Перевод статьи:
Fomin S.A., Belousov R.L. Detecting semantic duplicates in short news items
Business Informatics. 2017. No. 2 (40). P. 47–56.

# Поиск семантических дубликатов в коротких новостных сообщениях

**С.А. Фомин**
*бакалавр технических наук*
*оператор лаборатории научно-исследовательского центра*
*Академия гражданской защиты МЧС России*
*Адрес: 141435, Московская область, г. Химки, мкр. Новогорск*
*E-mail: sergio-dna@yandex.ru*

**Р.Л. Белоусов**
*кандидат технических наук, научный сотрудник научно-исследовательского центра*
*Академия гражданской защиты МЧС России*
*Адрес: 141435, Московская область, г. Химки, мкр. Новогорск*
*E-mail: romabel-87@mail.ru*

**Аннотация**

В статье рассмотрена задача, связанная с обнаружением публикаций, схожих по смыслу, а также публикаций, посвященных одному событию. Особенность решаемой задачи заключается в том, что в качестве публикаций рассматриваются короткие новостные сообщения, средняя длина которых составляет 40 слов. Для решения указанной задачи разработан алгоритм, в основу которого положена векторная модель семантики, где каждый текст рассматривается как точка в многомерном пространстве. Преобразование корпуса текстов в матрицу производится с помощью меры TF-IDF. Необходимо отметить, что даже для небольших корпусов (объемом порядка 800 сообщений) размерность векторного пространства может превосходить 2000 компонент, а в среднем размерность составляет около 8500 компонент. Для сокращения размерности пространства используется метод главных компонент. Его применение позволяет рационально сократить размерность пространства и оставить около трех процентов компонент от их исходного количества.

В сокращенном пространстве для объединения векторов в кластеры применяется агломеративная иерархическая кластеризация по алгоритму Ланса–Уильямса, который запускает процесс слияния кластеров. Слияние кластеров производится с помощью вычисления расстояния между ближайшими элементами этих кластеров. Процесс слияния кластеров прекращается в том случае, если расстояние между двумя кластерами превышает некоторое значение $r$.

При проведении численного эксперимента построена регрессионная модель, позволяющая найти наиболее подходящее значение параметра $r$ для каждого корпуса сообщений. В качестве исходных данных для проведения численного эксперимента использовалась коллекция коротких новостей, общий объем которых составляет около 135 тысяч сообщений.

Разработанный алгоритм имеет достаточно высокие показатели качества, которые учитывают, с одной стороны, способность классифицировать пары текстовых сообщений как семантические дубликаты, а с другой – способность объединять найденные дубликаты в группы.

**Ключевые слова:** коллекция коротких текстовых сообщений, кластеризация текстов, нечеткие дубликаты, векторная модель семантики, нейронная сеть.

**Цитирование:** Fomin S.A., Belousov R.L. Detecting semantic duplicates in short news items // Business Informatics. 2017. No. 2 (40). P. 47–56. DOI: 10.17323/1998-0663.2017.2.47.56.

Перевод статьи:
Fomin S.A., Belousov R.L. Detecting semantic duplicates in short news items
Business Informatics. 2017. No. 2 (40). P. 47–56.

## Введение

В июне 2014 года был опубликован отчет «Российский медиа-ландшафт: телевидение, пресса, Интернет [1]. Особого внимания в этом отчете за-служивает тот факт, что примерно треть населения страны (34 %) использует Интернет для того, чтобы «следить за последними новостями» и 20 % – чтобы «разобраться, что происходит в стране и за рубежом».

Публичные коммерческие компании и государственные организации не оставляют без внимания приведенные цифры и факты, поскольку для создания позитивного имиджа и безупречной деловой репутации необходимо вести активную информационную политику в Интернете.

Для реализации этих потребностей создаются информационные системы, которые позволяют в автоматическом режиме производить сбор, обработку и анализ информации из различных источников. Одним из ключевых требований, предъявляемых к подобным системам, является их способность детектировать схожие публикации, а также публикации, посвященные одному событию (http://www.mlg.ru/solutions/pr/analysis/, https://pressindex.ru/#technology).

В данной статье рассматривается алгоритм поиска дубликатов в коротких новостных сообщениях, полученных из RSS-лент различных новостных порталов или иным способом.

Дубликатами являются одинаковые по смыслу сообщения, у которых может наблюдаться частичное или полное лексическое совпадение. Таким образом, дубликаты обладают семантической схожестью.

Задача поиска дубликатов, в том числе, в коротких текстовых документах не является новой, решению этой задачи посвящены работы [2–5].

## 1. Исходные данные

Короткие новостные сообщения, как правило, состоят из заголовка и лида – первого абзаца, отвечающего на вопросы «что произошло?», «когда произошло?» и «где произошло?».

Для проведения исследования использовалась коллекция коротких новостных сообщений за 20 дней. Объем коллекции составляет около 135 тысяч сообщений. Если сообщения относятся к одному и тому же событию, то они имеют одинаковую метку дубликата *dup*.

В *таблице 1* представлены некоторые статистические характеристики собранной коллекции новостных сообщений.

*Таблица 1.*

**Статистические характеристики коллекции новостных сообщений**

| Характеристика | Коллекция | | |
|---|---|---|---|
| | Среднее | Медиана | Мода |
| Количество слов в сообщении | 39,72 | 37 | 33 |
| Количество уникальных слов в сообщении | 33,77 | 32 | 30 |
| Количество сообщений, имеющих одну и ту же метку дубликата *dup* | 14,73 | 4 | 1 |
| Количество сообщений в день | 6725,95 | 6487 | – |

Средняя длина сообщений составляет 39,72 слов. В сообщениях возможны грамматические ошибки и опечатки. Поиск дубликатов ведется среди сообщений, опубликованных в один и тот же день. Структура и типы исходных данных представлены в *таблице 2.*

*Таблица 2.*

**Структура и типы исходных данных**

| *id* | *head* | *description* | *time* | *dup* |
|---|---|---|---|---|
| hash | char | char | smalldatetime | char |

Здесь *id* – уникальный идентификатор сообщения, *head* – заголовок сообщения, *description* – основная часть сообщения (лид), *time* – дата и время публикации сообщения, *dup* – метка дубликата. Если два сообщения имеют одинаковую метку *dup*, то они являются семантическими дубликатами.

Сообщения, которые имеют одну и ту же метку *dup*, образуют группы. В *таблице 3* представлена информация о количестве таких групп. Следует обратить внимание на тот факт, что процент уникальных сообщений незначителен, таких сообщений всего 2385 из 135 тысяч. В коллекции имеется 108 групп, состоящих из 100 или более сообщений (самая крупная группа состоит из 1039 сообщений).

*Таблица 3.*

**Группы сообщений**

| Количество сообщений в группе | Количество групп |
|---|---|
| 1 | 2385 |
| 2 | 1026 |
| 3 | 649 |
| 4 | 522 |
| 5 | 433 |
| 6 | 351 |
| 7 | 278 |
| 8 | 275 |
| 9 | 229 |
| 10 | 195 |
| > 10 | 2792 |

## 2. Постановка задачи

Проблеме обнаружения дубликатов в текстовых документах посвящена статья Ю.Г. Зеленкова и И.В. Сегаловича, где дается сравнительное исследование наиболее популярных современных методов обнаружения нечетких дубликатов [6]. Важно отметить, что нечеткие дубликаты не всегда семантически схожи, т.е. имеют один и тот же смысл. Кроме того, методы, которые позволяют обнаруживать нечеткие дубликаты, не всегда корректно работают на текстах малой длины.

Поэтому задача исследования сформулирована следующим образом: разработать алгоритм обнаружения семантических дубликатов в коротких новостных сообщениях и объединения их в группы.

## 3. Описание алгоритма

За основу реализации алгоритма взята идея векторной модели семантики, где каждый текст рассматривается как точка в многомерном пространстве. Близко лежащие друг к другу точки соответствуют семантически схожим документам [7]. Рассмотрим описание каждого этапа алгоритма и инструментов для их реализации.

**Первый этап – предобработка.** Сообщения, опубликованные за один день, агрегируются в корпусы меньших размеров. Затем все слова в сообщениях приводятся к нормальной форме с использованием морфологического анализатора Mystem (https://tech.yandex.ru/mystem/). Таким образом, исходная коллекция сообщений $C$ может рассматриваться как объединение агрегированных корпусов $c_i$:

$$C = \bigcup_i c_i. \quad (1)$$

Исследуемая коллекция коротких новостных сообщений разбивается на 20 корпусов, поскольку содержит сообщения за 20 дней.

**Второй этап – построение векторной модели.** Сообщения из корпуса $c_i$ необходимо преобразовать в матрицу. Для решения этой задачи применяется мера TF-IDF [8].

Для каждого слова $t$ в конкретном сообщении $d$ рассчитывается TF-мера по формуле (2):

$$tf(t,d) = \frac{n_i}{\sum_k n_k}, \quad (2)$$

где $n_i$ – количество вхождений слова $t$ в сообщение $d$;

$\sum_k n_k$ – общее количество слов в сообщении $d$.

Для каждого слова $t$ в корпусе текстов $c_i$ рассчитывается IDF-мера по формуле (3):

$$idf(t,c_i) = \log \frac{|c_i|}{|(d_i \supset t)|}, \quad (3)$$

где $|c_i|$ – количество сообщений в корпусе $c_i$;

$|(d_i \supset t)|$ – количество сообщений, в которых встречается слово $t$.

Таким образом, каждое сообщение преобразуется в вектор, компонентами которого является мера TF-IDF каждого слова в этом сообщении. Для конкретного слова мера TF-IDF определяется как произведение мер TF и IDF.

Мера TF для слова определяется локально в каждом сообщении, а мера IDF – глобально для корпуса и не зависит от конкретного сообщения. Значение TF-IDF интерпретируется как «вклад» определенного слова в смысл сообщения.

Результат выполнения второго этапа представляется в виде матрицы (*таблица 4*), где каждый столбец определяет отдельное слово $t$ из корпуса $c_i$, а каждая строка соответствуют некоторому сообщению $d$.

*Таблица 4.*

**Матрица TF-IDF**

| | $t_1$ | $t_2$ | ... | $t_m$ |
|---|---|---|---|---|
| $d_1$ | $tf$ - $idf(d_1, t_1)$ | 0 | ... | $tf$ - $idf(d_1, t_m)$ |
| $d_2$ | 0 | 0 | ... | $tf$ - $idf(d_1, t_2)$ |
| ... | ... | ... | ... | ... |
| $d_n$ | $tf$ - $idf(d_n, t_1)$ | ... | ... | $tf$ - $idf(d_n, t_m)$ |

При построении векторной модели игнорируются слова с высокой степенью встречаемости и такие, которые встречаются один раз. Для представленных исходных данных высокочастотными словами считаются такие, которые встречаются более чем в 90 % сообщений.

Для построения векторной модели использовалась библиотека машинного обучения scikit-learn (http://scikit-learn.org/stable/) для языка программирования Python.

**Третий этап – уменьшение числа векторных компонент.** Суть данного этапа сводится к сокращению размерности данных, поскольку в среднем каждая матрица, соответствующая корпусу сообщений $c_i$, содержит 8547 столбцов. Для сокращения количества столбцов в *таблице 4* с минимальной потерей информативности применяется метод главных компонент.

Каждый столбец *таблицы 4* – это переменная $t_i$, а строка – номер наблюдения. Все переменные $t_i$ центрируются по формуле (4):

$$x_i = t_i - \overline{t_l}, \quad (4)$$

где $\overline{t_l}$ – среднее значение переменной $t_i$.

Затем осуществляется переход к новым переменным – главным компонентам по формуле (5):

$$pc_j = \sum_i^m v_{ij} \cdot x_i, \quad (5)$$

при этом сумма квадратов весовых коэффициентов $v_{ij}$ должна быть равна единице.

Новые переменные $pc_1, pc_2, \ldots, pc_m$ создаются таким образом, чтобы выполнялись следующие условия [9]:

✦ первая главная компонента $pc_1$ имеет максимально возможную выборочную дисперсию $sVar(pc_1)$;

✦ переменная $pc_2$ некоррелирована с $pc_1$ и имеет максимально возможную выборочную дисперсию $sVar(pc_2)$;

✦ переменная $pc_3$ некоррелирована с $pc_1$, $pc_2$ и имеет максимально возможную выборочную дисперсию $sVar(pc_3)$;

✦ и т.д.

Для уменьшения количества столбцов в *таблице 4* достаточно отбросить переменные, которые обладают наименьшими весами в линейной комбинации (5).

Количество столбцов в новой таблице рассчитывается как произведение количества столбцов в старой таблице и параметра $m$, где $m \in (0, 1]$. Установлено, что при изменении параметра $m$ в интервале от 0,02 до 0,1 качество работы алгоритма изменяется незначительно. Значения $m$ больше 0,1 увеличивают вычислительную сложность последующих операций. Наиболее рациональное значение параметра $m$ равно 0,03.

Метод главных компонент (principal component analysis, PCA) также реализован в библиотеке машинного обучения scikit-learn.

**Четвертый этап – измерение схожести двух векторов.** После реализации третьего этапа в *таблице 4* сократилось количество столбцов, при этом количество строк осталось прежним. Каждая строка соответствует конкретному тексту (документу) $d$ и рассматривается как вектор.

Самый популярный способ измерения схожести двух векторов – это нахождение косинуса угла между ними [7]. Чем больше значение косинуса, тем большей схожестью обладают векторы.

Для удобства дальнейшего использования алгоритмов кластеризации значение косинуса вычитается из единицы. В результате получается матрица косинусовых расстояний **A**:

$$a_{ij} = 1 - \cos(d_i, d_j). \quad (6)$$

Нахождение косинуса угла между двумя векторами реализовано в библиотеке анализа данных scikit-learn для языка программирования Python.

Необходимо отметить, что значение косинуса еще не позволяет судить о том, являются ли два сообщения семантическими дубликатами.

**Пятый этап – объединение векторов в кластеры.** Сообщения, которые попадают в одни и те же группы, семантически схожи, а соответствующие им векторы образуют кластеры.

Для объединения векторов **d** в кластеры применяется агломеративная иерархическая кластеризация. Суть такой кластеризации сводится к следующему. Сначала каждый вектор, соответствующий текстовому сообщению, рассматривается как отдельный кластер. Расстояния между этими кластерами содержатся в матрице **A**, полученной на четвертом этапе алгоритма.

Затем запускается процесс слияний. На каждой итерации вместо пары самых близких кластеров $U$ и $V$ образуется новый кластер $W = U \cup V$. Расстояние от нового кластера $W$ до любого другого кластера $S$ вычисляется по алгоритму Ланса–Вильямса [10]:

$$R(W,S) = \alpha_U R(U,S) + \alpha_V R(V,S) + \\ +\beta R(U,V) + \gamma \left| R(U,S) - R(V,S) \right|, \quad (7)$$

где расстояния $R(U,S)$, $R(V,S)$, $R(U,V)$ и числовые параметры $\alpha_U$, $\alpha_V$, $\beta$, $\gamma$ вычисляются методом ближайших соседей [10]:

$$R(W,S) = \min_{w \in W, s \in S} \rho(w,s), \\ \alpha_U = \frac{1}{2},\ \alpha_V = \frac{1}{2},\ \beta = 0,\ \gamma = -\frac{1}{2}. \quad (8)$$

Процесс слияния кластеров прекращается в том случае, если расстояние между двумя кластерами превышает некоторое значение параметра *r*.

Данный алгоритм агломеративной кластеризации реализован в библиотеке машинного обучения scikit-learn.

В представленном алгоритме агломеративная кластеризация позволяет найти пары семантических дубликатов и объединить их в группы.

### 4. Численный эксперимент

В рамках численного эксперимента проводилась оценка качества работы алгоритма, которая учитывает, с одной стороны, способность классифицировать пары текстовых сообщений как семантические дубликаты, с другой – способность объединять найденные дубликаты в группы.

Для оценки способности алгоритма объединять сообщения в смысловые группы, т.е. оценки качества кластеризации, применяется исправленный индекс Ранда (Adjusted Rand Index, ARI) [11] и исправленный индекс взаимной информации (Adjusted Mutual Information, AMI) [12].

Индекс ARI и индекс AMI представляют собой меру согласия и меру сходства между двумя разбиениями множества объектов соответственно.

Классификация сообщений оценивалась по следующим метрикам: точность (*P*), полнота (*R*) и *F*-мера – гармоническое среднее между точностью и полнотой [13]. С помощью этих метрик определяется способность алгоритма классифицировать пары текстовых сообщений как семантические дубликаты. Для удобства восприятия результатов классификации цифрой 1 обозначим класс дубликатов, цифрой 0 – класс недубликатов.

Приведем пример. Если два сообщения, которые попали в одну и ту же группу, имеют одинаковую метку *dup* в *таблице 2* (класс 1), то классификация проведена верно – обнаружены семантические дубликаты (класс 1). Такая классификация называется истинно-положительной (*TP*).

Другой пример. Два сообщения классифицированы как семантические дубликаты (класс 1), т.е. попали в одну группу. При этом эти сообщения имеют разные метки *dup* в *таблице 2* (класс 0). Это свидетельствует о том, что алгоритм неверно классифицировал данную пару сообщений. Такое решение называется ложно-положительным (*FP*).

Также выделяют истинно-отрицательную *TN* и ложно-отрицательную *FN* классификацию.

Приведенные типы классификации и выбранные метрики связаны соотношением (9):

$$P = \frac{TP}{TP + FP},\ R = \frac{TP}{TP + FN},\ F = 2\frac{P \cdot R}{P + R}. \quad (9)$$

Рассмотренные метрики и индексы качества реализованы в библиотеке машинного обучения scikit-learn.

Главную роль в оценке качества работы алгоритма играет параметр кластеризации *r*, который определяется при проведении агломеративной кластеризации. Очевидно, что оптимальное значение параметра *r* зависит от индивидуальных особенностей корпуса коротких новостных сообщений. Оптимальным значением *r* для отдельного корпуса считается такое значение, которое максимизирует *F*-меру по классу 1. Следует отметить, что термин «оптимальность» используется в узком смысле. Это значит, что получаемые значения *r* оптимальны только для определенных параметров настройки алгоритма. Для других значений параметров настройки оптимальные значения могут изменяться.

Таким образом, практический интерес представляет ответ на вопрос: насколько сильно значение параметра *r* зависит от этих особенностей и можно ли его предсказать?

Первая стадия эксперимента заключалась в эмпирическом подборе такого значения параметра кластеризации *r*, при котором *F*-мера по классу 1 достигает своего наибольшего значения. Именно класс 1 определяет оценку качества алгоритма, поскольку в силу специфики исследуемых данных полнота и точность по классу 0 всегда близки к единице.

На *рисунке 1* изображен график изменения значений выбранных метрик (точность, полнота, *F*-мера по классу 1) от значений параметра кластеризации *r* для случайного корпуса текстов из рассматриваемой коллекции. График обладает ярко выраженной дискретностью. Это связано с особенностями агломеративной кластеризации: поскольку каждое значение параметра *r* определяет расстояние между

кластерами, а количество кластеров ограничено, то значения метрик качества могут изменяться только на некотором ограниченном наборе значений параметра кластеризации *r*.

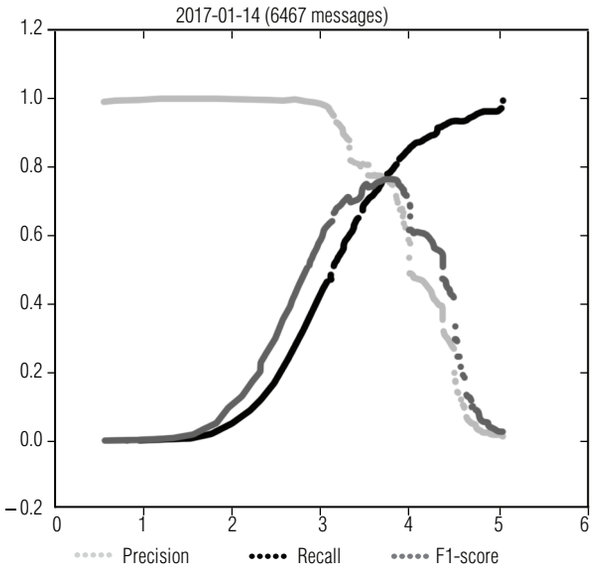


*Рис. 1.* Качество классификации (класс 1) пар сообщений корпуса

*Таблица 5.*

**Значения метрик**

| $\|c_i\|$ | Качество алгоритма | | | | | $r$ |
|---|---|---|---|---|---|---|
| | Классификация | | | Кластеризация | | |
| | *P* | *R* | *F* | *ARI* | *AMI* | |
| 855 | 0,93 | 0,75 | 0,83 | 0,83 | 0,83 | 2,05 |
| 2117 | 0,78 | 0,91 | 0,84 | 0,84 | 0,90 | 2,79 |
| 6056 | 0,66 | 0,67 | 0,67 | 0,67 | 0,76 | 3,59 |
| 7553 | 0,87 | 0,65 | 0,74 | 0,74 | 0,75 | 3,68 |
| 4142 | 0,69 | 0,83 | 0,75 | 0,75 | 0,769 | 3,88 |
| 2934 | 0,80 | 0,76 | 0,78 | 0,77 | 0,84 | 3,02 |
| 5093 | 0,75 | 0,77 | 0,76 | 0,76 | 0,82 | 3,44 |
| 6478 | 0,79 | 0,69 | 0,74 | 0,74 | 0,73 | 3,35 |
| 6869 | 0,72 | 0,76 | 0,74 | 0,74 | 0,83 | 3,89 |
| 6350 | 0,77 | 0,63 | 0,69 | 0,69 | 0,79 | 3,61 |
| 7097 | 0,73 | 0,72 | 0,73 | 0,73 | 0,77 | 3,78 |
| 6496 | 0,73 | 0,62 | 0,67 | 0,67 | 0,68 | 3,31 |
| 8366 | 0,71 | 0,73 | 0,72 | 0,72 | 0,81 | 4,04 |
| 11745 | 0,76 | 0,67 | 0,71 | 0,71 | 0,72 | 4,01 |
| 11106 | 0,79 | 0,75 | 0,77 | 0,77 | 0,79 | 4,20 |
| 7568 | 0,77 | 0,64 | 0,70 | 0,70 | 0,70 | 3,47 |
| 12221 | 0,76 | 0,74 | 0,75 | 0,75 | 0,80 | 4,46 |
| 10472 | 0,71 | 0,76 | 0,73 | 0,73 | 0,78 | 4,15 |
| 6467 | 0,76 | 0,78 | 0,77 | 0,77 | 0,83 | 3,72 |
| 4534 | 0,86 | 0,76 | 0,81 | 0,81 | 0,82 | 3,29 |

В *таблице 5* приведены значения метрик классификации, значения метрик кластеризации для каждого корпуса $c_i$ и соответствующие значения параметра *r*.

Анализ данных *таблицы 5* позволяет сделать вывод, что если значение параметра кластеризации *r* подобрано правильно, то качество работы алгоритма можно оценить как хорошее. Однако значение параметра *r* может сильно варьироваться для каждого корпуса.

**Вторая стадия** эксперимента заключалась в поиске зависимости значения параметра кластеризации *r* от индивидуальных особенностей корпусов коротких новостных сообщений.

Индивидуальные особенности корпуса коротких новостных сообщений могут определяться значениями следующих параметров:

– $p_1$ – количество сообщений в корпусе;

– $p_2$ – средняя длина сообщений;

– $p_3$ – наиболее часто встречаемая (мода) длина сообщений;

– $p_4$ – среднее количество уникальных слов в сообщениях;

– $p_5$ – мода количества уникальных слов в сообщениях;

– $p_6$ – общее количество уникальных слов в корпусе;

– $p_7$ – количество столбцов матрицы TF-IDF (*таблица 4*);

– $p_8$ – параметр кластеризации *r*.

Среди значений параметров, полученных в ходе эксперимента, были обнаружены некоторые закономерности.

Например, с увеличением количества сообщений в корпусе $p_1$ параметр кластеризации *r* в целом также растет (*рисунок 2*).

На *рисунке 3* продемонстрирована тенденция роста оптимального значения параметра *r* при увеличении количества уникальных слов в корпусе $p_6$.

На *рисунке 4* представлен график зависимости оптимального значения параметра *r* от $p_7$ – количества столбцов в матрице TF-IDF.

Для прогнозирования значений параметра кластеризации *r* по значениям входных переменных $p_1$, ..., $p_7$ будут использованы две модели: модель множественной линейной регрессии и нейросетевая модель – многослойный персептрон (MLP *i*-*h*-*o*, *hidden*, *output*), где *i* – размерность входного вектора значений, *h* – количество нейронов в скрытом слое,

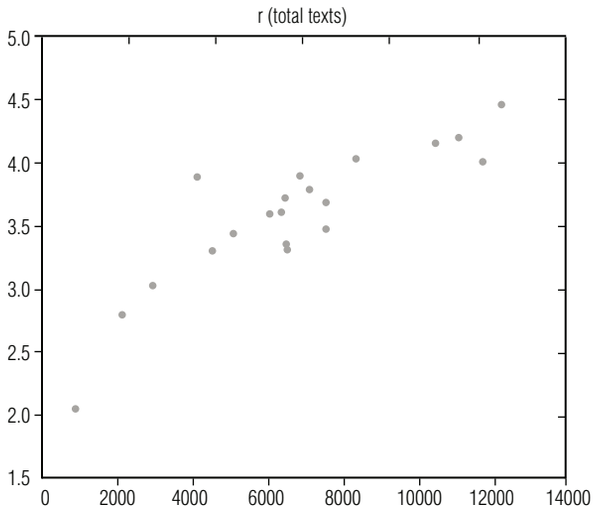


*Рис. 2.* Зависимость оптимального значения параметра *r* от количества сообщений

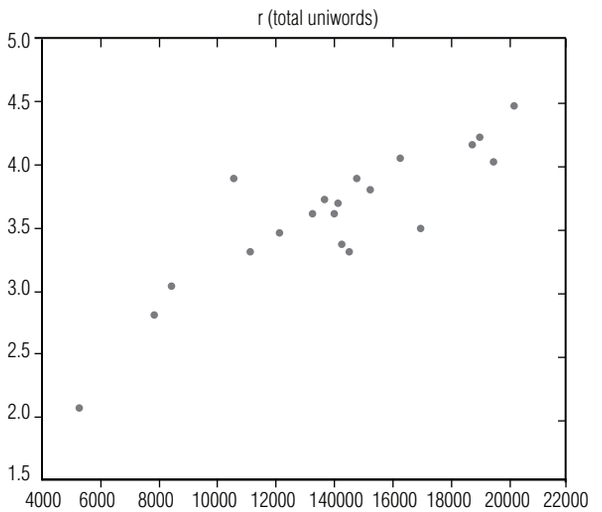


*Рис. 3.* Зависимость оптимального значения параметра *r* от количества уникальных слов в корпусе

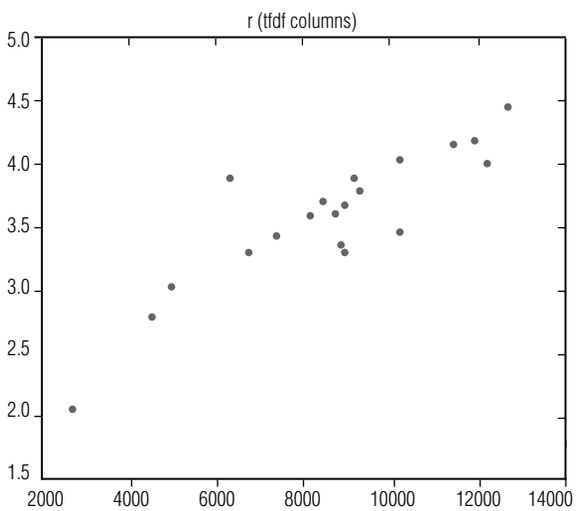


*Рис. 4.* Зависимость оптимального значения параметра *r* от количества столбцов в матрице TF-IDF

*o* – размерность выходного вектора, *hidden* – функция активации нейронов скрытого слоя, *output* – функция активации нейронов выходного слоя.

**Модель множественной линейной регрессии** имеет следующий вид:

$$p_8 = \beta_0 + \beta_1 p_1 + \beta_2 p_2 + \cdots \beta_7 p_7 + \varepsilon, \qquad (10)$$

где $\varepsilon$ – случайная составляющая, ошибка;

$\beta_i$ – неизвестные параметры.

Оценка неизвестных параметров $\widetilde{\beta}_i$ модели (10) производится методом наименьших квадратов. Для тестирования качества модели применялась процедура скользящего контроля по отдельным объектам (leave-one-out CV) [14, 15]: 19 наблюдений, где каждое наблюдение содержит значения параметров $p_1, p_2, \ldots, p_8$, использовались как обучающая выборка для построения модели множественной линейной регрессии, 1 наблюдение использовалось для контроля – прогнозирования параметра *r*.

С учетом того, что в коллекции имеется 20 корпусов общим объемом около 135 тысяч сообщений, было проведено 20 итераций кросс-валидации. В *таблице 6* приведены результаты кросс-валидации модели множественной линейной регрессии (10).

*Таблица 6.*

**Результаты кросс-валидации модели множественной линейной регрессии**

| $\lvert c_i \rvert$ | $R_2$ | Оценка прогноза | |
|---|---|---|---|
| | | $r_i$ | $\widetilde{r}_i$ |
| 855 | 0,85 | 2,051 | 2,783 |
| 2117 | 0,89 | 2,793 | 2,480 |
| 6056 | 0,90 | 3,599 | 3,326 |
| 7553 | 0,91 | 3,689 | 4,160 |
| 4142 | 0,92 | 3,889 | 3,217 |
| 2934 | 0,89 | 3,029 | 2,986 |
| 5093 | 0,90 | 3,441 | 3,631 |
| 6478 | 0,90 | 3,354 | 3,65 |
| 6869 | 0,90 | 3,895 | 3,588 |
| 6350 | 0,90 | 3,611 | 3,466 |
| 7097 | 0,91 | 3,787 | 4,159 |
| 6496 | 0,91 | 3,314 | 3,675 |
| 8366 | 0,90 | 4,043 | 3,787 |
| 11745 | 0,91 | 4,014 | 4,432 |
| 11106 | 0,89 | 4,203 | 4,057 |
| 7568 | 0,90 | 3,476 | 3,617 |
| 12221 | 0,88 | 4,465 | 4,319 |
| 10472 | 0,90 | 4,159 | 3,884 |
| 6467 | 0,90 | 3,721 | 3,616 |
| 4534 | 0,89 | 3,297 | 3,296 |

В приведенной *таблице* $|c_i|$ – количество сообщений в корпусе $c_i$; $R^2$ – коэффициент детерминации модели, построенной по 19 наблюдениям; $r_i$ –реальное оптимальное значение параметра кластеризации в корпусе $c_i$; $\tilde{r}_i$ – прогнозное значение.

Нейросетевая модель представлена трехслойной архитектурой нейронной сети, включающей входной слой, скрытый слой и выходной слой. Обучение нейронной сети производилось с помощью алгоритма Бройдена–Флетчера–Гольдфарба–Шанно (BFGS), который позволяет минимизировать сумму квадратов ошибок (*sos*).

Для использования нейросетевой модели проводится предварительная обработка исходных данных, которая заключается в масштабировании каждой входной и выходной переменной по формуле (11), таким образом, чтобы все значения переменной принадлежали интервалу [0, 1]:

$$\delta = \frac{1}{p_i^{max} - p_i^{min}}, \qquad (11)$$
$$p_i' = 0 - \delta \cdot p_i^{min} + \delta \cdot p_i,$$

где $p_i^{max}$, $p_i^{min}$ – минимальное и максимальное значения переменной $p_i$;

$p_i'$ – масштабированная переменная.

В *таблице 7* представлены результаты кросс-валидации нейросетевой модели с архитектурой (MLP *i-h-o*, *hidden*, *output*).

В *таблице 8* представлено сравнение показателей качества алгоритма, полученных на оптимальных значениях параметра кластеризации *r,* и на значениях, спрогнозированных по двум моделям соответственно.

Анализ полученных результатов позволяет сделать следующие выводы.

Значения параметра кластеризации *r*, полученные с помощью нейросетевой модели, позволяют повысить качество работы алгоритма и приблизиться к оптимальным показателям с учетом заданных настроек параметров алгоритма.

Результаты численного эксперимента подтвердили тот факт, что предложенный алгоритм, с одной стороны, способен классифицировать сообщения как семантические дубликаты, а с другой стороны – объединять найденные дубликаты в группы, основываясь лишь только на частотных характеристиках корпусов и текстов.

*Таблица 7.*

**Результаты кросс-валидации нейросетевой модели**

| $\|c_i\|$ | Конфигурация сети MLP | $R^2$ | Оценка прогноза | |
|---|---|---|---|---|
| | | | $r_i$ | $\tilde{r}_i$ |
| 855 | 7-12-1, log, log | 0,97 | 2,051 | 2,818 |
| 2117 | 7-7-1, exp, exp | 0,97 | 2,793 | 2,778 |
| 6056 | 7-6-1, tanh, tanh | 0,97 | 3,599 | 3,597 |
| 7553 | 7-8-1, tanh, ident | 0,85 | 3,689 | 3,732 |
| 4142 | 7-7-1, tanh, log | 0,86 | 3,889 | 3,662 |
| 2934 | 7-4-1, ident, ident | 0,88 | 3,029 | 2,992 |
| 5093 | 7-10-1, log, tanh | 0,80 | 3,441 | 3,585 |
| 6478 | 7-6-1, exp, tanh | 0,96 | 3,354 | 3,444 |
| 6869 | 7-8-1, ident, log | 0,75 | 3,895 | 3,855 |
| 6350 | 7-12-1, tanh, log | 0,90 | 3,611 | 3,617 |
| 7097 | 7-12-1, exp, ident | 0,98 | 3,787 | 3,785 |
| 6496 | 7-9-1, exp, exp | 0,97 | 3,314 | 3,494 |
| 8366 | 7-11-1, tanh, exp | 0,92 | 4,043 | 3,916 |
| 11745 | 7-11-1, log, ident | 0,85 | 4,014 | 4,113 |
| 11106 | 7-5-1, exp, log | 0,93 | 4,203 | 4,203 |
| 7568 | 7-10-1, ident, tanh | 0,90 | 3,476 | 3,539 |
| 12221 | 7-10-1, exp, log | 0,95 | 4,465 | 4,202 |
| 10472 | 7-5-1, tanh, log | 0,91 | 4,159 | 4,004 |
| 6467 | 7-4-1, exp, exp | 0,95 | 3,721 | 3,720 |
| 4534 | 7-4-1, log, tanh | 0,92 | 3,297 | 3,286 |

*Таблица 8.*

**Сравнение показателей качества алгоритмов**

| Оптимальное значение $r_i$ | | | | | |
|---|---|---|---|---|---|
| | *P* | *R* | *F* | *ARI* | *AMI* |
| Мин. | 0,66 | 0,62 | 0,67 | 0,67 | 0,69 |
| Макс. | 0,93 | 0,91 | 0,84 | 0,84 | 0,90 |
| Среднее | 0,77 | 0,73 | 0,75 | 0,74 | 0,79 |
| **Множественная линейная регрессия** | | | | | |
| | *P* | *R* | *F* | *ARI* | *AMI* |
| Мин. | 0,29 | 0,39 | 0,44 | 0,43 | 0,68 |
| Макс. | 0,88 | 0,95 | 0,82 | 0,81 | 0,84 |
| Среднее | 0,68 | 0,71 | 0,67 | 0,67 | 0,77 |
| **Нейросетевая модель** | | | | | |
| | *P* | *R* | *F* | *ARI* | *AMI* |
| Мин. | 0,55 | 0,63 | 0,64 | 0,64 | 0,69 |
| Макс. | 0,86 | 0,95 | 0,84 | 0,84 | 0,90 |
| Среднее | 0,72 | 0,74 | 0,73 | 0,72 | 0,79 |

## Заключение

В данной статье приведен алгоритм поиска семантических дубликатов в коротких новостных сообщениях, в основу которого положена идея векторной модели семантики [7]. При таком подходе каждое новостное сообщение рассматривается как точка в многомерном пространстве.

Для оценки качества введены метрики, которые оценивают способность алгоритма классифицировать сообщения как семантические дубликаты и способность объединять найденные дубликаты в группы.

Установлено, что качество алгоритма сильно зависит от параметра кластеризации *r*. В статье предложены модели, которые позволяют прогнозировать параметр *r* на основе характеристик исследуемого корпуса текстов.

Разработанный алгоритм показал вполне приемлемое качество работы.

Предполагается, что улучшение качества работы алгоритма может быть достигнуто за счет использования методов, учитывающих контекст, например, word2vec и doc2vec [16].

Для практического применения предложенного алгоритма также требуется разработать метод оптимизации, который позволит сократить время работы алгоритма и уменьшить требования к памяти. В текущей реализации алгоритма требования ко времени и памяти растут как квадрат от числа сообщений в корпусе текстов. ■

## Литература

1. Волков Д., Гончаров С. Российский медиа-ландшафт: телевидение, пресса, Интернет. [Электронный ресурс] (дата обращения 14.10.2015).
2. Rangrej A., Kulkarni S., Tendulkar A.V. Comparative study of clustering techniques for short text documents // Proceedings of the 20th International World Wide Web Conference (WWW 2011). Hyderabad, India, 28 March – 01 April 2011. P. 111–112.
3. Errecalde M.L., Ingaramo D.A., Rosso P. A new AntTree-based algorithm for clustering short-text corpora // Journal of Computer Science and Technology. 2010. Vol. 10. No. 1. P. 1–7.
4. Petersen H., Poon J. Enhancing short text clustering with small external repositories // Proceedings of the 9th Australasian Data Mining Conference (AusDM'11). Ballarat, Australia, 01–02 December 2011. P. 79–89.
5. Кириченко К.М., Герасимов М. Б. Обзор методов кластеризации текстовой информации. [Электронный ресурс]: http://www.dialog-21.ru/en/digest/2001/articles/kirichenko/ (дата обращения 17.01.2017).
6. Зеленков Ю.Г., Сегалович И.В. Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Труды 9-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2007). Переславль-Залесский, Россия. 15–18 октября 2007 г. С. 166–174.
7. Turney P.D., Pantel P. From frequency to meaning: Vector space models of semantics // Journal of Artificial Intelligence Research. 2010. No. 37. P. 141–188.
8. Interpreting TF-IDF term weights as making relevance decisions / H.C. Wu [et al.] // ACM Transactions on Information Systems. 2008. Vol. 26. No. 3. P. 13.1–13.37.
9. Прикладная статистика. Классификация и снижение размерности / С.А. Айвазян и [др.]. М.: Финансы и статистика, 1989.
10. Воронцов К.К. Лекции по алгоритмам кластеризации и многомерного шкалирования. [Электронный ресурс]: http://www.ccas.ru/voron/download/Clustering.pdf (дата обращения 28.09.2016).
11. Hubert L., Arabie P. Comparing partitions // Journal of Classification. 1985. No. 2. P. 193–218.
12. Vinh N.X., Epps J., Bailey J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance // Journal of Machine Learning Research. 2010. No. 11. P. 2837–2854.
13. Соколов Е. Семинары по выбору моделей. [Электронный ресурс]: http://www.machinelearning.ru/wiki/images/1/1c/Sem06_metrics.pdf (дата обращения 17.01.2017).
14. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection // Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95). Montreal, Quebec, Canada, 20–25 August 1995. Vol. 2. P. 1137–1145.
15. Refaeilzadeh P., Tang L., Liu H. Cross-validation // Encyclopedia of Database Systems. Springer, 2009. P. 532–538.
16. Шурига Л. Современные методы анализа тональности текста. [Электронный ресурс]: http://datareview .info/article/sovremennyie-metodyi-analiza-tonalnosti-teksta/ (дата обращения 01.02.2017).