data.frame(x_m)
x_f2 <-  convert(x_f, to = "data.frame")
x_f2 <- as.data.frame(x_f)
```
Корреляционный анализ:
```
library(outliers)
grubbs.test(emot_int$neg, type = 10)

grubbs.test(emot_int$pos, type = 10)
grubbs.test(emot_int$neut, type = 10)
library(ggplot2)
p = ggplot(emot_int[,-1], aes(x=self))
(p <- p+geom_density(aes(fill=gnd), alpha=1/2))

# Sample data
data <- emot_int[, 2:4] # Numerical variables
groups <- as.factor(emot_int[, 5]) # Factor variable (groups)
# Plot correlation matrix
pairs(data)

# Equivalent with a formula
pairs(~ neg+pos+neut, data = emot_int)


pairs(data,                    # Data frame of variables
      labels = colnames(data),  # Variable names
      pch = 1,                 # Pch symbol
      bg = rainbow(2)[groups],  # Background color of the symbol (pch 21 to 25)
      col = rainbow(2)[groups], # Border color of the symbol
      main = "",    # Title of the plot
      row1attop = TRUE,         # If FALSE, changes the direction of the diagonal
      gap = 1,                  # Distance between subplots
      cex.labels = NULL,        # Size of the diagonal text
      font.labels = 1)          # Font style of the diagonal text

panel.hist <- function(x, ...) {
  usr <- par("usr")
  on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5))
  his <- hist(x, plot = FALSE)
  breaks <- his$breaks
  nB <- length(breaks)
  y <- his$counts
  y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = rgb(0, 1, 1, alpha = 0.5), ...)
  # lines(density(x), col = 2, lwd = 2) # Uncomment to add density lines
}

# Creating the scatter plot matrix
pairs(data,
      upper.panel = NULL,         # Disabling the upper panel
      diag.panel = panel.hist)    # Adding the histograms
# Function to add correlation coefficients
```

```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {
  usr <- par("usr")
  on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  Cor <- abs(cor(x, y)) # Remove abs function if desired
  txt <- paste0(prefix, format(c(Cor, 0.123456789), digits = digits)[1])
  if(missing(cex.cor)) {
    cex.cor <- 0.4 / strwidth(txt)
  }
  text(0.5, 0.5, txt,
       cex = 1 + cex.cor * Cor) # Resize the text by level of correlation
}

# Plotting the correlation matrix
pairs(data,
      upper.panel = panel.cor,    # Correlation panel
      lower.panel = panel.smooth) # Smoothed regression lines

# install.packages("gclus")
library(gclus)

# Correlation in absolute terms
corr <- abs(cor(data))

colors <- dmat.color(corr)
order <- order.single(corr)

cpairs(data,                 # Data frame of variables
       order,                # Order of the variables
       panel.colors = colors,   # Matrix of panel colors
       border.color = "grey70", # Borders color
       gap = 0.45,              # Distance between subplots
       main = "Ordered variables colored by correlation", # Main title
       show.points = TRUE,      # If FALSE, removes all the points
       pch = 21,                # pch symbol
       bg = rainbow(2)[groups]) # Colors by group


#install.packages("psych")
library(psych)

pairs.panels(data,
             smooth = TRUE,      # If TRUE, draws loess smooths
             scale = FALSE,      # If TRUE, scales the correlation text font
             density = TRUE,     # If TRUE, adds density plots and histograms
             ellipses = TRUE,    # If TRUE, draws ellipses
             method = "spearman", # Correlation method (also "spearman" or "kendall")
             pch = 21,           # pch symbol
             lm = FALSE,         # If TRUE, plots linear fit rather than the LOESS (smoothed)
fit
             cor = TRUE,         # If TRUE, reports correlations
             jiggle = FALSE,     # If TRUE, data points are jittered
```

```
        factor = 2,         # Jittering factor
        hist.col = 4,       # Histograms color
        stars = TRUE,       # If TRUE, adds significance level with stars
        ci = TRUE)          # If TRUE, adds confidence intervals

library(psych)

corPlot(data, cex = 1.2)

library(ggplot2)
# install.packages("ggExtra")
library(ggExtra)
p_base <- ggplot(emot_int,aes(x=neg,y=pos,color=gnd)) + geom_point()
ggExtra::ggMarginal(p_base, groupColour = TRUE, groupFill = TRUE)
library(otrimle)
clus <- otrimleg(emot_int[,c(1,2)], G=2:5, monitor=1) # параметр monitor позволяет видеть ход выполнения



# Equivalent but using the plot function
plot(data)
library(tidyverse)
cor(w_dict_dfm_full[,2:4] %>% w_dict_dfm_mycorp_vk$gnd=="m")
w_dict_dfm_mycorp_vk  %>%
  group_by(gnd) %>%
  plot(w_dict_dfm_mycorp_vk$self)




w_dict_dfm_full$gnd <- as.factor(w_dict_dfm_full$gnd)
w_dict_dfm_full_01 <- w_dict_dfm_full
w_dict_dfm_full_01$gnd01[w_dict_dfm_full$gnd=="m"] <- 1
w_dict_dfm_full_01$gnd01[w_dict_dfm_full$gnd=="f"] <- 0

fitglm <- glm(w_dict_dfm_full_01$gnd01 ~ w_dict_dfm_full_01$personal - w_dict_dfm_full_01$you +
                w_dict_dfm_full_01$we +w_dict_dfm_full_01$self)
```

12. Перечень учебной литературы и ресурсов сети Интернет

а) основная литература:

– Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Большакова Е.И. и др.–М.: МИЭМ, 2011.

– Когнитивная и компьютерная лингвистика / Ред.: Р.Г.Бухараев, В.Д.Соловьев, Д.Ш.Сулейманов. - Казань: КГУ, 1994. - 112 с

– Баранов А. Н. Введение в прикладную лингвистику. М., 2001. URL: https://dislyget.ru/index.php?r=item/view&id=21065

б) дополнительная литература:

– Болховитянов А.В., Гусев А.В., Чеповский А.М. Морфологические модели компьютерной лингвистики: учеб. пособие –М. МГУП, 2010.

–Леонтьева Н. Н. Автоматическое понимание текстов: Системы, модели, ресурсы: Учебное пособие –М.: Академия, 2006.

–JurafskyD., Martin J. Speech and Language Processing.An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.Prentice Hall, 2000

в) ресурсы сети Интернет:

– открытые онлайн-курсы

– Журнал «Эксперт» - http://www.expert.ru

– Официальный сайт Федеральной службы государственной статистики РФ - www.gsk.ru

– Официальный сайт Всемирного банка - www.worldbank.org

– Общероссийская Сеть КонсультантПлюс Справочная правовая система. http://www.consultant.ru

– Официальный сайт языка программирования R - www.r-cran.com

13. Перечень информационных технологий

а) лицензионное и свободно распространяемое программное обеспечение:

– Microsoft Office Standart 2013 Russian: пакет программ. Включает приложения: MS Office Word, MS Office Excel, MS Office PowerPoint, MS Office On-eNote, MS Office Publisher, MS Outlook, MS Office Web Apps (Word Excel MS PowerPoint Outlook);

– публично доступные облачные технологии (Google Docs, Яндекс диск и т.п.).

– язык программирования R (RStudio) и Python;

б) информационные справочные системы:

– Электронный каталог Научной библиотеки ТГУ – http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system

– Электронная библиотека (репозиторий) ТГУ – http://vital.lib.tsu.ru/vital/access/manager/Index

– ЭБС Лань – http://e.lanbook.com/

– ЭБС Консультант студента – http://www.studentlibrary.ru/

– Образовательная платформа Юрайт – https://urait.ru/

– ЭБС ZNANIUM.com – https://znanium.com/

– ЭБС IPRbooks – http://www.iprbookshop.ru/

в) профессиональные базы данных *(при наличии)*:

– Университетская информационная система РОССИЯ – https://uisrussia.msu.ru/

– Единая межведомственная информационно-статистическая система (ЕМИСС) – https://www.fedstat.ru/

14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения занятий семинарского типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

Лаборатории, оборудованные компьютерами (не ниже i3, RAM 8Gb), проектором

Аудитории для проведения занятий лекционного и семинарского типа индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации в смешанном формате («Актру»).

15. Информация о разработчиках

Степаненко Андрей Александрович, НИ Томский государственный университет, ассистент кафедры общего славяно-русского языкознания и классической филологии