Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Соболева Дарья Михайловна

Языковое моделирование в задаче построения вопрос-ответной системы

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д.ф.-м.н.
К. В. Воронцов

Москва, 2018

Содержание

Аннотация

С каждым днем количество информации, накопленной человечеством, растет все более и более быстрыми темпами. Сегодня людям уже тяжело самостоятельно находить ответы в таком потоке информации. В связи с этим появляется необходимость в построении систем, которые позволят людям находить нужную им информацию – вопрос-ответных систем. В данной работе предложен новый способ построения архитектуры вопрос-ответной системы, значительно ускоряющий ее работу. В работе произведены эксперименты по сравнению данного способа с его предшественниками и продемонстрированы способы построения пространства признаков для такой архитектуры.

1 Введение

В данной работе рассмотрена задача построения вопрос-ответной системы на русском языке. Вопрос-ответная система – это информационная система, способная принимать вопросы и отвечать на них на естественном языке. «Когда родилась Елизавета II?» – пример вопроса, ответ на который должен быть представлен в виде упорядоченного набора слов, фраз или предложений. Пример ответа: «в 1926».

В наши дни люди все больше времени тратят на анализ текстов в поиске нужной для них информации. Для того чтобы сократить это время, разрабатываются различные автоматические способы анализа текста и поиска необходимой информации. Наиболее предпочтительным для многих пользователей видом извлечения требуемой информации из большого объема текстов является диалог с вопросно-ответной системой, которой можно задать вопрос на естественном языке и ожидать ответа в такой же естественно-языковой форме. Большинство существующих в настоящее время реализаций вопросно-ответных систем ориентировано на один из самых распространенных языков мира – английский.

В данной работе поиск ответов на вопросы осуществлялся по коллекции русскоязычной Википедии. Википедия – это открытая многоязычная универсальная интернет-энциклопедия, являющаяся уникальным источником знаний. Тексты Википедии предназначены именно для чтения их человеком, что позволяет использовать их в задаче языкового моделирования.

В работе [1] была предложена модель англоязычной вопрос-ответной системы, состоящей из двух последовательных этапов. Первый этап – это построение модели поиска по вопросу набора из k релевантных документов. Второй этап состоял в построения модели машинного понимания текста, способной находить правильный ответ на вопрос среди k релевантных документов. Более конкретно, второй этап осуществлял поиск фразы, являющейся ответом на вопрос. Для этого была использована модель, предсказывающая для каждой пары вопроса и документа начало и конец ответа в тексте данного документа.

Данный подход построения вопрос-ответной системы является общим, и может быть использован для построения вопрос-ответной системы на любой коллекции текстовых документов для любого языка. К сожалению, данный метод не лишен недостатков. Модели второго этапа приходится строить классификаторы, определяющие начало и конец ответа, для каждого слова из k документов. Если документов много и они большие по размеру, на это тратится существенное количество времени работы модели. При этом, правильный ответ на вопрос обычно представлен в виде короткой фразы и зачастую содержится в одном предложении.

В настоящей работе был рассмотрен способ построения аналогичной модели на русскоязычной коллекции текстовых документов с добавлением этапа ранжирования предложений в наборе из k релевантных документов. За счет добавления этапа ранжирования удалось ускорить этап применения модели в 16 раз. Для этого в модель третьего этапа подавались первое или небольшое количество первых предложений в ранжированном списке. Если ве-

роятность наличия ответа в поданном предложении была выше фиксированного порога, в таком случае все остальные предложения не рассматривались.

С помощью добавления в модель дополнительных признаков, характеризующих лексическую и семантическую схожесть слов документа и слов вопроса удалось также повысить качество ее работы на 15%.

Настоящая работа будет организована следующим образом. В разделе 2 будет приведен обзор существующих подходов построения подобных систем. В разделе 3 будет описан способ построения вопрос-ответной системы, разработанный в рамках настоящей работы. В разделе 4 будут продемонстрированы результаты экспериментов.

2 Обзор литературы

Еще в начале 1960-х годов под вопрос-ответной системой понимали поиск ответа на вопрос в коллекции неструктурированных данных. Ученые предполагали, что компьютеры должны помочь человеку отвечать на вопросы с использованием естественного языка. В то время система вопросов и ответов рассматривалась как набор элементарных правил, придуманных человеком [14].

Сегодня, в связи активным ростом сети информационных технологий и баз знаний, вопрос-ответная система не укладывается в набор элементарных правил. Многие компании (Microsoft, Facebook, IBM, Google) и научно-исследовательские группы участвуют в разработке современных вопрос-ответных систем [1–5].

Не так давно для построения вопрос-ответных систем активно использовались тематические модели [6–8]. Такие системы выделяли в документах куски текста (пассажи) и ранжировали эти пассажи по тематической близости к вопросу. Такой подход был актуален в то время, когда существовало небольшое количество размеченных данных для построения такой системы. В связи с чем, все модели были основаны на кластеризации, как в случае с тематическими моделями. В настоящее время, появились большие корпусы размеченных данных как на английском языке, так и на русском.

Одна из самых последних работ была опубликована компанией Facebook [1]. В ней была предложена модель англоязычной вопрос-ответной системы, состоящая из двух последовательных этапов. На рисунке 1 продемонстрирована схема работы данной модели. Предложенная модель предусматривала два последовательных этапа.

Первый этап заключался в поиске для вопроса набора из k наиболее релевантных документов. Данный этап позволял авторам статьи эффективнее решать задачу вопрос-ответной системы, сужая пространство поиска. Это позволяло сосредоточиться на чтении только тех документов, которые, вероятно, будут релевантными к данному вопросу. Поиск документов осуществлялся при помощи модели, основанной на использовании обратного индекса и функции оценки релевантности документа к вопросу. В качестве функции релевантности документа было использовано количество общих биграмм в вопросе и документе. Для уско-

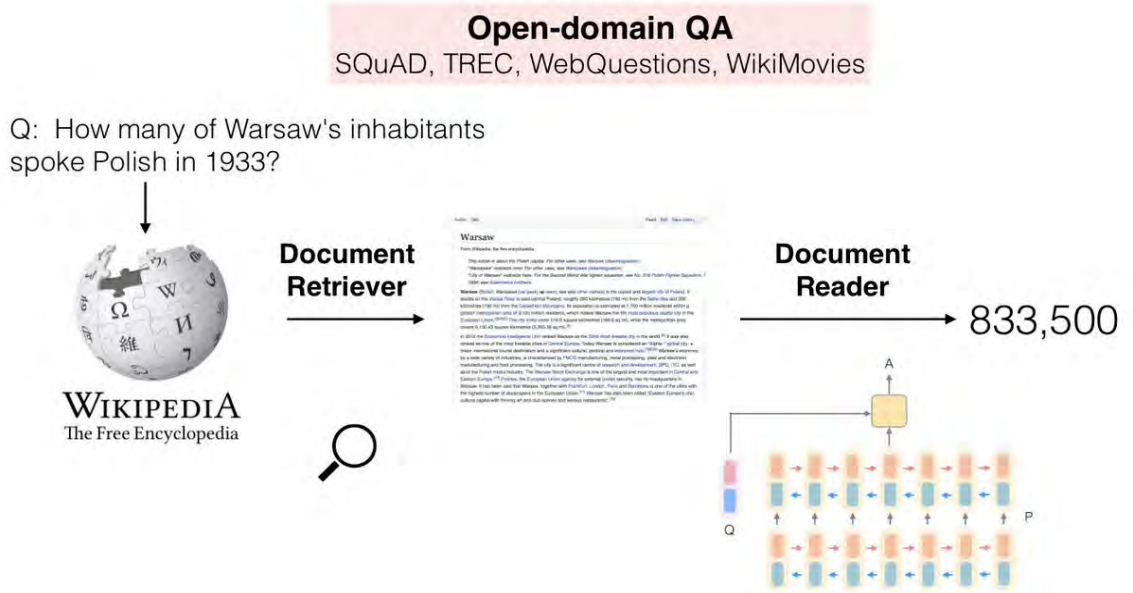


Рис. 1: Модель англоязычной вопрос-ответной системы

рения работы авторы использовали метод хэширования биграмм murmur3 [15].

Второй этап был основан на поиске точной фразы, являющейся ответом на вопрос среди k наиболее релевантных документов, найденных на первом этапе. Для этого вопрос и документ кодировались при помощи двух двунаправленных рекуррентных нейронных сетей LSTM [10].

Обозначим все слова из документа через $d_1, \ldots, d_m$. Тогда последовательность $\hat{d}_1, ..., \hat{d}_m$ представляет собой набор признаков, полученных для каждого слова d_i. Данная последовательность признаков подавалась на вход нейронной сети. В результате получался новый вектор признаков для каждого слова d_i, характеризующий контекст, в котором встречалось данное слово.

$$\{\mathrm{d}_1, \ldots, \mathrm{d}_\mathrm{m}\} = RNN(\hat{d}_1, ..., \hat{d}_m)$$

В качестве базовых признаков использовалось предобученное на корпусе текстов англоязычной Википедии Word2Vec [11] представление слова из документа ($f_{emb}(d_i)$). В качестве дополнительных признаков были использованы части речи, именованные сущности, частоты появления слова во всей коллекции документов (f_{token}), индикаторы присутствия слова из документа в вопросе ($f_{exact-match}$) и признак, учитывающий нелинейную близость между словами из вопроса $q_j, \forall j$ и словом документа (f_{align}). Для каждого слова все признаки были склеены в единый вектор и поданы на вход модели кодирования.

1. $f_{emb}(d_i) = Word2Vec(d_i)$. Размер вектора 300.

2. $f_{exact-match}(d_i) = I(d_i \in q)$.
   Учитывались индикаторы присутствия слов из вопроса в документе в исходной форме, лемматизированной, а также в приведенной к нижнему регистру.

3. $f_{token}(d_i) = (POS(d_i), NER(d_i), TF(d_i))$.

POS – часть речи, NER – тип именованной сущности, TF – частота появления слова d_i всей коллекции.

4. $f_{align}(d_i) = \sum_j a_{ij} \cdot E(q_j)$, где $a_{ij} = \frac{\exp(\alpha(E(f'_{emb}(d_i))) \cdot \alpha(E(f'_{emb}(q_j))))}{\sum_{j'} \exp(\alpha(E(f'_{emb}(d'_j))) \cdot \alpha(E(f'_{emb}(q_{j'}))))}$.
$f'_{emb}(t)$ векторное представление слова t, закодированное при помощи одного полносвязного слоя с функцией нелинейности ReLU. Вес a_{ij} интерпретировался как близость между словом из документа d_i и вопросом q. Такой признак был необходим для того, чтобы уметь отличать похожие, но неидентичные слова. Например, автомобиль и транспортное средство.

Для кодирования вопроса использовалась еще одна нейронная сеть. Данная сеть принимала на вход Word2Vec представление вопроса ($f_{emb}(q_j)$). Вес b_j интерпретировался как важность слова q_j в вопросе.

$$q_1, \ldots, q_l \rightarrow \mathrm{q} = \sum_j b_j \cdot f_{emb}(q_j)$$

$$b_j = \frac{\exp(w \cdot f_{emb}(q_j))}{\sum_{j'} \exp(w \cdot f_{emb}(q_{j'}))}$$

Для предсказания начала и ответа было построено два независимых классификатора, определяющих вероятность того, что слово d_i является началом или концом ответа соответственно. Эти классификаторы принимали на вход закодированный вектор слов документа $\mathrm{d}_1, \ldots, \mathrm{d}_\mathrm{m}$ и вопроса q. В результате, выбиралось то слово d_i из документа, вероятность $P_{start}(i) \cdot P_{end}(i)$ которого максимальна.

$$P_{start}(i) \propto \exp(d_i W_s q)$$

$$P_{end}(i) \propto \exp(d_i W_e q)$$

Такой способ построения вопрос-ответной системы был протестирован авторами на таких корпусах как SQuAD [9] и CNN/Daily Mail [2]. Общий подход, а также высокие показатели качества работы на вышеуказанных данных, позволяют использовать и развивать такой подход к построению вопрос-ответной системы на русском языке.

3 Предложенный метод

В следующих разделах будет приведено поэтапное описание предложенного метода: (1) этап поиска для вопроса набора из k релевантных документов, (2) ранжирование предложений в наборе из k релевантных документов и (3) модель извлечения ответов из найденных предложений.

3.1 Поиск релевантных документов

При построении модели первого этапа будет рассмотрено несколько стандартных алгоритмов поиска, основанные на принципе обратного индекса с использованием системы оценки релевантности документа к вопросу.

Обратный индекс или инвертированный индекс – это структура данных, в которой для каждого слова коллекции документов перечислены все документы в коллекции, в которых оно встретилось. При обработке однословного поискового запроса ответ уже есть в инвертированном индексе, поэтому достаточно взять список, соответствующий слову из запроса. При обработке многословного запроса берется пересечение списков, соответствующих каждому из слов запроса. Заранее подготовленная структура обратного индекса, позволяет существенно ускорить модель поиска.

Релевантность документа к вопросу – это функция, определяющая насколько полученный документ соответствует вопросу. В данной работе будет рассмотрено несколько различных функций релевантности, учитывающих лексическое соответствие между вопросом и документом.

Первая модель (TF) определяет релевантность между вопросом и документом по частоте пересечения слов из вопроса и документа. Данный подход не лишен смысла, однако, не учитывает, что такие слова как предлоги, союзы, местоимения и так далее, то есть слова общей лексики, присутствуют в большом количестве во всех документах в коллекции. Пересечение по ним между вопросом и документом не является хорошим показателем релевантности.

$$\mathrm{TF}(D, Q) = \sum_{i=1}^{n} TF(q_i, D),\ |Q| = n$$
$$TF(t, D) = \frac{f_{t,d}}{\sum_{t' \in D} f_{t',d}}$$

Следующая модель (TF_unique) в качестве близости использует пересечения уникальных слов из вопроса и документа. Такая модель решает проблему дублирования слов из вопроса в документе, однако по-прежнему учитывает все слова из вопроса и документа с одинаковым весом. Предположим, что вопрос Q состоит из $q_1, \ldots, q_s$ уникальных слов. Обозначим через $\hat{D}$ документ D, а через $\hat{Q}$ вопрос Q, представленные набором уникальных слов из документа и вопроса соответственно. Ниже представлена формула для подсчета релевантности документа (TF_unique).

$$\mathrm{TF\_unique}(\hat{D}, \hat{Q}) = \sum_{i=1}^{s} I[q_i \in \hat{D}]$$

Модель (TF-IDF) в отличие от модели (TF) позволяет добавить информацию о степени важности слов в вопросе. Для этого для каждого слова подсчитывается количество появлений данного слова во всей коллекции. Если слово часто встречается во всей коллекции текстовых документов, то оно является не очень важным при подсчете релевантности документа к вопросу.

$$\text{TF-IDF}(D, Q) = \sum_{i=1}^{n} TF(q_i, D) \cdot IDF(q_i),\ |Q| = n$$
$$IDF(t) = \log \frac{N - n_t + 0.5}{n_t + 0.5},\ n_t = |d \in D : t \in d|$$

Последняя Модель (BM25) является модификацией модели (TF-IDF). В ней учитывается длина рассматриваемого документа. Добавление такой информации в модель можно интерпретировать как добавление априорной информации о релевантности документа к вопросу. С одной стороны, чем больше документ, тем больше в нем слов, а значит, скорее всего, найдется пересечение со словами из вопроса. С другой стороны, чем больше документ, тем больше в нем слов общей лексики, поэтому скорее всего он не является самым релевантным к вопросу.

$$BM25(D, Q) = \sum_{i=1}^{n} IDF(q_i) \frac{TF(q_i, D) \cdot (k_1 + 1)}{TF(q_i, D) + k_1 \cdot (1 - b + b \frac{|D|}{avgdl})}$$
$$b = 0.75, k_1 = 2$$
$$avgdl \text{ – mean document length},\ |Q| = n$$

В подобных моделях часто используется совпадение не только по словам, но и по n-граммам. В настоящей работе добавление n-грамм практически не повлияло на качество работы модели. Результатом работы первого этапа для каждого вопроса является набор из k релевантных документов, который затем передается модели следующего этапа. Число k было зафиксировано равным 10.

3.2 Ранжирование предложений в документах

Суть второго этапа заключается в ранжировании предложений k документов, полученных на первом этапе, по степени релевантности к вопросу. Введение этапа ранжирования предложений в документах необходимо для ускорения работы классификаторов, предсказывающих для каждого слова k документов вероятности того, что данное слово является началом правильного ответа и концом соответственно. Ранжирование предложений позволило подавать не все слова k документов, а только слова самого релевантного предложения или небольшого набора релевантных предложений.

Для построения модели ранжирования из каждого предложения-кандидата были извлечены признаки, характеризующие лексическую и семантическую похожесть между предложением-кандидатом и вопросом. Лексические признаки были основаны на текстовом совпадении вопроса и предложения. Семантические признаки основаны на сравнении семантик слов из вопроса и предложения.

Были рассмотрены следующие лексические признаки: BM25, TF-IDF, процент (TF_%) и количество (TF_count) общих слов между предложением и вопросом. Данные признаки были посчитаны по формулам, описанным в предыдущем разделе. За исключением того, что в качестве документа рассматривалось предложение.

Для построения семантических признаков была использована модель Word2Vec [11], предобученная на корпусе текстов русскоязычной Википедии [1]. Существует две архитектуры

[1] `http://rusvectores.org/ru/models/`

модели Word2Vec: Skip-gram и CBOW.

Архитектура Skip-gram (рисунок 3b) принимает на вход one-hot представление некоторого выделенного слова и для каждого слова в окрестности заранее заданного окна предсказывает насколько вероятно встретить данное слово в контексте выделенного слова. Иными словами, модель Skip-gram по выделенному слову предсказывает его контекст.

Архитектура CBOW (рисунок 3a), в отличие от Skip-gram, предсказывает слова по их контекстам.

Размер скрытого слоя (Hidden Layer), как и размер окна являются параметрами модели. Для представления слов в виде векторов, учитывающих контексты, в которых они встречаются, в данной работе будут использованы векторы слов, полученные на скрытом слое модели Skip-gram с размером вектора, равным 300 и шириной окна, равной 5.

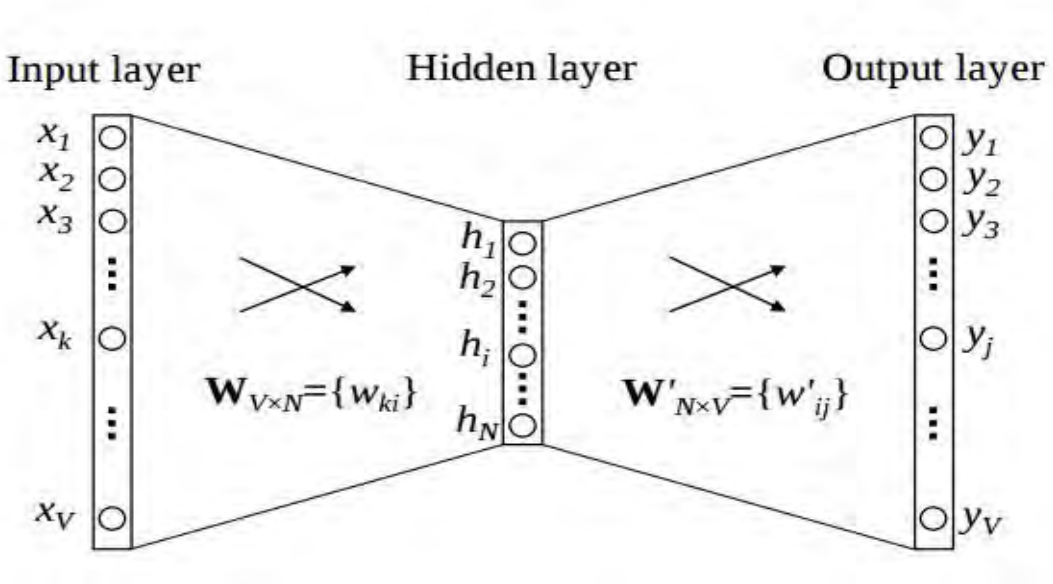
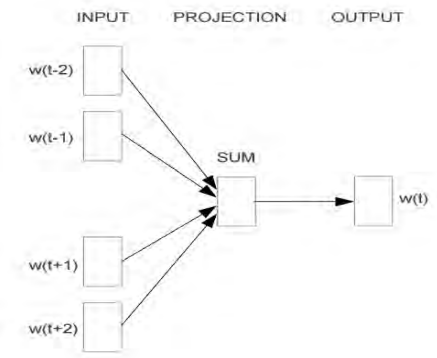
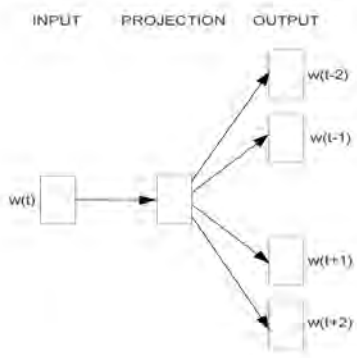


Рис. 2: Модель Word2Vec



(a) CBOW



(b) Skip-gram

В качестве семантического признака было использовано косинусное расстояние между средним вектором вопроса и средним вектором предложения.

$\text{cosine\_dist}(\vec{Q}, \vec{S}) = 1 - \frac{<\vec{Q},\vec{S}>}{||\vec{Q}||\cdot||\vec{S}||}$

В качестве вектора вопроса и предложения можно также рассматривать тематические векторы, представляющие собой вероятности присутствия набора тем в вопросе и в предложении соответственно. Для построения таких векторных представлений вопроса и предложения, была взята предобученная на коллекции русскоязычной Википедии тематическая модель со 100 темами. В качестве семантического признака была использована дивергенция Кульбака-Лейблера ($D(\vec{Q}||\vec{S})$) между тематическим профилем вопроса и предложения. Такой признак показывал, насколько тематический профиль вопроса схож с тематическим профилем предложения. $D(\vec{Q}||\vec{S}) = E_{\vec{Q}} \log \frac{\vec{Q}}{\vec{S}}$. Добавление в модель такого признака практически не повлияло на качество ее работы.

Был предложен способ построения семантического признака, учитывающего тип вопроса и наиболее часто встречаемых именованных сущностей [13] в ответах на данный тип вопроса.

Для определения типа вопроса было использовано несколько различных методик. В первой методике тип вопроса определялся по вопросительному местоимению из вопроса. Было выделено всего 13 различных типов вопроса. Был также рассмотрен метод, определяющий тип вопроса по вопросительному местоимению и следующему за ним слову. После фильтрации по встречаемости данного типа вопроса осталось несколько тысяч различных типов. Третий и наиболее успешный метод определения типа вопроса был основан на модели кластеризации. TF-IDF представление вопроса было сжато с помощью метода главных компонент до размерности 50. Для каждого вопросительного местоимения были добавлены индикаторы наличия их в вопросе. На получившемся наборе признаков была построена модель кластеризации kmeans. В результате типом вопроса являлся номер кластера. Количество используемых кластеров было зафиксировано равным 100. Семантический признак являлся индикатором наличия в предложении именованной сущности, характерной для ответа на данный тип вопроса.

Именованная сущность (NER) – это слово или словосочетание, обозначающее предмет или явление определенной категории. Примерами именованных сущностей являются имена людей, названия организаций и локаций. В настоящей работе было зафиксировано пять различных именованных сущностей. Имена, фамилии и отчества людей – как единая сущность, организация, локация, дата и число.

Все описанные признаки были объединены и поданы на вход модели бинарной классификации Logistic Regression, предсказывающей вероятность наличия ответа на вопрос в данном предложении. Далее ранжирование производилось по вероятности наличия в предложении ответа на вопрос. Подход с ранжированием предложений позволял нам не терять потенциально хорошие предложения-кандидаты. В результате, модели третьего этапа для каждого вопроса был передан ранжированный список предложений.

3.3 Извлечение ответов из предложений

Третий этап заключается в поиске точной фразы, являющейся ответом на вопрос, в предложениях, ранжированных на втором этапе. В качестве модели третьего этапа была использована модель рекуррентной двунаправленной нейронной сети, с архитектурой, аналогичной архитектуре, предложенной в работе [1]. На рисунке 4 продемонстрирована модель двунаправленной рекуррентной нейронной сети.

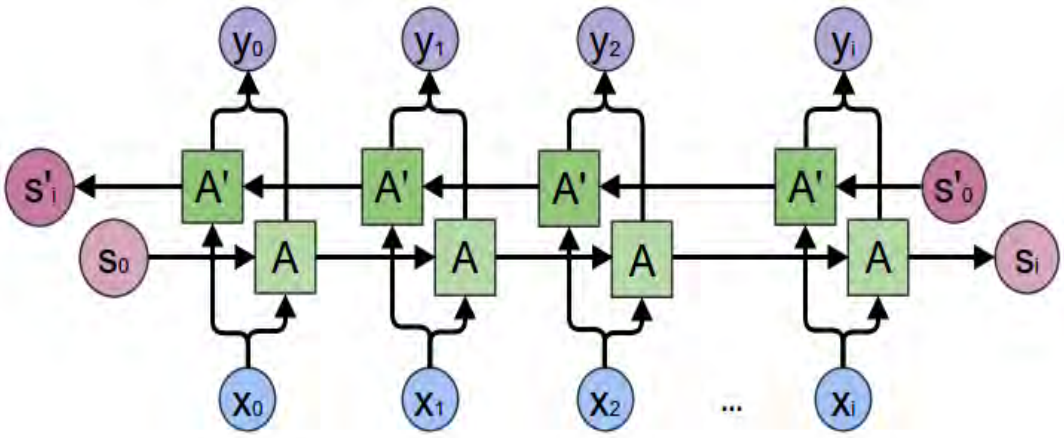


Рис. 4: Двунаправленная рекуррентная нейронная сеть

В базовой версии для кодирования предложения были использованы все признаки, описанные в [1], кроме именованных сущностей (NER) и частей речи (POS). Они были добавлены позднее. Признаки для кодирования вопроса были аналогичными. Далее будут описаны дополнительные признаки, добавленные к базовому набору.

В качестве лексических признаков для слов из предложения были использованы BM25 и TF-IDF, посчитанные между словом из предложения и вопросом. Формулы для данных моделей аналогичны формулам, описанным в разделе 3.1. Отличие заключается в том, что на данном этапе в качестве запроса выступает слово из предложения, а вопрос рассматривается в качестве документа.

Для построения семантических признаков было использовано косинусное расстояние между Word2Vec представлением слова из предложения и усредненным вектором вопроса. Предобученная модель Word2Vec была взята аналогичной модели, описанной в разделе 3.2.

Был предложен способ построения признака, учитывающий тип вопроса, аналогичный способу, продемонстрированному на втором этапе. За исключением того, что итоговый признак состоял из набора всех возможных комбинаций именованных сущностей слов из предложений и именованных сущностей, характерных для данного типа вопроса (Interaction).

Был проведен эксперимент, сравнивающий два способа построения вопрос-ответной системы: с учетом второго этапа, ранжирующего предложения по релевантности их вопросу и без него. В первом способе модель предсказывала начало и конец ответа для всех предложений среди k документов. Во втором способе в модель подавались первое или небольшое количество первых предложений в ранжированном списке. Если вероятность наличия ответа

в поданном предложении была выше порога, подобранного на валидации, в таком случае, все остальные предложения не рассматривались. Такой подход позволил ускорить этап применения модели вопрос-ответной системы примерно в 16 раз без существенных потерь в качестве.

4 Эксперименты

В следующих разделах будут приведены результаты экспериментов с моделями каждого из этапов. Все эксперименты проводились на данных, предоставленных организаторами конкурса Sberbank Data Science Journey[2]. Датасет содержал 50K уникальных вопросов, ответы на которые были выделены асессорами в параграфах русскоязычной Википедии. Вопросы были представлены на естественном языке и сформированы самими асессорами.

4.1 Поиск релевантных параграфов Википедии

На данном этапе для каждого вопроса необходимо было найти набор из k наиболее релевантных документов. В качестве метрики качества первого этапа была выбрана точность по первым k документам.

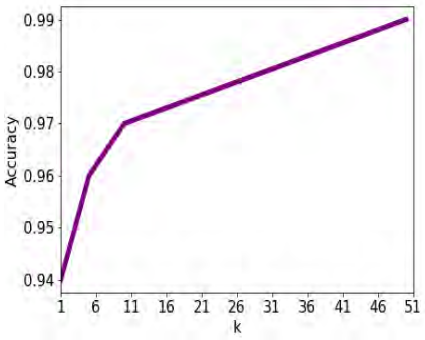
В таблице 1 продемонстрировано качество работы моделей, использующих различные системы оценки релевантности на подвыборке из 1K уникальных вопросов. Количество релевантных документов было зафиксировано равным $k = 10$. Модель BM25 оказалась наиболее конкурентоспособной. Ее качество было близко к 100%, поэтому решено было зафиксировать ее как лучшую.

Также было проведено исследование влияния качества работы модели BM25 для разных значений параметра k. Очевидно, что чем больше k, тем выше качество (рисунок 5a). Однако, для того чтобы модель второго этапа могла быстро и эффективно произвести ранжирование предложений в наборе выделенных документов, был построен график зависимости от k среднего количества предложений на вопрос (5b). На построенных графиках видно, что качество модели достигает порядка 97% при $k = 10$, а среднее количество документов на вопрос равно примерно 70. Качество при $k = 50$ составляет уже порядка 99%, однако, среднее количество документов на вопрос – свыше 200. В связи с этим, выдача модели BM25 с $k = 10$ была передана на следующий этап.

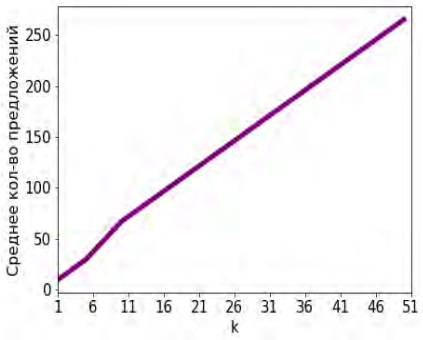
TF	**TF_unique**	**TF-IDF**	**BM25**
0.66	0.76	0.88	**0.97**

Таблица 1: Сравнение моделей поиска релевантных документов на подвыборке из 1K уникальных вопросов, $k = 10$.

[2] https://github.com/sberbank-ai/data-science-journey-2017



(a) Качество работы модели поиска в зависимости от k



(b) Среднее кол-во предложений в выдаче в зависимости от k

Рис. 5: Сравнение результатов работы модели BM25 для разных k

4.2 Ранжирование предложений в наборе релевантных параграфов

Данный этап производил ранжирование предложений в наборе из k релевантных к вопросу документов, найденных на первом этапе. В качестве метрики качества было зафиксировано AUC (Area Under Curve), усредненное по вопросам. Мера качества AUC интерпретируется как число правильно упорядоченных пар в ранжированном списке, поэтому логично взять ее в качестве метрики, оценивающей качество ранжирования. Результаты работы моделей второго этапа представлены в таблице 2. Модель ML представляла собой модель логистической регрессии, предсказывающей вероятность наличия в предложении ответа на вопрос. В качестве признаков были взяты все лексические признаки, качество ранжирования по которым представлено в таблице, а также семантические признаки, которые не сильно, но повлияли на качество ранжирования. Эти признаки более сильно окажут влияние на качество работы последней модели, которой необходимо будет выделять в предложении правильный ответ. На основании результата работы лучшей модели (ML), можно сказать, что в среднем, предложение, в котором содержится правильный ответ на вопрос, находится на 4 позиции в ранжированном списке.

TF-IDF	**BM25**	**Term_%**	**Term_count**	**ML**
0.94	0.94	0.94	0.93	**0.97**

Таблица 2: Сравнение моделей ранжирования предложений в документах на тестовых данных

4.3 Извлечение ответов из первых предложений в ранжированном списке

Данный этап является последним и самым сложным в построении вопрос-ответной системы. Необходимо среди ранжированного набора предложений выделить правильный ответ

на вопрос.

Для оценки качества третьего этапа были рассмотрены точность определения начала ответа (start), точность определения конца ответа (end) и точность по полному совпадению выделенного ответа с правильным (exact).

В первую очередь на данном этапе была исследована зависимость качества работы базовой модели, предложенной в работе [1] от добавления новых признаков, описанных в настоящей работе. В таблице 3 представлен результат работы модели на 10 эпохе. В каждой колонке продемонстрировано качество работы модели при добавлении соответствующего признака. Например, при добавлении именованных сущностей и частей речи для слов из предложения, точность определения начала увеличилась на 5% в абсолютном значении и на 8% в относительном. Модель, обученная на всех описанных признаках (ALL) продемонстрировала наилучшее качество. На 10 эпохе абсолютное улучшение относительно базовой модели составило порядка 8% в абсолютных величинах и 15% в относительных. Качество, зафиксированное на 40 эпохе составило порядка 60% точности определения обоих концов правильного ответа и порядка 50% точного совпадения.

	Base	**Ner, Pos**	**Cosine_dist**	**BM25, TF-IDF**	**Interaction**	**ALL**
start	45.57	49.36	50.16	50.88	51.31	**52.63**
end	42.25	47.03	47.72	47.89	49.76	**50.14**
exact	29.01	33.71	34.58	34.90	36.38	**36.88**

Таблица 3: Сравнение моделей извлечения ответов из предложений тестового множества на 10 эпохе.

Так как добавление признака (Interaction) продемонстрировало наибольшее увеличение в качестве работы модели, предлагается подробнее рассмотреть способ его построения и подбора параметров для его построения.

Для построения данного признака сначала определялся тип вопроса. Для этого было рассмотрено несколько методик. В первой методике в качестве типа вопроса бралось вопросительное местоимение, присутствующее в вопросе. Такой подход выделил 13 различных типов вопросов. Второй подход в качестве типа вопроса фиксировал вопросительное местоимение и следующее после него слово. После фильтрации типов вопросов по частоте встречаемости их в датасете, образовалось несколько тысяч разных типов. Наконец, последняя методика определения типа вопроса была основана на выделении TF-IDF признаков из вопросов. Затем использовался метод снижения размерности PCA и добавлялись индикаторы наличия в вопросе соответствующего вопросительного местоимения. Весь этот набор признаков подавался на вход модели кластеризации kmeans. В результате, в качестве типа вопроса брался номер кластера. Последний подход оказался самым успешным и более того интерпретируемым.

На рисунке 7 продемонстрирован результат кластеризации вопросов по типу на подвыбор-

ке из 10 уникальных типов вопросов по 1K случайных объектов внутри каждого кластера. В таблице 4 показаны примеры вопросов из этих кластеров. Общий цвет для вопросов означает общий кластер. На рисунке 7 и по таблице 4 видно, что кластеры получились интерпретируемыми и разнообразными.

Результаты подбора оптимального количества кластеров и размерности подпространства в модели PCA представлены на графиках 8 и 9 соответственно. Эксперименты проводились на подвыборке из 1K уникальных вопросов из тестового множества и 3K – из обучения. Качество, представленное на графиках, было зафиксировано на 10 эпохе.

На основании этих экспериментов для построения признака (Interaction) были зафиксированы количество кластеров, равное 100, размерность подпространства в модели PCA, равное 50.

Также было рассмотрено несколько способов построения модели третьего этапа. С учетом этапа ранжирования и без него. В первом способе модели третьего этапа подавались на вход последовательно предложения в том порядке, в котором они идут в ранжированном списке. Если вероятность $P_{start} \cdot P_{end}$ была выше фиксированного порога, тогда выделенный в данном предложении ответ брался в качестве итогового, а остальные предложения далее не рассматривались. Во втором способе в модель подавались все предложения. Внутри каждого предложения были выделены наиболее вероятные ответы. В результате, в качестве итогово ответа брался ответ с максимальной вероятностью $P_{start} \cdot P_{end}$. На графике 6 продемонстрирована зависимость точного совпадения выделенного ответа с правильным от порога. Видно, что оптимальное значение порога равно 0.25. При таком выборе порога, качество модели, рассматривающей только первое предложение или небольшой набор первых предложений в ранжированном списке, практически не отличается от качества модели, которая рассматривает все предложения подряд. При этом скорость работы модели увеличилась в 16 раз.

Данный результат достаточно легко интерпретируется. Модель ранжирования второго этапа продемонстрировала средний AUC по вопросам порядка 97%. Это означает, что в среднем, предложение, содержащее правильный ответ содержится на 4 позиции в ранжированном списке. В средним на вопрос приходилось порядка 70 предложений. Отсюда следует, что при идеальном подборе порога для модели третьего этапа, скорость работы модели должна увеличиться примерно в $16 - 17$ раз. Что и было подтверждено в данном эксперименте.

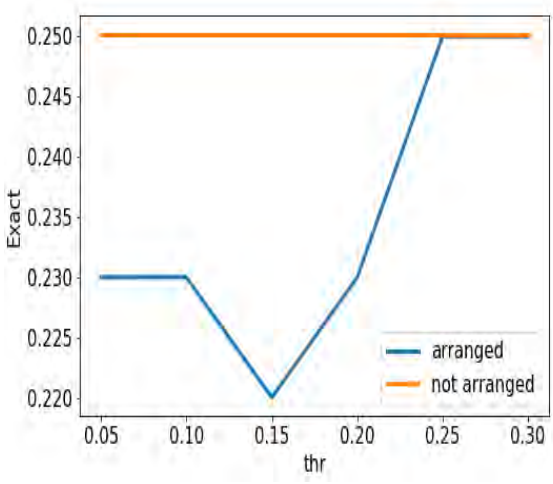


Рис. 6: Зависимость качества работы от порога принятия решения модели третьего этапа

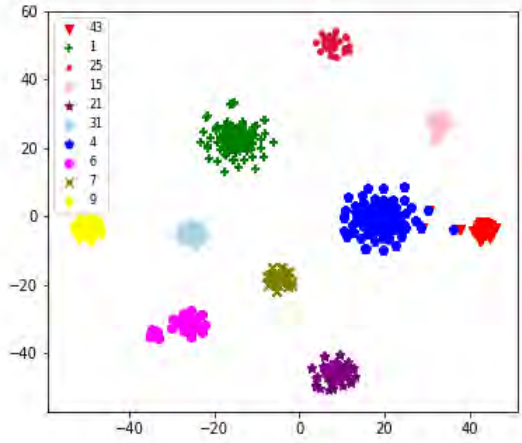


Рис. 7: Кластеризация вопросов по типу, kmeans. Число кластеров 100, размерность подпространства в модели PCA равна 50. Визуализация t-SNE. Подвыборка из 10 наиболее частотных типов вопросов. 1K случайных объектов внутри каждого кластера.

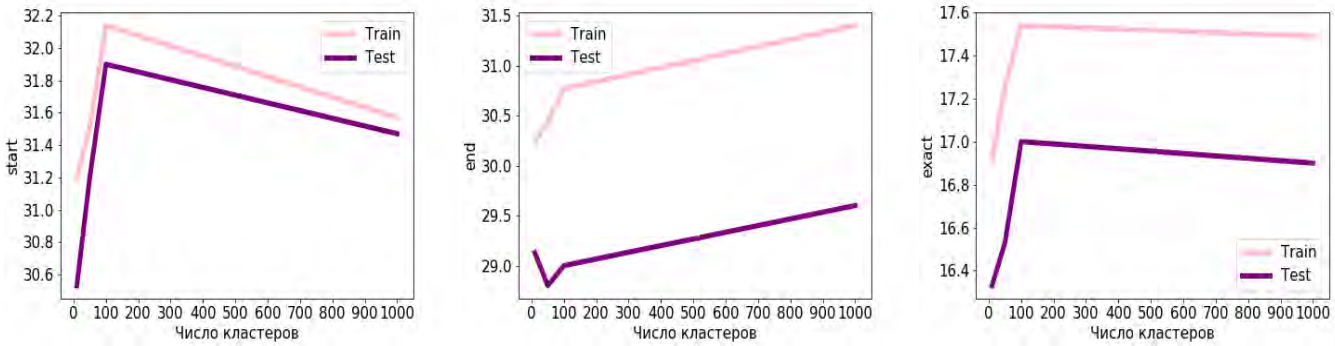


Рис. 8: Зависимость качества от числа кластеров в модели kmeans

• Когда были сформулированы многие из основных черт современного компьютера?	• Правительство какого штата в Индии выпустило документ о переходе всех местных школ на использование Linux?
• Когда было объявлено о создании Советской Латвии?	• В каком возрасте поступают в высшие школы Республики Корея?
• Когда в Вавилоне был сооружен первый известный подводный тоннель?	• В каком звании король Луи-Филипп назначил Лафайета Жильбера?
• Сколько всего проповедей зачитывается кардиналам?	• Что сделал Руссо?
• Сколько на 1 февраля 2014 года в городе насчитывается трамвайных маршрутов?	• Чем обладает любая полифоническая пьеса?
• Сколько сонат и вариаций написал Моцарт для скрипки и клавесина ?	• Что стало последним сочинением Шостаковича?
• Где применяют лазер для создания искусственных опорных звезд в верхних слоях атмосферы?	• Какие частоты слышат мыши ?
• Где встечаются крокодилы и черепахи?	• Какие вещества способны обратимо изменять светопоглощение?
• Где расположен Амстердам?	• Какой важнейший фактор, предпределяющий распространение бабочек?
• В каком году Еврипид оставил Афины?	• Кто стал луреатом Нобелевской премии в 1906 году вместо Менделеева?
• В каком году выпустили электрогитару Stratocaster?	• Кого принял Апостол Иоанн?
• В каком году открылась Галерея тысячелетия в Шеффилде?	• Кто предложил использовать подстановки (англ. substitution) и перестановки (англ. permutation)?

Таблица 4: Примеры вопросов, кластеризация kmeans. Число кластеров равно 100, размерность подпространства в модели PCA равна 50.

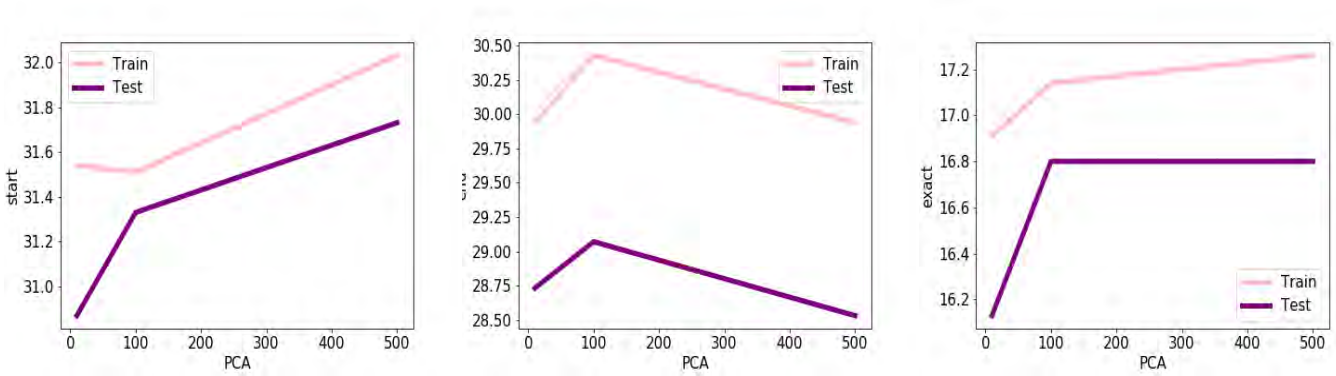


Рис. 9: Зависимость качества от размерности подпространства в модели PCA, число кластеров зафиксировано равным 100.

5 Заключение

В рамках проведенного исследования была построена вопрос-ответная система на русском языке. Представленное решение разбивается на три последовательных этапа. На первом этапе для каждого вопроса осуществляется поиск небольшого набора из k релевантных документов. На втором этапе производится ранжирование предложений внутри найденных документов по релевантности вопросу, а также ранжирование по вероятности наличия в предложении правильного ответа. Наконец, последний этап осуществляет поиск финального ответа в ранжированном списке предложений. Третий этап настоящей работы подразумевал, что предложения последовательно выбираются из ранжированного списка. Если модель находила ответ, в переданном ей предложении, тогда этот ответ выбирался в качестве итогового без просмотра остальных предложений. Также в ходе решения задачи третьего этапа было рассмотрено построение набора из лексических и семантических признаков и исследовано влияние каждого признака на качество модели.

Такой подход обладает следующими преимуществами:

1. Ускоряет работу вопрос-ответной системы на этапе ее применения в 16 раз без существенных потерь в качестве

2. Превосходит на 15% существующие методы извлечения ответов из текста за счет добавления набора из синтаксических и семантических признаков для слов из предложения

3. Метод универсален и может быть успешно использован для построения вопрос-ответной системы на любой коллекции документов

В настоящее время существует большое количество вопрос-ответных систем на английском языке, демонстрирующих высокие показатели точности определения правильного ответа в

тексте. Вопрос-ответная система на русском языке является более трудоемкой задачей как из-за сложности самого языка, так и из-за отсутствия больших размеченных датасетов. В настоящей работе был использован один из известных датасетов, состоящий из 50K уникальных пар вопросов и ответов, составленных на естественном языке.

К сожалению, рассмотренные данные были не самыми качественными с точки зрения модели. В них встречались вопросы, заданные к конкретному контексту, например, «Чему они уступили место?». Часть из таких вопросов была отсеяна моделью первого этапа, так как невозможно было найти для данного вопроса небольшой набор релевантных документов. Введение дополнительных семантических признаков для таких вопросов не приводило к улучшению в качестве. Например, признак, характеризующий часто встречаемую именованную сущность в ответах на данных тип вопроса, практически лишен смысла. Также, такой вопрос состоит почти полностью из слов общей лексики, а это значит, что скорее всего он будет близок по косинусному расстоянию почти к любому слову общей лексики предложения.

Поэтому, одним из важных направлений дальнейших исследований можно считать построение больших размеченных датасетов для русскоязычной вопрос-ответной системы с отсутствием вопросов, заданных в конкретном контексте.

Список литературы

[1] *Danqi Chen, Adam Fisch, Jason Weston and Antoine Bordes* Wikipedia to Answer Open-Domain Questions. // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada, 2017, P. 1870–1879.

[2] *Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, Phil Blunsom* Teaching Machines to Read and Comprehend. //NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, 2015. P. 1693–1701.

[3] *David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al* Building Watson: An overview of the DeepQA project. //AI magazine, 2010. P.59–79.

[4] *Huan Sun, Hao Ma, Wen-tau Yih, Chen-Tse Tsai, Jingjing Liu, and Ming-Wei Chang* Open domain question answering via semantic enrichment. //In Proceedings of the 24th International Conference on World Wide Web. ACM, Florence, Italy, 2015. P. 1045–1055.

[5] *Pum-Mo Ryu, Myung-Gil Jang, and Hyun-Ki Kim.* Open domain question answering using Wikipedia-based knowledge model. //Information Processing and Management: an International Journal archive Volume 50 Issue 5, NY, USA, 2014. P. 683–692.

[6] *Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu et al.* CQArank: jointly model topics and expertise in community question answering. //Proceedings of the 22nd ACM international conference on Information and Knowledge Management, San Francisco, California, USA. 2013. P. 99–108.

[7] *Asli Celikyilmaz, Dilek Hakkani-Tur, Gokhan Tur* LDA based similarity modeling for question answering. //SS '10 Proceedings of the NAACL HLT 2010 Workshop on Semantic Search, Los Angeles, California, 2010. P. 1–9.

[8] *Kyoung-Soo Han, Young-In Song, Hae-Chang Rim* Probabilistic model for definitional question answering. //SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA, 2006. P. 212–219.

[9] *Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang* SQuAD: 100,000+ questions for machine comprehension of text. //Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[10] *Sepp Hochreiter, Jürgen Schmidhuber* Long Short-Term Memory. //Journal Neural Computation archive Volume 9 Issue 8. MIT Press Cambridge, MA, USA, 1997. P. 1735–1780.

[11] *Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean* Distributed Representations of Words and Phrases and their Compositionality. //NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems – Volume 2, Lake Tahoe, Nevada, 2013. P. 3111–3119.

[12] *S. E. Robertson and S. Walker.* Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval // Proceedings of the 17th annual international ACM SIGIR conference on Research and develop- ment in information retrieval, 1994, Dublin, Ireland, P. 232–241.

[13] *Korobov M.* Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts, 2015, Yekaterinburg, Russia P. 320–332.

[14] *Phillips, A. V.* A question-answering routine. MIT AI Lab. 1960.

[15] *Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg.* Feature hashing for large scale multitask learning. // In International Conference on Machine Learning (ICML), 2009, Montreal, Quebec, Canada. P. 1113–1120.