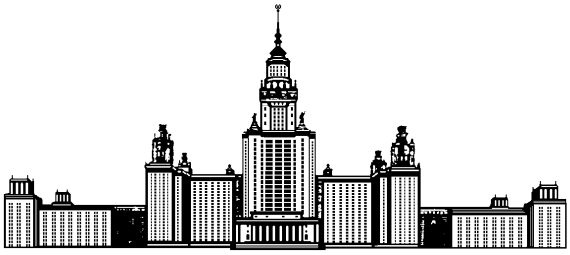

Московский государственный университет
имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Магистерская программа
«Логические и комбинаторные методы анализа данных»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Мультиметрические методы информационного поиска

Работу выполнил:
Сендерович Никита Леонидович

Научный руководитель:
к.ф-м.н.
Майсурадзе Арчил Ивериевич

Москва, 2017

Оглавление

Глава 1

Введение

Метрические методы широко применяются при решении задач распознавания образов. Поиск оптимальной в некотором смысле функции расстояния в пространстве анализируемых объектов является одной из фундаментальных задач анализа данных. На основе адекватно подобранных метрик строится модель пространства, позволяющая эффективно решать задачи классификации, кластеризации, ранжирования и другие. Методы, основанные на поиске ближайших соседей, и алгоритм k средних — классические примеры метрических подходов, успешно используемых в большом числе различных приложений.

Эффективный информационный поиск на основе сходства становится всё более важным инструментом связи с быстрым ростом объёмов хранимой и обрабатываемой информации. Как правило, поиск по точному совпадению во многих случаях не может удовлетворить запросам пользователя. Функции расстояния должны учитывать семантическое сходство запроса и кандидатов. Для задач информационного поиска в массивах неструктурированной информации различных видов (текстов, изображений, аудио, видео) разработано и разрабатывается большое число различных функций сходства между объектами, позволяющих учесть специфику решаемых задач и сложную структуру сопоставляемых сущностей [18], [14]. При этом зачастую использование отдельно взятой функции расстояния может быть недостаточным по двум причинам:

1. она учитывает лишь один аспект близости сопоставляемых объектов, не используя всю доступную информацию

2. пространство распознаваемых объектов неоднородно, и при вычислении функции сходства необходимо специальным образом учитывать область пространства, в которой находятся сопоставляемые объекты

К примеру, в задачах полнотекстового поиска по запросу стандартной функцией сходства является число слов запроса, встретившихся в документе. Однако релевантные запросу документы могут вовсе не содержать большинства слов запроса. В этом случае необходимо обогащать модель поиска путём учёта тематического сходства текстов запроса и документа. Неоднородность пространства объектов характерна для задач классификации изображений. В некоторых случаях использование разных метрик для различения разных типов изображений позволяет достичь более высоких результатов [12].

В этой связи востребованным является исследование различных способов комбинирования метрик, позволяющих компенсировать недостатки отдельно взятых функций сходства за счёт более полного использования пространственной и семантической метрической информации.

Другой важной задачей является собственно конструирование оптимальных функций для решения новых задач распознавания. Зачастую построение функций расстояния для решения новых задач осуществляется исследователями вручную путём эвристического поиска. Как правило, использование функций близости, подобранных с учётом экспертных знаний, позволяет достичь высоких показателей качества, однако процесс подбора весьма трудозатратен и не даёт гарантий оптимальности. Одной из целей данной работы является исследование способов автоматизированного построения функций расстояния на основе обучающих данных, позволяющих достичь наилучшего качества.

В рамках данной работы основное внимание уделяется применению метрических и мультиметрических подходов к решению прикладных задач информационного поиска в текстовых коллекциях. Особенностями текстовых данных являются сложная внутренняя структура и многомерность объектов распознавания, возможность использования различных представлений для вычисления функций сходства. При измерении степени похожести коротких текстов проблемой является отсутствие статистической информации о встречаемости слов, что приводит к необходимости использования дополнительных источников информации: экспертных знаний, семантических сетей, знаний, извлечённых из дополнительных корпусов данных.

Дальнейшее изложение построено по следующему плану. В главе 2 даётся общий обзор метрических и мультиметрических моделей и методов, дающий необходимую для дальнейшего исследования теоретическую базу. В главах 3 и 4 рассматриваются две прикладные задачи информационного текстового поиска и производится эмпирическое сравнение метрических методов на реальных данных. В частности, в главе 3 рассматривается задача классификации текстов, а в главе 4 рассматривается задача построения вопросно-ответной системы. Глава 5 содержит основные выводы и результаты, полученные в ходе исследования.

Глава 2

Мультиметрические методы анализа данных

2.1 Метрические пространства и задачи распознавания

Приведём общие соображения, исходя из которых при решении задач распознавания применяются метрические методы.

Рассматривается пространство объектов распознавания Ω с заданной на нём функцией расстояния $d(x, y)$, определённой для любой пары объектов. Пусть имеется размеченная обучающая выборка $X_{train} = \{(x_i, y_i)\}_{i=1}^{n}$, $x_i \in \Omega$, $y_i \in \{1, \ldots, K\}$ для задачи классификации, $y_i \in \mathbb{R}$ для задачи регрессии. Тогда для тестового объекта x с помощью некоторого метрического алгоритма a (например, алгоритма k ближайших соседей), исходя из подобранной функции расстояния и разметки обучающих объектов, восстанавливается метка $y = a(x)$. При этом чем лучше подобрана функция d, тем выше качество распознавания.

В случае необходимости применить данный подход на новой задаче, для подбора функции d можно идти следующими путями:

- экспертно выбирать d из набора известных функций, определённых в пространстве Ω
- подобрать функцию расстояния по обучающей выборке, используя некий алгоритм b для обучения функции расстояния $d = b(X_{train})$, принадлежащей некоторому параметрическому семейству функций расстояния

К достоинствам первого пути можно отнести простоту реализации, интерпретируемость и высокое качество результатов при достаточно тщательном подборе. Недостатками являются трудоёмкость, отсутствие гарантий оптимальности и отсутствие учёта информации о разметке. Последнее ограничение является весьма существенным, рассмотрим следующий пример, изображённый на рис. 2.1. Для изображений такого типа можно рассматривать две различных содержательных задачи распознавания:

- классификация изображения для определения поворота головы (анфас или вполоборота)

Рис. 2.1: Пример набора данных, для которого функция расстояния зависит от поставленной задачи, иллюстрация из [17]

- идентификация человека на изображении

Очевидно, что при метрическом подходе использовать одну и ту же функцию расстояния в обеих задачах нельзя, требуется дополнительный подбор. При тех же самых объектах распознавания постановка задачи отражается в информации о разметке, которую эффективно использует второй путь построения функции расстояния. Однако необходимо иметь в виду, что алгоритмы обучения метрик требуют достаточного количества обучающих данных, их специальной подготовки, дополнительной настройки алгоритма.

Классическим случаем, рассматриваемым всюду в этой главе, является следующий вариант метрического пространства:

1. $\Omega = \mathbb{R}^d$, каждый объект задаётся своим признаковым описанием в d-мерном евклидовом пространстве

2. функция расстояния — обобщённая евклидова метрика с неотрицательно определённой матрицей A:

$$d_A^2(x, y) = (x - y)^T A(x - y) \tag{2.1}$$

   В англоязычной литературе метрика заданного вида называется также *метрикой Махаланобиса* или *обобщённой метрикой Махаланобиса*.

Как правило, при поиске оптимальной функции расстояния d в Ω отталкиваются от некой исходной функции d_{init}. Часто подбирают только преобразование ψ исходного пространства:

$$d(x, y) = d_{init}(\psi(x), \psi(y)) \tag{2.2}$$

Отметим некоторые простые свойства обобщённой евклидовой метрики:

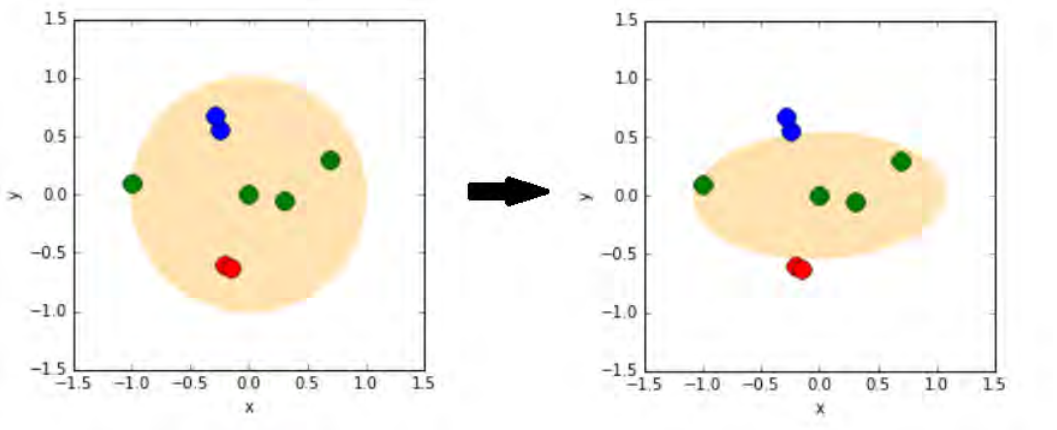

Рис. 2.2: Пример перехода от стандартной евклидовой метрики к обобщённой

- при $A = I$ получается стандартная евклидова метрика
- при $A = diag(w_1, w_2, \ldots, w_n)$, при $w_i \geqslant 0$ получаем взвешенное евклидово расстояние
- в силу неотрицательной определённости матрица A допускает разложение Холецкого и спектральное разложение:

  $$A = U^T U = Q^T \Lambda Q,$$

  при этом обобщённая евклидова функция принимает вид:

  $$d_A^2(x, y) = ||\psi(x) - \psi(y)||^2, \text{ где } \psi(x) = Ux = \left(Q\Lambda^{\frac{1}{2}}\right) x$$

Таким образом, использование обобщённой евклидовой функции соответствует линейному преобразованию ψ исходного пространства (рис. 2.2). Важно отметить, что линейно неразделимые данные после трансформации остаются линейно неразделимыми. Тем не менее, линейные преобразования пространства представляют собой достаточно богатый класс, удобны для аналитического исследования и позволяют строить мощные алгоритмы распознавания.

2.2 Метрический тензор и его оценка

Рассмотрим теперь один из вариантов наиболее общей теоретической постановки задачи моделирования метрического пространства, для того чтобы далее рассматривать с общих позиций частные случаи её решения, применимые на практике.

Свяжем с каждой точкой пространства x свою неотрицательно определённую матрицу $A_x = MT(x)$. Таким образом в пространстве $\mathbb{R}^n$ определён *метрический тензор*. Неформально говоря, можно считать, что в окрестности каждой точки обобщённое евклидово расстояние считается с использованием своей матрицы A. Определив метрический тензор, введём расстояние между точками x и y вдоль

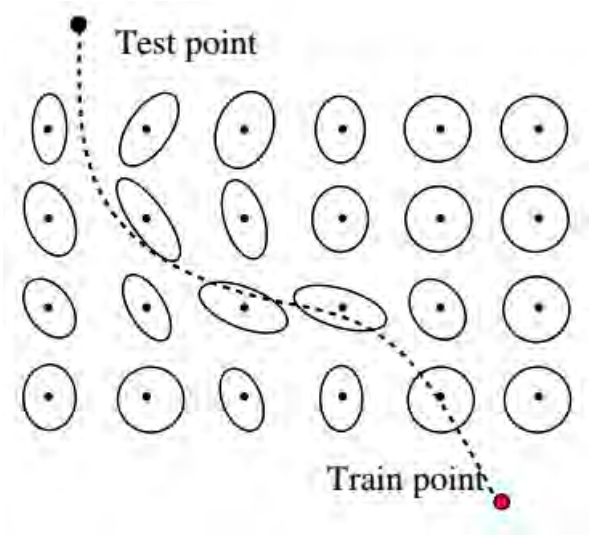

Рис. 2.3: Иллюстрация геодезического расстояния между двумя точками, иллюстрация из [23]

параметрически заданной кусочно дифференцируемой кривой $\mathbf{c}(\lambda), \lambda \in [0,1]$ следующим образом:

$$L(c) = \int_0^1 \left[\frac{d\mathbf{c}}{d\lambda}\right]^T MT(\mathbf{c}(\lambda))\frac{d\mathbf{c}}{d\lambda}d\lambda \tag{2.3}$$

Далее, естественно определить расстояние между парой точек как минимальное из расстояний вдоль всевозможных кривых, соединяющих эти точки:

$$D_{geo}(x,y) = \min_{\mathbf{c}(0)=x,\mathbf{c}(1)=y} L(\mathbf{c}(\lambda)) \tag{2.4}$$

Эта функция называется *геодезическим расстоянием*. Нетрудно видеть, что для геодезического расстояния определены аксиомы псевдометрики.

Задачу оценки метрического тензора можно отождествить с задачей поиска функции расстояния в пространстве распознаваемых объектов: зная метрический тензор, для вычисления расстояний между объектами можно использовать геодезическое расстояние. В общем случае задача вычисления геодезического расстояния сложна, на практике, как правило, имеют дело с частными случаями метрического тензора:

- во всех точках пространства метрический тензор является константой — в этом случае расстояние определяется, достаточно оценить матрицу A
- группе объектов обучающей выборки X_i сопоставляется матрица A_i

Последний вариант является примером мультиметрической модели, учитывающей неоднородность пространства распознаваемых объектов. При выборе разбиения обучающей выборки на группы X_i широко используются следующие варианты:

- группы соответствуют классам (для задач классификации)

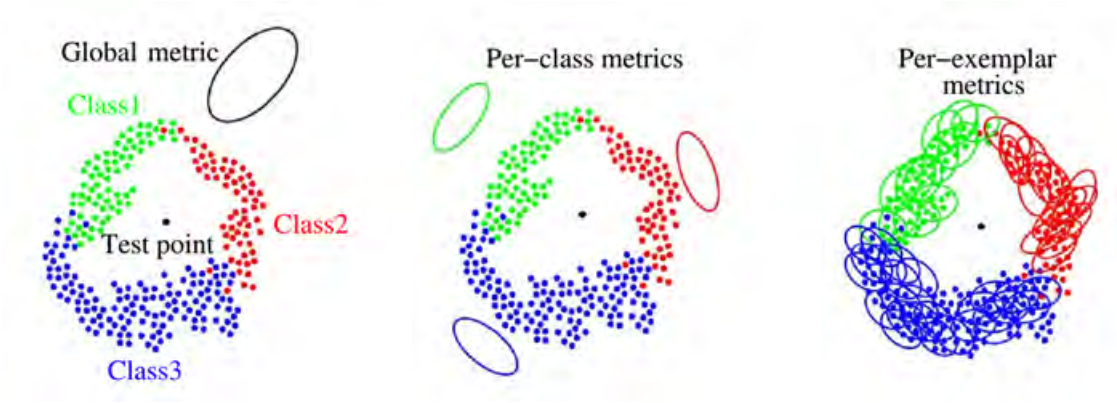

Рис. 2.4: Варианты обучения с различным числом метрик для задачи классификации, иллюстрация из [23]

- группы соответствуют некоторым образом выделенным кластерам обучающей выборки
- каждый объект относится к своей группе

Иллюстрация описанных ситуаций приведена на рис. 2.4

Сразу же поясним, как при наличии нескольких функций расстояния может производиться оценка расстояний от тестового объекта y до обучающих объектов. Распространённой является следующая схема: при вычислении расстояния до объекта x_i используется соответствующая ему метрика d_{A_j}. Таким образом, вычисление расстояния происходит в предположении, что y находится в окрестности объекта x_i.

В работе [23] предложено ещё несколько вариантов, включая «онлайн»-стратегию, при которой метрика для тестового объекта оценивается по достаточному числу обучающих объектов в его окрестности, а также метод для получения оценки сверху на геодезическое расстояние при движении вдоль прямой с учётом диаграммы Вороного.

Отметим, что при использовании нескольких метрик для оценки расстояния до объектов обучающей выборки полученные значения расстояний, вообще говоря, не находятся в одной и той же шкале. Некоторые мультиметрические методы (например, mmLMNN, описанный в разделе 2.4.1) автоматически решают данную проблему в ходе процесса оптимизации. Более общий подход для решения этой проблемы состоит в предварительной нормализации матриц, определяющих обобщённое евклидово расстояние. Одним из вариантов, является нормализация следа: деление всех значений в матрице A на её след.

Перейдём теперь к вопросу о методах построения обобщённых евклидовых расстояний по обучающим данным.

2.3 Базовые методы построения метрик

Сперва рассмотрим некоторые простейшие методы оценки обобщённого евклидова расстояния по обучающим данным.

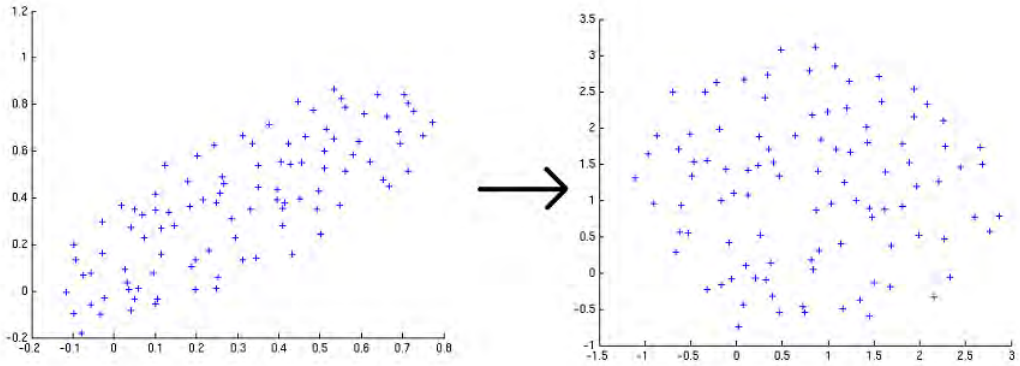

Рис. 2.5: Преобразование пространства при использовании метрики Махаланобиса

2.3.1 Метрика Махаланобиса

Простейшим примером такой функции расстояния является метрика Махаланобиса в оригинальной трактовке [20], т.е. оценка матрицы A через ковариацонную матрицу обучающей выборки:

$$d^2_{Mahalanobis}(x,y) = (x-y)^T\Sigma^{-1}(x-y), \text{ где } \Sigma = \frac{1}{N-1}\sum_i (x_i-\mu)(x_i-\mu)^T \quad (2.5)$$

Геометрически использование метрики Махаланобиса соответствует линейному преобразованию пространства, приводящему данные к единичной матрице ковариации, см. рис. 2.5.

Отметим, что данный метод никак не учитывает разметку. Также отметим, что на практике матрица ковариации может оказаться вырожденной или близкой к вырожденной, поэтому при её обращении необходимо следить за численной устойчивостью. Простейшим способом избежать ошибок является регуляризация: добавка небольшого положительного значения λ ко всем диагональным элементам выборочной матрицы ковариации.

2.3.2 Методы понижения размерности

Как было показано в разделе 2.1, любое линейное преобразование исходного пространства $x' = Bx$ соответствует использованию матрицы $A = B^TB$ при вычислении расстояния между исходными векторами. Ещё одной базовой идеей построения обобщённой евклидовой метрики является использование линейного преобразования, соответствующего переходу к подпространству меньшей размерности. Геометрически это означает, что после преобразования мы вычисляем евклидово расстояние на некой гиперплоскости.

Для отыскания преобразования могут быть использованы следующие широко известные методы:

1. метод главных компонент [26] и его вариации
2. метод сингулярного разложения

Отметим, что это семейство методов, как и использование матрицы ковариации, не позволяет использовать информацию о разметке.

2.3.3 Линейный дискриминантный анализ

Ещё одно семейство методов построения матрицы A связано с линейным дискриминантным анализом [28]. Для решения задачи классификации методы ЛДА предполагают нормальное распределение и равенство ковариационных матриц для объектов всех классов. В этих предположениях ЛДА позволяют оценить общую матрицу ковариации, которую затем можно использовать, как в метрике Махаланобиса.

Геометрически методы ЛДА позволяют произвести преобразование пространства, в котором внутриклассовые расстояния минимизируются, тогда как межклассовые расстояния максимизируются.

В отличие от двух предыдущих семейств метрик, методы ЛДА существенно опираются на разметку классов.

2.4 Методы обучения метрик

За последние полтора десятилетия было предложено большое число новых подходов к задаче оценки метрического тензора в пространстве по обучающей выборке, позволяющих учитывать различные виды информации о разметке. В данном разделе будут описаны несколько семейств моделей обучения метрик, использованные для целей исследования. Подробные обзоры можно также найти в работах [17], [1].

В качестве данных для обучения метрик могут выступать:

1. разметка обучающих объектов на классы
2. пары похожих (S) и непохожих объектов (D)
3. тройки: объект x похож на объект z больше, чем y

Аналогичные соображения закладываются в ограничения на искомые метрики, используемые в моделях обучения:

- $d_A(x_i, x_j) \leqslant l$ для похожих объектов и $d_A(x_i, x_j) \geqslant u$ для непохожих объектов
- для тройки (x_i, x_j, x_k) похожие объекты должны быть значимо ближе друг к другу, чем непохожие $d_A(x_i, x_j) \leqslant d_A(x_i, x_j) - m$
- близкие объекты должны быть близко в сумме:

$$\sum_{(i,j)\in S} d_A(x_i, x_j) \leqslant 1$$

Ещё одной важной составляющей модели являются регуляризаторы $r(A)$ для матрицы A обобщённой евклидовой метрики. Перечислим широко применяемые регуляризаторы:

- $tr(A)$ — аналогичен l_1-регуляризации, ведёт к матрице A низкого ранга
- $||A||_F^2$ — аналог l_2 регуляризации
- $tr(A) - \log\det(A) = \sum_i(\lambda_i - \log\lambda_i)$
- $tr(AC)$ — регуляризатор специального вида

Комбинирование регуляризаторов и накладываемых на искомую метрику ограничений позволяет формулировать метрические модели с различными свойствами. Общий вид модели такого рода следующий:

$$r(A) + \lambda \sum_{m=1}^{M} c_i(A) \longrightarrow \min_A \tag{2.6}$$

Опишем далее ряд моделей, в разной степени соответствующих данному представлению.

2.4.1 Модель Large Margin Nearest Neighbor

Предложенная в работе [33] модель LMNN не является исторически первой, в которой были предложены основные идеи обучения метрик (до неё были работы [27] [19]). Однако эта модель зарекомендовала свою практическую применимость во многих прикладных задачах и по сей день она остаётся популярной среди исследователей — в частности, с ней часто сравнивают новые методы.

Основная идея модели состоит в следующем. Пусть для каждого объекта i фиксированы K ближайших соседей из того же класса (обозначаемые переменной j), а индекс k пробегает по всем объектам других классов. Потребуем, чтобы в обучаемой метрике близкие объекты оказывались близко друг к другу, а далёкие — значимо дальше. Эта идея формализуется следующей оптимизационной задачей:

$$\sum_{(i,j)\in S} d_A(x_i, x_j) + \lambda \sum_{(i,j,k)\in R} [1 + d_A(x_i, x_j) - d_A(x_i, x_k)]_+ \longrightarrow \min_A \tag{2.7}$$

Таким образом, в модели задаются локальные ограничения на глобальную метрику. Принцип работы метода проиллюстрирован на рис. 2.6.

При работе метода возникает огромное число ограничений: к каждому объекту должны «притягиваться» его ближайшие соседи и отталкиваться все объекты других классов. Для решения этой проблемы авторы LMNN разработали эффективный метод оптимизации, который специальным образом учитывает все ограничения [34].

LMNN допускает мультиметрическое обобщение, при котором для каждой группы объектов (стратегии формирования групп были описаны в разделе 2.2) подбирается своя метрика:

$$\sum_{(i,j)\in S} d_{A_{c_j}}(x_i, x_j) + \lambda \sum_{(i,j,k)\in R} [1 + d_{A_{c_j}}(x_i, x_j) - d_{A_{c_k}}(x_i, x_k)]_+ \longrightarrow \min_{A_1,\ldots,A_m} \tag{2.8}$$

В данном методе проблема несогласованности масштаба метрик решается по построению: поскольку все метрики обучаются совместно и фигурируют в общих уравнениях, в результате они имеют одинаковый масштаб.

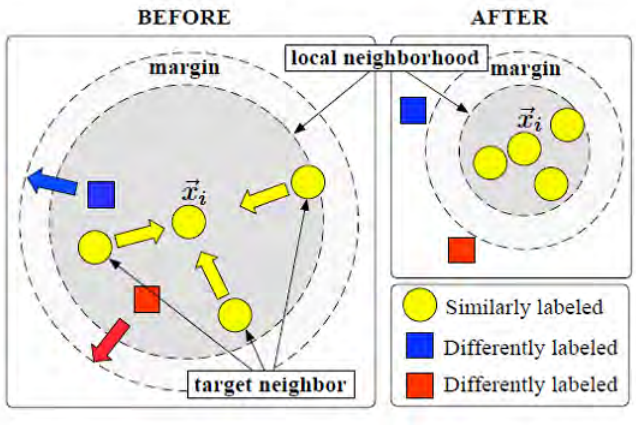

Рис. 2.6: Принцип работы метода LMNN, иллюстрация из оригинальной статьи

2.4.2 Модель Information-Theoretic Metric Learning

Модель ITML была предложена в работе [8]. Ставится оптимизациионная задача минимизации LogDet-дивергенции между матрицей исходной обобщённой евклидовой метрики A_0 и искомой матрицей A:

$$tr(AA_0^{-1}) - \log\det(AA_0^{-1}) \longrightarrow \min_A \tag{2.9}$$

при ограничениях $d_A(x_i, x_j) \leqslant l$ для похожих объектов и $d_A(x_i, x_j) \geqslant u$ для непохожих объектов. Часто выбирают $A_0 = I$, что соответствует использованию стандартной евклидовой метрики в качестве исходной. Для того, чтобы минимизировать нарушение ограничений, применяется стандартный приём —добавление slack-переменных, по которым производится оптимизация. Авторы предлагают решать поставленную оптимизационную задачу методом проекций Брегмана со стохастическим обновлением.

2.4.3 Модель Neighborhood Component Analysis

Предложенная в работе [16] модель основывается на использовании расстояния до фиксированного объекта для оценки вероятности принадлежности его классу. Это позволяет оптимизировать обобщённую евклидову метрику для минимизации вероятностной аппроксимации LOO-ошибки для метода ближайших соседей. Модель выглядит следующим образом:

$$\sum_i \sum_{j \in C_i, j \neq i} p_{ij} \longrightarrow \max_A, \text{ где } p_{ij} = \frac{exp(-d_A(x_i, x_j))}{\sum\limits_{k \neq i} exp(-d_A(x_i, x_k))} \tag{2.10}$$

Данная задача безусловной оптимизации не является выпуклой по A. В этом случае локальный минимум ищется в пространстве матриц желаемого ранга. Если же заменить функционал на KL-дивергенцию между $KL(p_{ij}, [C_i = C_j]) \longrightarrow \min$ — получится модель MCML [10] (Maximally Collapsing Metric Learning), функционал уже является выпуклым по A.

Познакомившись с классическими примерами метрических моделей, далее рассмотрим несколько идей построения мультиметрических моделей.

2.4.4 Модель Parametric Local Metric Learning

Модель PLML была предложена в работе [32]. Основная идея модели состоит в построении гладкого пространства матрикс-функций на основе множества базисных метрик $\{M_{b_1}, M_{b_2}, \ldots, M_{b_m}\}$, связанных с эталонными точками в пространстве. При этом эталонные точки выбираются из объектов обучающей выборки до начала работы алгоритма с помощью какого-либо метода.

Метрика для объекта x_i определяется как выпуклая комбинация базисных метрик:

$$M_i = \sum_{b_k} W_{ib_k} M_{b_k}, \ \sum_{b_k} W_{ib_k} = 1, W_{ib_k} \geqslant 0 \tag{2.11}$$

Матрица W в данной модели получается из соображений аппроксимации каждого примера линейной комбинацией эталонных точек. Наконец, подбираются базовые метрики путём решения следующей оптимизационной задачи:

$$\alpha_1 \sum_{b_l} ||M_{b_l}||_F^2 + \alpha_2 \sum_{(i,j)\in S} \sum_{b_l} W_{ib_l} d_{M_{b_l}}(x_i, x_j) + \\ \sum_{(i,j,k)\in R} [1 + \sum_{b_l} W_{ib_l}(d_{M_{b_l}}(x_i, x_j) - d_{M_{b_l}}(x_i, x_k)]_+ \longrightarrow \min_{M_{b_1},\ldots,M_{b_m}} \tag{2.12}$$

После того как эти шаги проделаны, для вновь пришедшей тестовой точки определяется соответствующая ей метрика, после чего в этой метрике может быть вычислено расстояние до объектов обучающей выборки. Таким образом, PLML предлагает метод оценки метрического тензора путём интерполяции базовых метрик на всё пространство.

2.4.5 Модель дерева метрик

Модель дерева метрик была предложена в работе [12] для решения задач иерархической классификации. В данном подходе каждому узлу известного таксономического дерева сопоставляется обобщённая евклидова метрика и соответствующий KNN-классификатор. Авторы предлагают совместно обучать метрики во всех вершинах, исходя из эвристики для использования различных признаков родителями и детьми: значение $||diag(A_t) + diag(A'_t)||$ должно быть как можно меньше. Кроме того, авторы используют регуляризатор-след. Получаемая задача оптимизации во многом аналогична задаче (2.8).

2.4.6 Модель Similarity Component Analysis

Модель SCA [5] использует вероятностную модель (см. рис. 2.4.6) для обучения нескольких независимых обобщённых евклидовых метрик, описывающих сходство двух заданных в векторном виде объектов.

Стандартным образом вводится расстояние между парой объектов u и v по k-ой метрике:

$$d_k = (u - v)^T M_k (u - v) \tag{2.13}$$

Далее, выписывается вероятность сходства двух объектов по k-ой метрике:

$$p(s_k = 1|u, v) = (1 + e^{-b_k})[1 - \sigma(d_k - b_k)] \tag{2.14}$$

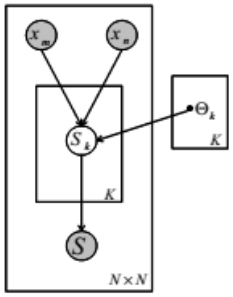

Рис. 2.7: Графическая модель метода SCA, иллюстрация из оригинальной статьи

Следующим предположением модели является использование т.н. Noisy-OR правила:

$$p(s = 1|s_1, s_2, \ldots, s_K) = 1 - \prod_{k=1}^{K} \theta_k^{\mathbb{I}[s_k=1]} \tag{2.15}$$

Маргинализуя данное выражение по всевозможным значениям $\mathbf{s}$, получаем оценку вероятности «несходства» для пары объектов (u, v):

$$p(s = 0|u, v) = \prod_{k=1}^{K} [1 - (1 - \theta_k)p(s_k = 1|u, v)] \tag{2.16}$$

Опираясь на обучающую выборку с помощью EM-алгоритма далее можно оценить все неизвестные параметры модели: M_k, b_k, θ_k.

Обученные таким образом метрики должны захватывать различные аспекты сходства и таким образом давать возможность наиболее полного учёта метрической информации.

2.5 Мультиметрические методы анализа текстов

Для расчёта расстояния между парами текстов разработано большое число различных функций близости. Подробный обзор классических способов вычисления расстояний между текстами приведён в [11].

Особенностью и главной проблемой при работе с текстовыми данными является их сложная многомерная структура. Так, при использовании классической модели мешка слов тексты представляются в виде векторов размерности d, равной мощности словаря. Как правило, эти векторы достаточно разреженные, при этом размер словаря составляет десятки тысяч слов. В этом случае обучение $d \times d$ матрицы A полного ранга не представляется возможным. В работе [7] предлагается метод для поиска матрицы A в обобщённой евклидовой метрике ранга не более k. При фиксации небольшого значения k предложенный авторами метод масштабируется линейно по d и показывает высокие значения точности и полноты в задачах информационного поиска.

Необходимым условием для того, чтобы использовать описанные выше методы для обучения матрицы A, является использование более компактных векторных

представлений текстов. Простейшим вариантом получения таких представлений является использование методов отбора признаков или методов понижения размерности (метод главных компонент, SVD и т.д.) в применении к модели мешка слов. Другим подходом для получения представлений текстов является использование следующих инструментов:

- вероятностных тематических моделей (моделей дистрибутивной семантики): LDA [2], PLSA [13] и др.
- моделей контекста: GloVe [22], Word2Vec [21]

Классический сценарий использования этих моделей для решения задач информационного поиска состоит в том, чтобы получить векторные представления слов словаря $x_w \in \mathbb{R}^d$, опираясь на которые далее можно получить векторное представление (профиль) любого текста. Стандартными схемами перехода являются следующие:

- простое усреднение векторных представлений
- взвешивание векторных представлений с весами, равными значениям tf-idf
- использование какого-либо метода оптимизации для подбора оптимального профиля текста по профилям отдельных слов

Полученные таким образом представления текстов могут быть использованы для вычисления метрик. Как и в случае модели мешка слов, для пары векторов может быть вычислена любая стандартная метрика в d-мерном пространстве. Типовыми вариантами являются обобщённая евклидова метрика и косинусное расстояние:

$$d_{cos}(x, y) = 1 - \frac{x^T y}{||x||||y||} \tag{2.17}$$

Как правило, при решении задач распознавания с текстами следует учитывать сразу несколько функций расстояния разной природы. На практике плодотворным оказывается подход, при котором для расчёта функций расстояния используются разные представления текстов:

- как последовательность символов
- как последовательность слов
- как последовательность векторных представлений слов
- как единое векторное представление документа

Помимо учёта семантической структуры, достоинством векторных представлений слов и текстов (в особенности компактных) является возможность применять мощные инструменты машинного обучения для подбора функций расстояния, такие как описанные выше методы обучения метрик, а также нейронные сети [30].

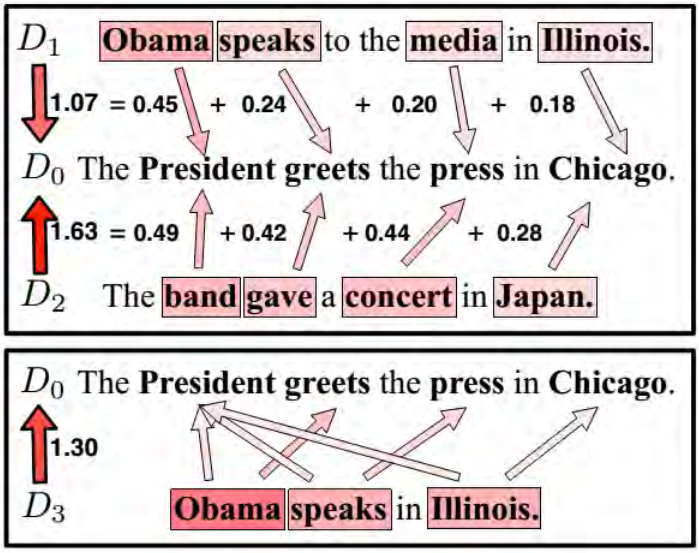

Рис. 2.8: Пример транспортной задачи для WMD из оригинальной статьи [18]

2.5.1 Модель Word Mover's Distance

В работе [18] был предложен новый инструмент для анализа текстов, рассматривающий поиск расстояния как транспортную задачу. Рассматриваются нормированные tf-idf представления двух текстов d и d' — в сумме эти значения для каждого документа дают единицу «массы». Для каждой пары слов из разных документов на основе векторного представления (например, word2vec) вычисляется евклидово расстояние, которое определяет стоимость переноса единицы «массы» по данному каналу (см. рис. 2.5.1). Требуется перенести всю единицу «массы» с наименьшими затратами. Более формально, рассматривается следующая задача линейного программирования:

$$\sum_{i,j=1}^{n} T_{ij} c(i,j) \longrightarrow \min_{T \geqslant 0} \tag{2.18}$$

При ограничениях:

$$\sum_{j=1}^{n} T_{ij} = d_i, \qquad \sum_{i=1}^{n} T_{ij} = d'_j, \tag{2.19}$$

где

$$d_i \geqslant 0, \quad d'_j \geqslant 0 \quad \sum_i d_i = 1 \sum_j d'_j = 1, \tag{2.20}$$

а функция $c(i,j)$ соответствует евклидову расстоянию между векторными представлениями слов i и j.

В такой постановке задача решается с неприемлемой асимптотикой $O(q^3 \log q)$, где q — число уникальных слов в двух документах. Поэтому авторы предлагают способ ускорения вычислений за счёт отбрасывания части ограничений, добиваясь таким образом квадратичной асимптотики. Авторы показывают, что предложенный ими метод показывает высокие результаты в задачах информационного поиска.

Развитием данной идеи является работа [15]. В модель (2.18) была добавлена возможность обучения по разметке обобщённой евклидовой функции для вычисления расстояний между словами:

$$\sum_{i,j=1}^{n} T_{ij}||A(x_i - x_j)||^2 \longrightarrow \min_{T \geqslant 0} \tag{2.21}$$

При этом для подбора матрицы A используется модель NCA 2.4.3.

Глава 3

Эмпирический анализ мультиметрических методов

В данной главе будут рассмотрены два приложения мультиметрических методов разных типов к классическим задачам информационного поиска:

- задаче классификация документов
- задаче поиска ответа ответа на вопрос в корпусе текстов

В первом случае задача исследования состояла в проверке возможности получения высокого качества в задачах классификации текстов за счёт применения методов обучения метрик и специального учёта пространственной информации с помощью мультиметрических подходов. Исходными данными выступили общедоступные размеченные текстовые коллекции на английском языке.

Во второй задаче основной трудностью является проблема краткости текста вопроса, диктующая необходимость использовать различные функции сходства для извлечения максимума семантической информации при оценке ответов-кандидатов. Исходными данными в этом исследовании выступали вопросы и ответы из нефтегазовой области на английском языке.

3.1 Задача классификации текстов

Рассматривается классическая постановка задачи классификации: для каждого из текстов обучающей выборки известна метка класса, к которому он относится, для вновь пришедшего тестового текста необходимо предсказать, к какому из классов он относится.

Для решения использовался метод k ближайших соседей, описанный в разделе 2.1. Различие между сопоставляемыми в ходе исследования методами состоит в используемой функции расстояния.

Во всех экспериментах качество работы методов оценивалось по уровню ошибки классификации (в процентах).

3.1.1 Исходные данные

Сравнение алгоритмов производилось на 4 публичных корпусах текстов на английском языке, кратко опишем каждый из них. Характеристики наборов данных

Набор данных	Число классов	Число текстов	Среднее число слов
BBCSPORT	5	737	345
TWITTER	3	3424	16
CLASSIC	4	7095	106
REUTERS	8	7674	102

Таблица 3.1: Количественные характеристики наборов данных

приведены в таблице 3.1.

Дадим качественную характеристику каждой коллекции текстов:

- BBCSPORT — коллекция газетных статей о 5 видах спорта
- TWITTER — коллекция коротких сообщений из социальной сети twitter, посвящённых IT-тематике. Классы соответствуют тональности: положительный, отрицательный, нейтральный
- CLASSIC — коллекция научных текстов по 4 различным темам (известна также как CLASSIC4)
- REUTERS — стандартным образом сформированная подвыборка из 8 классов классической коллекции REUTERS21578. Была собрана из новостей, появившихся в ленте агентства Reuters.

Для получения векторных представлений текстов использовалась предобученная модель word2vec[1], предоставляющая для слова соответствующий 300-мерный вектор. Для перехода в векторное пространство каждый текст проходил следующие этапы предобработки:

- в тексте оставлялись только слова, состоящие из буквенных символов
- каждому слову сопоставлялось векторное представление
- векторное представление текста получалось как среднее арифметическое векторных представлений слов

Кроме того, для каждого набора данных было заготовлено 5 фиксированных разбиений на обучающую и тестовую выборки для последующей валидации алгоритмов.

3.1.2 Результаты экспериментов с обучением метрик

Для 4 наборов данных были проведены эксперименты с различными описанными ранее методами построения метрик. Для каждого алгоритма на каждом наборе производилось 5 запусков обучения и тестирования на соответствующих разбиениях, после чего вычислялись среднее значение и стандартное отклонение ошибки на тесте. Для методов с гиперпараметрами использовались значения параметров, рекомендованные авторами. После построения соответствующих метрик по обучающей выборке, для определения тестового объекта использовался метод

[1] https://code.google.com/archive/p/word2vec/

	BBCSPORT	TWITTER	CLASSIC	REUTERS
EUCL	6.78 ± 1.21	29.8 ± 0.68	7.28 ± 0.2	4.83 ± 0.36
COS	7.18 ± 1.31	29.4 ± 0.74	6.08 ± 0.47	4.58 ± 0.54
COV	11.2 ± 3.49	29.6 ± 0.99	8.82 ± 0.7	5.81 ± 0.23
LFDA	3.52 ± 0.98	28.7 ± 1.13	6.13 ± 0.63	**3.83 ± 0.43**
ITML	5.69 ± 1.07	28.7 ± 0.85	7.7 ± 0.27	4.57 ± 0.47
NCA	5.95 ± 2.25	**28.2 ± 1.41**	7.78 ± 0.69	4.99 ± 0.25
LMNN	**2.7 ± 1.63**	29.5 ± 1.14	**5.28 ± 0.24**	4.15 ± 0.5
mmCOV	6.78 ± 1.19	29.1 ± 1.03	5.62 ± 0.3	4.37 ± 0.18
mmLFDA	3.79 ± 1.37	30.1 ± 1.45	6.58 ± 0.53	5.27 ± 0.55
mmITML	5.28 ± 0.75	32.7 ± 1.47	9.72 ± 1.95	4.86 ± 0.45
mmNCA	7.73 ± 0.69	34.4 ± 2.2	9.33 ± 0.68	16.0 ± 1.59

Таблица 3.2: Доля ошибок различных метрических методов, %

3 ближайших соседей. При наличии в методе нескольких метрик, использовалась нормализация следа всех матриц A в обобщённом евклидовом расстоянии. Для методов, требующих обучения матриц, производилось понижение размерности исходного пространства с 300 до 50 с помощью метода PCA.

В экспериментах участвовали следующие базовые алгоритмы, не использующие информации о разметке для вычисления расстояний:

- евклидово расстояние (EUCL)
- косинусное расстояние (COS)
- метрика Махаланобиса (см. раздел 2.3.1) (COV)

Следующая группа алгоритмов использует разметку обучающей выборки для построения единой для всего пространства обобщённой евклидовой метрики:

- метод локального линейного дискриминантного анализа [28] (LFDA)
- метод ITML
- метод NCA
- метод LMNN

Наконец, в сравнении участвуют мультиметрические версии методов. Эти методы строят для каждого класса собственную метрику, после чего для классификации тестового объекта применяется классическая схема, описанная в разделе 2.2.

Результаты запусков приведены в таблице 3.2. Обратим внимание, что на наборе данных TWITTER, состоящем из коротких текстов, качество значительно ниже, чем для коллекций с более длинными текстами. Видим, что побеждают методы, обучающие единую для всего пространства метрику, опирающиеся на разметку обучающих данных. Использование нескольких метрик только ухудшает

	BBCSPORT	TWITTER	CLASSIC	REUTERS
LFDA	4.6	30.3	5.98	4.4
NCA	8.0	28.2	7.18	5.3
LMNN	5.3	30.0	5.14	5.0
WMD EUCL	3.3	**27.4**	4.2	**3.8**
WMD COS	**2.0**	27.6	**3.0**	4.4

Таблица 3.3: Доля ошибок для различных версий WMD, %

качество работы соответствующих методов во всех случаях, кроме метода, использующего матрицы ковариации. По-видимому, это связано с проблемой переобучения: при большой размерности данных для каждого из классов по имеющейся обучающей выборке не удаётся адекватно подобрать матрицу для соответствующей обобщённой евклидовой метрики. Следует отметить, что эта проблема проявляется и в том, что в некоторых случаях качество метода, обучающего единую для всего пространства метрику, оказывается хуже базовых алгоритмов (так, NCA и ITML работают хуже евклидова и косинусного расстояния на наборе данных CLASSIC).

3.1.3 Результаты экспериментов с WMD

В своей статье разработчики WMD использовали евклидово расстояние для расчёта стоимости перемещения единицы «массы» между словами. При этом классическим способом измерения расстояния в векторном пространстве является косинусная метрика (2.17).

Между этими двумя функциями расстояния есть очевидная связь: евклидово расстояние между векторами на единичной сфере равно удвоенному косинусному расстоянию:

$$||x|| = ||y|| = 1 \Longrightarrow ||x - y||^2 = ||x||^2 + ||y||^2 - 2x^T y = 2 - 2x^T y \tag{3.1}$$

Тем не менее, косинусная метрика считается более предпочтительной, поскольку при вычислении похожести позволяет не учитывать масштаб векторов. Согласно исследованиям [25], масштаб векторов связан с частотой слова в коллекции, по которой обучались векторные представления, и не является существенным для оценки сходства. Естественным направлением исследования является проверить качество метода WMD, если использовать косинусное расстояние вместо евклидова.

В силу значительных объёмов вычислений, которых требует WMD, проверка осуществлялась только на одном разбиении на обучающую и тестовую выборки. Результаты для лучших метрических методов на этом разбиении и вариантов WMD приведены в таблице 3.3.

Отметим, что функции расстояния семейства WMD оказываются лучше по качеству, чем победившие методы обучения метрик. Далее, видим, что для двух наборов данных использование косинусного расстояния в WMD вместо евклидова позволяет более чем на 1% понизить ошибку, при том что она и так является низкой. В двух других случаях использование косинусного расстояния не улучшает

качество по сравнению с евклидовой метрикой, однако ухудшение не столь значительно. Эти наблюдения позволяют сделать вывод о том, что версия WMD с косинусным расстоянием также является работоспособным инструментом, имеет смысл пробовать данную модификацию метода при решении практических задач.

3.2 Задача построения вопросно-ответной системы

Задачи автоматического извлечения фактов (англ. Information Extraction) и построения вопросно-ответных систем (англ. Question Answering) являются классическими задачами информационного поиска, работа над которыми ведётся с 60-х годов 20 века. В последние десятилетия эти задачи приобрели особое значение в связи со значительным увеличением объёмов доступной неструктурированной и частично структурированной информации, требующей обработки. Сегодня автоматизированный поиск по обширным коллекциям научно-технических текстов востребован специалистами различных предметных областей, поскольку без помощи специализированных систем извлечения информации, обнаружение ответа на интересующий вопрос в массиве из тысяч релевантных документов становится слишком трудоемким.

Необходимость построения моделей и интеллектуальных экспертных систем (ИЭС) для обработки информации в нефтегазовой отрасли обусловлена заинтересованностью специалистов в использовании актуальных исследований при принятии решений в области разведки и добычи углеводородов.

В области вопросно-ответных систем большое число работ было посвящено методам для поиска ответов на фактические вопросы — вопросы, ответом на которые ялвляются конкретные факты. Как правило, они требуют ответа, состоящего из одного или нескольких слов (например, «Где расположена гора Монблан?», «Когда родился Моцарт?» и т.д.) [31] [6] [4].

В нефтегазовой области, как правило, задаваемые аналитиком вопросы требуют более развёрнутого ответа, поэтому в данном исследовании рассматривались модели для поиска ответов на т.н. не-фактические (англ. *non-factoid*) вопросы. К решению данной задачи исследователями были успешно применены различные модели, включая линейные модели [29] и градиентый бустинг [35] над широким набором признаков, а также нейронные сети [30].

Рассмотрим теперь формальную постановку задачи поиска ответа на вопрос в текстовой коллекции. Имеется множество текстов X, среди которых производится поиск. На вход модели поиска подаются поисковые запросы из множества Q. Задача поиска ответа рассматривается как задача ранжирования текстов: по данному запросу модель должна выдать список текстов из X, упорядоченный по вероятности наличия в данном тексте ответа на заданный вопрос.

Для настройки модели ранжирования имеется размеченная обучающая выборка: множество троек $(x, q, y) \in X \times Q \times Y$, где y — числовая оценка релевантности текста x запросу q. В данном исследовании множество Y состояло из оценок релевантности от 0 до K.

3.2.1 Исходные данные

В качестве текстовой коллекции используется выборка из 449 документов нефтегазовой тематики.

Для обучения модели ранжирования необходимо большое количество размеченных обучающих данных. Для учёта специфики нефтегазовых текстов используется размеченная экспертами выборка пар вопрос-ответ (список вопросов приведён в разделе 4). Суммарно экспертами были размечены для 25 вопросов 580 пар вопрос-ответ в двух шкалах: по шкале от 0 до 5, а также в бинарной шкале (0 или 1).

Однако полученный от экспертов объем обучающих примеров недостаточен для обучения модели поиска ответов на вопросы. Поэтому, были задействованы следующие дополнительные источники размеченных данных, не связанные с нефтегазовой тематикой:

- Выборка Yahoo Non-Factoid Question Dataset (nfL6)[2] – выборка из более чем 80.000 вопросов, заданных в интернете и около 500.000 ответов. Для каждого из вопросов дан перечень ответов, данных пользователями, причём среди них выбран ответ, признанный наилучшим.

- Выборка Web Answer Passages (WebAP)[3] – набор данных, состоящий из 82 вопросов, к каждому из которых приложен набор размеченных документов. Для каждого предложения каждого документа отмечено, в какой степени это предложение является ответом на заданный вопрос. Всего размечено 3428 документов, около 1.000.000 предложений. Данная выборка является специальным образом подготовленным подмножеством набора данных TREC Gov2.

Обе выборки состоят из не-фактических вопросов, на которые необходимо в первую очередь затачивать систему.

Выборка nfL6 имеет значительный объём, однако её применение осложняют следующие факторы:

- Многие вопросы и ответы написаны людьми на разговорном, неформальном интернет-языке, с использованием сленга, без соблюдения правил орфографии и т.д.

- Разметка имеет бинарный характер: для каждого вопроса ровно один ответ отмечен как правильный, тогда как на самом деле все написанные пользователями тексты в той или иной степени являются ответом на заданный вопрос.

Эта выборка была подготовлена к применению следующим образом: для каждой пары вопрос-ответ была проставлена метка 1, если данный ответ является наилучшим ответом на соответствующий вопрос, или метка 0, если данный ответ не является наилучшим ответом на заданный вопрос. Всего получилось 498.862 обучающих примеров, из них 86.595 положительных примеров.

Выборка WebAP более приближена к решаемой задаче поиска ответов в массиве релевантных публикаций, поскольку:

[2] выборка доступна по адресу https://ciir.cs.umass.edu/downloads/nfL6/

[3] выборка доступна по адресу https://ciir.cs.umass.edu/downloads/WebAP/

Исходная отметка	«perfect»	«excellent»	«good»	«fair»	«none»
Числовая отметка	5	4	3	2	0
Бинарная отметка	1	1	1	1	0

Таблица 3.4: Соответствие оценок релевантности для набора данных WebAP

- Ответы на заданные вопросы выделены книгах и статьях.
- Оценка каждого предложения принимает 5 возможных значений, обозначенных словами «perfect», «excellent», «good», «fair», «none». При этом оценка «none» гарантирует, что данный отрывок документа не имеет никакого отношения к заданному вопросу.

Существенным недостатком данного набора данных является небольшое количество вопросов и положительных примеров. Оценки для набора данных WebAP были переведены в числа и бинарные метки по правилу, описанному в таблице 3.4.

Затем данные были подготовлены следующим образом:

- Идущие подряд предложения, имеющие положительную оценку, были объединены и далее рассматривались как единый ответ.
- Последовательности, не имеющие положительной оценки, были разбиты на группы по 3 предложения, каждая из которых рассматривалась как единый ответ.

Таким образом, было получено 328.644 пар вопрос-ответ, среди которых 3843 положительных примера.

Многочисленные эксперименты с обучением моделей на этих выборках показали, что выборка nfL6 непригодна для решения рассматриваемой задачи. Во всех экспериментах её использование в качестве составной части обучающей выборки приводило к значительному снижению качества поиска по сравнению с использованием выборки WebAP. По-видимому, это вызвано указанными выше проблемами набора данных. Всюду далее в качестве дополнительных данных используется только выборка данных WebAP.

3.2.2 Методика оценки качества

Для оценки качества результатов, выдаваемых алгоритмами поиска ответов, использовались следующие метрики, общепринятые в задачах информационного поиска:

- Normalized Discounted Cumulative Gain at N, $nDCG(N)$, с экспоненциальными весами.

  Данный показатель качества вычисляется следующим образом. Сперва по оценкам релевантности первых N ответов в выдаче вычисляется Discounted Cumulative Gain:

$$DCG(N) = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{3.2}$$

где rel_i — истинная оценка релевантности ответа i-го ответа. Затем вычисляется наилучшее возможное значение $DCG(N)$ (т.е. для случая, когда все ответы упорядочены по убыванию релевантности):

$$IDCG(N) = \sum_{i=1}^{N} \frac{2^{rel_i^*} - 1}{log_2(i+1)} \tag{3.3}$$

где rel_i^* истинная оценка релевантности ответа i-го ответа, если ответы упорядочены по убыванию релевантности.

Наконец, производится нормировка:

$$nDCG(N) = \frac{DCG(N)}{IDCG(N)} \tag{3.4}$$

Данный показатель качества штрафует выдачу за несоответствие результата ранжирования и истинных оценок релевантности ответов. Для оценки качества рассматривались значения $nDCG(5)$, $nDCG(10)$.

- Mean Reciprocal Rank (MRR) — усреднённое по вопросам значение величины, обратной к позиции первого релевантного ответа в выдаче:

$$MRR = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{r_i} \tag{3.5}$$

где Q — общее число вопросов, а r_i — позиция первого релевантного ответа на i-ый вопрос. Данная метрика характеризует, на каком месте, в среднем, располагается первый релевантный ответ на вопрос.

- $Precision(N)$ —- доля релевантных ответов среди первых N ответов:

$$Precision(N) = \frac{1}{N} \sum_{i=1}^{N} isrel_i \tag{3.6}$$

где значение $isrel_i$ равна единице для релевантного ответа и равна нулю для нерелевантного. Данная метрика характеризует, насколько релевантные документы успешно находятся в общей массе кандидатов. Для оценки качества рассматривались значения $Precision(5)$ и $Precision(10)$.

Как следует из предыдущих разделов, для обучения моделей могут быть использованы следующие наборы размеченных данных:

- Данные по 25 вопросам, полученные в результате разметки экспертами
- Данные по 82 вопросам в выгрузке WebAP

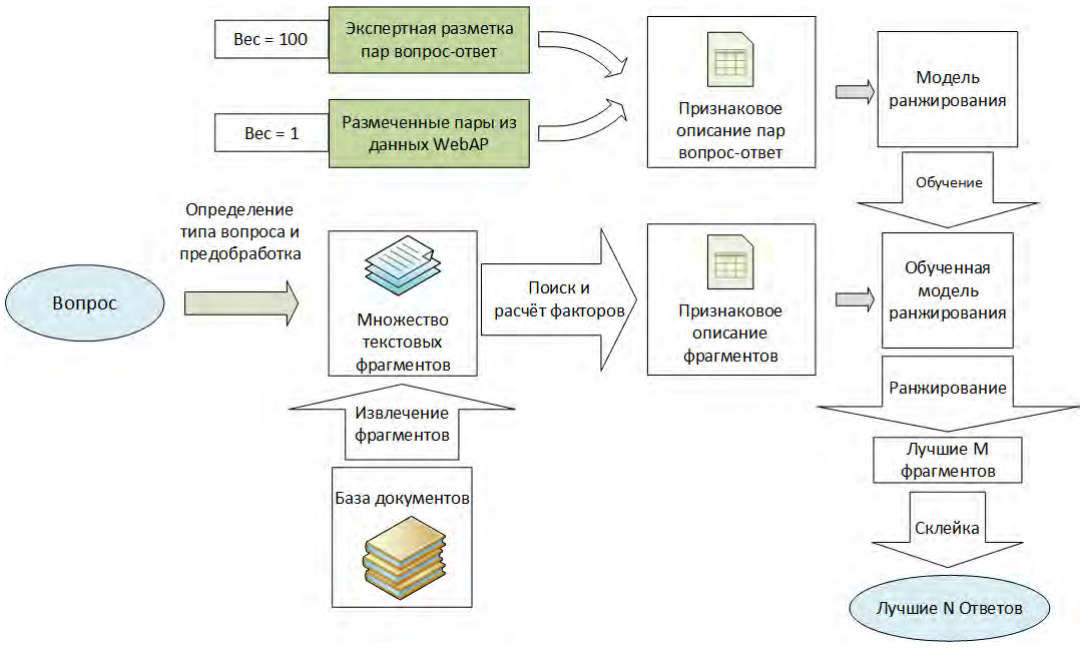

Рис. 3.1: Архитектура системы поиска ответов на вопросы

Для получения наиболее точной оценки качества поиска каждой модели вычислялись два следующих независимых набора показателей:

- Leave One Out-оценки по ручной разметке (LOO): модель обучается на выборке состоящей из данных WebAP и данных по 24 вопросам, полученным в результате разметки экспертами, а затем применяется для поиска ответов на оставшийся вопрос. При этом предполагается, что все неразмеченные ответы являются нерелевантными. Эта процедура повторяется 25 раз, после этого полученные 25 наборов значений Precision, Reciprocal Rank и nDCG усредняются.

- Кросс-валидация по WebAP: модель обучается на выборке, состоящей из данных по 72 вопросам WebAP. После этого она применяется для поиска ответов на оставшиеся 10 вопросов из WebAP. Процедура повторяется 8 раз, после чего полученные 80 наборов значений Precision, Reciprocal Rank и nDCG усредняются.

Отметим, что благодаря такому дизайну эксперимента модель на этапе обучения никогда не имеет возможности запомнить ответы на вопросы, для поиска ответов на которые затем применяется, поэтому полученные таким образом оценки качества корректны.

3.2.3 Модель поиска ответов на вопросы

Общая архитектура модели поиска приведена на рис. 3.1.

Чтобы найти ответы на вопрос во множестве документов, рассматривается следующий подход:

- Из исходных документов извлекаются фрагменты — всевозможные последовательности из 3 подряд идущих предложений.
- Исключаются фрагменты, относящиеся к спискам литературы внутри статей, поскольку они зачастую содержат многие ключевые слова вопроса, но не несут искомого ответа.
- Для каждого фрагмента строится мультиметрическое признаковое описание, каждый признак по-своему оценивает близость фрагмента к вопросу.
- Обученная модель использует признаковые описания, полученные для каждого фрагмента для ранжирования фрагментов по их релевантности заданному вопросу.
- Выбираются лучшие фрагменты, и если некоторые из них идут в тексте статьи подряд, то они склеиваются, образуя итоговый сниппет; при этом позиция итогового сниппета в ранжировании определяется рангом наилучшего из составляющих сниппет фрагментов.

Для решения задачи ранжирования фрагментов текста по вопросу необходимо уметь вычислять метрики, характеризующие семантическое сходство вопроса и текста. В качестве вариантов для признакового описания пары вопрос-ответ рассматривался набор из 20 классических функций близости для текстов (включая мощный признак BM25, широко используемый в задачах информационного поиска [24]), а также косинусное расстояние между word2vec-представлениями вопроса и кандидата и значение Word Mover's Distance.

Для получения векторных представлений текстов использовалась предобученная модель word2vec[4].

Следует отметить, что каждая из упомянутых функций близости может быть использована для ранжирования независимо от всех остальных. Однако, эффективнее может оказаться использовать некую композицию факторов для формирования итоговой формулы ранжирования. Эта формула может быть построена как вручную на основе экспертных знаний, так и специально подобрана с помощью машинного обучения, что позволяет точнее учесть специфику конкретной задачи ранжирования.

Для ранжирования рассматривались следующие классы моделей:

- модель ранжирования по одному фактору
- модель логистической регрессии [36]
- модель градиентного бустинга [9]
- модель случайных лесов [3]

При использовании моделей ранжирования по одному фактору, по заданному вопросу для каждого ответа-кандидата вычисляется значение рассматриваемого фактора, после чего тексты ранжируются в соответствии с полученной величиной.

[4] https://code.google.com/archive/p/word2vec/

Для использования таких моделей не требуется применять методы машинного обучения.

Остальные три класса моделей учитывают значения всех факторов для вычисления итоговых оценок релевантности для каждого ответа-кандидата. При использовании этих моделей, на этапе обучения по обучающей выборке, подбирается наилучшая формула ранжирования, переменными в которой являются значения всех факторов. После этого с помощью полученной формулы можно ранжировать тексты: по заданному вопросу вычисляются значения всех признаков, после чего вычисляется значение подобранной формулы ранжирования для каждого ответа-кандидата, в соответствии с которым производится ранжирование. От конкретного класса модели зависит семейство функций, среди которых производится поиск оптимальной ранжирующей формулы. Для экспериментов с моделями логистической регрессии и случайных лесов использовались реализации, предоставляемые соответствующими классами из открытой библиотеки машинного обучения scikit-learn[5]; для работы с моделью градиентного бустинга использовалась реализация xgboost[6].

Все эти классы моделей предназначены для поиска ответов на вопросы, заданные в произвольной форме. Для класса вопросов, связанных с поиском определений в тексте (т.е. вопрос, попадающий под шаблон «What is SUBJECT?»), используется дополнительная процедура, позволяющая, с помощью набора регулярных выражений, выделить фрагменты, наиболее похожие на определение понятия, содержащегося в вопросе. В дальнейшем, только эти фрагменты поступают на вход алгоритму ранжирования.

Таким образом, итоговый результат ранжирования зависит от:

- класса вопроса
- мультиметрического признакового пространства, в которое помещены пары вопрос-ответ
- используемой для обучения выборки
- используемого класса моделей
- конкретных параметров модели

Во всех экспериментах для каждого класса моделей были подобраны наилучшие значения гиперпараметров.

3.2.4 Результаты экспериментов

Таблица 3.5 содержит основные результаты, для каждого из методов валидации выделены два наилучших значения (в скобках указано, какое место занял метод).

Наиболее сильными отдельными признаками являются признаки из семейства BM25, и основанный на модели word2vec признак WMD. Последний на выборке

[5] http://scikit-learn.org/stable/index.html
[6] https://github.com/dmlc/xgboost

Validation method	NDCG(5)	NDCG(10)	MRR	Prec(5)	Prec(10)
BM25a LOO	0.23(2)	0.23(2)	0.44(2)	0.23	0.20
BM25a WebAP	0.069	0.077	0.21	0.093	0.099
word2vec WMD LOO	0.088	0.089	0.21	0.096	0.092
word2vec WMD WebAP	0.17(1)	0.15	0.39(1)	0.22(1)	0.18
Logistic Regression LOO	0.27(1)	0.30(1)	0.53(1)	0.28(1)	0.26(1)
Logistic Regression WebAP	0.16(2)	0.15(1)	0.36(2)	0.21(2)	0.18(1)
Gradient Boosting LOO	0.18	0.17	0.35	0.20	0.16
Gradient Boosting WebAP	0.13	0.13	0.32	0.19	0.17
Random Forest LOO	0.22	0.23	0.40	0.25(2)	0.22(2)
Random Forest WebAP	0.15	0.15(2)	0.35	0.20	0.18(1)

Таблица 3.5: Результаты оценки качества моделей

WebAP по $nDCG(5)$ показывает результаты выше моделей с использованием машинного обучения. Тем не менее, модели ранжирования по одному фактору могут вести себя неустойчиво в зависимости от конкретных исходных данных. Это видно и из таблицы, где уже на валидации по LOO признак WMD показывает гораздо более скромный результат по сравнению с другими моделями. Аналогично, признак BM25a демонстрирует достаточно высокое качество на валидации по LOO, но существенно уступает на WebAP. Поэтому использование комбинаций признаков является более надёжным решением.

В целом, наиболее высокое и устойчивое качество показывает модель логистической регрессии (линейная модель). Модель случайного леса также даёт неплохой результат, тогда как модель градиентного бустинга значительно отстаёт на обеих валидациях.

Помимо замеров показателей, для визуальной оценки качества использовался программный стенд, изображённый на рис. 3.2. Для заданного вопроса выводятся документы и тексты сниппетов, содержащие ответ на вопрос.

Значения показателей качества для оптимальной версии модели ранжирования с использованием логистической регрессии говорят о том, что в среднем модель:

- гарантирует наличие ответа среди первых 5 результатов
- ставит наиболее релевантный ответ на второе место в результатах поиска

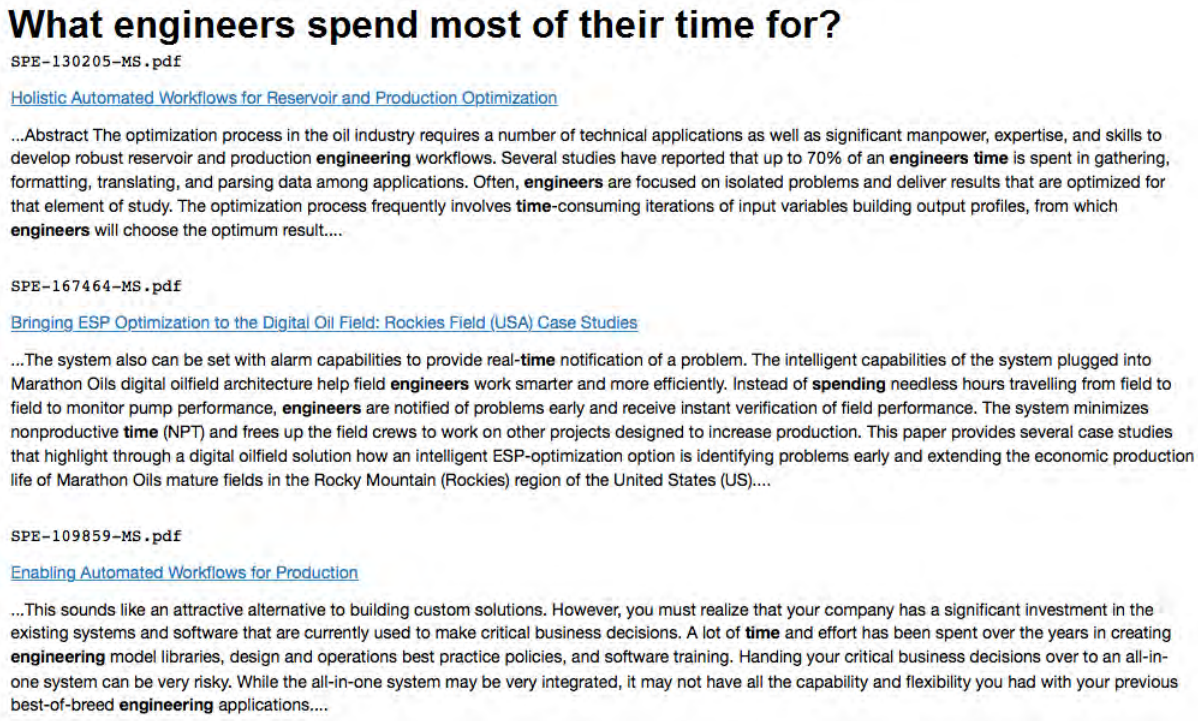

Рис. 3.2: Программный стенд для сравнения методов поиска ответа на вопрос

Глава 4

Заключение

В ходе исследования были достигнуты следующие результаты:

1. Произведён обзор различных постановок задач в области мультиметрического анализа данных, проанализированы различные варианты получения и использования множества функций расстояния для повышения качества работы алгоритмов текстового информационного поиска.

2. Рассмотрены два приложения мультиметрических методов разных типов к классическим задачам информационного поиска: задаче классификация документов и задаче поиска ответа ответа на вопрос в корпусе текстов.

3. Для задачи классификации текстов проведено сравнение качества работы большого числа метрических и мультиметрических моделей, основанных на обучении разных метрик для разных областей пространства.

4. Исследованы варианты вычисления расстояния между словами для модели Word Mover's Distance, показано, что косинусное расстояние в некоторых случаях позволяет достичь высокого качества.

5. Проведено исследование мультиметрических моделей поиска ответа на вопрос в корпусе текстов, позволяющих учесть различные аспекты семантического сходства между вопросом и текстом-кандидатом.

6. Проведено эмпирическое сравнение способов подбора оптимальной формулы ранжирования с помощью методов машинного обучения разных типов: логистической регрессии, градиентного бустинга, случайных лесов. Проанализированы модели ранжирования на основе отдельно взятых функций сходства.

7. Результаты работы моделей поиска ответа на вопрос валидированы на двух наборах данных.

8. Показано, что модель логистической регрессии в совокупности со специально подобранным набором функций сходства позволяет достичь наилучших результатов поиска.

9. Реализован программный стенд для анализа результатов работы моделей, представляющий собой вопросно-ответную систему: по заданному пользователем вопросу с помощью настроенной модели в корпусе отыскиваются релевантные документы и отрывки текста, содержащие ответ.

Приложение 1: список вопросов для вопросно-ответной системы

1. What is the Digital Oilfield (DOF)?
2. Which machine learning approaches allow for more than 70% accuracy in prediction of ESP (electric submersible pump) failure?
3. Which parameters of streaming sensor data are most important for ESP failure prediction?
4. What examples of artificial intelligence workflows are known to be used for oil production?
5. Which artificial intelligence workflows are developed at Schlumberger?
6. What types of traditional proxy models are used in smart oilfield?
7. What types of machine learning proxy models are considered in smart oilfield technology?
8. Which workflows are used for optimization of fracture design?
9. What issues are expected in preparation of training data for machine learning models within DOF?
10. What is Digital Rock?
11. What resolution limit does modern micro CT machines have?
12. What technique provides the maximal resolution for imaging of the core samples?
13. List the most common approaches for modeling the multiphase flows on porous media.
14. How to preserve edges and fine features at filtering (de-noising) images of the porous rocks?
15. What approach could be utilized for modeling the complex flows with multiphase multicomponent chemical fluids?
16. What are the limitations of methods for studying the rock structures?
17. How does gas lift work?

18. List key problems with multilateral wells?
19. List intelligent solutions invented for reservoir engineering?
20. What are the key problems with well production?
21. What are the purposes of using image analysis in Digital Rock?
22. What approaches exist for pore-space analysis?
23. What is DRP workflow in detail?
24. List the most common problems with micro CT?
25. What fraction of carbon could be found in produced oil?

Литература

[1] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Eric Brill, Susan Dumais, and Michele Banko. An analysis of the askmsr question-answering system. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 257–264. Association for Computational Linguistics, 2002.

[5] Soravit Changpinyo, Kuan Liu, and Fei Sha. Similarity component analysis. In *Advances in Neural Information Processing Systems*, pages 1511–1519, 2013.

[6] Hoa Trang Dang, Diane Kelly, and Jimmy J Lin. Overview of the trec 2007 question answering track. In *Trec*, volume 7, page 63, 2007.

[7] Jason V Davis and Inderjit S Dhillon. Structured metric learning for high dimensional problems. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 195–203. ACM, 2008.

[8] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[9] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[10] Amir Globerson and Sam Roweis. Metric learning by collapsing classes. In *Nips*, volume 18, pages 451–458, 2005.

[11] Wael H Gomaa and Aly A Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13), 2013.

[12] Kristen Grauman, Fei Sha, and Sung Ju Hwang. Learning a tree of metrics with disjoint visual features. In *Advances in neural information processing systems*, pages 621–629, 2011.

[13] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[14] Steven CH Hoi, Wei Liu, Michael R Lyu, and Wei-Ying Ma. Learning distance metrics with contextual constraints for image retrieval. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2072–2078. IEEE, 2006.

[15] Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. Supervised word mover's distance. In *Advances in Neural Information Processing Systems*, pages 4862–4870, 2016.

[16] SamRoweis JacobGoldberger and RuslanSalakhutdinov GeoffHinton. Neighbourhood components analysis. *NIPS'04*, 2004.

[17] Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

[18] Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, pages 957–966, 2015.

[19] James T Kwok and Ivor W Tsang. Learning with idealized kernels. In *ICML*, pages 400–407, 2003.

[20] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[22] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[23] Deva Ramanan and Simon Baker. Local distance functions: A taxonomy, new algorithms, and an evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):794–806, 2011.

[24] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[25] Adriaan MJ Schakel and Benjamin J Wilson. Measuring word significance using distributed representations of words. *arXiv preprint arXiv:1508.02297*, 2015.

[26] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[27] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *NIPS*, volume 1, page 2, 2003.

[28] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 905–912. ACM, 2006.

[29] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online qa collections. In *ACL*, volume 8, pages 719–727, 2008.

[30] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*, 2015.

[31] Ellen M Voorhees and L Buckland. Overview of the trec 2003 question answering track. In *TREC*, volume 2003, pages 54–68, 2003.

[32] Jun Wang, Alexandros Kalousis, and Adam Woznica. Parametric local metric learning for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2012.

[33] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

[34] Kilian Q Weinberger and Lawrence K Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the 25th international conference on Machine learning*, pages 1160–1167. ACM, 2008.

[35] Liu Yang, Qingyao Ai, Damiano Spina, Ruey-Cheng Chen, Liang Pang, W Bruce Croft, Jiafeng Guo, and Falk Scholer. Beyond factoid qa: effective methods for non-factoid answer sentence retrieval. In *European Conference on Information Retrieval*, pages 115–128. Springer, 2016.

[36] Tong Zhang and Frank J Oles. Text categorization based on regularized linear classification methods. *Information retrieval*, 4(1):5–31, 2001.