САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ — ПРОЦЕССОВ УПРАВЛЕНИЯ
КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Зайцев Андрей Алексеевич

Выпускная квалификационная работа бакалавра

Кластеризация с поиском дубликатов на примере патентов

Направление 010400

Прикладная математика, фундаментальная информатика

и программирование

Научный руководитель,
Старший преподаватель,
Уланов А. В.

Санкт-Петербург

2016

Содержание

Введение

Патент — документ, удостоверяющий исключительное право и авторство на изобретение. Патент содержит следующие данные:

- Библиографические данные (номер патента, дата подачи заявки, дата выдачи патента, категории и т.д.);
- Название;
- Описание изобретения;
- Патентную формулу;
- Чертежи;
- Аннотацию.

Предметом данной работы является поиск дубликатов в патентной базе с помощью кластеризации. Количество поданных заявок на патенты каждый год неуклонно увеличивается в соответствии с прогрессом в защите интеллектуальной собственности. Многие правительства и компании оформляют результаты своих исследований и разработанных устройств в виде патентов. Отделы научно-исследовательской деятельности постоянно анализируют базы патентов, чтобы отслеживать текущие тенденции и вектор развития новых технологий. Это позволяет корректировать исследовательскую политику и развивать приоритетные направления. Однако патенты содержат огромное количество технической и юридической терминологии, что затрудняет процесс анализа изобретения или технологии для тех, кто не знаком с данной областью. Необходимы простые методы для нахождения полезной информации среди такого количества документов. Классификация и кластеризация

являются популярными методами в анализе патентов. Техники анализа патентов базируются на структурированной информации, которая доступна в информации о патенте. Например, название, дата подачи заявки, аннотация, описание изобретения и многое другие.

Основная проблема при анализе патентов заключается в том, что они содержат большое количество данных, и, при использовании стандартных алгоритмов кластеризации, таких как метод K—средних [7], появляются проблемы, связанные с большой размерностью. Однако существуют различные методы для кластеризации данных большой размерности. В частности, в 2012 году был построен алгоритм кластеризации патентов, который базировался на Байесовском анализе [6]. Проблемой данного подхода является вычислительная сложность и сложность в подготовке и обработке данных, связанная с выбором распределения и функции правдоподобия.

В 2008 году был предложен новый метод визуализации для анализа патентов [21]. Данный метод извлекал из патентов слова, связанные с определённой технологией. После этого с помощью метода K-средних производилась кластеризация патентов. Далее, используя полученные кластеры, строилась семантическая сеть ключевых слов без использования данных о дате подачи заявки на патент. Затем формировалась карта патентов, в которой каждое ключевое слово перестраивалось в соответствии с наиболее ранней датой подачи заявки и частоты встречаемости данного термина в коллекции патентов.

Поиск дубликатов среди больших массивов данных также является большой проблемой. В 2007 году были рассмотрены различные методы для поиска дубликатов в базе данных, начиная от простых методов, таких как посимвольное сравнение, до более сложных, например, построение классификаторов [1].

Постановка задачи

Пусть $x^i \in G$ - патент, $X = \{x^1, x^2, ..., x^n\}$ — множество патентов, а $Y = \{y_1, y_2, ..., y_m\}$ — множество категорий патентов.

Определение 1 . *Будем говорить, что $p(\tilde{x}, \hat{x})$ является метрикой схожести двух патентов $\tilde{x}, \hat{x} \in X$, если:*

1. $p(\tilde{x}, \hat{x}) \geqslant 0$;
2. $p(\tilde{x}, \hat{x}) = 1$, *если* $\tilde{x} = \hat{x}$;
3. $p(\tilde{x}, \hat{x}) = p(\hat{x}, \tilde{x})$;
4. $p(\tilde{x}, \hat{x}) \in [0; 1]$.

Определение 2 . *Будем говорить, что $\hat{x}$ является дубликатом $\tilde{x}$, если $p(\tilde{x}, \hat{x}) \geqslant \beta$, где $\beta \in [0; 1]$ - произвольный параметр, задаваемый пользователем.*

Необходимо среди патентов из множества X найти дубликаты, т.е. построить алгоритм $\alpha(X) = \{(\tilde{x}, \hat{x}) \mid p(\tilde{x}, \hat{x}) \geqslant \beta\}$.

Основные определения

Определение 3 [9]. *TF(Term frequency) - отношение числа вхождения некоторого слова к общему количеству слов в документе:*

$$TF(t,d) = \frac{n_i}{\sum_k n_k},$$

где n_i — число вхождений i-ого слова в документ.

Определение 4 [9]. *IDF(Inverse document frequency) - инверсия частоты, с которой слово встречается в коллекции документов:*

$$IDF(t,D) = \frac{|D|}{|d \in D : t \in d|}$$

Определение 5 [9]. *Матрица документ-термин - матрица, описывающая частоту вхождения терминов в коллекции документов. Строки данной матрицы соответствуют документам, а столбцы - терминам:*

$$DTM = \begin{bmatrix} w_{t_1,d_1} & w_{t_2,d_1} & \dots & w_{t_m,d_1} \\ w_{t_1,d_2} & w_{t_2,d_2} & \dots & w_{t_m,d_2} \\ \dots & \dots & \dots & \dots \\ w_{t_1,d_n} & w_{t_2,d_n} & \dots & w_{t_m,d_n} \end{bmatrix},$$

где $w_{t,d} = TF(t,d) * IDF(t,D)$.

Определение 6 [9]. *Инвертированным индексом называется структура данных, в которой каждому термину соответствует список документов, в которых он встречается.*

Кластеризация с поиском дубликатов на примере патентов

Идея, которая лежит в основе поиска дубликатов при помощи кластеризации, состоит в том, что патенты, могут распределяться экспертами в несколько категорий. Используя алгоритм кластеризации хотим добиться разделения множества патентов на кластеры, чтобы в дальнейшем, сравнивая патенты в каждом кластере, а именно, сравнивая их категории, найти подозрительные на дубликат патенты и построить алгоритм для оценки их схожести с другими патентами в кластере.

В следующих главах будет построен метод для кластеризации патентов и алгоритм поиска дубликатов. Будут приведены несколько метрик для оценки качества кластеризации.

Предобработка данных

В данной работе для сбора данных использовался парсер, написанный на Python, который загружал архивы из Google Patents [19], разархивировал их и обрабатывал XML файлы, содержащие патенты. Для хранения полученных данных использовалась система управления базами данных (СУБД) MySQL. Для обработки данных использовался арендованный в Amazon сервер типа c4.4xlarge.

Для того, чтобы использовать данные, полученные из United States Patent and Trademark Office (USPTO) [18] и Google Patents [19] в алгоритме кластеризации, их необходимо обработать и перевести в нужный формат. Из каждого патента извлекается следующая информация: идентификационный номер

(ID) патента, дата подачи заявки, название, аннотация, описание, патентная заявка и категории. Текстовые данные разбиваются на отдельные слова, удаляются стоп-слова и пунктуация, а также производится лемматизация для построения матрицы документ-термин.

Для обработки текста использовалась библиотека Natural Language Toolkit (NLTK) [2], которая производит разбиение текста на отдельные слова. Для лемматизации использовалась база данных английских слов WordNet [20].

Кластеризация

Одним из основных и популярных методов кластеризации является метод K—средних [7]. Однако данный метод имеет ряд недостатков, которые не позволяют применять его напрямую для кластеризации патентов:

1. Необходимо загружать все данные в оперативную память, что в случае кластеризации патентов является критичным;
2. Метод чувствителен к выбору начальных точек кластеров;
3. Необходимо задавать число кластеров.

Для решения проблемы кластеризации данных большой размерности в 2000 году был предложен *Canopy clustering algorithm* [14]. Пусть $dist(\tilde{x}, \hat{x})$ — приближённая метрика в множестве X.

Определение 7 [14]. *Canopy называется такое множество элементов, у которого расстояние от центральной точки из этого множества до всех остальных точек в множестве не превышает некоторого значения, т.е.*

$$Canopy = \{x \mid dist(x_0, x) \leqslant T_1\},$$

где x_0 — центральная точка, а T_1 — некоторая константа.

Идея алгоритма заключается в том, чтобы разделить кластеризацию на 2 этапа. На 1 этапе множество патентов разбивается на несколько подмножеств, которые называются *Canopy*. При этом данное разбиение происходит с использованием приближённой метрики, которая вычисляется достаточно быстро. Данный процесс позволяет получить приближение к исходным кластерам. Рассмотрим алгоритм создания *Canopies*. Пусть $T_1 \geqslant T_2 > 0$, выберем патент из множества данных случайным образом, данный патент будет

являться центральной точкой *Canopy*, далее сравним расстояние от данного патента до всех остальных патентов в множестве, используя приближённую метрику. Тогда все патенты, которые оказались на расстоянии ближе, чем T_1, помещаются в данный *Canopy*, но при этом они всё ещё могут учавствовать в образовании новых подмножеств. Патенты, которые оказались на расстоянии ближе, чем T_2, не могут учавствовать в образовании новых подмножеств. Более формально алгоритм представлен в Приложении 1.

Следует заметить, что используя подобный алгоритм разбиения и упрощённую метрику, один и тот же патент может оказаться в нескольких *Canopy*. Однако из-за специфики патентов, а именно распределения патентов в нескольких различных категорий экспертами, будем считать, что данная особенность не создаёт проблем для кластеризации и отображает исходные данные.

Рассмотрим выбор приближённой метрики для задачи кластеризации патентов. Пусть $\tilde{x}, \hat{x} \in X, \tilde{x} = \{\tilde{w_1}, \tilde{w_2}, ..., \tilde{w_{k_1}}\}, \hat{x} = \{\hat{w_1}, \hat{w_2}, ..., \hat{w_{k_2}}\}$, где
$\tilde{w_i}, \hat{w_j}$ — термы, извлечённые из аннотации к патенту. Поместим оба патента в инвертированный индекс. Тогда будем вычислять приближённую метрику для алгоритма *Canopy clustering* как количество общих термов между патентами $\tilde{x}$ и $\hat{x}$, т.е.

$$dist(\tilde{x}, \hat{x}) = \sum_{w \in V(\tilde{x}, \hat{x})} \tilde{F}(w, \tilde{x}, \hat{x}),$$

где $V(\tilde{x}, \hat{x})$ — множество терминов, которые входят в оба патента,

$$\tilde{F}(w, \tilde{x}, \hat{x}) = \begin{cases} 1, & \text{если } F(w, \tilde{x}) = F(w, \hat{x}); \\ 0, & \text{в противном случае,} \end{cases}$$

где F(w, x) — количество вхождений терма w в патенте x.

Для использования данной метрики в Алгоритме 1 необходимо изменить условие добавления патента в *Canopy*. Будем помещать патент в *Canopy* в случае, если количество общих слов превышает T_1 и если количество общих слов больше T_2, то патент больше не будет принимать участие в образовании новых подмножеств.

После того, как получено разбиение на множества, на каждом множестве можно произвести независимую кластеризацию и объединить результаты. Для их кластеризации будет использоваться модификация метода К—средних [16]. Идея алгоритма состоит в использовании небольшого количества случайных объектов из оригинального множества X вместо всего множества X. Таким образом обеспечивается эффективное использование памяти. На каждой итерации алгоритм выбирает случайное подмножество документов фиксированного размера из X и использует их для минимизации целевой функции:

$$min \sum_{x \in X} \|f(C, x) - x\|^2,$$

где $f(C, x)$ — возвращает ближайший кластер для патента x.

Алгоритм представлен в Приложении 2.

Оценка качества кластеризации

Для оценки качества кластеризации будет использоваться ряд критериев, которые осуществляют проверку на кластеризованных данных без использования множества Y. Основной мотивацией данного решения является высказанное ранее предположение о том, что дубликаты могут находиться в разных категориях. Приведённые ниже критерии используют следующие критерии для оценки качества кластеризации:

- Компактность — показывает насколько близко расположены элементы в кластере;
- Разделённость — показывает насколько далеко находятся друг от друга кластеры.

Пусть $C = \{c_1, c_2, ..., c_k\}, c_k \in X$ - множество кластеров, которые были получены методом, описанным в главе 4.2, а $\tilde{x}_i$ — центральный элемент в кластере (центроид).

Критерий Davies–Bouldin

Критерий был представлен David L. Davies и Donald W. Bouldin в 1979 году [4] и вычисляется следующим образом:

$$BD = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \frac{d(x^i) + d(x^j)}{d(\tilde{x}_i, \tilde{x}_j)},$$

где $d(x^i)$ — расстояние от патента i до центрального элемента в кластере, а $d(\tilde{x}_i, \tilde{x}_j)$ — расстояние между двумя центроидами.

Критерий Dunn

Данный критерий был представлен J. C. Dunn в 1974 году [5] и вычисляется следующим образом:

$$Dunn = \frac{\min\limits_{1 \leqslant i \leqslant j \leqslant k} d(i,j)}{\max\limits_{1 \leqslant z \leqslant k} diam(z)},$$

где $d(i,j)$ — расстояние между кластерами i и j, а $diam(z)$ — диаметр кластера z.

Критерий Silhouette

Впервые данный критерий был представлен Peter J. Rousseeuw в 1986 году. Его можно вычислить следующим образом:

$$s_i^j = \frac{b_i^j - a_i^j}{\max\{a_i^j, b_i^j\}},$$

где a_i^j — средняя расстояние между i-м патентом в кластере c_j и остальными патентами из этого кластера, а b_i^j — минимальное среднее расстояние между i-м патентом в кластере c_j и остальными кластерами.

Таким образом критерий для кластера j можно вычислить так:

$$S_j = \frac{1}{|c_j|} \sum_{i=1}^{|c_j|} s_i^j$$

где $|c_j|$ — мощность кластера j.

И для всех кластеров данный критерий можно вычислить следующим образом:

$$S = \frac{1}{k} \sum_{j=1}^{k} S_j \tag{1}$$

Значения данного критерия лежат в промежутке от -1 до 1. Значения меньше 0 показывают, что патент возможно был отнесён к неправильному кластеру, значение 0 показывает, что кластеры находятся очень близко друг к другу, а значения близкие к 1 показывают, что кластеры находятся на значительном удалении. В дальнейшем будем использовать эти свойства для нахождения количества кластеров для метода K—средних. Целью является выбор такого числа k, чтобы кластеры находились на достаточном удалении друг от друга, т.е. значение данного критерия стремились к 1. В главе 5 будет показано, как с помощью данного критерия выбрать количество кластеров.

В качестве метрики для вычисления расстояния для всех приведённых критериев можно использовать Евклидову метрику.

Поиск дубликатов

Определение 8 . *Элемент $x \in c_i$ называется подозрительным на дубликат, если $y_i \neq \tilde{y}_i$, где y_i и $\tilde{y}_i$ - оригинальные категории патентов.*

Т.е. патент является подозрительным на дубликат, если его оригинальная категория отличается от оригинальной категории центрального элемента в кластере. Необходимо построить алгоритм, который с заданной точностью будет сравнивать 2 патента, и окончательно решать, являются ли они дубликатами.

В качестве развития векторной модели пространства была представлена относительная частотная модель [15]. Было показано, что косинусовое сходство не зависит от точного вхождения терма w_i в документ. Для исправления этого была введена новая метрика, которая строится следующим образом. Пусть $C(\tilde{x}, \hat{x})$ - множество, которое содержит термины w_i, встречающиеся в обоих документах одинаковое количество раз.

Тогда $w_i \in C(\tilde{x}, \hat{x})$, если

$$\varepsilon - (\frac{F_i(\tilde{x})}{F_i(\hat{x})} + \frac{F_i(\hat{x})}{F_i(\tilde{x})}) > 0,$$

где $\varepsilon \in [0; 1]$ — задаваемый параметр, который определяет отклонение,
$F_i(x)$ — количество вхождений w_i в документ x.

Введём меру сходства между общими термами в двух документах:

$$subset(\tilde{x}, \hat{x}) = \frac{\sum_{w_i \in C(\tilde{x}, \hat{x})} F_i(\tilde{x}) F_i(\hat{x})}{\sum_{i=1}^{k} F_i^2(\tilde{x})}$$

Данная мера необходима для выявления случая, когда один документ полностью содержится в другом. Этой проблеме подвержено косинусовое сход-

ство, которое даёт крайне малые значения для подобного случая. Таким образом сходство между двумя документами определяется следующим образом:

$$sim(\tilde{x}, \hat{x}) = \min\{1, \max\{subset(\tilde{x}, \hat{x}), subset(\hat{x}, \tilde{x})\}\}$$

Введём параметр $\delta \in [0; 1]$.

Определение 9 . *Будем говорить, что $\hat{x}$ является дубликатом $\tilde{x}$ с точностью ε, если $sim(\tilde{x}, \hat{x}) \geqslant \delta$.*

Для вычисления значений $F_i(x)$ можно воспользоваться инвертированным индексом. Использование инвертированного индекса потребует шага инициализации, в котором оба документа будут обработаны и добавлены в индекс, но затем можно будет вычислять $F_i(x)$ за время $\mathcal{O}(1)$.

Эксперимент

Инициализация

В качестве рабочего множества будет рассматриваться множество из 490637 патентов за 2011-2015 года, которые были загружены из Google Patents [19]. Полученные данные были обработаны — из каждого патента была извлечена информация об его идентификационном номере, название, аннотация, описание, патентная формула и категории. Для хранения полученных данных используется система управления базами данных (СУБД) MySQL.

Для *Canopy clustering algorithm* необходимо определить константы T_1 и T_2. Для этого было произведено сравнение 5000 случайно выбранных патентов, используя метрику, построенную в главе 4.2. Среднее количество общих слов между аннотациями к патентами составило 17, а максимальное — 82.

Как можно заметить на Рис. 1, большинство патентов имеет мало общих признаков друг с другом. Экспериментальным путём было выяснено, что для того, чтобы разбить патенты на подмножества с достаточным количеством элементов, необходимо взять $T_1 = 20$, а $T_2 = 40$.

Далее необходимо определить количество кластеров k, которые будут использоваться в методе K-средних. Для этого воспользуемся Silhoutte analysis [12] и Elbow method [17]. В главе 4.4 был введён критерий Silhoutte, который лежит в основе Silhoutte analysis. Для того, чтобы определить количество кластеров, произведём кластеризацию, используя в качестве k значения от 1 до 500. По окончанию кластеризации для каждого значения k вычисляется Silhoutte index и отображается график зависимости Silhoutte index от k. Точками интереса являются значения k при которых Silhoutte index принимает

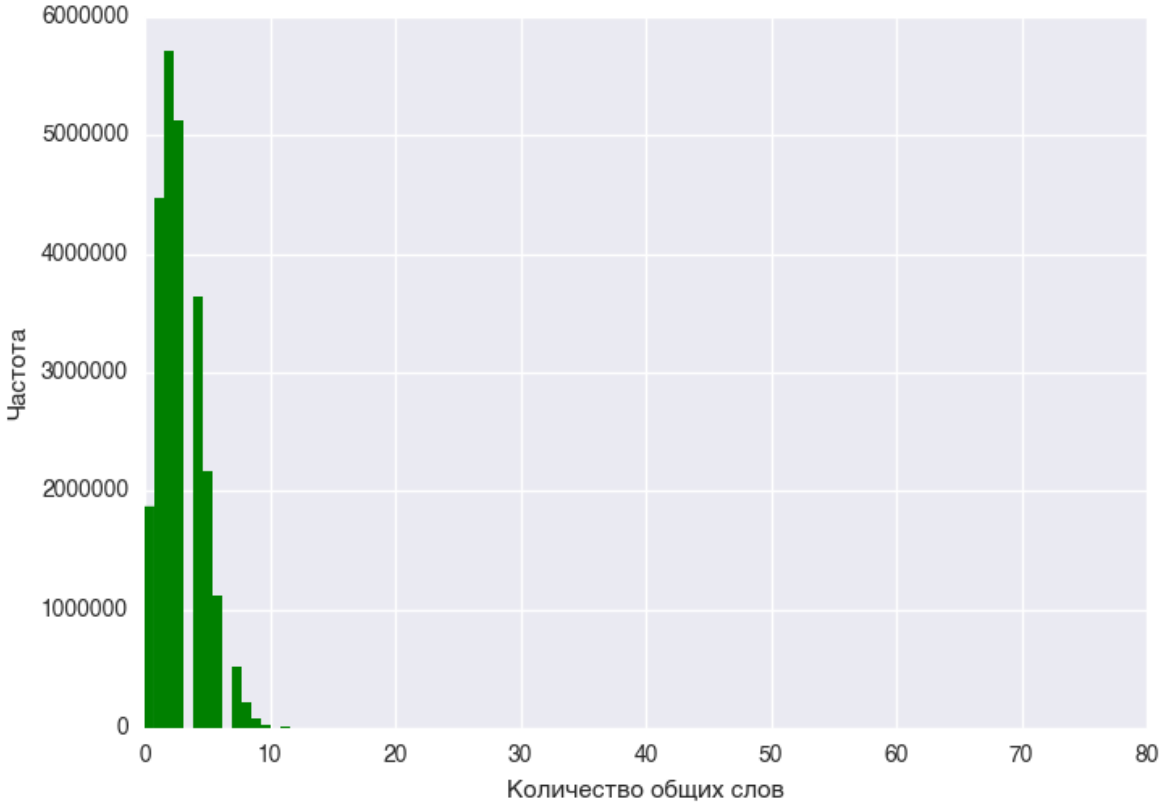

Рис. 1: Гистограмма количества общих слов между аннотациями к патентам

значения близкие к 1 и при этом наблюдается излом кривой, что свидетельствует о том, что при увеличении числа k качество кластеризации практически не улучшится.

Аналогичный анализ проводится при применении Elbow method. Точками интереса являются такие значения k при которых происходит сильный излом кривой и средняя сумма квадратов увеличивается, что позволяет говорить о том, что при увеличении числа k качество кластеризации практически не улучшится.

На Рис. 3. и Рис. 4. видно, что достаточно $k \in [350; 400]$ для данной задачи, т.к. в данном интервале происходит перелом кривой и на следующем шаге качество кластеризации не улучшается.

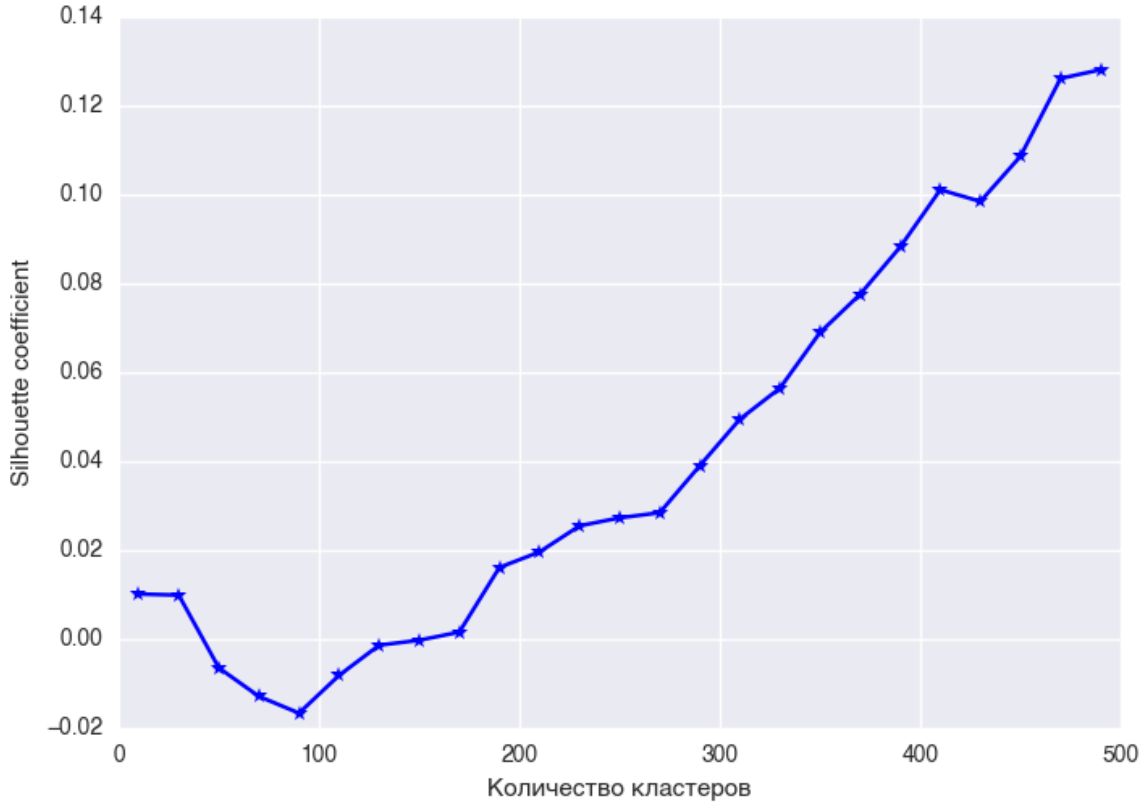

Рис. 2: Зависимость Silhouette coefficients от k

Кластеризация

Выберем из множества патентов 100000 случайных патентов. Будем кластеризировать данное множество патентов. Рассмотрим 500 случайных патентов из данного множества, чтобы понять как распределены данные. Для отображения используется алгоритм t-SNE [8]. Как мы видим на Рис. 4 данные расположены достаточно плотно и нет участков, похожих на отдельные кластеры.

Произведём кластеризацию 100000 патентов и снова выберем 500 случайных патентов, которые спроецируем на двухмерную плоскость с помощью алгоритма t-SNE. После первого этапа кластеризации было получено 5758 под-

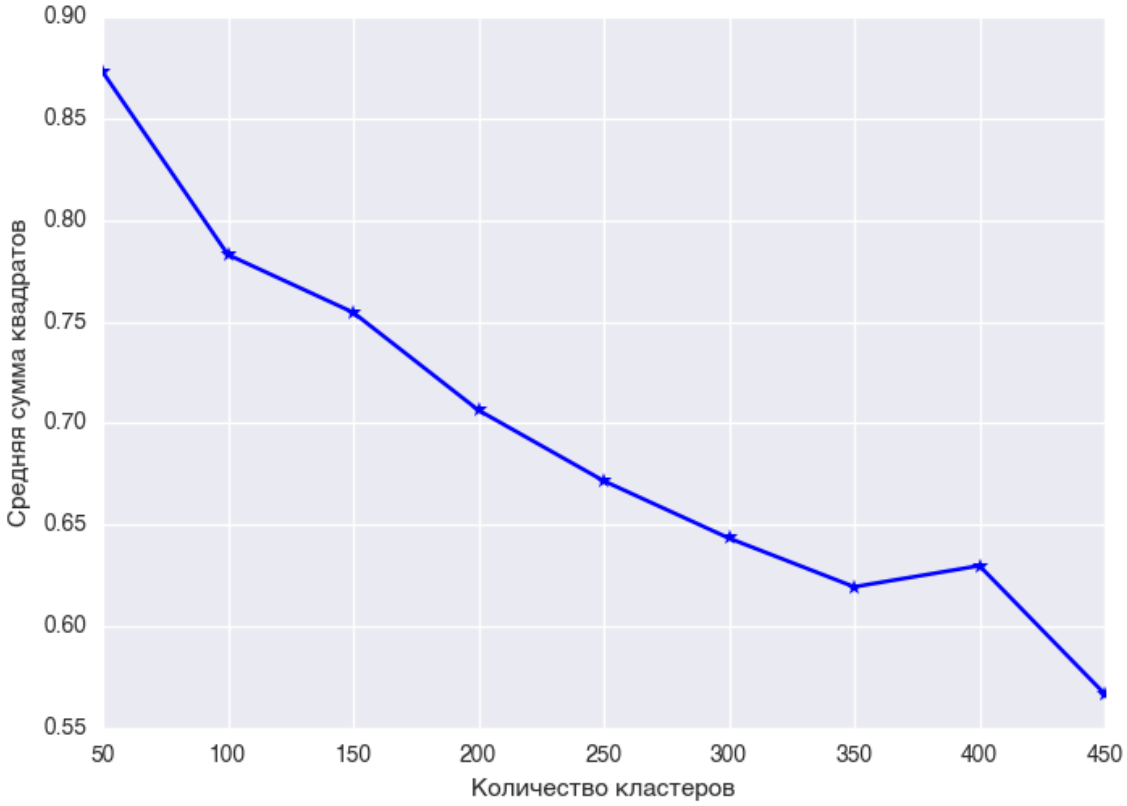

Рис. 3: Elbow method

множеств, называемых Canopy. Наименьшее из этих подмножеств содержит 3 патента, наибольшее — 1244. Далее данные подмножества были независимо кластеризированы, используя модифицированный алгоритм K—средних. Было получено 11127 кластеров. Значения каждого из критериев оценки качества кластеризации представлены в следующей таблице:

Davies-Bouldin	0.416218
Dunn	0.980366
Silhoutte	0.555022

Значения Silhoutte index 0.555022 показывает, что было получено разбиение на кластеры, которые находятся на достаточном расстоянии друг от друга.

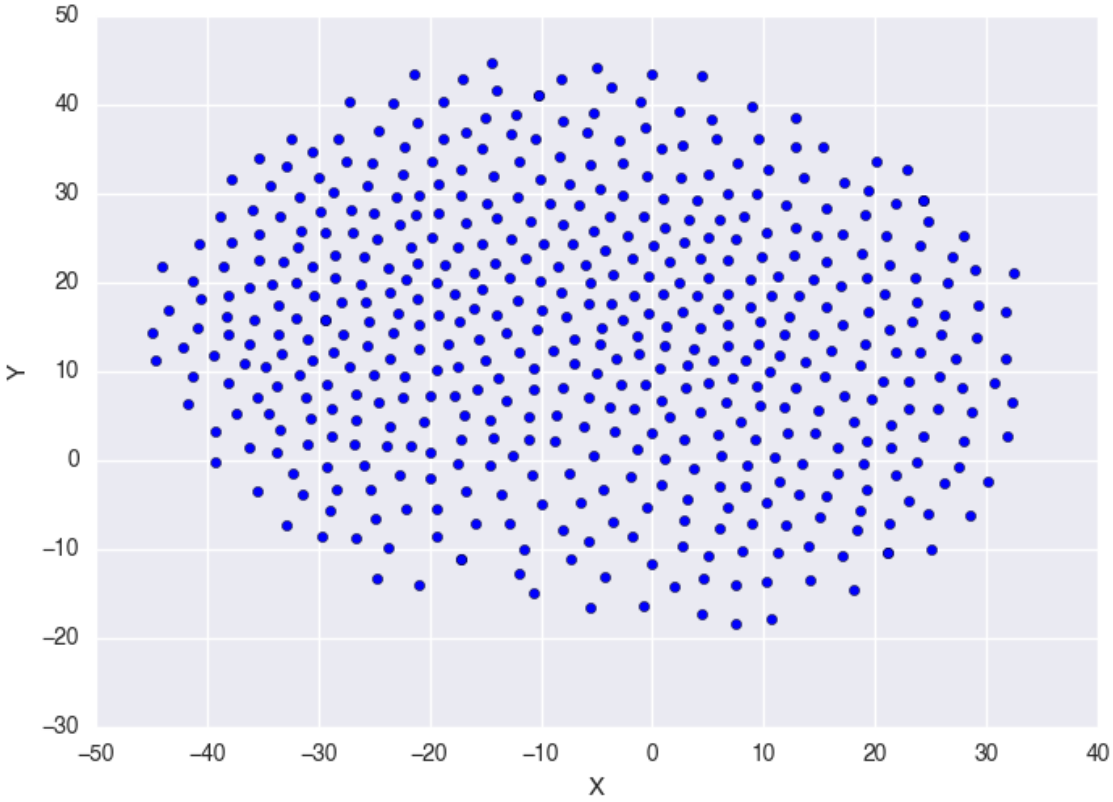

Рис. 4: Проекция 500 патентов на двухмерную плоскость

Используя полученное разбиение на кластеры можно отобразить 500 патентов, выбранных ранее, с помощью цветов для разных кластеров. Данные патенты содержатся в 30 различных кластерах. На Рис. 5 представлена их проекция на двухмерную плоскость с помощью алгоритма t-SNE после кластеризации.

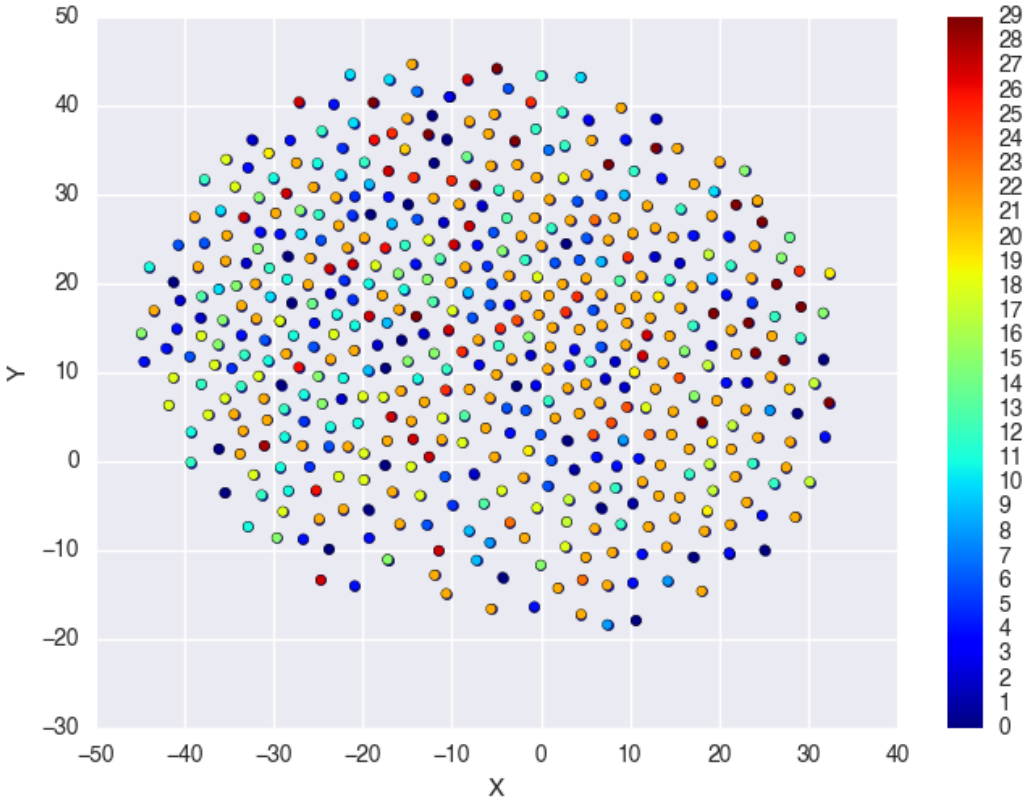

Рис. 5: Проекция 500 патентов на двухмерную плоскость

Поиск дубликатов

В качестве ε и δ для алгоритма поиска дубликатов взяты значения $\varepsilon = 2.01$ и $\delta = 0.6$. Используя данные, полученные после кластеризации, и алгоритм из главы 4.5. удалось найти 970 пар патентов, подозрительных на дубликат. Минимальная мера схожести среди найденных пар равна 0.6, максимальная — 0.82.

Часть полученных данных были проанализированы вручную. В частности оказалась интересной пара патентов US 20140165327 (Canister vacuum cleaner, дата подачи заявки 13 декабря 2013 года) - US 20140101888 (Canister vacuum cleaner, дата подачи заявки 16 октября 2013 года). Мера схожести двух патен-

тов оказалась равна 0.82. US 20140165327 находится в категориях A47L9/24, A47L5/36, в то время как US 20140101888 в категориях A47L9/24, A47L9/10, A47L5/36. Именно категория A47L9/10 повлияла на проверку схожести между этими патентами. Если мы обратимся к патентной базе, то можем заметить, что описание данных патентов и чертежи схожи. К сожалению, для более глубокой оценки и точного утверждения, что данные патенты являются дубликатами, необходимо привлечение экспертов в патентной области.

Заключение

В данной работе удалось построить алгоритм кластеризации, который использовался как составная часть алгоритма для поиска дубликатов в базе патентов. Было показано, что задачу поиска дубликатов можно решить используя алгоритм кластеризации. Было проанализировано 100000 патентов, полученных с помощью сбора данных из Google Patents. Найдены патенты, подозрительные на дубликат и вычислены оценки схожести между парами оригинал—дубликат.

В качестве направлений дальнейших исследований отметим задачу нахождения оценки точности полученного алгоритма, например, с помощью проверочного множества, разработанного экспертами в патентной области, а также разработку поисковой системы, используя результаты кластеризации.

Список литературы

1. Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, Vassilios S. Verykios. Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data // Engineering, Volume 19 Issue 1, January 2007

2. Bird, Steven, Edward Loper and Ewan Klein, Natural Language Processing with Python. // O'Reilly Media Inc., 2009

3. Christopher I., Lin S., Spieckermann S., Automated Patent Classification

4. Davies, David L.; Bouldin, Donald W, A Cluster Separation Measure // IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-1 (2): p. 224–227, 1979.

5. Dunn, J. C., A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters // Journal of Cybernetics 3 (3): p. 32–57, 1973.

6. Jun S., A Clustering Method of Highly Dimensional Patent Data Using Bayesian Approach // IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012.

7. Lloyd S. Least square quantization in PCM's // Bell Telephone Laboratories Paper. 1957.

8. L.J.P. van der Maaten, G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. // Journal of Machine Learning Research 9 November, p. 2579-2605, 2008.

9. Manning C. D., Raghavan P., Schutze H. Introduction to Information Retrieval // "Scoring, term weighting, and the vector space model". p. 100

10. Medvedev T., Ulanov A., Company Names Matching in the Large Patents Dataset // HP Laboratories HPL-2011-90R1

11. McCallum, A., Nigam, K., Ungar L.H. "Efficient Clustering of High Dimensional Data Sets with Application to Reference Matching"// Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, p. 169-178. 2000.

12. Rousseeuw P. Jr., Silhouettes: A graphical aid to the interpretation and validation of cluster analysis // Journal of Computational and Applied Mathematics, Volume 20, p 53-65, November 1987.

13. Salton G., Wong A., Yang C. S., A Vector Space Model for Automatic Indexing // Communications of the ACM, vol. 18, nr. 11, p. 613–620.

14. Sharma A., A Survey On Different Text Clustering Techniques For Patent Analysis // International Journal of Engineering Research & Technology. 2012.

15. Shivakumar N., Garcia-Molina H. SCAM: A Copy Detection Mechanism for Digital Documents // 2nd International Conference in Theory and Practice of Digital Libraries (DL 1995), June 11-13, 1995.

16. Sculley D., Web-scale k-means clustering // WWW '10 Proceedings of the 19th international conference on World wide web, p. 1177-1178. 2010.

17. Trupti M. Kodinariya, Dr. Prashant R. Makwana, Review on determining number of Cluster in K-Means Clustering // International Journal of Advance

Research in Computer Science and Management Studies, Volume 1, Issue 6, November 2013.

18. United States Patent and Trademark Office http://www.uspto.gov

19. United States Patent and Trademark Office Bulk Downloads https://www.google.com/googlebooks/uspto-patents-grants-text.html

20. WordNet https://wordnet.princeton.edu/

21. Young Gil Kim, Visualization of patent analysis for emerging technology // Expert Systems with Applications, Volume 34, Issue 3, p. 1804–1812, April 2008.

Приложение

Приложение 1

Алгоритм разбиения множества патентов на независимые подмножества.

Algorithm 1 Разделение множества X на *Canopies*

Входные данные: X, $dist(\tilde{x}, \hat{x})$, T_1, T_2

```
canopies ← {}
points ← X
for each p1 in points do
    canopy ← {p1}
    points.remove(p1)
    for each p2 in points do
        distance ← dist(p1, p2)
        if distance > T1 then
            continue
        end if
        canopy.add(p2)
        if distance < T2 then
            points.remove(p2)
        end if
    end for
end for
return canopies
```

Приложение 2

Модифицированный алгоритм K—средних.

Algorithm 2 Mini-batch K-Means

Входные данные: k, b размер случайного множества, t количество итераций, X

Инициализировать $c \in C$ с помощью случайного выбранного $x \in X$

$v \leftarrow 0$

for $i \leftarrow 1$ to t **do**

$M \leftarrow b$ документов, выбранных из X

for $x \in M$ **do**

$d[x] \leftarrow f(C, x)$

end for

for $x \in M$ **do**

$c \leftarrow d[x]$

$v[c] \leftarrow v[c] + 1$

$\eta \leftarrow \frac{1}{v[c]}$

$c \leftarrow (1 - \eta)c + \eta x$

end for

end for