Contents

1 Выделение информации

На входе мы имеем несколько тесктов, в которых содержатся рекомендации, советы и т.д. по поводу достижения некоторой единой для всех текстов цели. На выходе мы хотим получить на основе этих текстов общую схему действий для достижения этой цели.

На данный момент мы предполагаем получение, что все советы прописаны достаточно явно. Мы не предполагаем, что нам нужно читать какой-либо личный рассказ о том, как человек достигал этой цели, и в соответствии с анализом этого текста вносить изменения в схему.

1.1 Основные этапы выделения сценария

Обозначим основные этапы выделения сценария:

1. Выделение из каждого текста информации, возможно относящейся к сценарию.
2. Струтуризация информации для каждого текста согласно единому шаблону.
3. Выбор предложений, удовлетворяющих необходимым условиям для вхождения в сценарий.
4. Формирование окончательного сценария на основе полученных на предыдущем шаге выборок.

Теперь разберем каждый из этих этапов. Анализ, насколько идея и реализация успешны, будет в сецкии Результаты И Анализ.

1.2 Выделение Информации Из Исходных Текстов

Рассмотрим отдельный текст со входа. Мы предполагаем, что полезную информацию могут содержать только те элементы текста, которые относятся к одному из двух типов:

- Предложения, содержащие глаголы и отглагольные части речи,
- Списки внутри текста.

Разберем, почему выделили именно эти два типа.

Схема, которую мы хотим получить в конце и которую мы называем сценарием, по сути есть компиляция из всех советов, рекомендаций и указаний, находящихся в текстах. Эти элементы есть побуждения к

действию, а действие всегда за редким исключением выражаются через глагол и отглагольные части речи. По сути это объясняет и почему мы пренебрегаем остальными предложениями.

Однако есть такие элементы, как списки. В некотором смысле, список достаточно часто есть продолжение предыдущих предложений и содержит особо важную выделенную информацию. Это может быть список того, что нужно взять, или того, какие вопросы следует задать. И потому нам не следует ими пренебрегать.

О реализации. Нахождение предложений с глаголами сводится к задаче сегментации текста на предложение и опреднию того, является ли слово глаголом. Для этого существует достаточно большое разнообразие парсеров, в нашем случае мы использовали isanlp. Далее любой найденный глагол или отглагольная часть речи вместе со своими зависимостями будет называться действием.

Теперь поговорим о списках. В нашем представлении список выглядит следующем образом: это череда абзацев, которые начинаются с определенных символов - для нумерованного списка это числа или буквы, для ненумерованного - различные символы, например, '*', '-', '+' и т.д. Так же мы учитываем возможность, что после каждого маркированного абзаца, являющегося элементом списка, идет еще один абзац, который является разъясняющим к первому. Использование этих критериев позволяет легко выделять списки.

Здесь стоит заметить, что список в тексте это необязательно несколько однотипно начинающихся обзацев и он может быть более сложно устроен. В частности, каждый элемент списка может быть представлен более, чем двумя абзацами, или список может содержать вложенные списки. Однако мы пренебрегаем данным фактом, поскольку мы предполагаем, если такие сложные структуры имеют место быть, то каждый элемент списка и так должен содержать некоторые дополнительную информацию, которая будет учтена, как предложения с глаголами. И в результате мы не проиграем сильно от того, что не знаем, что это список.

1.3 Структуризация Информации

Данный этап является надстройкой над предыдущим. В результате предыдущего этапа мы научились выделять действия и списки. В результате этого этапа мы получаем следующую структуру данных: каждый элемент это либо абзац, либо список. Элементы упорядочены согласно встречанию в

тексте. Каждый элемент-абзац содержит все действия, содержащиеся в нем, и название секции, к которой он принаджежит. Каждый элемент-список содержит составляющие его элементы-абзацы и название секции, к которой он принадлежит.

Немного обсудим название секций. Большинство текстов содержат некоторую внутренную структуру, которая выражается в первую очередь за счет заголовков разных уровней (заголовки, подзаголовки, подзаголовки для подзаголовков и т.д.). И логично предположить, что эти заголовки смогут помочь при дальнейшем объединении извлеченной из разных текстов информации в единый сценарий. Для элемента-абзаца, который является частью списка, мы считаем названием секции первое предложение соответствующего элемента списка. Для элементов-списков и всех остальных элементов-абзацев мы считаем названием секции заголовок самого нижнего уровня.

Выделение заголовков построено на крайне простой идеи: каждый заголовок представляет собой абзац, состоящий из единственного предложения. Причем это предложение удовлетворяет двум условиям: оно не очень длинное (эмперически подобранно ограничение не более 10 слов) и оно либо заканчивается обычными для окончания предложения знаками припинания (тока, вопросительный или восклицательный знак), либо в конце не стоит никакого знака припинания.

1.4 Необходимые Условия Вхождения В Сценарий

В результате предыдущих этапов мы получили все действия, которые есть в тексте, и структуризировали их. На этом этапе мы определимся как, используя морфологическую и синтаксическую информацию, оставить то, что нам с крайне высокой вероятностью подходит.

За время анализа текстов были выделены следующие условия:

- Глагол в повелительной форме
- Глагол в форме инфинитива, причем этот глагол зависит от таких слов, как 'можно', 'нужно' и т.д.
- Глаголы во втором лице

Все эти условия так же проверяются, используя инструменты из библиотеки isanlp.

В результате отработки всех предыдущих и этого этапов для каждого текста мы имеем экстракт, который отчасти уже является отдельным сценарием для этого текста. Он имеет два важных следующих недостатка:

- содержит некоторые действия, которые не несут никакой смысловой информации
- не рассматривает возможность ветвления сценария

Устранение первого недостатка, предположительно, произойдет при сравнении сценариев, извлеченных из разных текстов. В частности этому посвящены следующие этапы.

1.5 Результаты И Анализ

Мы работали с выборкой текстов из интернета, которые есть инструкции по покупке автомобиля. Всего в выборке находится 35 документов. Вся выборка состоит из трех частей: инструкции для подержанных автомобилей, инструкции для новых и остальные тексты. Каждая из этих выборок содержит 13, 17 и 5 документов соответственно.

В данном разделе, мы обсудим качество информации, выделяемой выше методом. Заметим, что эти результаты во многом зависят от качества используемого парсера.

О выделении списков. Среди всех 35 документов не было обнаружено такого, что текст в нем содержит список и он не был бы обнаружен. Поэтому критерии выделения списка, описанные в Выделение Информации Из Исходных Текстов, можно считать сформулированными корректно.

О качестве получаемой выборки из текста. Для начала, давайте определимся, как исследовать это качество. Результат первых трех этапов есть выборка, состоящая из действий и списков. Давайте считать каждый элемент списка отдельной единицей информации. Так же пусть у нас имеется выделенный в ручную сценарий. Нас будут интересовать две величины:

- сколько из вручную выделенного сценария элементов было выделено алгоритмически,
- сколько из выделеных алгоритмически элементов входит во множество вручную выделенных элементов.

Т.е. какую часть истинного сценария мы смогли найти и какая часть выборки является нужной. Легко заметить, что это хорошо известные метрики Recall и Precision. Было выбрано два текста и результаты вы можете найти в таблице 1.

col1	Recall,%	Precision, %
Text 1	97.4	97.6
Text 2	83.7	87.8
Mean	90.6	92.7

Table 1: Качество выбираемой информации

Как можно видеть, наш метод показывает достаточно хорошие результаты. При непосредственном ознакомлении с текстами, можно заметить, что больше всего выделяется ненужной информации в начале (предисловии, которое достаточно часто содержится в текстах и не несет сильной смысловой нагрузки) и в конце (пожелания, напутственные слова и т.д.).

2 Об идентификации шагов

В данном разделе мы хотим изучить, насколько хорошо возможно отличить один шаг от другого и насколько хорошо можно определить, что два шага из разных документов есть один и тот же шаг.

2.1 Гипотеза Об Корректной Идентификации Шагов

Мы сделаем следующие гипотезы:

- Каждому тексту соответствует некоторый вектор в $\mathbb{R}^n$;
- Каждому шагу соответствует некоторый центральный вектор;
- Вектора текстов, которые воплощают именно этот шаг, ближе всего к центрального вектора именно этого шага.

Обсудим эти гипотезы. Соответствие из первой гипотезы строится следующим образом: возьмем некоторую обученную модель word2vec (в нашем случае мы использовали готовые модели RusVectores), каждому слову в тексте поставим в соответствие вектор из этой модели, и тогда

вектор, соответствующий этому тексту, есть среднее арифметическое векторов всех слов входящих в него.

Соответствие из второй гипотезы есть следущее: центральный вектор конкретного шага это центр всех векторов для текстов, которые воплощают этот шаг

Подтверждение же последней гипотезы равносильно подтверждению гипотезы, что все шаги легко идентифицируемы. Проверим насколько это верно при помощи следующего эксперимента:

1. Каждый текст вручную разбиваем на множество текстов, каждый из которых есть воплощение одного шага. В результате этого пункта, у нас есть множество текстов, каждый из которых есть отдельный шаг и мы знаем, что это за шаг;

2. При помощи word2vec для каждого текста мы находим соответствующий ему вектор;

3. Для каждого типа шага сценария, мы находим центр соответствующих векторов. В результате, для каждого типа шага сценария у нас есть центральный вектор;

4. Определим для каждого текста-шага тип шага по ближайшему центру и сравним полученную разметку с исходной разметкой.

Информацию о том, какие шаги были выделены и сколько текстов для каждого шага в нашем наборе, Вы можете найти в таблице 2.

№ шага	Имя шага	Количество текстов
0	Ваши деньги	5
1	Цены	4
2	Объявлеия	4
3	Телефонный разговор	7
4	Документы на машину	6
5	Мониторинг Сайтов	9
6	ДКП	8
7	Осмотр	8
8	Тест-драйв	5
9	Определиться с маркой и моделью	7
10	Дигностика	4
11	Год выпуска и пробег	3

Table 2: Информация о выделенных шагах

Для оценки качества распознавания каждого шага мы будем использовать

две метрики: Precision и Recall (см. таблица 3). Стоит заметить, что некоторые шаги распознаются значительно хуже (к примеру Recall четвертого шага составляет всего 66.7%, т.е. около трети текстов, относящихся к этому шагу не было распознано). Это связано с двумя факторами. Во-первых, шаги не являются сильно детализированнами. Это приводит к тому, что, к примеру, в разделе 'Телефонный разговор' могут быть советы, связанные к документации, что приближает такие тексты к пятому шагу. А во-вторых, некоторые шаги близки по смыслу сами по себе. К примеру, очевидно, что близкими являются пары 4-6 и 7-8. Первый фактор устраняется достаточно легко - большая детализация. А ухудшения качества, вызванных вторым фактором, пожалуй можно и не считать сильных ухудшением. Ведь если шаги достаточно близки, то небольшая путаница в текстах, скорей всего, не повлияет на качество финального результата.

№ шага	Recall,%	Precision, %
0	100.0	100.0
1	100.0	100.0
2	100.0	80.0
3	85.7	66.7
4	66.7	75.0
5	83.3	83.3
6	75.0	85.7
7	87.5	87.5
8	80.0	100.0
9	100.0	100.0
10	100.0	100.0
11	100.0	100.0

Table 3: Качество идентификации шагов

Для глобальной оценки идентификации шагов, мы используем усредненные Precision и Recall, а так же Accuracy (см. таблица 4). Заметим, что выборка является достаточно сбалансированной - размер всех шагов, кроме последнего, отличается от среднего не более, чем в полтора раза. Мы получили достаточно высокие показатели - порядка 90%.

На основе полученных выше результатов, можно считать гипотезу об корректной идентификации шагов выполненной с высокой точностью.

Метрика	Значение, %
Mean Recall	89.9
Mean Precision	89.9
Accuracy	87.1

Table 4: Качество идентификации шагов, средние показания

Далее обсудим растояние между шагами и зависимость качества разделяемости шагов от количества текстов, на основе которых строится центральный вектор.

2.2 Расстояние Между Шагами

Данный раздел содержит информацию, о близости между шагами. В данном случае близость понимается как порожденное евклидовой нормой расстояние между соответствующими векторами в 5 для каждого шага представлены ближайший и наиболее удаленный шаги, а так же среднее расстояние до всех остальных шагов.

№ Шага	MinDistance	ArgMin	MaxDistance	ArgMax	MeanDistance
0	0.157634	10.0	0.266157	8.0	0.181316
1	0.157767	10.0	0.299288	8.0	0.190498
2	0.124397	3.0	0.230869	11.0	0.159127
3	0.120623	10.0	0.241608	8.0	0.156652
4	0.102595	6.0	0.271278	8.0	0.170855
5	0.149185	10.0	0.278785	8.0	0.179390
6	0.102595	4.0	0.274943	8.0	0.170093
7	0.119390	8.0	0.259894	1.0	0.189826
8	0.119390	7.0	0.299288	1.0	0.226551
9	0.156122	10.0	0.258760	8.0	0.175332
10	0.120623	3.0	0.254001	8.0	0.155111
11	0.215655	7.0	0.271436	1.0	0.219019

Table 5: Статистическая информация о расстоянии между шагами

Из таблицы 5 следует два интересных вывода. Во-первых, расстояние между всеми парами шагов (кроме двух очень близких пар, обозначенных в предыдущем разделе), больше чем 0.12. Данный порог можно использовать

для проверки корректности нахождения центра. Во-вторых, все шаги можно вписать в некоторую сферу с радиусом не более чем 0.3. Второй факт является достаточно естественным, поскольку все шаги сценария раскрывают некоторую одну тему, однако его можно использовать, чтобы отсеить ненужные предложения, не несущие особой информативности.

2.3 Влияние Размера Выборки Тестов

Выше описанные эксперименты и результаты говорят о том, насколько хорошо возможно идентифицировать шаги. Рассмотрим более реальную задачу:

- На входе n текстов, из которых для $m \leq n$ нам известен номер шага к которому они принадлежат.
- На выходе разметка для всех n шагов.

В данном разделе изучим, как зависит от качество результата от m. Для этого мы сделаем следущее:

- Оставим из исходной выборки, описанной в таблице 2, только 4 шага и тексты, относящиеся только к ним. В нашем случае мы оставили шаги 4, 6, 7, 9.
- Сделаем так, чтобы для каждого шага было только k текстов. В нашем случае мы взяли $k = 7$.

Выделим из полученных данных обучающую (та часть, на основе которой мы будем строить центральные вектора для шагов) выборку: выберем r текстов для каждого шага. Таким образом r - это количество текстов для каждого шага. Сделаем всемозможные выборки для фиксированного r, решим выше сформулированную задачу для них, измерим качество, усредним метрики и полученные значение метрик примем за метрики качества для r текстов на шаг. Полученные результаты представлены в таблице 6. Значения Recall и Accuracy достаточно близки поэтому на графике 2.3 находится данные только для метрик Recall и Precision.

мы получаем следующие результаты. Во-первых, увеличение размера исходной разметки ведет к улучшению качества по любой метрике. Данный момент был достаточно очевиден и без данного эксперимента, поскольку это есть следствие того, что чем больше текстов для каждого шага дано,

r	Recall, %	Precision, %	Accuracy, %
1	63.4	73.1	63.4
2	75.8	79.7	75.8
3	83.5	85.5	83.5
4	88.6	89.8	88.6
5	92.2	93.0	92.2
6	95.2	95.7	95.2
7	96.4	96.9	96.4

Table 6: Влияние Размера Выборки Тестов На Качество Идентификации

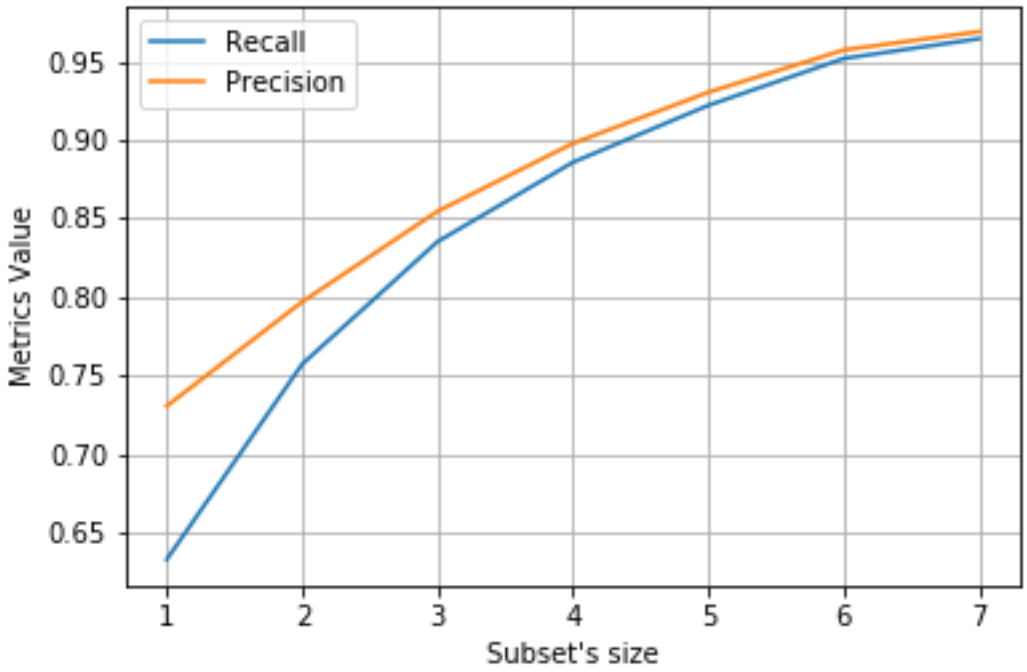
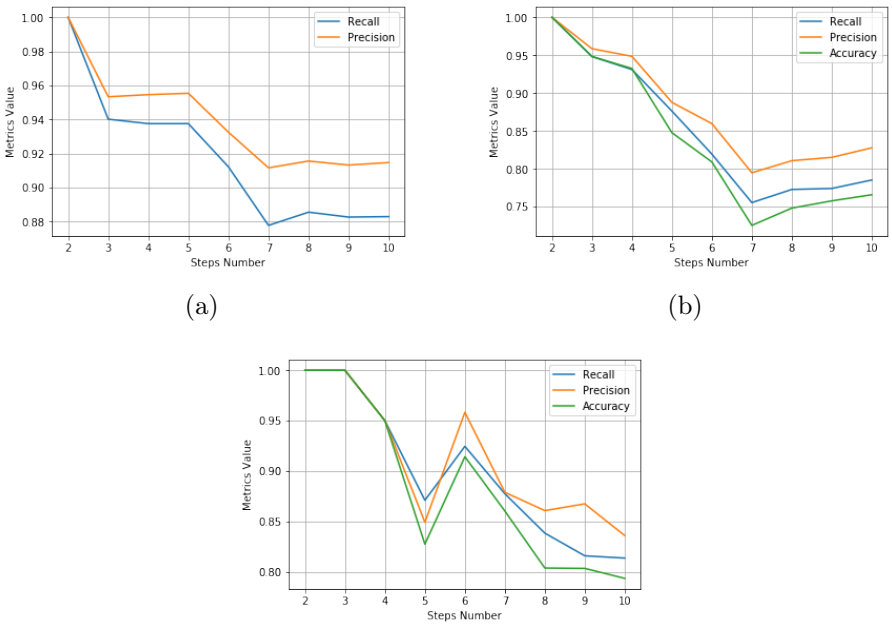


Figure 1: Зависимость качества от размера выборки

тем точнее мы определим центральный вектор этого шага. Во-вторых, при шести текстах достигается качество 95% и далее активный рост прекращается. Таким образом, шесть текстов на шаг, когда всего четыре

шага, можно считать необходимым и достаточным количеством для качественной идентификации. Теперь зададимся вопросом, насколько этот показатель качества ухудшится, если количество шагов продолжит расти.

2.4 Влияние Количества Шагов



(a) (b)

(c)

Figure 2: Влияние количества шагов на качество распознавания при фиксированном размере обучающей выборки: (a) 3/4; (b) 3/ALL; (c) 4/ALL.

Проведем серию экспериментов, измеряя качество на каждом этапе:

- **3/4**: Каждый шаг содержит 4 текста, 3 из них обучающие. Количество шагов от 2 до 11.
- **3/ALL**: Каждый шаг содержит столько текстов, сколько было в исходном, 3 из них обучающие. Количество шагов от 2 до 11.
- **4/ALL**: Каждый шаг содержит столько текстов, сколько было в исходном, 3 из них обучающие. Количество шагов от 2 до 11.

Результаты этих экспериментов представлены на графиках 2.

Из полученных результатов мы можем сделать следующие выводы:

1. Качество идентификации падает при увеличении количества шагов в начале. Этот вывод достаточно очевиден и ожидаем.

2. Падение качества замедляется после семи-восьми шагов.

3. Падение качества останавливается приблизительно на значении 80%.

Отсюда следует, что можно ожидать, что шесть текстов на шаг (количество, установленное в предыдущем подразделе для семи шагов) будет и далее показывать хорошие результаты.

3 Классификация

В предыдущем разделе мы показали, что выделенные человеком сегменты текста можно классифицировать с достаточно высокой точностью, имея разметку для достаточно небольшого количества сегментов. В данном разделе мы изучим следующую задачу классификации:

- Объекты - программно выделенные сегменты класса(см. 3.1),
- Метка класса - номер шага.

3.1 Сегментация текста

Перед тем, как посмотрим на результаты различных методов классификации, рвзберемся с тем, как выделяются сегменты.

Предлагается следующий алгоритм:

1. Определить предложения, удовлетворяющие необходимым условиям вхождения в сценарий (см. 1.4). Далее эти предложения называем предложения-инструкции (ПИ).

2. Все предложения преобразовать в вектора. Мы использовали для этого обученную модель word2vec, вектор предложения находился как срелнее арифметическое векторов слов.

3. Каждое ПИ рассматривается как центр будующего сегмента. Каждое предложение, не являющееся ПИ, находится между двумя центрами или оно стоит либо перед первым ПИ, либо после последнего. В последних случаях предложение будет относиться либо к первому центру, либо к последнему. Когда предложение находится между двумя центрами, оно

будет отнесено к центру, ближайшему по выбранной метрике. В нашем случае это $l2$-метрика для соответствующих векторов.

4. Кажому получившемуся сегменту поставить в соответствие вектор, как среднее арифметическое векторов предложений.

5. Объединить два соседних сегмента, если расстояние между соответствующими векторами меньше порога ϵ. Если есть такая ситуация, что $\rho(t_{i-1}, t_i) < \epsilon$ и $\rho(t_i, t_{i+1}) < \epsilon$, то объединим ту пару, которая ближе.

Дадим несколько комментариев по поводу алгоритма.

Во-первых, зачем он нужен, если большинство информации для инструкции содержится в предложениях-инструкциях и прилегающих списках? Информация, содержащаяся в этих предложениях, является достаточной для понимания человеком. Но этой информации не достаточно для качественной классификации. Дополнительные предложения в сегменте уточняют вектор семгента, что должно положительно сказаться на качестве классификации сегментов.

Во-вторых, получение векторов предложений и сегментов, как сренее арифмитечкое, является достаточно наивным, однако в работе [нужна ссылка] было показано, что для коротких текстов данный подход дает хорошие результаты, что оправдывает такой подход в нашей работе.

В-третьих, об объединении сегментов. При сегментации текста на шагах до объединения, возникает некоторое множество сегментов. Но некоторые из них можно было бы объединить, как близкие по смыслу - это должно увеличить вероятность правильной классификации. Однако при объединении сегментов могут объединиться не близкие по смыслу сегменты. Очевидно, что для классификации, объединить далекие сегменты гораздо хуже, чем не объеденить близкие. Исходя из этого, выбирается порог ϵ. По этой же причине, проводится только одна итерация процедуры объединения.

3.2 Наивная сегментация

3.3 SVM