**Московский Физико-Технический Институт**
**Факультет Инноваций и Высоких Технологий**
**Кафедра алгоритмов и технологий программирования**

# Темы и задания по курсу «Анализ текстов»

## Список тем

1. Обзор задач анализа текстов
2. Извлечение признаков: bag of words, term frequencies, tf*idf, n-граммы
3. Классификация текстов – примеры задач, используемые на практике классификаторы. Подробней о линейных классификаторах.
4. Кластеризация текстов. Напоминание методов кластеризации.
5. Матричные разложения: SVD, NMF. SGD и ALS.
6. Тематическое моделирование
7. Word2vec, GloVe, word embeddings
8. Рекуррентные нейросети
9. Дополнительные темы

## Система оценки

Итоговый балл складывается из оценок за задания, «летучки» (мини-контрольные на 5-7 минут из 2-3 вопросов на занятиях), оценке на контрольной по теории и баллов за опциональные задания: доклад про один из методов анализа текстов (не входящий в программу курса) и творческое задание.

Таблица с распределением баллов, 100% и больше - отлично(10):

| | | |
|---|---|---|
| 1 | Семинарские мини-задания | 10% |
| 2 | "Летучки" | 5% |
| 3 | Задание 1: классификация | 10% |
| 4 | Задание 2: кластеризация | 10% |
| 5 | Задание 3: word embeddings | 10% |
| 6 | Задание 4: RNN | 15% |
| 7 | Доклад | 10% |
| 8 | Творческое задание: подготовить задание на тему любой задачи из NLP, про которую не было задания в курсе | 15% |
| 9 | Контрольная по теории | 40% |
| **Сумма:** | | 125% |

## Организационные вопросы

Для сдачи задания выложите IPython/Jupiter notebook с кодом на github или nbviewer, и пришлите на почту viktor.kantor@phystech.edu письмо со ссылкой на код. Тема письма должна иметь вид [МФТИ Тексты 2017] Фамилия Имя – Задание N

# Задание 1
# Классификация текстов: спам-фильтр для SMS

**Срок сдачи: 16 марта 2017**

## Описание

В этом задании вам предстоит взять открытый датасет с SMS-сообщениями, размеченными на спам ("spam") и не спам ("ham"), построить на нем классификатор текстов на эти два класса, оценить его качество с помощью кросс-валидации, протестировать его работу на отдельных примерах, и посмотреть, что будет происходить с качеством, если менять параметры вашей модели.

## Задание

1. Загрузите датасет по ссылке: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/smsspamcollection.zip (описание датасета можно посмотреть здесь)

2. Считайте датасет в Python (можете сразу грузить все в память, выборка небольшая), выясните, что используется в качестве разделителей и как проставляются метки классов.

3. Подготовьте для дальнейшей работы два списка: список текстов в порядке их следования в датасете и список соответствующих им меток классов. В качестве метки класса используйте 1 для спама и 0 для "не спама".

4. Используя sklearn.feature_extraction.text.CountVectorizer со стандартными настройками, получите из списка текстов матрицу признаков X.

5. Оцените качество классификации текстов с помощью LogisticRegression() с параметрами по умолчанию, используя sklearn.cross_validation.cross_val_score и посчитав среднее

арифметическое качества на отдельных fold'ах. Параметр cv задайте равным 10. В качестве метрики качества используйте f1-меру. Получившееся качество – ответ в этом пункте.

6. А теперь обучите классификатор на всей выборке и спрогнозируйте с его помощью класс для следующих сообщений:

"FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! Subscribe6GB"
"FreeMsg: Txt: claim your reward of 3 hours talk time"
"Have you visited the last lecture on physics?"
"Have you visited the last lecture on physics? Just buy this book and you will have all materials! Only 99$"
"Only 99$"

Выпишите через пробел прогнозы классификатора (0 – не спам, 1 – спам)

7. Задайте в CountVectorizer параметр ngram_range=(2,2), затем ngram_range=(3,3), затем ngram_range=(1,3). Во всех трех случаях измерьте получившееся в кросс-валидации значение f1-меры, округлите до второго знака после точки, и выпишете результаты через пробел в том же порядке. В данном эксперименте мы пробовали добавлять в признаки n-граммы для разных диапазонов n - только биграммы, только триграммы, и, наконец, все вместе - униграммы, биграммы и триграммы. Обратите внимание, что статистики по биграммам и триграммам намного меньше, поэтому классификатор только на них работает хуже. В то же время это не ухудшает результат сколько-нибудь существенно, если добавлять их вместе с униграммами, т.к. за счет регуляризации линейный классификатор не склонен сильно переобучаться на этих признаках.

8. Повторите аналогичный п.7 эксперимент, используя вместо логистической регрессии MultinomialNB(). Обратите внимание, насколько сильнее (по сравнению с линейным классификатором) наивный Байес страдает от нехватки статистики по биграммам и триграммам.

9. Попробуйте использовать в логистической регрессии в качестве признаков Tf*idf из TfidfVectorizer на униграммах. Повысилось или понизилось качество на кросс-валидации по сравнению с CountVectorizer на униграммах? Обратите внимание, что результат перехода к tf*idf не всегда будет таким - если вы наблюдаете какое-то явление на одном датасете, не надо сразу же его обобщать на любые данные.

10. * Попробуйте получить как можно более высокое качество на кросс-валидации. Напишите, что пробовали и какое качество получилось.

11. Какие наблюдения и выводы можно сделать из этого задания?

## Задание 2
## Кластеризация писем
**Срок сдачи: 6 апреля**

### Описание
В этом задании вам предстоит взять открытый датасет с электронными письмами Хиллари Клинтон и поэкспериментировать с построением более-менее интерпретируемых кластеров.

### Задание

a. Загрузите датасет с kaggle: https://www.kaggle.com/kaggle/hillary-clinton-emails

b. Изучите, из чего состоит датасет.

c. Предобработайте тексты как сочтете правильным для первых экспериментов. Опишите, как вы его предобрабатываете, и почему так в блокноте в markdown ячейке

d. Выясните, какие биграммы чаще всего встречаются в датасете

e. Попробуйте выделить коллокации из двух слов по PMI с помощью nltk (примеры можно найти по ссылке: http://www.nltk.org/howto/collocations.html)

f. Выполните любую несложную кластеризацию писем (не тратьте на этот шаг много времени)

g. Придумайте, как визуализировать содержание кластеров. Например, можно выводить самые частые слова из каждого кластера (но, вероятно, это не самая удачная идея). Визуализируйте ту кластеризацию, которая у вас уже получилась.

h. Поработайте с признаками и методом кластеризации так, чтобы

кластеры выглядели наиболее интерпретируемыми.

i. Придумайте, как оценить интерпретируемость кластеров с помощью асессоров (какие вопросы задавать, как подсчитать качество на основе ответов). Для эксперимента воспользуйтесь кем-то из однокурсников в качестве асессора, и оцените интерпретируемость вашей кластеризации. Имейте ввиду, что такая оценка разумеется не статзначима и по-хорошему нужно привлекать более одного асессора, но протестировать придуманный вами способ оценки до какой-то степени так можно. Опишите ваш способ оценить интерпретируемость кластеризации и результаты в markdown ячейке в вашем ipython notebook.

## Задание 3
## Word embeddings

**Срок сдачи: 27 апреля**

Условие задания будет выдано отдельно. Содержание задания: эксперименты с постановкой оптимизационных задач, приводящих к получению word embedding'ов, и реализацией их решения.

## Задание 4
## Рекуррентные нейросети

**Срок сдачи: 18 мая**

Разобрать IPython notebook и сделать задание в конце: https://github.com/vkantor/MIPT_Data_Mining_In_Action_2016/blob/master/trends/hw5/clf_rnn.ipynb