ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Московский институт электроники и математики им. А.Н. Тихонова

Кряжова Анастасия Александровна

# МЕТОДЫ ОЦЕНИВАНИЯ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ФРАЗ ДЛЯ КЛАССИФИКАЦИИ ТЕКСТОВЫХ СООБЩЕНИЙ

Выпускная квалификационная работа
студентки образовательной программы бакалавриата
«Прикладная Математика»

по направлению подготовки Прикладная Математика

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

Москва, 2019

**Abstract**

Short text or single phrase classification task exists in client inquiries online routing systems and in intelligent conversational systems to identify client intents. One of the crucial problems in this task is the identification and matching of paraphrases. The same intent can be explained by several different phrases, even with completely different words. This problem dramatically complicates the classification task and makes conventional methods of text classification based on word frequency useless. In this work, we propose using probabilistic topic modeling with partial supervised learning and automatic detection of the informative word collocations together with neural vector interpretation of words to evaluate semantic similarity of phrases. These methods was tested on a corpus consisting of question pairs with binary indication and was compared with the Word2Vec and FastText models.

**Аннотация**

Задача классификации коротких текстовых сообщений возникает в системах маршрутизации обращений клиентов, а также в системах разговорного интеллекта при распознавании намерений собеседника. Важной проблемой при решении данной задачи является распознавание парафраз. В коротких сообщениях одно и то же намерение может быть перефразировано многими способами, иногда вообще не имеющими одинаковых слов. Данное обстоятельство существенно усложняет задачу и делает неприменимыми обычные методы классификации текстов, основанные на частотных признаках слов. В данной работе предлагается совместно использовать вероятностные тематические модели с частичным обучением и автоматическим выделением информативных словосочетаний и нейросетевые векторные представления слов для оценивания семантической близости коротких сообщений. Рассматриваемые методы протестированы на коллекции пар вопросов с бинарным целевым признаком, и проведено их сравнение с конкурирующими моделями Word2Vec и FastText.

**Ключевые слова:** *короткий текст, классификация текстов, тематическое моделирование, Word2Vec, FastText, векторные представления слов.*

# Содержание

# 1 Введение

Классификация текстовых сообщений – фундаментальная задача обработки естественного языка (Natural Language Processing). К задачам *NLP* относятся построение диалоговых, рекомендательных, вопрос-ответных и информационно-поисковых систем. Взаимодействие человека с данными системами оптимизирует процесс в консультационном, образовательном и развлекательном бизнесе. Данное взаимодействие – непростая задача даже для человека, так как в ходе работы требуется переключаться с одной предметной области на другую, понимать ситуацию в целом, исходя из её краткого нечёткого описания.

Автоматическая классификация текстов осложняется также наличием в пользовательских в запросах (коротких текстах) ошибок, слов-паразитов, жаргонизмов. Распространённая проблема при классификации коротких текстов заключается в том, что в естественном языке различные фразы могут обладать схожим смыслом, даже не имея пересечений по словам. И, наоборот, одно слово может обладать различными значениями в контекстах разных тем.

Определим основные термины, используемые в работе. *Тема* – предметная область письменного или устного изложения, имеющая характерную устойчивую терминологию. *Тематика* – это множество тем. *Тематизацией* будем называть определение тематики коллекции в целом и каждого документа в отдельности. *Дистрибутивная семантика* – область лингвистики, которая вычисляет степень семантической близости между лингвистическими единицами (словами) на основании их распределения (дистрибуции) в текстовых коллекциях. *Коротким текстом* называют документ, длина которого не достаточна для надёжного определения тематики [1]. *Полисемия* (многозначность) – наличие у слова нескольких значений. *Контекст* слова – часть текста, окружающая данное слово. *Парафразы* – различные по словам фразы, имеющие одинаковый смысл.

Цель данной работы — построение модели для определения семантической близости двух коротких текстов без использования предобученных векторных представлений слов, с учетом многозначности слов.

На сегодняшний день для определения семантической близости коротких текстов используются модели, основанные на идеях дистрибутивной семантики, которые представляют слова в векторной форме с использованием контекста слов. Так как документы в коллекциях коротких текстов лишены объемного контекста, обучение векторных представлений только за счёт данного корпуса не приносит модели высокую точность прогноза. В данном случае в модели используют векторные представления слов, предварительно обученные по большому референтному корпусу текстов, например, по Википедии[1]. Проблема использования предобученных векторов слов в том, что они не учитывают или плохо учитывают многозначность слов. Предобученный вектор слова отражает все его смыслы, встретившиеся в референтном корпусе, независимо от того, что в каждом конкретном контексте данное слово употребляется только в одном определённом смысле.

Задача работы состоит в выявлении семантической близости между парой коротких

[1] http://www.cs.upc.edu/ nlp/wikicorpus/

текстов и последующей бинарной классификацией данных текстов для выделения парафраз.

Для решения поставленной задачи предлагается разделить коллекцию текстов на конечное число монотематичных подколлекций, в которых каждое слово обладает единственным смыслом, определяемым тематикой данной подколлекции, и обучать векторные представления слов по этим подколлекциям. Для преобразования исходной текстовой коллекции в набор монотематичных подколлекций предлагается использовать вероятностное тематическое моделирование с частичным обучением, которое предполагает, что появление слова в тексте связано с конкретной темой. После тематизации коллекции векторные представления слов строятся отдельно для каждой темы с использованием архитектуры Skip-gram [8].

Два коротких текста предлагается сравнивать с помощью векторных представлений слов каждой модели Skip-gram, обученной по коллекции, которая является общей коллекцией для обоих текстов.

Качество модели будет оцениваться с помощью меры *precision* (точность). Результаты предложенного метода предлагается сравнивать с нейросетевыми моделями Word2Vec [7] и FastText [10].

Работа предполагает изучение методов предобработки текста и их влияние на точность предсказания модели Skip-gram.

# 2 Тематическое моделирование

На сегодняшний день оценивание семантической близости слов и фраз активно исследуется и применяется в компьютерной лингвистике. За несколько десятков лет предложено множество методов. Первые из них, как и машинное обучение в целом, основаны на математической статистике, где выделение нескольких элементарных признаков обеспечивает модели высокое качество. С течением времени было определено, что классическими методами машинного обучения, основанными на гипотезе «мешка слов», задача выявления схожести фраз решается плохо.

Модели дистрибутивной семантики основываются на описании слова как единицы корпуса данных и *дистрибутивной гипотезе*: лингвистические единицы, встречающиеся в схожих контекстах, имеют близкие значения [2]. В роли контекста в тематических моделях выступает весь документ.

## 2.1 Постановка задачи

Введём три конечных множества: множество (коллекция) документов $D$, множество (словарь) термов $W$ и множество тем $T$. Для каждой темы $t$ определим словарь $W_t$. Коллекция текстовых документов представляется выборкой троек $(w, d, t)$, генерируемых случайно и независимо из распределения $p(w, d, t)$, заданного на дискретном вероятностном пространстве $W \times D \times T$. Обозначим через $n$ объём этой выборки, равный суммарной длине всех документов коллекции. Каждый документ $d \in D$ описывается дискретным распределением $p(t|d)$ на множестве тем $t \in T$, каждая тема $p(w|t)$ на множестве терминов $w \in W_d$. Определим $n_{dw}$ – число вхождений терминов $w_1, w_2, ..., w_{n_d}$, $w \in W$ в документ $d$ длины $n_d$.

Тогда задачу тематического моделирования можно определить как нахождение темы $t \in T$ для каждого термина $w \in W_d$.

## 2.2 Тематическая модель

*Вероятностный латентный семантический анализ* (probabilistic latent semantic analysis, PLSA) предложен в [3]. Вероятностная модель появления термина в документе определяется на основе гипотезы условной независимости определения условной и полной вероятности:

$$p(w|d) = \sum_{t \in T} p(t|d)p(w|t).$$

Предполагаем, что количество тем много меньше числа документов и числа терминов. Тогда можно представить задачу как поиск приближенного представления заданной матрицы частот $F = (\hat{p}(w|d))_{W \times D} = (\frac{n_{dw}}{n_d})_{W \times D}$ в виде произведения $F \approx \Phi\Theta$. Обозначим: $\varphi_{wt} = p(w|t)$ и $\theta_{td} = p(t|d)$. Построение тематической модели – нахождение стохастических матриц терминов тем $\Phi = (\varphi_{wt})_{W \times T}$ и тем документов $\Theta = (\theta_{td})_{T \times D}$ по коллекции D.

Для нахождения параметров $\Phi$ и $\Theta$ по исходным данным $n_{dw}$ применяется EM-алгоритм, максимизирующий логарифм правдоподобия вероятностной модели порождения данных (1)

при условиях (2).

$$L(\Phi,\Theta) = \ln \prod_{d\in D}\prod_{w\in d} p(w|d)^{n_{dw}} = \sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} \to \max_{\Phi,\Theta}; \tag{1}$$

$$\sum_{w\in W} \varphi_{wt} = 1; \quad \varphi_{wt} > 0; \qquad\qquad \sum_{t\in T} \theta_{td} = 1; \quad \theta_{td} > 0. \tag{2}$$

Таким образом, тематическая модель основана на предположениях, что каждое слово в документе связано с определённой темой, при этом порядок слов в документе не важен (гипотеза «мешка слов») и порядок документов в коллекции также не важен.

В работе [4] описывается подход Word Network Topic Model (WNTM), с помощью которого набор предложений заменяется последовательностью псевдо-документов $d_u$, которые строятся для каждого слова из словаря как объединение всех его контекстов в коллекции. В данном случае моделируются не документы, а связи между словами:

$$p(w|d_u) = \sum_{t\in T} p(w|t)p(t|d_u).$$

Данный подход компенсирует размер документов коллекции. При этом псевдо-документы построены исходя из смешанного контекста слова.

## 2.3 Регуляризация тематических моделей

Нахождение разложения $\Phi\Theta$ осложняется его неопределенностью и неединственностью, т.к. $\Phi\Theta$ определено с точностью до невырожденного преобразования: $\Phi\Theta = (\Phi S)(S^{-1}\Theta)$, где $(\Phi S)$ и $(S^{-1}\Theta)$ – также стохастические матрицы.

*Регуляризация* – метод решения задач, которые имеют неединственное или неустойчивое решение. Данный подход предполагает наложение дополнительных условий на $\Phi, \Theta$ для сужения множества решений.

Наложение разного рода ограничений при построении тематической модели позволяет учитывать дополнительные данные и требования к модели, и в итоге строить модели с заданными свойствами. Метод аддитивной регуляризации тематических моделей (additive regularization of topic modeling, ARTM), предложенный в [6], позволяет комбинировать несколько параметрических ограничений для одной тематической модели.

Каждое ограничение, из дополнительных $n$ критериев $R_i(\Phi,\Theta)$, $i = 1,\ldots,n$, называется регуляризатором. Для решения задачи многокритериальной оптимизации максимизируется линейная комбинация критериев $L(\Phi,\Theta)$ и $R_i(\Phi,\Theta)$, с неотрицательными коэффициентами регуляризации $\tau_i$ при условиях (2):

$$L(\Phi,\Theta) + R(\Phi,\Theta) \to \max_{\Phi,\Theta}, \qquad R(\Phi,\Theta) = \sum_{i=1}^{n} \tau_i R_i(\Phi,\Theta). \tag{3}$$

Наиболее популярной тематической моделью является – *латентное размещение Дирихле*

(latent Dirichlet allocation, LDA) предложенное Дэвидом Блейем в 2003 году [5]. Вероятностная модель та же, что предлагал Хофманн [3], при дополнительном предположении, что векторы документов и векторы тем порождаются распределениями Дирихле при условиях (2):

$$P(\theta_d, \alpha) = \frac{\Gamma(\alpha_0)}{\prod_t \Gamma(\alpha_t)} \prod_t \theta_{td}^{\alpha_t - 1}, \qquad P(\varphi_t, \beta) = \frac{\Gamma(\beta_0)}{\prod_w \Gamma(\beta_w)} \prod_w \varphi_{wt}^{\beta_w - 1},$$

где $\alpha_t > 0, \quad \beta_w > 0, \quad \Gamma(x)$ – гамма-функция.

Таким образом, получаем регуляризатор:

$$R(\Phi, \Theta) = \sum_{t,w} \beta_w \ln p(w|t) + \sum_{d,t} \alpha_t \ln p(t|d).$$

Для тематической модели важными мерами качества являются разреженность и интерпретируемость тем. Слова, принадлежащие теме, должны описывать одну предметную область. Такую тему $t \in S$ назовем предметной. Фоновая тема $t \in B$ в основном содержит слова общей лексики, которые не могут образовать предметную область. Тогда формализуем множество всех тем: $T = S \sqcup B$.

Подход ARTM [6] позволяет комбинировать регуляризаторы разреженности и интерпретируемости тем. Рассмотрим основные регуляризаторы.

Будем использовать *дивергенцию Кульбака-Лейблера* между двумя дискретными распределениями $P = (p_i)_{i=1}^n$ и $Q = (q_i)_{i=1}^n$:

$$KL(P||Q) \equiv KL_i(p_i||q_i) = \sum_{i=1}^n p_i \ln \frac{p_i}{q_i}.$$

Основные свойства KL-дивергенции:

1. $KL(P||Q) \geqslant 0$;
2. $KL(P||Q) = 0 \Longleftrightarrow P = Q$;
3. Минимизация KL эквивалентна максимизации правдоподобия вероятностной модели $q_i(\alpha)$ :
$$KL(P||Q(\alpha)) = \sum_{i=1}^n p_i \ln \frac{p_i}{q_i(\alpha)} \to \min_\alpha \iff \sum_{i=1}^n p_i \ln q_i(\alpha) \to \max_\alpha;$$
4. Если $KL(P||Q) < KL(P||Q)$, то $P$ сильнее вложено в $Q$, чем $Q$ в $P$.

Выше рассмотрен метод LDA, где сглаженные частотные оценки условных вероятностей получаются через априорные распределения Дирихле и байесовский вывод. Гипотезу сглаженности можно представить так же в виде регуляризатора.

*Гипотеза сглаженности* предполагает близость по дивергенции Кульбака-Лейблера распределений $p(w|t)$ и $p(t|d)$ к заданным распределениям $\beta_{wt}$ и $\alpha_{td}$. Имеем два регуляризатора:

$$\sum_{t \in T} KL_w(\beta_{wt}||\varphi_{wt}) \to \min_\Phi; \quad (4) \qquad\qquad \sum_{d \in D} KL_t(\alpha_{td}||\theta_{td}) \to \min_\Theta; \quad (5)$$

Максимизируем сумму регуляризаторов (4) и (5):

$$R(\Phi, \Theta) = \beta_0 \sum_{t \in T} \sum_{w \in W} \beta_{wt} \ln \varphi_{wt} + \alpha_0 \sum_{d \in D} \sum_{t \in T} \alpha_{td} \ln \theta_{td} \to \max,$$

где $\alpha_0, \beta_0$ – коэффициенты регуляризации.

Данное ограничение используется для выделения фоновых тем, представляющих общую лексику языка в конкретной коллекции и фоновых слов в каждом документе.

*Гипотеза разреженности* предполагает, что $p(w|t)$ и $p(t|d)$ – сильно разреженные распределения, так как в общем случае документ $d \in D$ относится к небольшому количеству тем $t \in T$, то большая часть условных вероятностей $p(w|t)$ и $p(t|d)$ должна обращаться в нуль. Равномерное распределение, в отличие от сильно разреженных распределений, является менее разреженным и обладает максимальной энтропией.

Максимизируем функцию расстояния между $p(w|t)$, $p(t|d)$ и равномерным распределением:

$$R(\Phi, \Theta) = -\beta \sum_{t \in T} \sum_{w \in W} \ln \varphi_{wt} - \alpha \sum_{d \in D} \sum_{t \in T} \ln \theta_{td} \to \max,$$

где $\alpha, \beta$ – коэффициенты регуляризации.

*Гипотеза декоррелированности тем* базируется на предположении, что при повышении различности тем повышается интерпретируемость модели. Это достигается за счет минимизации ковариации между столбцами $p(w|t)_{w \in W}$:

$$R(\Phi) = -\gamma \sum_{t \in T} \sum_{s \in T \setminus t} \sum_{w \in W} \varphi_{wt} \varphi_{ws} \to \max,$$

где $\gamma$ – коэффициент регуляризации.

Данный регуляризатор помогает избавляться в тематической модели от похожих тем. За счет этого документы коллекции имеют более адекватную кластеризацию.

*Гипотеза частичного обучения* предполагает, что для некоторых слов и документов темы могут быть указаны экспертами в явном виде:

1. Документы $d \in D_0$ принадлежат темам $T_d \in T$;
2. Термины $W_t \in W$ принадлежат темам $t \in T_0$;
3. $\alpha_{td}$ и $\beta_{wt}$ – распределения, равномерные на $T_d$ и $W_t$, тогда

$$R(\Phi, \Theta) = \beta_0 \sum_{t \in T_0} \sum_{w \in W_t} \beta_{wt} \ln \varphi_{wt} + \alpha_0 \sum_{d \in D_0} \sum_{t \in T_d} \alpha_{td} \ln \theta_{td} \to \max.$$

Использование экспертных знаний о релевантности слов для конкретной темы или значимости тем документа, повышает устойчивость и интерпретируемость тем.

# 3 Методы определения семантической близости слов

Для классификации текстовых сообщений требуется построение семантических признаков. Существует множество моделей дистрибутивной семантики, различающихся по типу контекста, оценке частоты встречаемости термина в контексте, мере расстояний между векторами, методом уменьшения размерности. Современные методы оценивания семантической близости фраз описаны в работах [9]-[12]. Особенно привлекает внимание работа Миклова в 2013 году [7]. После ее выхода, и представления компанией Google нейросетевой модели Word2Vec, позволяющей эффективно решать задачи определения семантической близости слов и выявления аналогий в парах слов, всего за несколько лет появилось множество современных интерпретирующих Word2Vec моделей: Adaptive skip-gram [9], FastText [10], GloVe [11], StarSpace [12] и др.

*One-hot encoding* – метод, который преобразует слово в вектор длины словаря, с ненулевым значением на позиции, соответствующей номеру данного слова в словаре.

*Векторное представление слова* (word embedding) – сопоставление текстовому слову некоторого вещественного вектора фиксированной невысокой размерности. В данной главе работы будут рассмотрены нейросетевые модели определения близости слов, которые представляют слова в форме эмбеддингов.

Одна из самых популярных моделей *Word2Vec* представляется двумя архитектурами: Skip-gram и Continuous Bag of Words (CBOW, непрерывный мешок слов) [7].

Модель CBOW (рисунок 1.a) принимает на вход последовательно 2k+1 наборов one-hot представлений слов из текста, где центральное слово требует прогнозирования контекста, а длина локальных контекстов с каждой стороны k задана заранее. Таким образом, слово предсказывается по контексту:

$$p(v|w) = \frac{exp(B_v^T(\sum_j A_{w_j}))}{\sum_s exp(B_s^T(\sum_j A_{w_j}))},$$

где $B_v$ – вектор предсказываемого термина,
$\sum_j A_{w_j}$ – вектор контекста.
Входные и выходные представления обучаются в ходе оптимизации целевой функции:

$$F(A, B) = \sum_t \log p(w_t|w_{t-1}, ..., w_{t-K}).$$

Архитектура модели Skip-gram (рисунок 1.b) похожа на CBOW, но вместо прогнозирования некоторого слова по его контексту, она пытается максимизировать классификацию слова на основе слов в локальной окрестности заданного окна. Данная модель независимо прогнозирует контекстные слова для заданного слова:

$$p(v|w) = \frac{exp(B_v^T A_w))}{\sum_s exp(B_s^T A_w))},$$

где $B_v$ – вектор предсказываемых терминов,

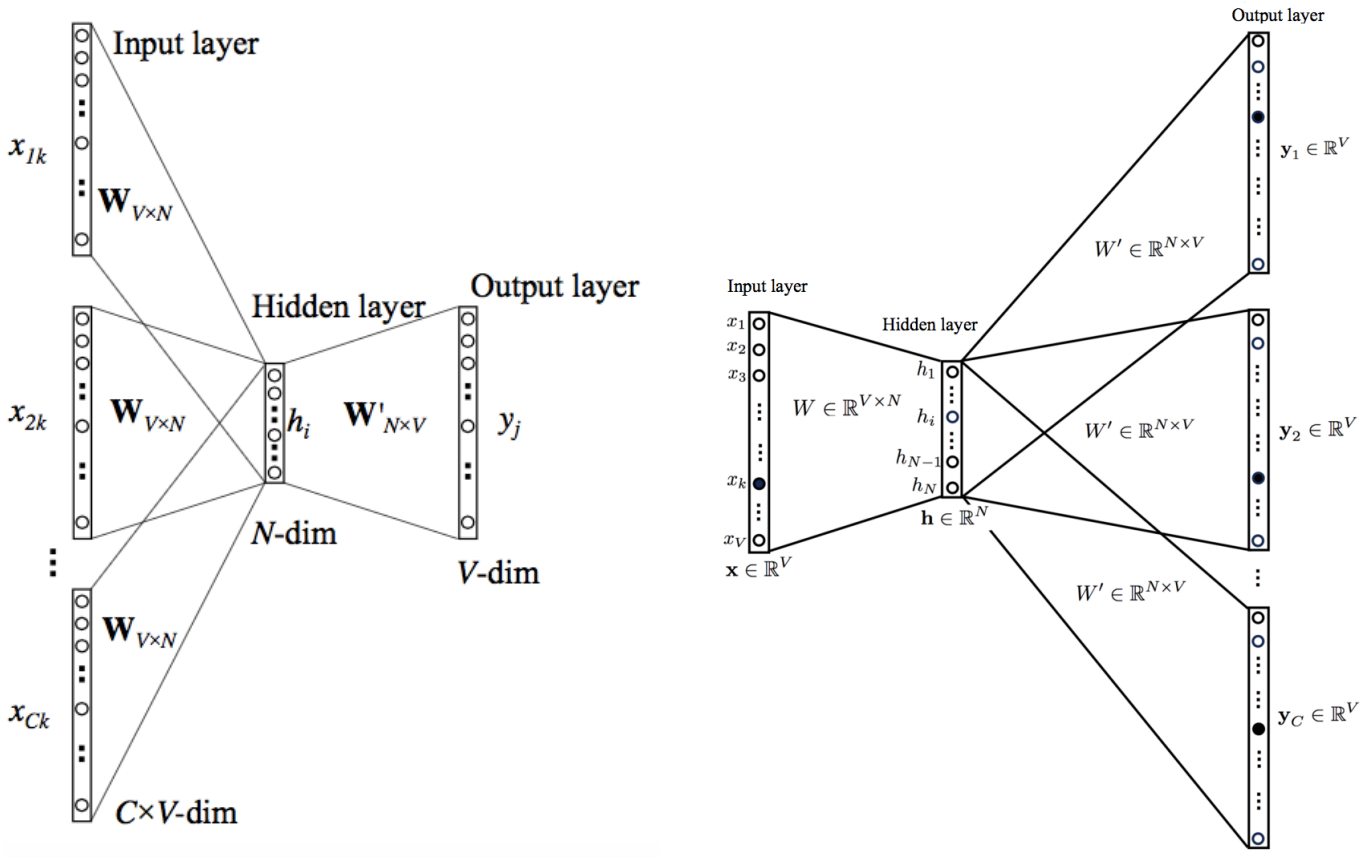


Рис. 1: Модель Word2Vec

1.a CBOW 1.b Skip-gram

$A_w$ – вектор слова, подающийся на скрытый слой.

Входные и выходные представления обучаются в ходе оптимизации целевой функции:

$$F(A, B) = \sum_t \sum_{j \in c(t)} \log p(w_j | w_t).$$

Оба алгоритма обучения на первом этапе строят словарь терминов по коллекции. На втором этапе для каждого термина вычисляется векторное представление $B_v$.

Таким образом, модель совсем не учитывает различные смыслы терминов, в которых они употребляются в различных контекстах и вычисляет вектор для смешанного смысла, смещенного в сторону наиболее частого.

Вторая модель *Adaptive skip-gram* [9] – непараметрическое расширение Skip-gram представляет решение проблемы многозначности слов с помощью введения скрытой переменной $z$:

$$p(y, z|x) = \prod_{i=1}^{N} p(z_i) \prod_{j \in c(t)} p(y_{ij} | z_i, x_i),$$

где $x_i$ – $i$ -ое слово в тексте длины N,

$y_i$ – контекст слова $x_i$,

$z_i$ – номер смысла слова $x_i$.

Адаптирующая модель прогнозирует контекстные слова так же, как Skip-gram, проделывая данную процедуру для всех возможных смыслов слова.

Обучение векторов оптимизируется за счет оценки максимального правдоподобия с шагом по градиенту функции:

$$F_i(A, B) = \sum_j \sum_k [z_i = k] \log p(y_{ij}|k, x_i).$$

Метод предполагает, что слово потенциально имеет неограниченное число смыслов. Для автоматического определения числа смыслов слов применяется процесс Дирихле.

Преимущества модели:

1. Автоматическое определение числа смыслов для всех слов;
2. Хорошая интерпретируемость слов для всех найденных смыслов.

Недостатки модели:

1. Низкая производительность за счет хранения частот слов и их контекстов;
2. Невысокое качество оценок семантической близости документов, при использовании в модели всех контекстов слов с кластеризацией по частоте употребления.

Подход *FastText*, основанный также на модели Skip-gram, придает значение внутренней структуре слова [10].

Каждое слово в словаре представляется в виде мешка символьных $n$-грамм. Само слово включается в набор его $n$-грамм для полного представления. Например, представление для слова *where*, где $n = 3$: <wh, whe, her, ere, re>, и специальное представление слова <where>. На практике для представления слова перебираются все $n$ -граммы, $3 \leqslant n \leqslant 6$.

Для словаря $n$-грамм размера $G$, каждое слово $w$ будет иметь множество $n$ -грамм $\Upsilon_w \subset 1, ..., G$. Слово представляется суммой всех векторных представлений его $n$ -грамм. Скоринг функция сравнения слов:

$$s(w, c) = \sum_{g \in \Upsilon_w} z_g^T v_c,$$

где $z_g$ – векторное представление $n$ -граммы $g$ слова $w$,
$v_c$ – векторное представление слова $c$.

Преимущество модели – поиск векторных представлений для редких слов. Поскольку редкие слова все еще можно разбить на $n$ -граммы символов, они могут разделить эти $n$ -граммы с другими словами.

Недостатки модели:

1. Метод неэффективен в задачах, где поиск редких слов не влияет на качество модели;
2. Интерпретируемость обычных слов сильно уступает моделям Word2Vec и Ada-gram.

Метод *StarSpace* позволяет оценивать близость объектов различных типов, сравнивая их в одном векторном пространстве.

Каждый объект $d \in D$ имеет характеристики $w \in W$ (bag-of features), тогда вектор объекта:

$$v_d = \sum_{w \in d} v_w,$$

где $D$ и $W$ – коллекции объектов и характеристик, $v_w$ – вектор характеристики.

Тренировка алгоритма заключает в себя минимизацию потерь:

$$\sum_{\substack{(d,d^+) \in E^+ \\ d^- \in E^-}} L(sim(d, d^+), sim(d, d_1^-), ..., sim(d, d_k^-)),$$

где $E^+$ – набор близких пар объектов и $E^-$ – набор сильно различающихся пар объектов специфичных для каждой задачи;
$sim(a, b)$ – функция близости объектов $a$ и $b$ (обычно косинусное расстояние).

Метод показывает результаты чуть лучше *FastText*, но есть и недостатки:

1. Долгое время обучения (в 2-3 раза дольше *FastText*);
2. Интерпретируемость обычных слов уступает моделям Word2Vec и Ada-gram.

# 4 Методы предварительной обработки текста

Задачи краткого изложения текста и определения их схожести относятся к классу задач доступных для решения только человеку. Рассмотрим техники, которые использует человек для понимания и сравнения текста, с целью последующего переноса этого понимания на методы автоматической обработки текста.

## 4.1 Техника перефразирования

Техника перефразирования пришла из психологии, и заключается она в понимании собеседника посредством перефразирования его речи более простой, констатирующей формой без потери смысла.

Основные моменты техники:

- Удаление эмоционально окрашенной лексики;
- Исключение речевого мусора;
- Сокращение объема текста.

Техника перефразирования легла в основу техники упрощения текста *Text Simplification* [13].

**Пример 1.**

*Исходный текст.* «Этот фильм оказался фантастически красивым и интересным. Он произвел на меня неизгладимое впечатление. Думаю, что по приезду домой, я начну читать книгу, по мотивам которой сняли эту экранизацию.»

*Парафраз.* «Фильм интересный. Он сподвигнул меня к прочтению книги.»

В данном парафразе первое предложение является сокращенной версией предложения в исходном тексте. Второе и третье предложение объединены, и не содержат слов из исходной версии.

## 4.2 Предобработка текста

Современные европейские языки содержат в своих словарях порядка 340000 слов[2]. При этом словарь, который люди задействуют в речи насчитывает около 1000000 слов, так как в нем дополнительно содержится специальная речь, например, названия продуктовых марок. Зачастую новые слова образуются на основе существующих. Также одно слово имеет несколько форм за счет морфологических правил его употребления. Человек легко узнает значение слова по любой его форме. Для сравнения системой двух текстов требуется их предварительная обработка.

[2]https://rg.ru/2014/10/10/slovari.htmli

### 4.2.1 Токенезация

*Токенезация* – метод предобработки текста, заключающийся в разбиении текста на слова, токены.

- Текст переводится в нижний регистр;
- Исключаются символы, отличные от букв, цифр и пробела.

**Пример 2.** Токенезируем исходный тест из примера в пункте 4.1:
«этот», «фильм», «оказался», «фантастически», «красивым», «и», «интересным», «он», «произвел», «на», «меня», «неизгладимое», «думаю», «что», «по», «приезду», «домой», «я», «начну», «читать», «книгу», «по», «мотивам», «которой», «сняли», «эту», «экранизацию».
Токенезируем парафраз текста из примера в пункте 4.1:
«фильм», «интересный», «он», «сподвигнул», «меня», «к», «прочтению», «книги».

Из примера 2 видно, что обработка документов с помощью токенезации необходима, но её недостаточно для сравнения двух текстов на предмет парафраза.

Техника несложная для алгоритмической реализации, подходит для текстов на любом языке. Исключением являются специальные тексты, например, медицинские, где символы, отличные от букв и цифр также имеют смысл.

### 4.2.2 Лемматизация

Лемма – нормальная форма слова.

Для существительных лемма:

- именительный падеж;
- единственное число.

Для прилагательных лемма:

- именительный падеж;
- единственное число;
- мужской род.

Для глагольных форм лемма:

- инфинитив;
- несовершенный вид.

Лемматизация слов – приведение словоформы к лемме. Этот метод предобработки тестов применяется для снижения размерности словаря и увеличения объёма статистики по частотам отдельных слов.

### 4.2.3 Стемминг

Базовые морфологические признаки для слов можно отобразить в нескольких десятках правилах:

- множественное число существительного;
- форма глагола;
- степени прилагательного.

Применяя данные правила в обратную сторону находится основа для слова. *Стемминг* – нахождение основы слова. Простыми словами: исходное слово усекается, если его окончание попадает под свод правил.

Стемминг отличается самой простой реализацией в отличии от предыдущих методов, не уступает по качеству лемматизации, при этом дает низкие результаты для флективных языков, таких как русский.

## 4.3 Обработка словосочетаний

Использование информации о совместной встречаемости слов даёт множество преимуществ для автоматической обработки текста.

*Коллокация* – устойчивое словосочетание, которым считается последовательность слов, имеющих смысл преимущественно находясь рядом друг с другом. Понятие «рядом» подбирается под конкретную задачу и корпус документов, обычно рассматриваются слова в диапазоне одного предложения, при условии нахождения в окне 3-6 слов.

Поиск коллокаций можно производить с помощью несложных действий над коллекцией:

1. Подсчет количества употребления слов и словосочетаний в коллекции;
2. Рассчет значимости фразы по определенной формуле.

Например, метод последовательной фильтрации основан на расчете частоты совместной встречаемости (Pointwise Mutual Information, PMI) [14] и формуле:

$$PMI(w1, w2) = log\frac{p(w1, w2)}{p(w1)p(w2)},$$

где $p(w1)$ и $p(w2)$ — априорные вероятности появления терминов $w1$ и $w2$ соответственно в коллекции, $p(w1, w2)$ — вероятность совместной встречаемости терминов $w1$ и $w2$.

Эффективное извлечение повторяющихся последовательностей слов достигается за счет оценивания качества фраз с помощью метода расчета средней обратной чистоты документа (Inverse Document Frequency, IDF). IDF вычисляется по словам.

$$IDF(w) = log\frac{|S|}{|d \in [D] : w \in C|},$$

где $S$ – размер корпуса, $C$ – последовательность слов документа d корпуса D.

При оценивании словосочетания на предмет коллокации, можно учитывать информацию о частоте слова в документах коллекции. Если слово нечастое для коллекции, то объединение его с другим словом не будет эффективным термином.

Метод извлечения устойчивых словосочетаний TopMine [17] на первом этапе строит словарь частот слов и словосочетаний в корпусе. На втором этапе, используя информацию о частоте слов и словосочетаний, алгоритм преобразует коллекцию текстов в порядке значимости в мешок фраз.

Алгоритм TopMine объединяет слова в термин, если они употребляются вместе чаще, чем это могло произойти чисто случайно.

Эффективность алгоритма обеспечивают два правила:

1. Любая фраза, содержащая нечастую фразу, заведомо является не частой;
2. Любой текст, который не содержит частых фраз длины n, не может содержать частых фраз длины $>$ n.

Метод C-value [18], направленный на улучшение извлечения многословных терминов в целом, основан на гипотезе, что статистическая информация, без какой-либо лингвистической фильтрации, не является достаточной для получения полезных результатов при выделении коллокаций. Алгоритм обрабатывает только словосочетания, состоящие из прилагательных и существительных, размеченных предварительно. Для этих словосочетаний вычисляется C-значение:

$$Cvalue(a) = \begin{cases} log_2|a|f(a), & \text{если a не вложено в другое словосочетание;} \\ log_2|a|(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)), & \text{иначе;} \end{cases}$$

где $a$ – проверяемое словосочетание, $|a|$ его длина в словах, $f(a)$ – частота словосочетания в корпусе, $T_a$ – набор терминов, содержащих $a$, $P(T_a)$ – число этих терминов.

В результате получается список терминов-кандидатов, ранжированных по C-value как их вероятность быть терминами. Построенная из статистических характеристик контекстных слов (прилагательных, существительных и глаголов) для каждого термина-кандидата весовая функция влияет на ранжирование терминов-кандидатов. В итоге каждому словосочетанию присваиваются веса контекста, который будет суммой весов его контекстных слов:

$$Weigth(a) = \sum_{b \in C} weigth(b) + 1,$$

где $b$ – слово из контекста $C$ для термина $a$,

$$weigth(b) = 0.5 \left( \frac{t(b)}{n} + \frac{ft(b)}{f(b)} \right),$$

где $n$ – число терминов, которые содержат $b$, $t(b)$ – количество терминов-кандидатов, с которыми появляется слово $b$, $ft(b)$ – полная частота появления $b$ с термином кандидатом, $f(b)$ – общая частота $b$ в корпусе.

Итоговая оценка словосочетания для определения его устойчивости:

$$NCvalue(a) = \frac{1}{\log N} Cvalue(a) Weigth(a),$$

где $N$ – количества слов корпуса.

Метод C-value использует больше статистической информации, чем чистая частота возникновения устойчивых словосочетаний, за счет чего улучшает точность извлеченных вложенных терминов, и использует информацию из контекста терминов-кандидатов, что улучшает их распределение в извлеченном списке, реальные термины находятся ближе к вершине, в то время как «ложные» словосочетания концентрируются ближе к нижней части списка.

# 5 Тематические векторные представления слов

В следующих разделах описывается метод оценивания семантической близости между фразами, состоящий из трех этапов: (1) – тематизация текстовых сообщений коллекции, (2) – обучение нейросетевых моделей слов для каждой темы, (3) – применение моделей тематических векторных представлений слов для текстов общих сегментов.

## 5.1 Тематизация коллекции

Модели, основанные на дистрибутивной семантике и векторном представлении слов, рассмотренные в предыдущих главах, имеют некачественное представление о многозначности слов в коллекции.

В этих моделях темы формируются, основываясь на контексте, который так же неоднозначен, как и само слово. Таким образом, контекст для слова формируется некорректно, так как изначально любое слово представляется в словаре коллекции в единственном смешанном смысле для всех контекстов.

**Гипотеза:** *Предметное слово однозначно для одной темы.*

Исходя из гипотезы, коллекция разделяется на монотематические сегменты по предложениям. Для каждого предложения определяется его главная тема исходя из тематики большинства слов и терминологических словосочетаний в предложении. Если предложение не может быть однозначно отнесено к единственной теме, то оно включается в сегменты нескольких тем. Пороговые коэффициенты принадлежности подбираются для каждой темы эмпирически.

Зачастую, в качестве начальных распределений $p_0(w|t)$ и $p_0(t|d)$ берут случайные распределения. Также при инициализации можно использовать стратегии частичного обучения [16], задавая равномерные распределения на определенных подмножествах термов для тем и равномерные распределения на определенных подмножествах тем для документов. При инициализации эти распределения смешиваются со случайными, затем в процессе итераций ЕМ-алгоритма они используются в качестве регуляризаторов сглаживания. Данный метод позволяет фиксировать интерпретируемость тем.

В общем случае подход частичного обучения (semi-supervised learning) для тематического моделирования представляется так:

1. Для построенных предметных тем асессоры исключают термины, которые имеют низкую вероятность темы;
2. После нечёткой кластеризации терминов асессоры настраивают центры кластеров;
3. Эксперт определяет однозначные термины, из которых формируется семантическое ядро темы.

В работе применяется инициализация с частичным обучением. Существует набор общепринятых областей (тем), которые имеют фиксированный словарь. На вход модели

подается описание тем. Для неклассифицированных документов строится тематическая модель, где в качестве начального распределения $p_0(w|t)$ используется равномерное распределение на $W_t$:

$$\beta_{wt} = \frac{1}{|W_t|}[w \in W_t].$$

## 5.2 Предобработка текста

Для контроля качества тем предусматривается возможность формирования наборов тематических слов и словосочетаний, а также отрицательных значений слов в каждой конкретной теме. Векторные представления строятся не только для слов, но и для словосочетаний. Если пара слов встречается вместе гораздо чаще, чем это могло бы происходить чисто случайно, то данная пара слов образует коллокацию, и она добавляется к словарю как один терм. Вероятность совместного появления слов оценивается с помощью статистического критерия significance score или PMI (поточечной взаимной информации), для формирования всех коллокаций используется алгоритм TopMine [17]. Также предлагается использовать собственный метод, основанный на простой метрике:

$$significance = \frac{N(w1, w2)}{N(w1)} + \frac{N(w1, w2)}{N(w2)}. \tag{6}$$

Алгоритм за две итерации объединяет фразы, для которых significance больше заданного порога. Данный метод помогает выделять не только устойчивые словосочетания, но и обогащать словарь отрицательными смыслами.

Например, имея в коллекции слова «not», «beautiful» и «ugly». При построении векторных представлений слов, мы получим картину: «beautiful» близко к «ugly». В действительности, эти слова являются антонимами. Они используются с похожими сущностями, но в разных ситуациях. Имея в словаре дополнительный термин «not_beautiful», слова в векторной форме должны располагаться относительно друг друга более интерпретируемо: «not_beautiful» близко к «ugly».

## 5.3 Обучение векторов

После тематизации коллекции и обогащения каждой подколлекции коллокациями и отрицательными смыслами слов, получаем $K$ независимых коллекций.

Каждая коллекция представляет собой монотематичный корпус коротких текстов, в котором каждое слово в словаре представляется в одном единственном смысле.

Для каждой подколлекции обучается Skip-gram модель. На вход модель принимает one-hot представление слова, для которого предсказывается контекст с целью обучения модели. В итоге получается $K$ моделей Skip-gram, а любое слово $w$ раскладывается для каждой из тем может быть представлено в виде вектора $w_k$.

## 5.4 Выделение парафраз

Обученная модель предназначена для определения семантической близости текстов. Принимая на вход два коротких текста, их близость предлагается измерять, основываясь на векторном представлении слов каждой общей для текстов темы. Если тексты не имеют общих тем, их близость приравнивается к нулю.

Векторное представление документа $d$ по теме $k$ можно представить, как сумму представлений всех его слов:

$$V_k(d) = \sum_{w \in d} w_k.$$

В качестве признака для сравнения текстов используется косинусное расстояние между векторами документов:

$$s(\overrightarrow{V_k}(d_1), \overrightarrow{V_k}(d_2)) = 1 - \frac{\langle \overrightarrow{V_k}(d_1), \overrightarrow{V_k}(d_2) \rangle}{||\overrightarrow{V_k}(d_1)|| \cdot ||\overrightarrow{V_k}(d_2)||}. \quad (7)$$

Таким образом, с помощью данной модели есть возможность сравнивать предложения разной длины.

Сравнение текстов делается по каждой из тем отдельно. Рассчитанные параметры подаются в модель бинарной классификации для определения, обладают ли тексты идентичным смыслом.

# 6 Определение качества

Оценка качества в задачах *NLP* – самостоятельная задача. На интуитивном уровне успешность алгоритма можно определять, как отношение числа верных ответов к величине выборки. Такая мера качества называется *Accuracy*.

Данный метод оценки алгоритма не является качественным для несбалансированных выборок, когда число представителей одного класса много больше числа представителей другого.

Работу системы можно представить четырьмя параметрами:

- TP (true positive) – количество верных срабатываний;
- TN (true negative) – количество верных пропусков;
- FP (false positive) – количество ложных срабатываний;
- FN (false negative) – количество ложных пропусков.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

Более релевантной оценкой являются точность и полнота.

Точность (precision) определяет сколько представителей класса действительно таковыми являются:

$$P = \frac{TP}{TP + FP}.$$

Полнота (recall) определяет сколько представителей класса были определены:

$$R = \frac{TP}{TP + FN}.$$

Оценивать алгоритм двумя параметрами не всегда эффективно. Большинство задач требуют высокой точности и полноты. Для усреднения параметров используют гармоническое среднее (F-мера):

$$F = \frac{(Q+1)^2 PR}{Q^2 P + R}.$$

Определение важности точности и полноты зависит от постановки конкретной задачи. В случае одинаковой для оценки значимости точности и полноты, коэффициент Q принимается единицей, и формула принимает вид:

$$F_1 = \frac{2PR}{P + R}.$$

Для построения диалоговых систем, в частности для задачи определения парафраз, важно получать корректный результат решения системы. Пользователь может уточнить запрос в любой момент, для получения более точного результата, при этом количество ложных срабатываний критично. Исходя из этого, в данной работе качество моделей будет сравниваться с помощью их оценки по мере precision.

# 7 Эксперимент

В следующих разделах описаны эксперименты для проверки предложенных подходов каждого этапа работы, описанных в главе 5, направленных на снятие многозначности и повышение интерпретируемости векторных представлений слов для эффективности выделения парафраз в коротких текстах.

Вычислительные эксперименты проводятся на размеченной коллекции коротких текстов Quora Question Pairs[3], в которой выделены пары текстов, являющиеся и не являющиеся парафразами.

Корпус представляет 342572 пар пользовательских запросов на английском языке. Тексты коллекции содержат грамматические и лексические ошибки, описки, слова на других языках, диалектные сокращения. Слова некоторых запросов намеренно заменены составителями случайным образом из других текстов.

## 7.1 Тематизация документов коллекции

Первый этап предложенного метода состоит в построении монотематических подколлекций. Для этого тексты коллекции полноценно распределяются по темам в зависимости от содержания. Так как текст – одно предложение, он содержит мало информации о связи слов внутри. Поэтому для тематизации эффективно использование начальных приближений. В данной работе проводятся эксперименты по подбору количества тем. Результаты экспериментов приводятся для количества тем 3 (эксперимент 1) и 27 (эксперимент 2). Каждая тема изначально содержит несколько слов (35 – 1250) в словаре. На рисунке ниже представлен фрагмент словарей (Рис. 2), они не были подобраны специально под корпус текстов, и являются списками слов для обучения английскому языку[4].

Риск не раскрытия одной из тем снимается последним этапом модели, классификатором, который при данной ситуации не будет учитывать признаки близости в данной теме, в связи с их неэффективностью.

До распределения текстов по подколлекциям, из словарей исключаются повторяющиеся слова. Затем каждый словарь преобразовывается в векторный вид (one-hot). Построение матрицы документ-тема основывается на факте пересечения некоторого слова из предложения и словаря. При пересечении со словарем, предложение полностью помещается в коллекцию с номером словаря.

По факту тематизации одно предложение исходной коллекции может присутствовать одновременно в нескольких подколлекциях, это указывает на полисемию данного текста.

Тексты, которые не покрываются словарями для тем, считаются фоновыми и составляют соответственно четвертую и двадцать восьмую подколлекции в экспериментах 1 и 2 соответственно.

Факт того, что предложения размечены асессором в корпусе как парафраз, не

[3]https://www.kaggle.com/c/quora-question-pairs/data
[4]https://www.memorysecrets.ru/english-words.html

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **4** | baths | bench | cathedral | corner | crossroads | crosswalk | cul | sac | firehouse | fountain | ... |
| **5** | fridge | wardrobe | yard | address | anteroom | electricity | storey | washing | heating | boulevard | ... |
| **6** | acorn | equinox | chestnuts | cloud | early | falling | feast | gloomy | halloween | harvest | ... |
| **7** | advent | angel | calendar | cane | chimney | christ | christian | christianity | bauble | sauce | ... |
| **8** | crossbow | aristocracy | mangonel | baron | baroness | baronet | bastion | noble | battle | axe | ... |
| **9** | acrobat | applaud | attractions | trampoline | balloons | giant | gymnast | swallower | petting | ride | ... |
| **10** | academic | art | attendants | audio | auditorium | blackboard | bookshelf | bulletin | campus | chalk | ... |

Рис. 2: Начальное приближение тем.

использовался при тематизации коллекции.

В первом эксперименте предложения коллекции были нечетко классифицированы на 3 подколлекции по близости (one-hot) вектора словаря темы и (one-hot) вектора предложения. В результате тематизации 28% текстов так и не были распределены между темами, то есть оказались подходящими для всех тем, а для 10% текстов не одна из тем не оказалась подходящей.

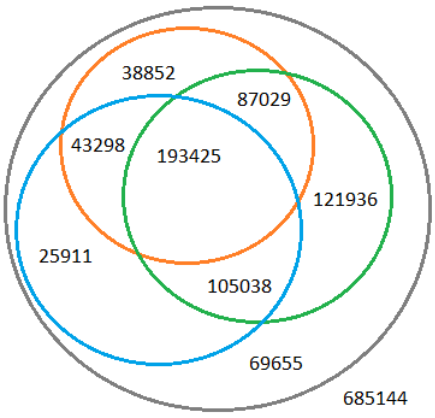


Рис. 3: Распределение текстов по трём темам.

При данном распределении документов по темам модель определяет парафразы с точностью 68.4%. Эксперимент показал, что данного разделения коллекции недостаточно для снятия многозначности слов и фраз.

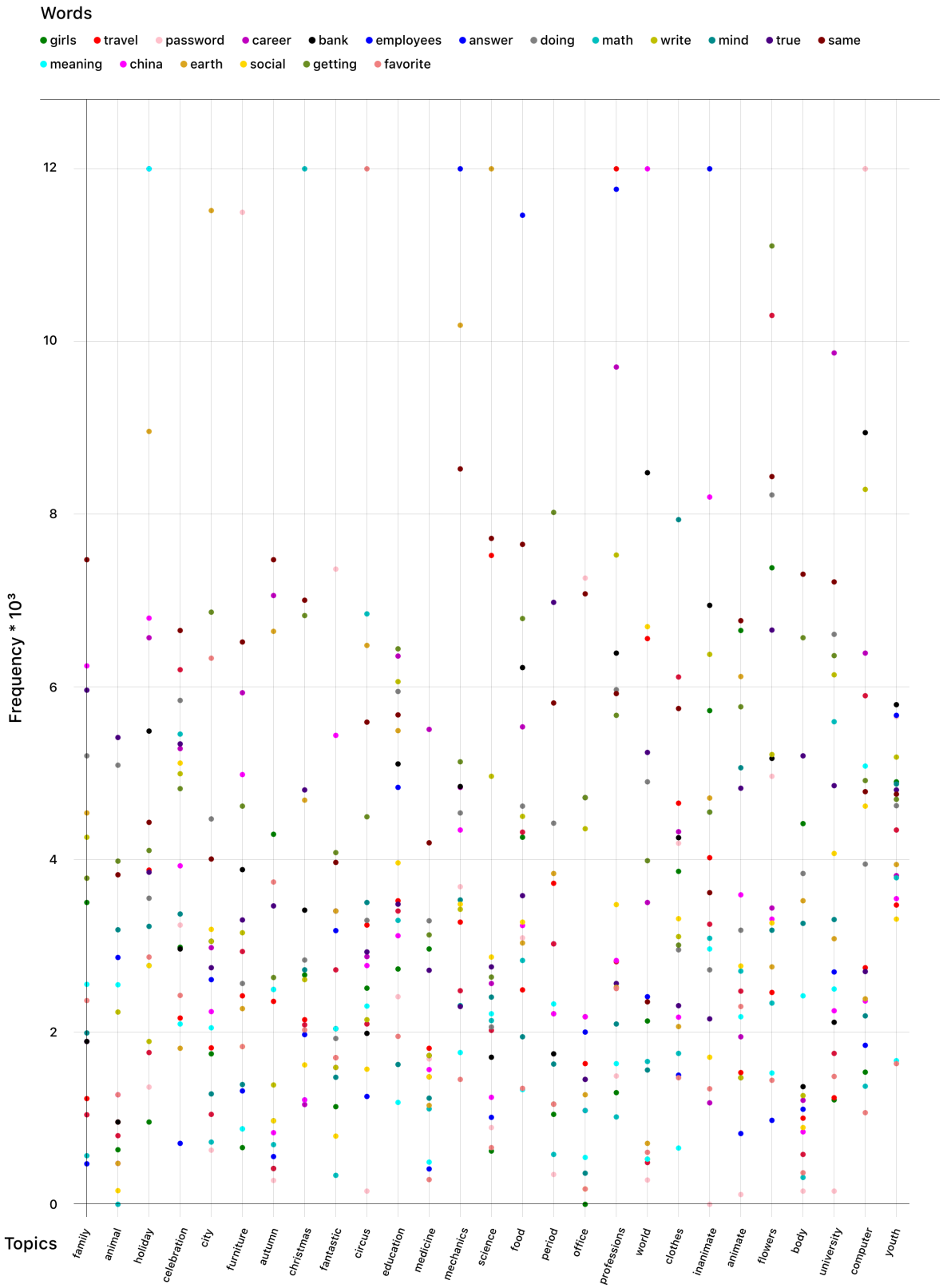


Рис. 4: Распределение слов по темам.

Второй эксперимент заключался в разбиении коллекции на 28 монотематических подколлекций. На вход классификатору были поданы все предложения корпуса в количестве

685144 штук. Результатом эксперимента являются независимые коллекции предложений, общий объем всех коллекций составляет 1724960 предложений. Таким образом, в среднем предложение относится к 5 корпусам для обучения векторных представлений слов. Данное разделение коллекции гарантирует семантическую оценку текста по отношению ко всем смыслам, которые оно содержит. При этом данная оценка независима от процентного содержания темы в самом документе.

Для слов, значимых относительно всего корпуса, то есть, имеющих среднюю частоту употребления в коллекции, подсчитана частота их употребления в каждой теме и отображена на рисунке 4. Можно сделать вывод, что слова имеют различную частоту употребления в разных темах (значениях).

## 7.2 Построение векторных представлений слов

Второй этап обучения модели подразумевает обучение независимых Word2Vec моделей, используя монотематическе коллекции, полученные в ходе эксперимента, описанного в пункте 7.1.

Каждая модель Skip-gram была обучена в однородных условиях. Для каждого слова каждой коллекции контекст предсказывался в окне 3 слов и записывался в вектор размерности 400.

```
Lead_Model.wv.most_similar(positive=['oil'])
```

```
[('olive', 0.7169071435928345),
 ('vegetable', 0.6990725994110107),
 ('coconut', 0.6920554637908936),
 ('gas', 0.6907387971878052),
 ('crude', 0.6883388757705688),
 ('cannabis', 0.6863281726837158),
 ('aloe', 0.665648341178894),
 ('plastic', 0.6629904508590698),
 ('juice', 0.6611119508743286),
 ('castor', 0.647707462310791)]
```

Рис. 5a: Контекст к oil по всему корпусу модели Word2Vec.

```
Print[Food]
```

```
[('juice', 0.7636476755142212),
 ('olive', 0.7599772214889526),
 ('cream', 0.7412714958190918),
 ('onion', 0.7316570281982422),
 ('cheese', 0.726305365562439),
 ('cooking', 0.723984956741333),
 ('coconut', 0.7210065126419067),
 ('baking', 0.719397246837616),
 ('vera', 0.719170868396759),
 ('lemon', 0.7158592939376831)]
```

Рис. 5b: Контекст к oil в теме Еда.

```
Print[Mechanics]
```

```
[('diesel', 0.8689920902252197),
 ('unreal', 0.8637183904647827),
 ('internal', 0.8636921644210815),
 ('petrol', 0.8613460659980774),
 ('2stroke', 0.8516882061958313),
 ('ic', 0.8502888679504395),
 ('steam', 0.8498152494430542),
 ('turbo', 0.8493829965591431),
 ('gas', 0.8473856449127197),
 ('fuel', 0.842423141002655)]
```

Рис. 5c: Контекст к oil в теме Механика.

```
fastText_model.most_similar('oil')
```

```
[('oi', 0.9468726515769958),
 ('foil', 0.8581773638725281),
 ('broil', 0.8306301832199097),
 ('boil', 0.8268641829490662),
 ('oitnb', 0.805431604385376),
 ('pedoil', 0.7904340028762817),
 ('turmoil', 0.7816019058227539),
 ('soil', 0.781015932559967),
 ('parboil', 0.7600042819976807),
 ('glucocil', 0.759742021560669)]
```

Рис. 6: Контекст к oil по всему корпусу модели FastText.

Разделение коллекции документов на монотематические сегменты и построение векторных представлений слов с помощью Word2Vec по этим сегментам обладает преимуществами. На рисунке 5 представлены контексты к слову «oil», полученные при обучении Skip-gram модели на всей коллекции и монотематических подколлекциях в частности. Можно убедиться, что интерпретируемость модели повышается за счет снятия многозначности данного слова.

## 7.3 Сравнение моделей

Разметка корпуса позволяет сравнивать различные модели семантической близости по критерию точности (*Precision*).

*Precision* – отношение верных срабатываний алгоритма ко всем его срабатываниям.

За эталон для сравнения была взята модель Word2Vec, обученная на коллекции пар вопросов. При классификации пар коротких текстов не использовались предобученные векторные представления слов. Близость текстов рассчитана как косинусное расстояние между векторными представлениями обоих предложений. Классификация предложений производилась при различных порогах близости $k$, подобранных эмпирически в ходе эксперимента. Лучший результат обученной на коллекции Skip-gram модели – около 68.7% точности.

Модель FastText аналогично модели Word2Vec, была обучена на всей коллекции. Для каждого слова из словаря контекст предсказывался на основании самого слова и его $n$-gram, где $n \in [3, 6]$. Близость текстов и выделение парафраз было произведено аналогично эксперименту с моделью Word2Vec. Точность модели – около 67.7%.

На рисунке 6 приведен пример использования модели FastText для поиска близких слов, для слова «oil». Можно сделать вывод, о том, что интерпретируемость модели FastText не конкурирует даже с моделью Word2Vec. При этом точность определения парафраз незначительно хуже.

Предложенный в работе метод более ресурсозатратный. Для оценивания семантической близости двух текстовых фрагментов сначала строятся их тематические модели и определяются общие доминирующие темы. Тексты, не имеющие пересечений по темам, далее не

сравниваются и отмечаются как различные. Из каждой общей для текстов темы берутся векторные представления слов. Затем, на их основе, строятся векторные представления обоих фрагментов. В качестве оценки семантической близости двух текстовых фрагментов принимается косинусная близость двух векторов в каждой теме. Данные значения подаются на вход классификатору, который определяет, являются ли текстовые фрагменты парафразами.

На рисунке 7 для каждой модели Skip-gram, обученной на конкретной монотематической коллекции, представлены пороги классификации $k$, при которых модель имеет наилучшее качество по мере precision. $k$ – коэффициент совпадения, при превышении которого текстовые фрагменты считаются парафразами.

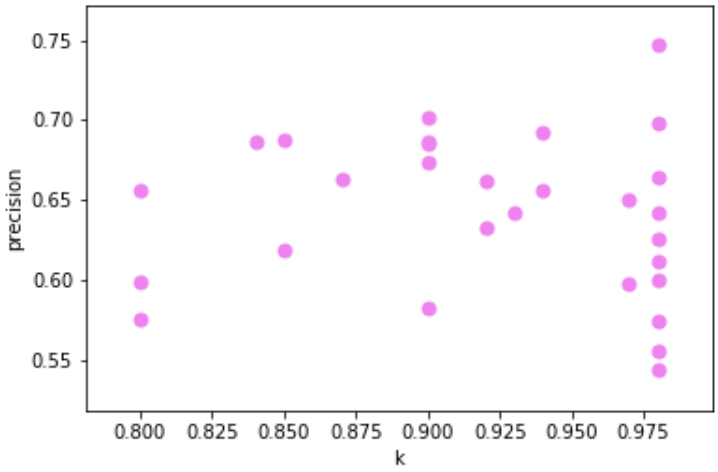


Рис. 7: Качество монотематичных моделей.

Предложенный метод, заключающийся в разбиении коллекции на монотематические подколлекции и обучения в каждой модели Skip-gram, во втором эксперименте дает 74.7% точности (черная точка на рисунке 8).

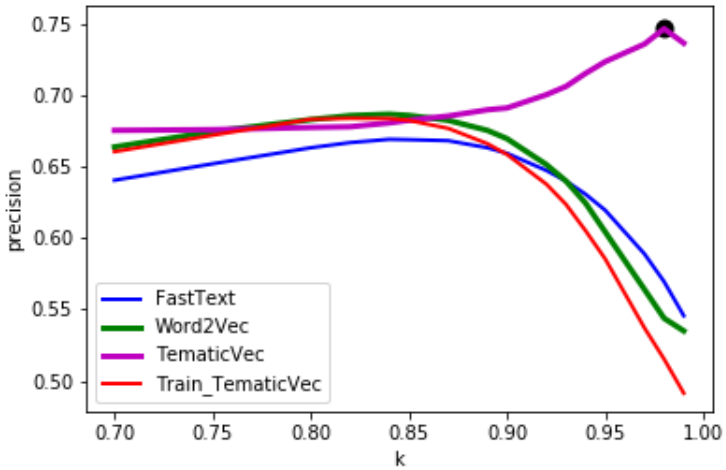


Рис. 8: Сравнение моделей векторных представлений слов.

На рисунке 8 представлено сравнение качества моделей при разных порогах косинусной близости текстовых фрагментов. Модели FastText и Word2Vec имеют похожее поведение. Точность обеих моделей достигает максимума при пороге косинусной близости $k = 0.84$.

Первый эксперимент (TrainTematicVec) – крупная тематизация с последующим обучением трех независимых моделей Skip-gram, также не отличается по качеству и поведению из-за недостаточного разделения коллекции.

Для второго эксперимента (TematicVec) коллекция разделена на 28 монотематичных подколлекций, и картина меняется. Отдельные нейросетевые модели векторных представлений слов достигают высшей точности при разных порогах косинусной близости (Рис. 7), при этом большинство из них имеют наивысшую точность при пороге классификации $k$ более 0.95. Данный факт объясняется тем, что в более мелких монотематичных коллекциях, тексты являются близкими по теме, и слова в векторной форме расположены ближе друг к другу изначально. Поэтому, чтобы документы были действительно похожи, их косинусная близость должна быть высокой.

| Model | Word2Vec | FastText | TrainTematicVec | TematicVec |
|---|---|---|---|---|
| Precision | 68.7% | 67.7% | 68.4% | 74.7% |

Таблица 1: Сравнение моделей.

## 7.4 Предобработка текстов

Большое влияние на качество классификации текстов оказывает их предобработка. Стандартный для английского языка метод лемматизиции лишает короткие предложения смысла. Для человека сравнивать лемматизированные тексты более сложная задача, так как люди основывают решение о схожести текстов по тонким речевым особенностям, которые не удается учитывать при лемматизации.

В эксперименте используется лемматизатор WordNet[5]. Для преобразования коллекции были лемматизированы отдельные части речи. Результаты эксперимента отображены в Таблице 2, они представляются точностью классификации текстов, основанной на близости их векторных представлений с помощью модели Skip-gram по лемматизированным коллекциям. Также было рассчитано отношение лемматизированых терминов ко всем терминам коллекции – count.

При лемматизации корпуса качество модели незначительно вырастает. Словарь коллекции становится меньше и для слов в нем формируются более полные контексты.

| Lemmatizer | Noun | Adverb | Verb | ALL |
|---|---|---|---|---|
| Precision | 68.8% | 68.7% | 69.2% | 69.1% |
| Count | 6.33% | 0.42% | 8.75% | 15.5% |

Таблица 2: Сравнение методов лемматизации.

[5]https://wordnet.princeton.edu/

При тематизации коллекции была реализована идея о том, что для сравнения текстов недостаточно информации о конкретном слове, текст должен оцениваться целиком. Результаты экспериментов показали, что многозначным является весь текст, а не конкретное слово в нем.

Действительно, некоторые слова не имеет смысл сравнивать в отдельности от контекста. Более качественную оценку семантической близости короткого текста можно получить, если при сравнении учитывать только значимые единицы речи.

```
model.wv.most_similar(positive=['aloe'])
```

```
[('vera', 0.9092367887496948),
 ('gel', 0.8310377597808838),
 ('juice', 0.7971554398536682),
 ('coconut', 0.7864762544631958),
 ('onion', 0.7634326815605164),
 ('mustard', 0.759655773639679),
 ('olive', 0.7563325762748718),
 ('lemon', 0.7448221445083618),
 ('moisturizer', 0.7377375960350037),
 ('vinegar', 0.7354497313499451)]
```

```
model.wv.most_similar(positive=['vera'])
```

```
[('aloe', 0.9092369079589844),
 ('gel', 0.8464951515197754),
 ('juice', 0.8012564182281494),
 ('coconut', 0.7744492292404175),
 ('mustard', 0.7732386589050293),
 ('moisturizer', 0.7500103712081909),
 ('olive', 0.7410944700241089),
 ('lemon', 0.7398724555969238),
 ('peanut', 0.7370172739028931),
 ('onion', 0.7345526814460754)]
```

9.a: Без учета коллокаций

```
model_D.wv.most_similar(positive=['aloe_vera'])
```

```
[('gel', 0.8541169166564941),
 ('nail_polish', 0.8099393248558044),
 ('coconut', 0.7976799607276917),
 ('olive', 0.7820346355438232),
 ('moisturizer', 0.7606412768363953),
 ('vinegar', 0.7590630054473877),
 ('powder', 0.7492128610610962),
 ('shampoo', 0.7461521625518799),
 ('butter', 0.7396477460861206),
 ('lotion', 0.7395588159561157)]
```

9.b: С учетом коллокаций

Рис. 9: Пример использования колокациий.

В первом эксперименте поиск коллокаций производился для всех пар слов. Для слов длинной более трёх букв, стоящих непосредственно рядом, оценивалась их самостоятельная и общая частота в коллекции. На основе данных параметров с использованием формулы (6) пары слов заменялись соответствующим термином. На рисунке 9 представлены контексты слов с использованием коллокаций и без в модели. Теоретически можно определить положительное влияние, при этом точность предсказания модели целевого признака становится немного меньше (на 0.8%). Обратно применению лемматизации, при увеличении словаря коллекции, векторные представления не так качественно предсказываются.

Также проводились эксперименты для поиска коллокаций в окне размерности 2 и 3. На результаты это практически не повлияло.

Во втором эксперименте поиск коллокаций производился для каждой подколлекции. Была выдвинута гипотеза, что одни и те же слова в одной теме будут являться коллокацией, а в другой нет. Также в разных монотематичных коллекциях слово может образовывать коллокации с разными предметными словами.

В таблице 3 представлены результаты эксперимента в виде отношения выделенных коллокаций ко всем терминам подколлекции (percent). Анализ выделения коллокаций в

независимых коллекциях показал, что некоторые темы, не подвержены объединению слов в устойчивые словосочетания. Больше всего коллокаций было выделено в теме "Одушевленные прилагательные".

| Topic | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Percent | 7% | 10.1% | 8.2% | 3.7% | 6.6% | 8.3% | 10.8% | 8% | 11.3% | 6.6% |
| Topic | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Percent | 3.1% | 8.2% | 6% | 6.8% | 3.6% | 8.9% | 10.7% | 5.1% | 6.7% | 4.5% |
| Topic | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | ALL | |
| Percent | 11.6% | 11% | 3.8% | 7.8% | 5.9% | 3.3% | 2.8% | 9.2% | 3% | |

Таблица 3: Сравнение количества коллокаций в подколлекциях.

Несмотря на высокую интерпретируемость модели (рис. 9), тематические векторные представления слов, обученные с учётом коллокаций, не смогли повысить точность классификации парафраз относительно базовой модели (рис. 10).

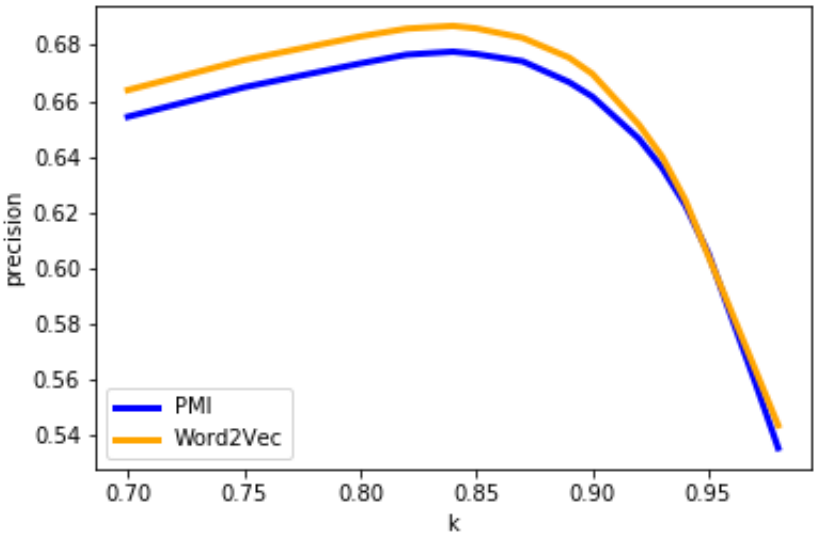


Рис. 10: Точность модели при выделении коллокаций и без.

# 8 Заключение

В работе проведен анализ методов предобработки данных, тематических моделей и моделей дистрибутивной семантики для построения векторных представлений слов, и предложен новый метод, основанный на объединении преимуществ нескольких подходов, нацеленный на решение проблемы полисемии слов при оценивании семантической близости коротких текстов.

Предложенный метод обучения тематических векторных представлений слов реализуется в два этапа. На первом этапе над текстами коллекции производится нечеткая классификация за счет близости документов и начальных приближений для словарей тем. В результате тематизации коллекции формируются N независимых подколлекций. Второй этап подразумевает построение векторных представлений слов, основываясь на контексте слова в конкретной теме.

Построенная модель определяет семантическую близость коротких текстов за три этапа. Первый этап заключается в определении тем для каждого документа. Второй этап – поиск общих тем документов. Последний этап использует обученные модели векторных представлений слов в каждой общей теме для оценивания близости текстов для всех их общих смыслов. Рассчитанные значения используются моделью бинарной классификации для выделения парафраз коллекции.

Экспериментальная часть работы выполнена с помощью языка программирования Python. Эксперименты производились на коллекции коротких текстов на английском языке. В результате проверки, было определено, что предложенный метод, основанный на выдвинутой в работе *гипотезе однозначности слов* в пределах монотематической коллекции обладает рядом преимуществ:

1. Предложение сравнивать документы в их конкретных значениях обеспечивает модели 74.7% точности в определении парафраз, что превосходит на 6.0% существующие методы;
2. Тематические векторные представления слов (TematicVec) имеют более высокую интерпретируемость, чем векторные представления слов моделей Word2Vec и FastText;
3. Семантическая модель TematicVec не требует ипользования предобученных векторных представлений слов;
4. Метод универсален и может быть успешно использован для построения модели выделения парафраз на любой коллекции текстовых документов английского языка.

Вторая часть экспериментов была посвящена анализу влияния методов предобработки текстов на качество базовой модели Word2Vec, обученной только за счет коллекции. Выяснено, что лемматизация увеличивает точность модели незначительно (до 0.5%). Предложенное в работе улучшение модели за счет выделения устойчивых словосочетаний не повышает точность модели, но делает модель более интерпретируемой при предсказании схожих терминов для слов и фраз.

Исходя из успешности предложенного метода TematicVec, дальнейшие исследования будут направлены на обучение семантических моделей, с усиленными связями слов в конкретной области.

# Литература


[1] Воронцов К. В. Обзор вероятностных тематических моделей // Автоматическая обработка текстов на естественном языке и анализ данных: учеб. пособие / Большакова Е. И., Воронцов К. В., Ионов М. И., Клышинский Э. С., Лукашевич Н. В. — М.: Изд-во НИУ ВШЭ, 2017. 235 — 272 с.

[2] Sahlgren M. The Distributional Hypothesis. From context to meaning (англ.) // Distributional models of the lexicon in linguistics and cognitive science (Special issue of the Italian Journal of Linguistics), Rivista di Linguistica : журнал. — 2008. — Vol. 20, no. 1. — P. 33-53.

[3] Thomas Hoffman. Probabilistic Latent Semantic Indexing // Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval. — 1999.

[4] Yuan Zuo, Jichang Zhao, and Ke Xu. Word network topic model: a simple but general solution for short and imbalanced texts. Knowledge and Information Systems, 48(2):379–398, 2016.

[5] David M. Blei, Andrew Y. Ng, Michael I. Jordan. Latent Dirichlet Allocation // Journal of Machine Learning Research. — 2003.

[6] Воронцов К. В. Аддитивная регуляризация тематических моделей коллекций текстовых документов // Доклады РАН. 2014. — Т. 455., №3. 268–271

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781

[8] 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.

[9] Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2015. Breaking sticks and ambiguities with adaptive skip-gram. arXiv:1502.07257.

[10] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5: 135–146, 2017.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. 2014.

[12] Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. Starspace: Embed all the things! arXiv preprint arXiv:1709.03856, 2017.

[13] Chandrasekar, R. Doran, C. and Srinivas, B., Motivations and Methods for Text Simplification, 1996.

[14] Robert M. Fano. Transmission of Information A Statistical Theory of Communication, ISBN:9780262561693, 1961. – p.27

[15] Porter, M.F. (1980), "An algorithm for suffix stripping", Program, Vol. 14 No.3, pp. 130-137.

[16] Воронцов К. В. Потапенко А. А. Регуляризация вероятностных тематических моделей для повышения интерпретируемости и определения числа тем // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 4–8 июня 2014 г.) Вып.13 (20). М: Изд-во РГГУ, 2014. С.676–687.

[17] El-Kishky A., Song Y., Wang C., Voss C. R., Han J. Scalable topical phrase mining from text corpora // Proc. VLDB Endowment. — 2014. — Vol. 8, no. 3. — pp. 305–316.

[18] *Frantzi K., Ananiadou S., Mima H.* Automatic recognition of multi-word terms: the C-value/NC-value method // Intl. J. of Digital Libraries Vol. 3 Issue 2, p. 117–132, 2000.

[19] Vorontsov K. V. Additive Regularization for Topic Models of Text Collections // Doklady Mathematics. 2014, Pleiades Publishing, Ltd. — Vol. 89, No. 3, pp. 301–304.

[20] Vorontsov K. V., Potapenko A. A. Additive Regularization of Topic Models // Machine Learning Journal. Special Issue "Data Analysis and Intelligent Optimization with Applications".

[21] Potapenko A. A., Popov A. S., Vorontsov K. V. Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks // Filchenkov A., Pivovarova L., Žižka J. (eds) Artificial Intelligence and Natural Language. AINL 2017, St. Petersburg, Russia, September 20-23, 2017. — Communications in Computer and Information Science, vol 789. Springer, Cham, 2017. — pp 167-180.

[22] Воронцов К. В. Вероятностное тематическое моделирование [Электронный ресурс]. URL: www.machinelearning.ru/wiki/images /2/22/Voron-2013-ptm.pdf.