Доверие, основанное на знаниях: Оценка достоверности веб ресурсов

Синь Луна Донг, Евгений Габрилович, Кевин Мерфи, Ван Данг Вилко Хорн,
Камилло Лугареси, Шаохуа Сун, Вей Чжан
Google Inc.

{lunadong|gabr|kpmurphy|vandang|wilko|camillol|sunsh|weizh}@google.com

РЕЗЮМЕ

Традиционно качество веб-ресурса оценивалось с использованием *внешних* сигналов, например, с использованием ссылок на сайт. Мы предлагаем новый подход, который основывается на *внутренних* сигналах, а именно, на корректности фактической информации, представленной на веб ресурсе. Ресурс, содержащий небольшое количество ложных фактов, может считаться достоверным.

Факты автоматически извлекаются из каждого ресурса с помощью методов извлечения информации, как правило используемых для создания баз знаний. Мы предлагаем различать ошибки, сделанные в процессе извлечения данных, от фактических ошибок, которые содержит ресурс сам по себе, с помощью метода комбинированного логического вывода, используемого в новейшей многослойной вероятностной модели.

Рассчитанную нами оценку достоверности мы назвали *«Доверием, основанным на знаниях» (Knowledge-Based Trust (KBT)*. Используя смоделированные данные, нам удалось показать, что наш метод позволяет надежно определить истинную достоверность ресурсов. Затем мы использовали наш метод применительно к базе данных, содержащей 2.8 миллиардов фактов, извлеченных из сети, и, таким образом, оценили достоверность 119 миллионов веб страниц. Оценка групп результатов, проведенная без использования автоматических средств, подтвердила эффективность метода.

1. ВВЕДЕНИЕ

«*Научиться доверять - одна из самых сложных задач в жизни.*»

– Исаак Уотс.

Оценка качества веб-ресурса[1] имеет огромное значение для веб-поиска. Традиционно оно оценивалось с помощью внешних сигналов, например, по ссылкам на сайты или истории посещений сайтов. Однако подобные сигналы, по большей части, отражают популярность ресурса. Например, сайты светской хроники, перечисленные в работе [16], преимущественно имеют высокие значения PageRank [4]. Однако такие сайты, как

[1] Мы используем термин «веб-ресурс» для обозначения конкретной веб страницы, например, wiki.com/page1, или целого веб-сайта, например, wiki.com. Более подробно различия обсуждаются в разделе 4.

правило, не считаются надежными. И наоборот, некоторые менее популярные сайты содержат очень точную информацию.

В данной работе мы изучили фундаментальный вопрос оценки того, насколько достоверным является отдельно взятый веб-ресурс. Мы произвольно определили достоверность или *надежность* веб-ресурса, как вероятность того, что он содержит правильное значение для факта (как пример используется гражданство Барака Обамы), допуская, что сайт упоминает любое значение для этого факта. (Таким образом, мы не штрафовали ресурсы, которые располагают немногочисленными фактами, при условии, что они верны.)

Мы предлагаем, используя оценку *«Доверия, основанном на знаниях» (Knowledge-Based Trust (KBT)*, определять достоверность ресурса следующим образом. Мы извлекли множество фактов со множества страниц с помощью методов извлечения информации. Затем мы совместно оценили корректность этих фактов и надежность ресурсов, используя логический вывод в вероятностной модели. Логический вывод представляет собой итеративный процесс, учитывая, что мы полагаем, что ресурс надежен, если факты, которые он содержит, правильны, а также полагаем, что факты правильные, если они извлечены из надежного ресурса. Мы использовали избыточность информации в сети для нарушения симметрии. Более того, мы показали, как инициализировать нашу оценку точности ресурса, основываясь на заслуживающей доверия информации, чтобы гарантировать, что этот итеративный процесс приводит к хорошему результату.

Процесс извлечения фактов, который мы использовали, основывается на проекте *«Хранилище знаний» (Knowledge Vault (KV)* [10]. KV использует 16 различных систем извлечения информации для извлечения *триад данных* (субъект, предикат, объект) с веб-ресурсов. Примером такой триады является *(Барак Обама, гражданство, США)*. Субъект представляет собой реальную сущность, идентифицированную с помощью ID, например, с помощью *машинного ID* в *Freebase* [2]; предикат заранее предопределен в *Freebase* и описывает конкретный атрибут сущности; объект может быть сущностью, строкой, численным значением, или датой.

Факты, извлеченные с помощью автоматических методов, таких как KV, могут быть ошибочными. Один из методов оценки того, являются ли они верными или нет, описан в работе [11]. Однако данная более ранняя работа не содержит различий между фактическими ошибками на странице и ошибками, совершенными системой извлечения информации. Как показано в работе [11], ошибки извлечения более распространены, чем ошибки, которые содержит ресурс. Игнорирование таких различий между ошибками, может привести к тому, что мы ошибочно не станем доверять ресурсу.

Другая проблема подхода, описанного в работе [11], заключалась в том, что он позволяет оценить надежность каждой веб-страницы отдельно. Это может привести к затруднениям, если данные рассеяны. Например, для более чем миллиарда веб-страниц KV может извлечь одну триаду информации (прочие системы извлечения данных имеют схожие ограничения). Это затрудняет надежную оценку достоверности таких ресурсов. С другой стороны, из некоторых страниц KV извлекает десятки тысяч триад, что может привести к затруднениям в вычислениях.

Метод KBT, предложенный в данной работе, позволяет устранить некоторые из ранее существовавших недостатков. Мы сделали три существенных дополнения. Первое,

основное дополнение заключается в использовании более сложной вероятностной модели, которая может различать два типа ошибок: некорректные факты на странице и некорректное излечение фактов, выполненное системой. Это позволяет более точно оценить надежность ресурса. Мы предложили эффективный, масштабируемый алгоритм для осуществления логического вывода и оценки параметров в предложенной вероятностной модели (Раздел 3).

Второе дополнение представляет собой новый метод адаптивного принятия решения о степени разбиения ресурсов, с которыми необходимо работать: если конкретная веб-страница приносит слишком мало триад, мы можем объединить ее с другими страницами этого же веб-сайта. И наоборот, если веб-страница содержит слишком много триад, мы можем разбить ее на более мелкие элементы, чтобы избежать сложностей при вычислении (Раздел 4).

Третье дополнение представляет собой детальную, крупномасштабную оценку эффективности модели. В частности, мы использовали 2.8 миллиардов триад, извлеченных из веб-ресурсов, что дало нам возможность надежно предсказать достоверность 119 миллионов веб-страниц и 5.6 миллионов веб-сайтов (Раздел 5).

Мы отметили, что достоверность ресурса дает дополнительный сигнал для оценки качества сайта. Мы обсудили новые исследовательские возможности для улучшения оценки и возможность комбинированного использования данного вида оценки с существующими видами, такими как PageRank (Раздел 5.4.2). Мы также отметили, что несмотря на то, что наш метод представлен в контексте извлечения информации, основной подход, предложенный нами, может быть использован применительно ко множеству других задач, связанных с интеграцией и очисткой данных.

2. ПОСТАНОВКА ПРОБЛЕМЫ И КРАТКОЕ ОПИСАНИЕ

В данном разделе мы дали формальное определение методу *KBT*. Затем мы вкратце описали нашу предыдущую работу, которая решала смежную проблему, а именно проблему *интеграции знаний* [11]. И наконец, мы дали краткое описание нашего подхода и обозначили его отличия от предыдущей работы.

2.1 Постановка проблемы

Нам был дан ряд веб-ресурсов *W* и ряд экстракторов *E*. Экстрактор представляет собой метод извлечения триады (субъект, предикат, объект) из веб-страницы. Например, один экстрактор может искать *шаблон типа «$A, президент $B, ...»*, из которого он может извлечь триаду *(A, гражданство, B)*. Конечно, этот факт не всегда является правильным (например, *A* может быть президентом компании, а не страны). К тому же экстрактор увязывает строковое представление сущностей с идентификаторами сущностей, например, с машинными идентификаторами Freebase, что порой он не может выполнить. Подобные ошибки экстрактора (неправильные запросы), которые необходимо отделять от ошибок, содержащихся на сайте сподвигли нас на осуществление данных исследований.

В дальнейшем в данной работе мы будем представлять триады, как пары (элемент данных, значение), где элемент данных будет иметь вид (субъект, предикат),

описывающий конкретный признак сущности, а объект будет служить в качестве значения для элемента данных. Мы резюмировали обозначения, используемые в данной работе, в Таблице 1.

Таблица 1: Сводная таблица основных обозначений, используемых в данном документе.

Обозначение	Определение
$w \in W$	Веб-ресурс
$e \in E$	Экстрактор
d	Элемент данных
v	Значение
X_{ewdv}	Бинарный признак того, извлекает ли e (d, v) из w
X_{wdv}	Все извлеченные данные из w относительно (d, v)
X_d	Все данные об элементе данных d
X	Все введенные данные
C_{wdv}	Бинарный признак того, содержит ли w (d, v)
T_{dv}	Бинарный признак того, является ли v корректным значением для d
V_d	Истинное значение для элемента данных d, при условии использования предположения о единственной истине
A_w	Надежность веб ресурса w
P_e, R_e	Точность и полнота экстрактора e

Мы ввели переменную X_{ewdv}. X_{ewdv} = 1, если экстрактор e извлекает значение v для элемента данных d из веб-ресурса w; если экстрактор не извлекает такое значение, X_{ewdv} = 0. Экстрактор также должен показывать степень уверенности, указывающий на то, насколько уверенным является экстрактор в правильности извлечения. Всю эту дополнительную информацию мы рассмотрели в Разделе 3.5. Мы использовали матрицу $X = \{X_{ewdv}\}$ для обозначения всех данных.

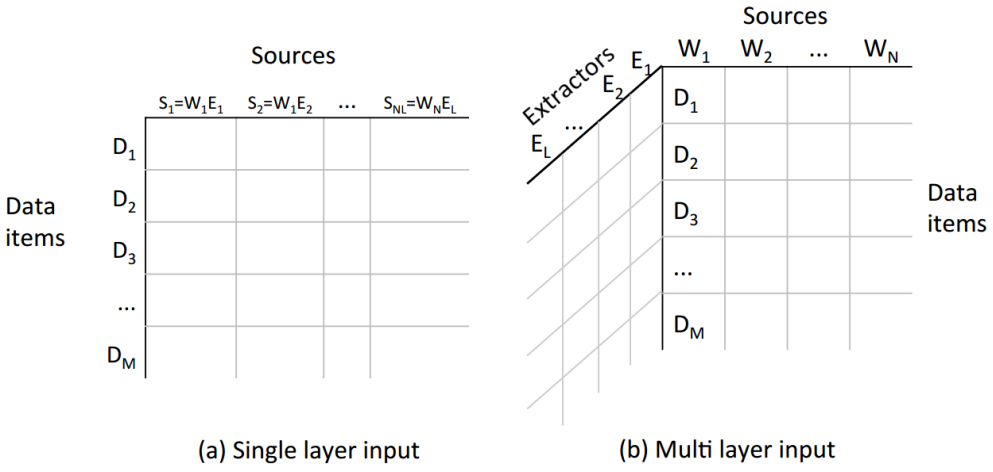

Рисунок 1: Форма вводных данных для (а) однослойной модели и (б) многослойной модели.

Мы можем представить X как «куб данных», как показано на Рисунке 1(б). В Таблице 2 показан пример одного горизонтального «среза» этого куба для случая, когда элемент данных - это $d^* = (Барак\ Обама, гражданство)$. Далее мы более подробно изучим данный пример.

Таблица 2: Информация о гражданстве Обамы, извлеченная 5-ью экстракторами из 8 веб-страниц. Столбец 2 (значение) содержит информацию о гражданстве, действительно представленную каждым ресурсом. Столбцы 3-7 содержат информацию о гражданстве, извлеченную экстрактором. Неверные извлечения показаны курсивом.

	Значение	E_1	E_2	E_3	E_4	E_5
W_1	США	США	США	США	США	*Кения*
W_2	США	США	США	США	*С. Амер.*	
W_3	США	США		США	*С. Амер.*	
W_4	США	США		США	*Кения*	
W_5	Кения	Кения	Кения	Кения	Кения	Кения
W_6	Кения	Кения		Кения	*США*	
W_7	-			*Кения*		*Кения*
W_8	-					*Кения*

ПРИМЕР 2.1. *Предположим у нас есть 8 веб-страниц, W1 – W8, и предположим, что нас интересует элемент данных (Обама, гражданство). Значение, указанное для этого элемента данных каждой веб-страницей показано во втором столбце слева в Таблице 2. Мы видим, что страницы W1 – W4 дают информацию о том, что Обама имеет гражданство США, в то время как страницы W5 – W6, предоставляют информацию о том, что Обама имеет гражданство Кении (ложное значение). Страницы W7 – W8 вообще не отображают информацию о гражданстве Обамы.*

Теперь предположим, что у нас 5 экстракторов различной надежности. Значение, которое они извлекают для данного элемента данных из каждой из 8-ми веб-страниц, показаны в таблице. Экстрактор E1 извлекает все триады правильно. Экстрактор E2 пропускает некоторые из представленных триад (ложноотрицательный результат), но все извлеченные им данные правильные. Экстрактор E3 извлекает все представленные триады, но ложно извлекает значение Кения со страницы W7, даже несмотря на то, что W7 не содержит данного значение (ложноположительный результат). Экстракторы E4 и E5 оба характеризуются плохим качеством, пропуская множество представленных триад и совершая большое количество ошибок.

Для каждого веб ресурса $w \in W$, мы определили его *надежность*, обозначенную через A_w, как вероятность того, что значение, которое он содержит для факта, является верным (то есть соответствует реальному). Мы использовали $A = \{A_w\}$ для обозначения набора всех параметров надежности. В итоге, мы можем официально определить задачу оценки KBT.

ОПРЕДЕЛЕНИЕ 2.2 (ОЦЕНКА KBT). Задача оценки достоверности, основанной на знаниях (Knowledge-Based Trust (KBT)) заключается в *оценке надежности веб-ресурса* $A = \{A_w\}$ *при наличии матрицы наблюдений извлеченных триад* $X = \{X_{ewdv}\}$.

2.2 Оценка истинности с использованием однослойной модели

Оценка КВТ тесно связана с проблемой *интеграции знаний*, которую мы рассматривали в нашей предыдущей работе [11], в которой мы оценили истинные (но скрытые) значения для каждого элемента данных в условии наблюдений при наличии помех. Мы ввели бинарные скрытые переменные T_{dv}, которые отражали, является ли v правильным значением для элемента данных d. Пусть $T = \{T_{dv}\}$. *При наличии матрицы наблюдения* $X = \{X_{ewdv}\}$ *метод интеграции знаний вычисляет апостериорную вероятность на основе скрытых переменных,* $p(T|X)$.

Один из способов решения данной проблемы заключается в «изменении формы» куба, то есть куб необходимо превратить в двухмерную матрицу, как показано на Рисунке 1(а), рассматривая каждую комбинацию веб-страницы и экстрактора в качестве обособленного ресурса данных. Теперь данные представлены в форме, с которой могут работать стандартные методы *интеграции данных* (изученные в работе [22]). Мы назвали такую модель *однослойной моделью*, так как она содержит всего один слой скрытых переменных (представляющих неизвестные значения для элементов данных). Теперь рассмотрим эту модель более подробно и сравним ее с нашей работой.

В нашей предыдущей работе [11] мы применили вероятностную модель, описанную в работе [8]. Мы предположили, что каждый элемент данных может иметь только одно истинное значение. Данное предположение справедливо для функциональных предикатов, таких как *гражданство* или дата *рождения*, но технически не действительно для предикатов, характеризующихся множеством значений, например, для предиката *ребенок*. Тем не менее, в работе [11] эмпирическим путем доказано, что предположение о «единственной истине» отлично работает на практике даже для нефункциональных предикатов. Так что мы используем данное предположение в текущей работе для простоты. (Смотрите работы [27, 33], в которых идет речь об атрибутах, характеризующихся множеством значений).

Основываясь на предположении о единственной истине, мы определили скрытые переменные $V_d \in \text{dom}(d)$ для каждого элемента данных, чтобы представить истинное значение для d, где $\text{dom}(d)$ – это интервал (набор возможных значений) для элемента данных d. Пусть $V = \{V_d\}$, и отметим, что мы можем вывести $T = \{T_{dv}\}$ из V, основываясь на предположении о единственной истине. Тогда определим следующую модель наблюдения:

$$p(X_{sdv} = 1|V_d = v^*, A_s) = \begin{cases} A_s & \text{if } v = v^* \\ \dfrac{1 - A_s}{n} & \text{if } v \neq v^* \end{cases} \quad (1)$$

где v^* - это истинное значение, $s = (w, e)$ - ресурс, $A_s \in [0, 1]$ – *надежность* этого ресурса данных, а n – число ложных значений для данного интервала (то есть мы предположили, что $|\text{dom}(d)| = n + 1$). Модель говорит о том, что вероятность того, что s гарантирует истинное значение v^* для d, является его надежностью, принимая во внимание то, что вероятность предоставления этим ресурсом одного из n ложных значений – это $1–A_s$ деленое на n.

Используя данную модель, можно легко применить правило Байеса для вычисления $p(V_d|X_d, A)$, где $X_d = \{X_{sdv}\}$ – это все данные, соответствующие элементу данных d (то есть ряд d в матрице данных), а $A = \{A_s\}$ – это набор всех параметров надежности. Предполагая

равномерное распределение априорной вероятности для $p(V_d)$, можно вычислить $p(V_d|X_d, A)$ следующим образом:

$$p(V_d = v|X_d, A) = \frac{p(X_d|V_d = v, A)}{\sum_{v' \in \text{dom}(d)} p(X_d|V_d = v', A)} \quad (2)$$

где функция правдоподобия может быть выведена из уравнения (1) при условии независимости ресурсов данных:[2]

$$p(X_d|V_d = v^*, A) = \prod_{s,v:X_{sdv}=1} p(X_{sdv} = 1|V_d = v^*, A_s) \quad (3)$$

Эта модель носит название модели ACCU [8]. Немного более совершенная модель под названием POPACCU исключает предположение о том, что неверные значения распределены равномерно. Вместо этого, модель использует эмпирическое распределение значений в наблюдаемых данных. Было доказано, что модель POPACCU монотонна. Это значит, что добавление большего количества ресурсов не снизит качество результатов [13].

В обеих моделях ACCU и POPACCU необходимо совместно оценить скрытые значения $V = \{V_d\}$ и параметры надежности $A = \{A_s\}$. Для реализации этой задачи было предложено использовать EM – подобный алгоритм([8]):

- Задаем счетчик итераций $t = 0$.
- Задаем для параметров A_s^t некоторое значение (например, 0.8).
- Рассчитываем $p(V_d \mid X_d, A^t)$ параллельно для всех d, используя уравнение (2) (Подобно шагу E). Получив эти данные, мы можем рассчитать наиболее вероятное значение $\widehat{V}_d = \text{argmax}\, p(V_d|X_d, A^t)$.
- Рассчитываем $\hat{A}_s^{(t+1)}$ следующим образом:

$$\hat{A}_s^{t+1} = \frac{\sum_d \sum_v \mathbb{I}(X_{sdv} = 1) p(V_d = v|X_d, A^t)}{\sum_d \sum_v \mathbb{I}(X_{sdv} = 1)} \quad (4)$$

  где $\mathbb{I}(a = b)$ - это 1, если $a = b$, в противном случае 0. Это уравнение говорит о том, что мы оцениваем надежность ресурса по средней вероятности фактов, которые она извлекает. Это уравнение подобно шагу M в алгоритме EM.

- Теперь мы возвращаемся к шагу E и повторяем итерации до схождения.

Теоретические свойства этого алгоритма обсуждаются в работе [8].

2.3 Оценка KBT с использованием многослойной модели

Несмотря на то, что оценка KBT тесно связана с интеграцией знаний, однослойная модель неспособна решать две проблемы. Первая проблема заключается в неспособности модели оценить достоверность веб ресурса отдельно от экстракторов.

[2] В предыдущих работах [8, 27] обсуждалась возможность определения копирования и корреляций между источниками при интеграции данных. Однако проблема, возникающая при масштабировании до миллиарда веб-ресурсов, остается неразрешенной.

Другими словами, A_s – это надежность пары (w, e), а не надежность ресурса как такового. В нашем примере мы можем сделать неверный логический вывод о том, что страница W_1 – плохой ресурс по причине извлечения значения *Кения*, хотя это ошибка является ошибкой извлечения.

Вторая проблема касается неспособности модели правильно оценить достоверность триад. В нашем примере имеется 12 ресурсов (то есть пар экстрактор – веб-страница) для *США* и 12 ресурсов для *Кении*. Кажется, что значения *США* и *Кения* могут в равной степени быть истиной. Однако интуитивно данное предположение выглядит необоснованным: экстракторы E_1 – E_3 демонстрируют согласие друг с другом при извлечении данных, оттого они выглядят надежными. Таким образом мы можем «объяснить» значения «Кения», извлеченные экстракторами E_4 – E_5, как возможные ошибки при извлечении.

Решение двух вопросов требует, чтобы мы отличали ошибки извлечения от ошибок ресурса. В нашем примере нам нужно было отличить правильно извлеченные истинные триады (например, *США* из W_1 – W_4), правильно извлеченные ложные триады (например, *Кения* из W_5 – W_6), неправильно извлеченные истинные триады (например, *США* из W_6), и неправильно извлеченные ложные триады (например, *Кения* из W_1, W_4, W_7 – W_8).

В данной работе мы представили новую вероятностную модель, которая может оценить надежность каждого-веб ресурса, исключая помехи, привнесенный экстракторами. Данная модель отличается от однослойной модели двумя особенностями. Во-первых, помимо скрытых переменным для представления истинного значения каждого элемента данных (V_d), новая модель вводит набор скрытых переменных для отображения того, был ли каждый из экстракторов верен или нет. Это позволило нам отличать ошибки извлечения от ошибок, содержащихся на самом ресурсе. Во-вторых, вместо использования параметра A для представления надежности пар (e, w), новая модель задает набор параметров надежности веб-ресурса и качества экстракторов. Это позволило нам отделить качество ресурсов от качества экстракторов. Мы назвали новую модель *многослойной моделью*, так как она содержит два слоя скрытых переменных и параметров (Раздел 3).

Основные различия между многослойной и однослойной моделями позволяют надежно оценить КВТ. В Разделе 4 мы описали, как динамически выбирать степень разбиения ресурса и экстрактора. И наконец, в Разделе 5 мы эмпирически показали, какую важную роль оба компонента играют в повышении эффективности модели по сравнению с однослойной моделью.

3. Многослойная модель

В данном разделе мы детально описали, как вычислить $A = \{A_w\}$ из нашей матрицы наблюдения $X = \{X_{ewdv}\}$ с использованием многослойной модели.

3.1 Многослойная модель

Мы расширили упомянутую выше однослойную модель в двух направлениях. Во-первых, мы ввели бинарные скрытые переменные C_{wdv}, которые показывают, действительно ли веб-ресурс w имеет триады (d, v) или нет. Так же, как и в уравнении (1) эти переменные зависят от истинных значений V_d и надежности каждого из веб-ресурсов A_w:

$$p(C_{wdv} = 1 | V_d = v^*, A_w) = \begin{cases} A_w & \text{if } v = v^* \\ \dfrac{1 - A_w}{n} & \text{if } v \neq v^* \end{cases} \quad (5)$$

Во-вторых, на основе работ [27, 33] мы использовали двухпараметрическую модель помех для наблюдаемых данных:

$$p(X_{ewdv} = 1 | C_{wdv} = c, Q_e, R_e) = \begin{cases} R_e & \text{if } c = 1 \\ Q_e & \text{if } c = 0 \end{cases} \quad (6)$$

где R_e - это *полнота* экстрактора. Говоря другими словами, это вероятность извлечения истинно имеющейся триады. Q_e - это 1 минус *специфичность*. Другими словами, это вероятность извлечения непредусмотренной триады. Параметр Q_e зависит от полноты (R_e) и точности (P_e) следующим образом:

$$Q_e = \frac{\gamma}{1 - \gamma} \cdot \frac{1 - P_e}{P_e} \cdot R_e \quad (7)$$

где $\gamma = p(C_{wdv} = 1)$ для любого $v \in \text{dom}(d)$, как поясняется в [27]. (В Таблице 3 представлен числовой пример вычисления Q_e из P_e и R_e.)

Таблица 3: Качество и итоговые оценки экстракторов в поясняющем примере. Мы предположили, что γ = .25 при выводе Q_e из P_e и R_e.

	E_1	E_2	E_3	E_4	E_5
$Q(E_i)$	.01	.01	.06	.22	.17
$R(E_i)$	.99	.5	.99	.33	.17
$P(E_i)$	.99	.99	.85	.33	.25
$Pre(E_i)$	4.6	3.9	2.8	.4	0
$Abs(E_i)$	-4.6	-.7	-4.5	-.15	0

Чтобы завершить описание модели, мы должны указать априорную вероятность различных параметров модели:

$$\theta_1 = \{A_w\}_{w=1}^{W}, \theta_2 = (\{P_e\}_{e=1}^{E}, \{R_e\}_{e=1}^{E}), \theta = (\theta_1 + \theta_2) \quad (8)$$

Для простоты мы использовали равномерно распределенные априорные вероятности параметров. По умолчанию мы задали A_w =0.8, R_e =0.8, и Q_e =0.2. В Разделе 5 мы обсудим альтернативный способ расчёта исходного значения A_w, исходя из относительного количества правильных триад, которые были извлечены из ресурса, используя внешнюю оценку корректности (основанную на *Freebase* [2]).

Пусть $V = \{V_d\}$, $C = \{C_{wdv}\}$, а $Z = (V,C)$ все будут скрытыми переменными. Наша модель определяет следующее совместное распределение:

$$p(X, Z, \theta) = p(0)p(V)p(C|V, \theta_1)p(X|C, \theta_2) \quad (9)$$

Мы можем представить предположения об условной независимости, которые мы сделали, с помощью графической модели, как показано на Рисунке 2. Закрашенный круг – это и есть наблюдаемая переменная, представляющая собой данные. Не закрашенные

круги – это скрытые переменные или параметры. Стрелочки показывают зависимость между переменными и параметрами. Прямоугольники в виде рамок представляют собой так называемые «границы» и отображают повторение включенных в них переменных, например, прямоугольных e повторяется для каждого экстрактора $e \in E$.

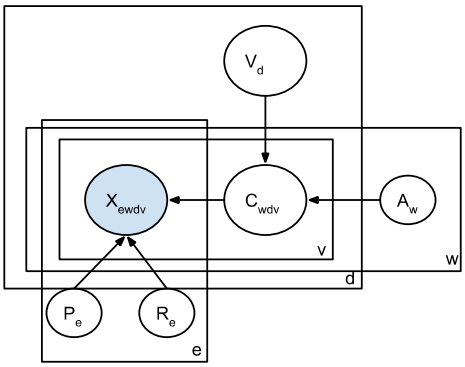

Рисунок 2: Представление многослойной модели с использованием графической модели с обозначением границ.

3.2 Логический вывод

Напомним, что оценка КВТ, в сущности, требует, чтобы мы вычислили апостериорную вероятность, касающуюся интересующих нас параметров $p(A|X)$. С точки зрения вычислений – это трудноразрешимая задача из-за присутствия скрытых переменных Z. Один из способов расчета заключается в использовании аппроксимации данных методом Монте-Карло, например, с помощью генерации выборки по схеме Гиббса, как показано в работе [32]. Однако процесс может протекать довольно медленно и со сложностями в модели распределенных вычислений Map-Reduce, которая требуется для объема данных, используемых нами в данной работе.

Более быстрый альтернативный способ состоит в использовании алгоритма EM, который даст точечную оценку всех параметров, $\hat{\theta} = \text{argmax}\, p(\theta|X)$. Поскольку мы используем равномерно распределенную априорную вероятность, этот вариант является эквивалентом оценки методом максимального правдоподобия $\hat{\theta} = \text{argmax}\, p(X|\theta)$. Отсюда мы можем вывести $\hat{A}$.

Как сказано в работе [26], безошибочный EM алгоритм характеризуется квадратической сложностью даже в случае однослойной модели, поэтому он не подходит для больших объемов данных из сети. Вместо этого алгоритма мы использовали итерационную «EM-подобную» процедуру оценки, в процессе выполнения которой мы задали параметры, как было описано ранее, а затем переходили от Z к оценке θ до полной сходимости.

В данной работе мы сначала приведем краткое описание EM-подобного алгоритма, а затем в следующих разделах перейдем к деталям.

В нашем случае Z состоит из двух «слоев» переменных. Мы скорректируем их последовательно следующим образом. Во-первых, пусть $X_{wdv} = \{X_{ewdv}\}$ - это все извлечения из веб-ресурса w, касающиеся конкретной триады $t = (d, v)$. Мы вычислили правильность извлечения $p(C_{wdv} \mid X_{wdv}, \theta_2^t)$, как описано в Разделе 3.3.1, а затем мы вычислили $\hat{C}_{wdv} = \text{argmax}\, p(C_{wdv}|X_{wdv}, \theta_2^t)$, что является наиболее вероятным предположением об

«истинном контенте» каждого веб-ресурса. Подобные вычисления могут быть выполнены одновременно относительно d, w, v.

Пусть $\hat{C}_d = \hat{C}_{wdv}$ – это все рассчитанные значения для d, охватывающие различные веб-сайты. Мы вычислили $p(V_d | \hat{C}_d, \theta_1^t)$, как описано в Разделе 3.3.2, а затем рассчитали $\hat{V}_d = \text{argmax}\, p(V_d | \hat{C}_d, \theta_1^t)$, что является наиболее вероятным предположением об «истинном значении» каждого элемента данных. Подобные вычисления могут быть выполнены одновременно относительно d.

Рассчитав скрытые переменные, мы рассчитали θ^{t+1}. Корректировка данного параметра также состоит из двух этапов (которые, однако, могут быть выполнены одновременно): выполняется оценка надежности ресурса $\{A_w\}$ и надежности экстракторов $\{P_e, R_e\}$, как описано в Разделе 3.4.

Алгоритм 1 дает краткий обзор псевдокода. Мы расскажем об этом более подробно ниже.

Алгоритм 1: МНОГОСЛОЙНАЯ МОДЕЛЬ (MULTILAYER(X, t_{max}))
Вводные данные: X: все извлеченные данные;
t_{max}: максимальное число логических выводов.
Результат: Оценка Z и θ.

```
1 Определение для θ значений по умолчанию;
2 для t ∈ [1, t_max] выполнить
3     Оценка C с помощью уравнений (15, 26, 31);
4     Оценка V с помощью уравнений (23-25);
5     Оценка θ1 с помощью уравнения (28);
6     Оценка θ2 с помощью уравнений (32-33);
7     если Z, θ сходятся, тогда
8         выход из цикла;
9 результат Z, θ;
```

3.3 Оценка скрытых переменных

Теперь мы переходим к подробному описанию того, как мы рассчитали скрытые переменные Z. Для упрощения обозначений, мы пропустили приведение к требуемым условиям относительно θ^t, за исключением случаев, где это необходимо.

3.3.1 Оценка корректности извлечения

Сначала мы описали, как вычислить $p(C_{wdv}=1|X_{wdv})$ на основе модели «множества истин» [27]. Мы обозначили априорную вероятность $p(C_{wdv} = 1)$ буквой α. В первоначальных итерациях мы присвоили значение $\alpha = 0.5$. Примите во внимание, что, используя фиксированную априорную вероятность, мы разрываем связь между C_{wdv} и V_d в графической модели, как показано на Рисунке 2. Таким образом, при последующих итерациях мы пересчитываем $p(C_{wdv} = 1)$, используя результаты V_d, полученные в предыдущей итерации, как описано в Разделе 3.3.4.

Мы применили правило Байеса следующим образом:

$$
\begin{aligned}
& p(C_{wdv} = 1 | X_{wdv}) \\
&= \frac{\alpha p(X_{wdv} | C_{wdv} = 1)}{\alpha p(X_{wdv} | C_{wdv} = 1) + (1 - \alpha) p(X_{wdv} | C_{wdv} = 0)} \\
&= \frac{1}{1 + \frac{1}{\frac{p(X_{wdv} | C_{wdv} = 1)}{p(X_{wdv} | C_{wdv} = 0)} \frac{\alpha}{1 - \alpha}}} \\
&= \sigma\left(\log \frac{p(X_{wdv} | C_{wdv} = 1)}{p(X_{wdv} | C_{wdv} = 0)} + \log \frac{\alpha}{1 - \alpha} \right)
\end{aligned} \tag{10}
$$

Где $\sigma(x) \triangleq \frac{1}{1+e^{-x}}$ – сигмовидная функция.

Предполагая независимость экстракторов и используя уравнение (6), мы можем вычислить отношение подобия:

$$
\frac{p(X_{wdv} | C_{wdv} = 1)}{p(X_{wdv} | C_{wdv} = 0)} = \prod_{e: X_{ewdv} = 1} \frac{R_e}{Q_e} \prod_{e: X_{ewdv} = 0} \frac{1 - R_e}{1 - Q_e} \tag{11}
$$

Другими словами, для каждого экстрактора мы можем рассчитать *оценку присутствия* Pre_e для триады, которую он извлекает, и *оценку отсутствия* Abs_e для триады, которую он не извлекает:

$$
\text{Pre}_e \triangleq \log R_e - \log Q_e \tag{12}
$$

$$
\text{Abs}_e \triangleq \log(1 - R_e) - \log(1 - Q_e) \tag{13}
$$

Для каждой триады (w, d, v) мы можем вычислить *итоговую оценку*, как сумму имеющихся оценок присутствия и оценок отсутствия:

$$
VCC(w, d, v) \triangleq \sum_{e: X_{ewdv} = 1} \text{Pre}_e + \sum_{e: X_{ewdv} = 0} \text{Abs}_e \tag{14}
$$

Соответственно, мы можем переписать уравнение (10) следующим образом:

$$
p(C_{wdv} = 1 | X_{wdv}) = \sigma\left(VCC(w, d, v) + \log \frac{\alpha}{1 - \alpha} \right) \tag{15}
$$

ПРИМЕР 3.1. *Рассмотрим экстракторы из поясняющего примера (Таблица 2). Предположим, мы знаем значения Q_e и R_e для каждого экстрактора e, как показано в Таблице 3. Тогда мы можем вычислить Pre_e и Abs_e, как показано в той же таблице. Мы видим, что в целом экстрактор с низким значением Q_e (малая вероятность извлечения непредусмотренной триады; например, E_1, E_2) зачастую имеет высокую оценку присутствия; экстрактор с высоким значением R_e (большая вероятность извлечения предусмотренной триады; например, E_1, E_3) зачастую имеет низкую (отрицательную) оценку отсутствия; а экстрактор низкого качества (например, E_5) зачастую имеет низкую оценку присутствия и высокую оценку отсутствия.*

Теперь рассмотрим уравнение 15 для вычисления вероятности того, что конкретный источник содержит триаду t^* = (Обама, гражданство, США) *при условии, что $\alpha = 0.5$. В*

случае ресурса W_1 мы видим, что экстракторы E_1 –E_4 извлекают t, поэтому итоговая оценка будет равна* (4.6+3.9+2.8+ 0.4) + (0) = 11.7. *Таким образом,* $p(C_{1,t^*} = 1|X_{w,t^*}) = \sigma(11.7) = 1$. *В случае ресурса W_6 мы видим, что только E_4 извлекает t*, поэтому итоговая оценка будет равна* (0.4) + (−4.6 − 0.7 − 4.5 − 0) = −9.4, *а* $p(C_{6,t^*} = 1|X_{6,t^*}) = \sigma(-9.4) = 0$. *Некоторые другие значения для* $P(C_{wt} = 1|X_{wt})$ *отображены в Таблице 4.*

Таблица 4: Корректность извлечения и распределение значений элемента данных для данных Таблицы 2 с использованием параметров извлечения, указанных в Таблице 3. Столбцы 2-4 показывают значения $p(C_{wdv} = 1|X_{wdv})$, описано в Примере 3.1. Последняя строка показывает значение $p(V_d|\hat{C}_d)$, описано в Примере 3.2. Примите во внимание тот факт, что данное распределение не дает в сумме 1.0, так как в таблице представлены не все значения.

	США	Кения	Сев. Америка
W_1	1	0	-
W_2	1	-	0
W_3	1	-	0
W_4	1	0	-
W_5	-	1	-
W_6	0	1	-
W_7	-	.07	-
W_8	-	0	-
$p(V_d\|\hat{C}_d)$	0.995	0.004	0

Вычислив $p(C_{wdv} =1|X_{wdv})$, мы можем рассчитать $\hat{C}_{wdv}$ = argmax $p(C_{wdv}|X_{wdv})$. Это значение послужит вводными данными на следующем этапе логического вывода.

3.3.2 Оценка истинного значения элемента данных

На этом этапе мы вычислили $p(V_d = v|C_d)$ на основе модели «единственной истины» [8]. По правилу Байеса, мы имеем:

$$p(V_d = v|\hat{C}_d) = \frac{p(\hat{C}_d|V_d = v)p(V_d = v)}{\sum_{v' \in \mathrm{dom}(d)} p(\hat{C}_d|V_d = v')p(V_d = v')} \tag{16}$$

Поскольку мы заранее не предполагали ничего о правильных значениях, мы предположили, что априорная вероятность распределена равномерно $p(V_d = v)$, поэтому нам просто необходимо было сконцентрироваться на подобии. Используя уравнение (5), мы получили:

$$p(\hat{C}_d|V_d = v)$$

$$= \prod_{w:\hat{C}_{wdv}=1} A_w \prod_{w:\hat{C}_{wdv}=0} \frac{1 - A_w}{n} \tag{17}$$

$$= \prod_{w:\hat{C}_{wdv}=1} \frac{nA_w}{1 - A_w} \prod_{w:\hat{C}_{wdv}\in\{0,1\}} \frac{1 - A_w}{n} \tag{18}$$

Поскольку последний член $\prod_{w:\hat{C}_{wdv}\epsilon\{0,1\}} \frac{1-A_w}{n}$ является константой по отношению к *v*, мы можем опустить его.

Теперь мы можем определить итоговую оценку следующим образом:

$$VCV(w) \triangleq \log \frac{nA_w}{1-A_w} \tag{19}$$

Объединив вместе данные со всех веб-ресурсов, которые содержат нужную триаду, мы получим:

$$VCV(d,v) \triangleq \sum_w \mathbb{I}\left(\hat{C}_{wdv}=1\right) VCV(w) \tag{20}$$

С этим выражением мы можем записать уравнение (16) следующим образом:

$$p\left(V_d = v|\hat{C}_d\right) = \frac{\exp\left(VCV(d,v)\right)}{\sum_{v'\in \mathrm{dom}(d)} \exp\left(VCV(d,v')\right)} \tag{21}$$

ПРИМЕР 3.2. *Предположим, что мы правильно решили, что триада, предлагаемая каждым веб-ресурсом, такая, какой мы ее показали в столбце «Значение» в Таблице 2. Предположим, каждый источник характеризуется одинаковой надежностью* A_w *=0.6, а n=10, таким образом, итоговая оценка* $\ln\left(\frac{10*0.6}{1-0.6}\right) = 2.7$. *Тогда значение* США *имеет итоговую оценку 2.7*4 = 10.8,* Кения *- 2.7*2 = 5.4, а непредусмотренное значение, такое как* Северная Америка*, имеет итоговую оценку 0. Поскольку интервал содержит 10 ложных значений, поэтому существует 9 непредусмотренных значений. В результате мы получаем* $p\left(V_d = USA\middle|\hat{C}_d\right) = \frac{\exp(10.8)}{Z} = 0.995$, *где Z =* exp(10.8) + exp(5.4) + exp(0) * 9. *Подобным образом* $p\left(V_d = Kenya\middle|\hat{C}_d\right) = \frac{\exp(5.4)}{Z} = 0.004$. *Эти значения показаны в последней строке в Таблице 4. Недостающая часть выражения* 1 – (0.995 + 0.004) *распределена (равномерно) между остальными 9 значениями, которые не рассматривались (но существуют в интервале).*

3.3.3 Усовершенствованная процедура оценки

Вплоть до этого момента мы предполагали, что сначала необходимо вычислять MAP оценку $\hat{C}_{wdv}$, которую мы затем использовали в качестве результата для оценки V_d. Однако это приводит к игнорированию неопределенности в $\hat{C}$. Было бы более правильно вычислить $p(Vd|Xd)$, ограничивая по C_{wdv}.

$$\begin{aligned} p(V_d|X_d) &\propto P(V_d)P(X_d|V_d) \\ &= P(V_d)\sum_{\vec{c}} p(C_d=\vec{c}|V_d)p(X_d|V_d) \end{aligned} \tag{22}$$

Мы можем рассмотреть $\vec{c}$, как возможный мир, в котором каждый элемент c_{wdv} указывает на то, содержит ли ресурс *w* триаду (d, v) (значение 1), или не содержит (значение 0).

В случае данного подхода мы заменили использованную ранее итоговую оценку взвешенной:

$$VCV'(w,d,v) \triangleq p(C_{wdv}=1|X_d)\log\frac{nA_w}{1-A_w} \tag{23}$$

$$VCV'(d,v) \triangleq \sum_w VCV'(w,d,v) \tag{24}$$

Затем мы вычислили:

$$p(V_d=v|X_d) \approx \frac{\exp\big(VCV'(d,v)\big)}{\sum_{v'\in \mathrm{dom}(d)}\exp\big(VCV'(d,v')\big)} \tag{25}$$

3.3.4 Переоценка априорной вероятности корректности

В Разделе 3.3.1 мы предположили, что $p(C_{wdv} = 1) = \alpha$ - известное значение, которое разрывает связи сижу V_d и C_{wdv}. Таким образом, мы корректировали эту априорную вероятность после каждой итерации, исходя из корректности значения и надежности ресурса:

$$\hat{\alpha}^{t+1} = p(V_d=v|X)A_w + \big(1-p(V_d=v|X)\big)(1-A_w) \tag{26}$$

Мы можем использовать эту улучшенную оценку в следующих итерациях. Приведем пример.

ПРИМЕР 3.3. *Рассмотрим вероятность того, что ресурс W_7 предоставляет триаду t'= (Обама, гражданство, Кения). Два экстрактора извлекают триаду t' из ресурса W_7 и итоговая оценка составляет* -2.65*. Таким образом, первоначальная оценка $p(C_{wdv} = 1|X) = \sigma(-2.65) = 0.06$. Однако после того, как была завершена предыдущая итерация, мы знаем, что $p(V_d$ = Кения$|X)$ = 0.04. Это дает нам следующую скорректированную априорную вероятность: $p'(C_{wt}$ =1)=0.004*0.6+(1−0.004)*(1−0.6)=0.4, исходя из того, что A_w =0.6. Таким образом, обновленная апостериорная вероятность задается $p'(C_{wt}=1|X) = \sigma\left(-2.65+\log\frac{1-0.4}{0.4}\right) = 0.04$, значение которой ниже, чем было раньше.*

3.4 Оценка параметров качества

Оценив скрытые переменные, теперь мы переходим к оценке параметров модели.

3.4.1 Качество ресурса

Используя работу [8], мы рассчитали надежность ресурса, вычислив среднюю вероятность того, что содержащиеся в нем значения, являются истинными:

$$\hat{A}_w^{t+1} = \frac{\sum_{dv:\hat{C}_{wdv}=1} p(V_d=v|X)}{\sum_{dv:\hat{C}_{wdv}=1} 1} \tag{27}$$

Мы можем принять во внимание неопределенность $\hat{C}$ следующим образом:

$$\hat{A}_w^{t+1} = \frac{\sum_{dv:\hat{C}_{wdv}>0} p(C_{wdv}=1|X)p(V_d=v|X)}{\sum_{dv:\hat{C}_{wdv}>0} p\,(C_{wdv}=1|X)} \tag{28}$$

Это ключевое уравнение, лежащее в основе оценки доверия, основанном на знаниях. Оно позволяет оценить надежность веб-ресурса, как средневзвешенное значение вероятности содержания (предоставления) им (ресурсом) фактов, где в качестве весовых коэффициентов выступает вероятность того, что ресурс действительно содержит эти факты.

3.4.2 Качество экстрактора

В соответствии с определением точности и полноты мы можем рассчитать эти параметры следующим образом:

$$\hat{P}_e^{t+1} = \frac{\sum_{wdv:X_{ewdv}=1} p(C_{wdv} = 1|X)}{\sum_{wdv:X_{ewdv}=1} 1} \quad (29)$$

$$\hat{R}_e^{t+1} = \frac{\sum_{wdv:X_{ewdv}=1} p(C_{wdv} = 1|X)}{\sum_{wdv} p\,(C_{wdv} = 1|X)} \quad (30)$$

Примите во внимание, что по причинам, указанным в работе [27], более надежным считается расчёт показателей P_e и R_e на основании данных, а затем уже вычисление Q_e с использованием уравнения (7), нежели попытка напрямую оценить показатель Q_e.

3.5 Обработка извлеченных значений, взвешенных по степени уверенности

До сих пор мы предполагали, что каждый экстрактор в качестве результата выдает бинарный ответ того, извлекает ли он триаду или нет, $X_{ewdv} \in \{0, 1\}$. Однако в реальной жизни экстракторы в качестве результата возвращают оценки доверия, которые мы можем интерпретировать как вероятность того, что эта триада присутствует на странице согласно экстрактору. Обозначим такое «субъективное доказательство» $p(X_{ewdu} = 1) = \bar{X}_{ewdu} \in [0,1]$. Простейшим способом обработки таких данных является их перевод в двоичный с помощью ввода пороговой величины. Однако этот метод приводит к потере информации, что показано в примере ниже.

ПРИМЕР 3.4. *Рассмотрим случай, когда экстракторы E_1 и E_3 не полностью уверены в извлеченных ими значениях из ресурсов W_3 и W_4. В частности, экстрактор E_1 присваивает каждому извлечению вероятность (то есть уверенность), равную .85, в экстрактор E_3 присваивает вероятность .5. Хотя ни один из экстракторов не уверен полностью в извлеченных значениях, после совместного изучения извлеченных ими значений, мы, пожалуй, можем быть вполне уверены в том, что ресурсы W_3 и W_4 на самом деле содержат триаду T =* (Обама, гражданство, США).

Однако, если бы мы применили порог в .7, мы бы проигнорировали значения, извлеченные экстрактором E_3, из ресурсов W_3 и W_4. По причине недостатка извлеченных значений, мы бы сделали вывод о том, что ни ресурс W_3, ни W_4 не содержат триаду Т. Тогда, так как значение США *содержат ресурсы W_1 и W_2, в то время как значение* Кения *содержат ресурсы W_5 и W_6, а все ресурсы характеризуются одинаковой точностью, мы бы вычислили равную вероятность для значений* США *и* Кения.

Используя тот же подход, который применялся в уравнении (23), мы предложили изменить уравнение (14) следующим образом:

$$VCC'(w,d,v) \triangleq \sum_e [p(X_{ewt}=1)\mathrm{Pre}_e + p(X_{ewt}=0)\mathrm{Abs}_e] \tag{31}$$

Подобным же образом мы изменили расчет точности и полноты:

$$\hat{P}_e = \frac{\sum_{wdv:\bar{X}_{ewdv}>0} p(X_{ewdv}=1)\, p(C_{wdv}=1|X)}{\sum_{wdv:\bar{X}_{ewdv}>0} p(X_{ewdv}=1)} \tag{32}$$

$$\hat{R}_e = \frac{\sum_{wdv:\bar{X}_{ewdv}>0} p(X_{ewdv}=1)\, p(C_{wdv}=1|X)}{\sum_{wdv} p(C_{wdv}=1|X)} \tag{33}$$

4. ДИНАМИЧЕСКИЙ ВЫБОР СТЕПЕНИ РАЗБИЕНИЯ

В данном разделе описан порядок выбора степени разбиения веб-ресурса. В конце раздела мы обсудим, как применить этот метод к экстракторам. Этот этап реализовывается до применения многослойной модели.

В идеале мы хотели разбить ресурс на как можно большее количество составляющих. Например, принято считать каждую веб-страницу в качестве отдельного источника, так как такая страница может характеризоваться надежностью, отличной от надежности других страниц. Мы же можем определить ресурс даже как конкретный предикат на конкретной странице, что позволит нам оценить, насколько достоверной является страница в пределах конкретного типа предиката. Однако, если определенные нами ресурсы будут слишком малы, мы можем получить слишком маленькое число данных для достоверной оценки надежности ресурсов. И напротив, могут существовать ресурсы, которые при всем своем маленьком размере, могут содержать огромное количество данных, что может привести к сложностям при вычислениях.

Чтобы решить данную проблему, мы решили динамически выбирать степень разбиения веб-ресурсов. В случае слишком маленьких веб-ресурсов мы можем «возвращаться» к предыдущему уровню иерархии, что позволит нам «занять статистической прочности» у смежных страниц. В случае более крупных ресурсов мы можем прибегнуть к их разбиению на множества ресурсов и независимой оценке их надежности. При объединении ресурсов наша цель состоит в улучшении статистического качества наших расчетов без ущерба для эффективности. При разбиении наша цель заключается в существенном улучшении эффективности при наличии расфазировки данных без существенного изменения наших расчетов.

Чтобы быть более точными, мы можем определить ресурс на множестве уровней разложения, путем задания следующих значений вектора признаков: ⟨веб-сайт, предикат, веб-страница), расположенных от наиболее общего к наиболее конкретному. Теперь мы можем выстроить эти ресурсы в иерархию. Например, ⟨wiki.com⟩ исходный элемент для ⟨wiki.com, дата рождения⟩, который в свою очередь является исходным элементом для ⟨wiki.com, дата рождения, wiki.com/page1.html⟩. Мы определили следующие две операции.

- **Разбиение:** При разбиении больших ресурсов, мы разбиваем их произвольно на подресурсы одинакового размера. Например, пусть W будет ресурсом размером $|W|$, а M – максимальный нужный нам размер. Мы равномерно распределяем триады из W на интервалы $\left\lceil \frac{|W|}{M} \right\rceil$, каждый из которых представляет собой подресурс. Положим M равным такому большому числу, которое с одной стороны не вызовет ненужного разбиения ресурсов, а с другой стороны не приведет к сложностям в вычислениях.
- **Объединение:** При объединении небольших ресурсов, мы хотим объединить только те ресурсы, которые обладают одинаковыми признаками, например, делят один предикат, или происходят от одного веб-сайта. Таким образом, мы объединяем только дочерние элементы одного материнского элемента в иерархии. Положим m равным такому наименьшему числу, которое с одной стороны не потребует ненужного объединения ресурсов, а с другой стороны позволит сохранить достаточную статистическую прочность.

ПРИМЕР 4.1. *Рассмотрим три ресурса:* ⟨website1.com, дата рождения⟩, ⟨website1.com, место рождения⟩, ⟨website1.com, пол⟩, *каждый с двумя триадами, чего возможно не вполне достаточно для оценки качества. Мы можем объединить эти ресурсы в их материнском ресурсе, удалив второй признак. Тогда мы получим ресурс* ⟨website1.com⟩ *размера* 2*3 = 6*, который предоставляет больше данных для оценки качества.*

Алгоритм 2: Разбиение и объединение (**W**, m, M)
Вводные данные: **W**: наиболее раздробленные ресурсы;
m/M: мин/макс размер ресурса по желанию.
Результат: **W′**: новый набор желаемого размера.

```
1  W′←∅;
2  для W ∈ W выполнить
3      W ← W \ {W};
4      если |W| > M тогда
5          W′ ← W′∪ SPLIT(W);
6      в противном случае если |W| < m тогда
7          W_par ← GETPARENT (W);
8          если W_par =⊥ тогда
9              // Уже достигнута вершина иерархии
               W′←W′∪{W};
10         в противном случае
11             W ← W ∪ {W_par};
12     в противном случае
13         W′ ←W′ ∪{W};
14 результат W′;
```

Примите во внимание тот факт, что, когда мы объединяем небольшие ресурсы, результирующий материнский ресурс может не достигнуть желаемого размера: он может быть по-прежнему слишком мал, или же он может быть слишком велик после

объединения нами большого числа небольших ресурсов. Как следствие, нам, возможно, потребуется итерационно объединять результирующие ресурсы до размера материнских, или же также разбивать чрезмерно крупные ресурсы, что мы и показали в полном алгоритме.

Алгоритм 2 представляет собой алгоритм РАЗБИЕНИЯ И ОБЪЕДИНЕНИЯ. Мы использовали обозначение **W** для ресурсов в целях проверки, а **W′** - для итоговых результатов. В начале ресурс **W** содержит ресурсы, разделенные мелкие части, а **W′** = ∅. Мы рассматриваем каждый $W \in \mathbf{W}$. Если W слишком велик, мы применяем процедуру РАЗБИЕНИЕ для разбиения ресурса на ряд подресурсов. Процедура РАЗБИЕНИЕ гарантирует, что каждый подресурс приобретет необходимый размер, поэтому мы добавляем подресурсы к **W′**. Если W слишком мал, мы получаем материнский ресурс. В случае, если W уже достиг вершины иерархии и у него не может быть материнского ресурса, мы добавляем его к **W′**; в противном случае мы добавляем W_{par} обратно к **W**. В итоге, ресурсы, достигнувшие нужного размера, мы перемещаем непосредственно в **W′**.

ПРИМЕР 4.2. *Рассмотрим массив из 1000 ресурсов* $\langle W, P_i, URL_i \rangle$, $i \in [1, 1000]$*; иными словами, они принадлежат одному веб-сайту, каждый имеет отличный предикат и отличный URL. Предположим мы хотим иметь ресурс размером* [5,500]*. Метод MULTILAYERSM предполагает три этапа.*

На первом этапе каждый ресурс рассматривается, как слишком маленький и заменяется материнским ресурсом $\langle W, P_i \rangle$*. На втором этапе каждый новый ресурс по-прежнему считается слишком маленьким и заменяется материнским ресурсом* $\langle W \rangle$*. На третьем этапе один оставшийся ресурс рассматривается, как очень большой, и равномерно разбивается на два подресурса. Выполнение алгоритма прекращается, когда мы получаем 2 ресурса, каждый размером 500.*

Мы хотим обратить внимание на то, что эти же методы также применимы к экстракторам. Мы задаем экстрактор, используя следующий вектор признаков, также располагая признаки в порядке от наиболее общего к наиболее конкретному: ⟨экстрактор, шаблон, предикат, вебсайт⟩. Наибольшая степень разбиения отображает качество конкретного шаблона экстрактора (различные шаблоны могут отличаться качеством), при извлечении для конкретного предиката (в некоторых случая, когда шаблон может извлекать триады различных предикатов, он может отличаться качеством), из конкретного веб-сайта (шаблон может отличаться качеством на разных веб-сайтах).

5. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

В данном разделе описаны результаты экспериментов, проведенных с использованием массива смоделированных данных (при этом нам известна реальные истинные значения), и с использованием большого объема реальных данных. Мы покажем, (1) что наш алгоритм может эффективно оценивать корректность извлечений, достоверность триад и надежность ресурсов; (2) наша модель существенно увеличивает эффективность

передовых методов, используемых для интеграции данных; и (3) KBT предлагает ценный дополнительный сигнал для оценки качества веб-ресурса.

5.1 Подготовка эксперимента

5.1.1 Метрические значения

Мы измерили то, насколько хорошо мы предсказали правильность извлечения, вероятность нахождения триад и надежность ресурса. В случае смоделированных данных у нас было преимущество в виде реальной истины, поэтому мы смогли точно измерить все три показателя. Количественное значение мы определили через *квадратичную функцию потерь*: чем меньше квадратичные потери, тем лучше. В частности, SqV определяет средние квадратичные потери между $p(V_d = v|X)$ и истинным значением ‖ $(V_d^* = v)$; SqC определяет средние квадратичные потери в интервале между $p(C_{wdv} = 1|X)$ и истинным значением ‖ $(C_{wdv}^* = 1)$; а SqA определяет средние квадратичные потери в интервале между $\hat{A}_w$ и истинным значением A_w^*.

В случае реальных данных, как будет показано ниже, у нас не было эталонного значения для определения достоверности ресурса. У нас был лишь частичный эталон для определения корректности триады и корректности извлечения. В результате, в случае реальных данных мы просто сконцентрировали усилия на измерении того, насколько хорошо нам удается предсказывать достоверность триады. Помимо SqV мы в этих же целях использовали следующие три показателя, которые ранее были использованы в работе [11].

- *Взвешенная ошибка (WDev)*: WDev определяет, насколько *выверенными* являются предсказанные вероятности. Мы разделили наши триады в соответствии с предсказанными вероятностями на интервалы [0, 0.01), . . . , [0.04, 0.05), [0.05, 0.1), . . . , [0.9, 0.95), [0.95, 0.96), . . . , [0.99, 1), [1, 1] (большинство триад попало в интервал [0, 0.05) и [0.95, 1], поэтому к этим интервалам мы применили наибольшую степень разбиения). Для каждого интервала мы вычислили надежность триад, используя эталон, которая может считаться реальной вероятностью существования триад. WDev вычисляет средние квадратичные потери из предсказанных вероятностей и реальных вероятностей, взвешенные по числу триад в каждой группе (чем это значение меньше, тем лучше).
- *Область под кривой точность-полнота (AUC-PR)*: показывает, являются ли предсказанные вероятности *монотонными*. Мы расположили триады по порядку в зависимости от вычисленных вероятностей и построили кривые точности-полноты (кривые PR). Ось X отображает значения полноты, ось Y- значения точности. AUC-PR вычисляет область под кривой (чем она больше, тем лучше).
- *Охват (Cov)*: показывает, для какого процента триад мы вычисляем вероятность (как будет показано ниже, мы можем игнорировать данные ресурсов, чье качество остается равным значению, присвоенному по умолчанию, на протяжении всех итераций).

Примите во внимание тот факт, что в случае смоделированных данных Cov = 1 для всех методов, а сравнение различных методов относительно AUC-PR и WDev очень схоже со сравнением относительно SqV, поэтому мы пропустили графики.

5.1.2 Методы для сравнения

Мы сравнили три основных метода. Первый, названный SINGLELAYER (ОДНОСЛОЙНЫЙ), задействует передовые методы, используемые для интеграции знаний [11] (вопрос рассматривается в Разделе 2). В частности, каждый ресурс или «источник» представляет собой набор из 4 элементов (экстрактор, веб-сайт, предикат, шаблон). Мы анализировали источник при интеграции, только если его точность не оставалась равной заданному по умолчанию значению на протяжении итераций по причине малого охвата. Мы задали n = 100 и выполняли итерации 5 раз. Эти вводные данные, продемонстрированные в работе [11], показали наилучшие результаты.

Второй метод, который мы назвали MULTILAYER (МНОГОУРОВНЕВЫЙ), задействует многослойную модель, описанную в Разделе 3. Чтобы добиться приемлемого времени расчета, мы использовали наибольшую степень разбиения, как описано в Разделе 4, для экстракторов и ресурсов: каждый экстрактор - это вектор (экстрактор, шаблон, предикат, веб-сайт), а каждый ресурс – это вектор (веб-сайт, предикат, веб-страница). После того, как мы определили корректность извлечения, мы проанализировали уверенность, демонстрируемую экстракторами, приведенную к [0, 1], как показано в Разделе 3.5. Если экстрактор не дает уверенности, мы принимаем уверенность =1. При определении достоверности триад по умолчанию мы использовали улучшенную оценку $p(C_{wdv} = 1|X)$, описанную в Разделе 3.3.3, вместо простого использования $\hat{C}_{wdv}$. Мы начали корректировать априорные вероятности $p(C_{wdv} = 1)$, как описано в Разделе 3.3.4, начав с третьей итерации, поскольку вероятности, рассчитанные нами, приобрели устойчивость после второй итерации. Для моделей помех мы задали n = 10 и γ = 0.25, но мы нашли другие вводные значения, приводящие точно к таким же результатам. Мы изменяли вводные данные и показали влияние таких изменений в Разделе 5.3.3.

Третий метод, который мы назвали MULTILAYERSM (МНОГОСЛОЙНЫЙ С РАЗБИЕНИЕМ И ОБЪЕДИНЕНИЕМ), задействует алгоритм разбиения и объединения, а также использует многослойную модель, как описано в Разделе 4. Мы задали соответствующие минимальные и максимальные размеры m = 5 и M = 10тыс., как значения по умолчанию, и изменяли их в Разделе 5.3.4.

Для каждого из методов существует по два варианта. Первый вариант определяет, какую из версий вероятностных моделей мы используем $p(X_{ewdv}|C_{wdv})$. Мы попробовали использовать обе модели: ACCU и POPACCU. Мы обнаружили, что результаты обоих вариантов при использовании однослойной модели были очень близкими, хотя модель POPACCU давала чуть более хорошие результаты. Но к удивлению, мы выяснили, что версия многослойной модели, использующей вариант POPACCU, оказалась хуже версии ACCU. Это объясняется тем, что мы до сих пор не нашли способ комбинирования модели POPACCU с улучшенной процедурой оценки, описанной в Разделе 3.3.3. В результате, ниже мы опубликовали только результаты версии ACCU.

Второй вариант определяет то, как мы присваиваем начальные значения качеству ресурса. Мы либо присваиваем значения по умолчанию (A_w = 0.8, R_e = 0.8, Q_e = 0.2), или же задаем значения качества в соответствии с эталоном, как описано в Разделе 5.3. В последнем случае мы добавили знак «+» к названию методов, чтобы отличать их от методов, в которых присваиваются исходные значения по умолчанию (например, SINGLELAYER+).

5.2 Эксперименты с использованием смоделированных данных

5.2.1 Массив данных

Мы произвольным образом сгенерировали массивы данных, содержащие 10 ресурсов и 5 экстракторов. Каждый экстрактор предоставляет 100 триад с надежностью $A = 0.7$. Каждый экстрактор извлекает из ресурса триады с вероятностью $\delta = 0.5$; для каждого ресурса он извлекает предусмотренную триаду с вероятностью $R = 0.5$; надежность среди извлеченных предметов (такая же и для предикатов, и для объектов) составляет $P = 0.8$ (другими словами, надежность экстрактора $P_e = P^3$). В каждом эксперименте мы изменяли один параметр от 0.1 до 0.9 и не меняли все остальные. Каждый эксперимент мы повторяли 10 раз и выводили среднее значение. Примите во внимание, что наши вводные значения по умолчанию отражают наиболее сложный случай, когда качество экстракторов и ресурсов находится на сравнительно низком уровне.

5.2.2 Результаты

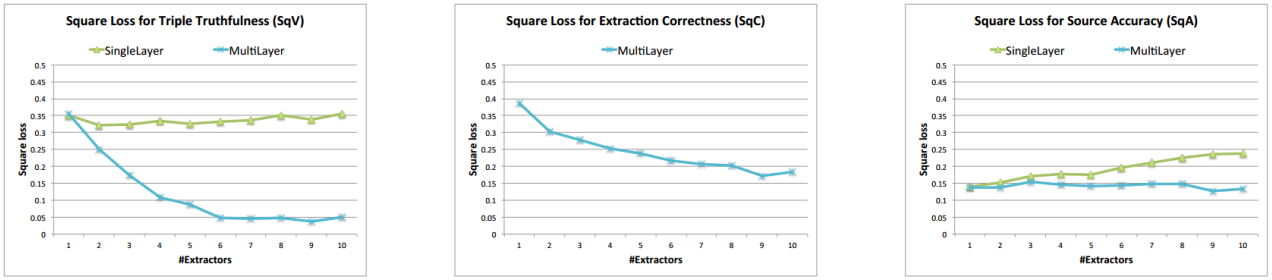

Рисунок 3. Ошибки при оценке V_d, C_{wdv} и A_w по мере изменения числа экстракторов в случае с использованием смоделированных данных. Многослойная модель демонстрирует значительно меньшее значение квадратичных помех, нежели однослойная модель. Однослойная модель не может оценить C_{wdv}, что приводит к отображению одного графика для SqC.

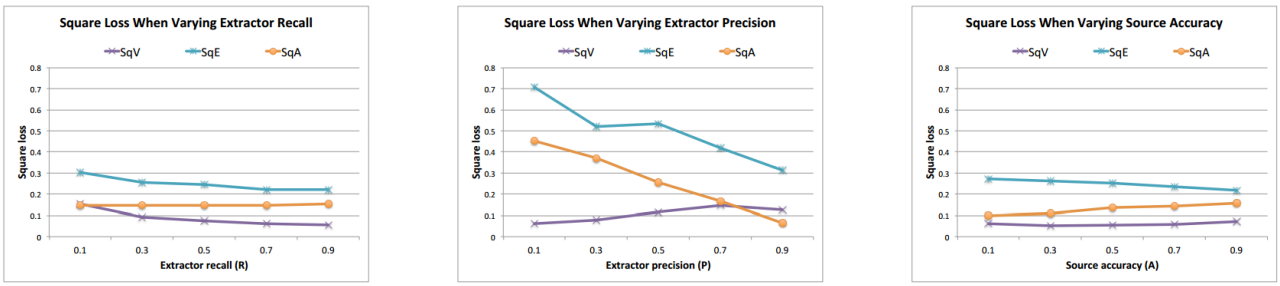

Рисунок 4: Ошибки при оценке V_d, C_{wdv} и A_w по мере изменения качества экстракторов (P и R) и качества ресурса (A) в случае с использованием смоделированных данных.

На рисунке 3 показаны три графика для SqV, SqC, и SqA, так как мы увеличили число экстракторов. Мы предположили, что метод SINGLELAYER анализирует все извлеченные триады при вычислении надежности ресурса. Мы увидели, что многослойная модель всегда дает лучшие результаты, нежели однослойная модель. По мере увеличения числа

экстракторов, график SqV начинает быстро стремиться вниз в случае многослойной модели. График SqC также стремится вниз, хотя и более медленно. Несмотря на то, что дополнительные экстракторы могут привносить гораздо больше извлечений-помех, график SqA остается стабильным в случае метода MULTILAYER, в то время, как он стремится вверх в случае с методом SINGLELAYER.

Далее мы изменяли качество ресурса и экстрактора. Метод MULTILAYER продолжал выглядеть лучше, чем метод SINGLELAYER. Поэтому на рисунке 4 продемонстрированы графики только для метода MULTILAYER с изменением значений R, P и A (график с изменением δ схож с графиком при изменении R). В целом, можно наблюдать, что чем выше качество, тем меньше потерь. Хотя существует несколько небольших отклонений от этого тренда. При возрастании полноты экстрактора (R), SqA не уменьшается, поскольку экстракторы также привносят больше помех. При возрастании точности (P), начинаем им больше доверять, что приводит к слегка более высокой (хотя все еще низкой) вероятности нахождения ложных триад. Поскольку ложных триад гораздо больше, чем истинных, SqV слегка растет. Точно также, при увеличении A, наблюдается небольшой рост SqA, поскольку мы немного больше доверяем ложным триадам. Однако, в целом, мы считаем, что эксперименты с использованием смоделированных данных показали, что наш алгоритм работает ожидаемым образом и может успешно приблизительно определять истинные значения параметров в условиях использования контролируемых вводных данных.

5.3 Эксперименты с данными KV

5.3.1 Массив данных

Мы экспериментировали с триадами, собранными в ходе реализации проекта «Хранилище знаний» (Knowledge Vault) [10] 24.7.2014. Для простоты мы назвали эти данные массивом данных *KV*. Было извлечено 2.8 миллиарда триад с более чем 2 миллиардов веб-страниц с помощью 16-ти экстракторов, включающих 40 миллионов шаблонов извлечения. По сравнению с предыдущей версией данных, собранных 2.10.2013 [11], данная выборка на 75% больше, используется на 25% больше экстракторов, на 8% больше шаблонов извлечения, было обработано в двое больше веб-страниц.

На Рисунке 5 показано распределение количества отдельных извлеченных триад из расчета на URL и на шаблон извлечения. С одной стороны, мы видим некоторое количество гигантских ресурсов и экстракторов: 26 URL привносят более 50 тысяч триад каждый (много за счет ошибок извлечения), 15 веб-сайтов привносят более 100 миллионов триад, а 43 шаблона извлечения извлекают более 1 миллиона триад каждый. С другой стороны, мы наблюдаем «длинные хвосты»: 74% URL привносят менее 5 триад каждый, и 48% шаблонов извлечения извлекают менее 5 триад каждый. Такие наблюдения приводят нас к использованию нашей стратегии SPLITANDMERGE (разбиения и объединения).

Чтобы определить, являются ли эти триады истинными или нет (маркеры эталона), мы использовали два метода. Первый метод называется *«Предположение о локально замкнутом мире» (Local-Closed World Assumption (LCWA))* [10, 11, 15], который работает

следующим образом. Триада (*s*, *p*, *o*) считается *истинной*, если она появляется в базе знаний Freebase. Если триада отсутствует в базе знаний, но имеется (*s*, *p*) для любого иного значения *o'*, мы предполагаем, что база знаний локально полная (для (*s*, *p*)), тогда мы помечаем триаду (*s*, *p*, *o*) как *ложную*. Остальные триады (где отсутствует (*s*, *p*)) помечаем как *неизвестные* и убираем из массива для оценки. Таким образом мы смогли определить достоверность 0.74 миллиарда триад (26% от *KV*), 20% из которых являются истинными (находятся в Freebase).

Второй метод позволяет нам использовать проверку типов для нахождения неправильных извлечений. В частности, мы считаем триаду (*s*, *p*, *o*) ложной, если 1) *s* = *o*; 2) тип *s* или *o* несовместим с типом, который требует предикат; или 3) *o* находится вне ожидаемых пределов (например, вес атлета составляет более 450 кг). Мы обнаружили 0.56 миллиардов триад (20% от KV), которые нарушают эти правила, и приняли их как за *ложные* триады, так и за ошибки извлечения.

Наш эталон включает триады, полученные в результате применения обоих методов маркировки. Он включает в сумме 1.3 миллиарда триад, среди которых 11.5% являются истинными.

5.3.2 Однослойная модель против многослойной

Таблица 5: Сравнение различных методов, используемых применительно к *KV*; лучшие результаты в каждой группе помечены жирным шрифтом. Для SqV и WDev, чем значение меньше, тем лучше; для AUC-PR и Cov, чем выше, тем лучше.

SqV	SqV	WDev	AUC-PR	Cov
SINGLELAYER	0.131	0.061	**0.454**	**0.952**
MULTILAYER	0.105	0.042	0.439	0.849
SINGLELAYERSM	**0.090**	**0.021**	0.449	0.939
SINGLELAYER +	0.063	0.0043	0.630	0.953
MULTILAYER +	**0.054**	0.0040	**0.693**	0.864
SINGLELAYERSM +	0.059	**0.0039**	0.631	**0.955**

В Таблице 5 приведено сравнение результатов использования трех методов. На рисунке 8 показан калибровочный график, а на Рисунке 9 график кривой PR. Мы видим, что все методы довольно хорошо выверены, но многослойная модель демонстрирует лучшую кривую PR. В частности, метод SINGLELAYER часто прогнозирует низкую вероятность для истинных триад, а следовательно предлагает большое количество ложноотрицательных результатов.

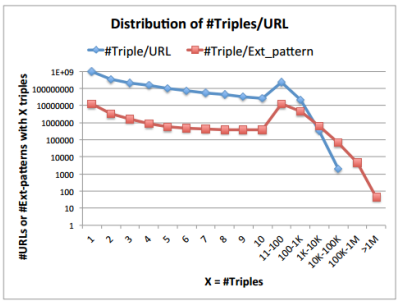

Рисунок 5: Распределение #триад в расчете на URL или шаблон извлечения приводит к использованию стратегии SPLITANDMERGE.

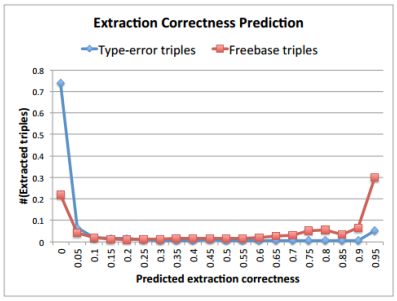

Рисунок 6: Распределение предсказанной корректности извлечения демонстрирует эффективность метода MULTILAYER+.

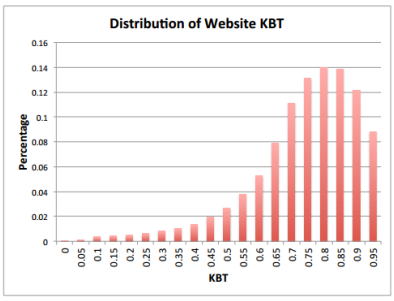

Рисунок 7: Распределение KBT для веб-сайтов, для которых характерно извлечение по меньшей мере 5 триад.

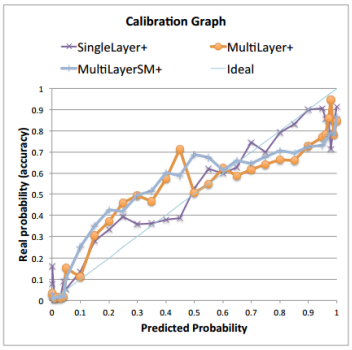

Рисунок 8: Калибровочные графики для различных методов, примененных к данным KV.

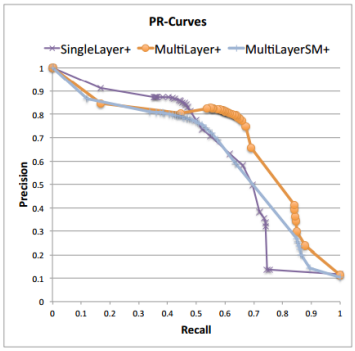

Рисунок 9: Кривые PR - для различных методов, примененных к данным KV. Метод MULTILAYER+ имеет наилучшую кривую.

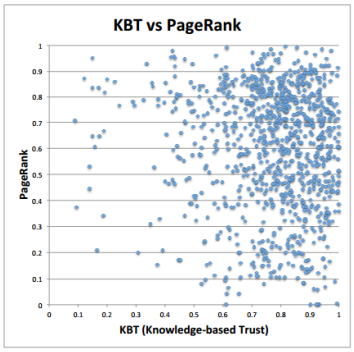

Рисунок 10: KBT и PageRank представляют собой ортогональные сигналы.

Мы видим, что метод MULTILAYERSM демонстрирует лучшие результаты по сравнению с MULTILAYER, но на удивление метод MULTILAYERSM+ характеризуется более низкой производительностью по сравнению с MULTILAYER+. Говоря иными словами, существует зависимость между степенью разбиения ресурсов и способом, с помощью которого мы задаем первоначальное значение их точности.

Причина кроется в следующем. Когда мы определяем первоначальные значения качества ресурса и экстрактора с помощью значений, используемых по умолчанию, мы применяем не проконтролированную выборку (без маркированных данных). При таком условии метод MULTILAYERSM объединяет небольшие ресурсы таким образом, чтобы он смог лучше спрогнозировать их качество, и поэтому этот метод лучше, чем стандартный метод MULTILAYER. Теперь рассмотрим случай, когда мы определяем качество ресурса и экстрактора, используя эталон. В таком случае мы используем наполовину проконтролированную выборку. «Умное» присвоение начальных значений помогает особенно тогда, когда мы разбиваем ресурсы и экстракторы на мелкие составляющие, так как в таких случаях мы часто имеем намного меньше данных в отношении конкретного ресурса или экстрактора.

В итоге, чтобы проверить качество сделанных нами прогнозов относительно корректности извлечения (напоминаем, что у нас нет эталона), мы построили график распределения прогнозов касательно триад с типовыми ошибками (в идеале мы хотели

бы предсказать для них вероятность равную 0), и касательно корректных триад (предположительно большая их часть извлечена правильно и мы должны спрогнозировать большую вероятность). На Рисунке 6 показаны результаты, полученные с использованием метода MULTILAYER+. Мы видим, что для триад с типовыми ошибками, метод MULTILAYER+ прогнозирует вероятность меньше 0.1 для 80% триад, а вероятность более 0.7 только для 8%. И напротив, для правильных триад, содержащихся в Freebase, MULTILAYER+ прогнозирует вероятность ниже 0.1 для 26% триад, а вероятность больше 0.7 для 54%, демонстрируя эффективность нашей модели.

5.3.3 Влияние изменения алгоритма логического вывода

В Таблице 6 показано влияние изменения различных элементов алгоритма логического вывода, используемого в случае многослойной модели.

Таблица 6: Вклад различных элементов. Худшие значения (по сравнению с базовыми) обозначены курсивом.

SqV	SqV	WDev	AUC-PR	Cov
MULTILAYER+	0.054	0.0040	0.693	0.864
$p(V_d \mid \hat{C}_d)$	*0.061*	0.0038	*0.570*	0.880
без корректировки α	0.055	*0.0057*	0.699	0.864
$p(C_{dwv} \mid \Vert (\bar{X}_{ewdv} > \varphi))$	0.053	0.0040	0.696	0.864

Строка $p(V_d \mid \hat{C}_d)$ показывает изменения, которые произошли в результате того, что C_d рассматривался, как наблюдаемые данные при выведении V_d (смотрите Раздел 3.3.2), в противовес использованию версии со взвешенным значением (Раздел 3.3.3). Мы видим существенное снижение значения в столбце AUC-PR и увеличение значения в столбце SqV в результате игнорирования неопределенности в C_d. В действительности, мы предсказали вероятность меньше 0.05 в отношении достоверности 93% триад.

Строка «без корректировки α» показывает изменения, которые произошли в результате того, что мы зафиксировали значение $p(C_{wdv} = 1)$ на уровне α, в противовес использованию схемы корректировки значения, описанной в Разделе 3.3.4. Мы видим, что большинство метрических значений одинаковы. Исключение составляет лишь заметно ухудшившееся значение WDev, что показывает, что вероятности недостаточно выверены. Выходит, что, если не изменять априорную вероятность, это зачастую приводит к чрезмерной уверенности при расчете $p(V_d \mid X)$, как показано в Примере 3.3.

Строка $p(C_{dwv} \mid \Vert (\bar{X}_{ewdv} > \varphi))$ показывает изменения, которые произошли в результате ограничения извлечений, взвешенных по степени уверенности, пороговым значением φ = 0, в противовес использованию извлечений, взвешенных по степени уверенности, описанных в Разделе 3.5. К удивлению, мы видим, что ввод ограничения позволяет получить чуть более лучший результат. Однако этот вывод совпадает с предыдущими наблюдениями, касающимися того, что некоторые экстракторы могут делать ошибки при прогнозировании уверенности [11].

5.3.4 Эффективность вычислений

Все алгоритмы реализованы с использованием FlumeJava [6], который основан на модели Map-Reduce. Абсолютное время счета может существенно различаться в зависимости от количества используемых нами машин. По этой причине в Таблице 7 показана только относительная эффективность алгоритмов. Мы отразили время, затрачиваемое на подготовку, включая применение механизма разбиения и объединения веб-ресурсов и экстракторов; время, затрачиваемое на процесс итераций, включая вычисление корректности извлечений, вычисление достоверности триад, вычисление точности ресурса, и вычисление качества экстрактора. Для каждого элемента в итерациях мы указали среднее время счета из пяти итераций. По умолчанию m = 5, M = 10 тыс.

Таблица 7

Задание		Обычное	Разбиение	Разбиение и объединение
Подготовка	Ресурс	0	0.28	0.5
	Экстрактор	0	0.50	0.46
	Итого	*0*	*0.779*	*1.034*
Итерация	I. Корректность извлечения	0.097	0.098	0.094
	II. Точность триады	0.098	0.079	0.087
	III. Надежность ресурса	0.105	0.080	0.074
	IV. Качество экстрактора	0.700	0.082	0.074
	Итого	*1*	*0.337*	*0.329*
Итого		5	2.466	2.679

Во-первых, мы отметили, что разбиение больших ресурсов и экстракторов может существенно ускорить время расчёта. В нашем массиве данных некоторые экстракторы, извлекали огромное количество триад из некоторых веб-сайтов. Разбиение таких экстракторов ускорило вычисление качества экстрактора в 8.8 раз. К тому же мы отметили, что разбиение больших ресурсов также ведет к сокращению времени расчёта надежности ресурса на 20%. В среднем расчёт каждого приближения был ускорен в 3 раза. Несмотря на большие затраты времени на разбиение, в целом, время расчёта сократилось в двое.

Во-вторых, мы заметили, что дополнительное применение механизма объединения не приводит к существенному временных затрат. Несмотря на то, что применение этого механизма увеличивает время на подготовку на 33%, время, затрачиваемое на каждый из этапов итерации, немного сокращается (на 2.4%), из-за меньшего числа ресурсов и экстракторов. Итоговое время расчёта превышает время расчёта с использованием только механизма разбиения всего лишь на 8.6%.

И наконец, мы изучили влияние параметров m и M. Мы заметили, что изменение M в диапазоне от 1 тыс. до 50 тыс. слабо влияет на качество прогноза. Однако значение M = 1 тыс. (более раздробленные) замедляет этап подготовки на 19%, а значение M = 50 тыс. (менее раздробленные) замедляет процесс логического вывода на 21%, поэтому в обоих случаях длительность расчета больше. С другой стороны, увеличение m до значения больше 5 не влияет существенно на эффективность, в то время как значение m = 2 (менее

объединенные) ускоряет вычисление wDev на 29% и замедляет логический вывод на 14%.

5.4 Результаты экспериментов применительно к KBT

Теперь мы можем вычислить, насколько хорошо мы оценили достоверность веб-страниц. Наш массив данных содержит более 2 миллиардов веб-страниц с 26 миллионов веб-сайтов. Наша многослойная модель считает, что мы имеем как минимум 5 правильно извлеченных триад из 119 миллионов веб-страниц и 5.6 миллионов веб-сайтов. На Рисунке 7 показаны распределения оценок KBT: мы видим, что пик находится на уровне 0.8, а 52% веб-сайтов обладают KBT (доверием) более 0.8.

5.4.1 KBT против PageRank

Поскольку мы не располагали реальной истиной относительно качества веб-страниц, мы сравнили наш метод оценки доверия с методом вычисления рейтинга PageRank. Мы рассчитали математический рейтинг методом PageRank для всех веб-страниц в сети и привели оценки к виду [0, 1]. На Рисунке 10 показаны графики KBT и PageRank для 2000 произвольно выбранных веб-сайтов. Как и ожидалось, два сигнала практически взаимно перпендикулярны друг другу. Затем мы изучили два случая, когда метод KBT существенно отличается от метода PageRank.

Маленькое значение PageRank, но высокое KBT (нижний правый угол на графике): Чтобы понять, какой ресурс может получить высокую оценку доверия KBT, мы произвольно отобрали 100 сайтов, значение KBT которых было более 0.9. Количество извлеченных из каждого веб-сайта триад варьировалось от сотен до миллионов. Для каждого сайта мы рассмотрели первых 3 предиката и произвольно выбрали из этих предикатов 10 триад, где вероятность того, что извлеченные данные являются корректными, превышала 0.8. Мы вручную оценили каждый из выбранных веб-сайтов по 4-ем критериям:

- *Корректность триады*: верны ли по меньшей мере 9 триад.
- *Правильность извлечения*: правильно ли извлечены по меньшей мере 9 триад (с этого момента мы можем оценить веб-сайт, исходя из того, что он на самом деле утверждает).
- *Релевантность темы*: мы определились с основными темами веб-сайтов, исходя из названия и на данных со страницы «О нас»; затем мы определили, релевантны ли по меньшей мере 9 триад данным темам (например, если веб-сайт посвящен бизнес директориям Южной Америки, однако, извлеченные данные, рассказывают о городах и странах Южной Америки, мы считали, что триады не релевантны теме сайта).
- *Нетривиальность*: мы определили, констатируют ли выбранные триады нетривиальные факты (то есть, если большинство выбранных с сайта об индийском кино триад утверждало, что фильмы сняты на индийском языке, тогда мы считали такие триады тривиальными).

Мы считали веб-сайт истинно достоверным, если он соответствовал всем четырем критериям. Из 100 сайтов 85 были названы достоверными; 2 не соответствовали темам,

12 не обладали достаточным количеством нетривиальных триад, а 2 содержали более 1 ошибки извлечения (один сайт был посвящён двум темам). Однако только 20 из 85 достоверных сайтов имели рейтинг, рассчитанный по методу PageRank, более 0.5. Этот факт демонстрирует, что оценка КВТ может определить ресурс с достоверными данными, даже если они имеют низкий рейтинг PageRank.

Высокое значение рейтинга, рассчитанного по методу PageRank, но низкая оценка КВТ (верхний левый угол): Мы рассмотрели 15 веб-сайтов, посвящённых светской хронике, перечисленных в работе [16]. Из них 14 имели высокий PageRank, поскольку такие веб-сайты очень популярны. Однако они имели очень низкие оценки КВТ. Также более низкие оценки КВТ имеют, как правило, форумы. Например, мы выяснили, что на сайте *answers.yahoo.com* утверждается, что *«Кэтрин Зета-Джонс родилась в Новой Зеландии»*[3], хотя согласно *Википедии* она была рождена в Уэльсе[4].

5.4.2 Комментарии

Несмотря на то, что мы видим, что оценка КВТ дает полезный сигнал о достоверности ресурса, который ортогонален по отношению к более традиционным сигналам, таким как PageRank, наши эксперименты также показали места, требующие дальнейших доработок. Так:

1. Необходимо избегать оценки КВТ на основании триад, не относящихся к теме ресурса. Нам необходимо определять основные темы веб-сайтов, фильтровать триады, чьи сущности или предикаты не относятся к этим темам.

2. Необходимо избегать оценки КВТ на основании тривиальных извлеченных триад. Нам необходимо решить, является ли информация, содержащаяся в триаде, тривиальной. Один из подходов – это рассмотрение предикатов с очень маленьким разнообразием объектов, как менее информативных. Другой подход - это связь триад с IDF, благодаря чему триады с низким значением IDF получат меньший вес при вычислениях КВТ.

3. Наши экстракторы (и большая часть современных экстракторов) по-прежнему характеризуются ограниченными возможностями извлечения данных, что ограничивает нашу возможность оценить КВТ для всех веб-сайтов. Мы бы хотели увеличить охват КВТ, расширив наш метод до возможности управления экстракторами типа open-IE, которые не соответствуют схеме[14]. Однако, несмотря на то, что эти методы позволяют извлекать больше триад, они привносят больше помех.

4. Некоторые веб-сайты копируют данные других сайтов. Нахождение таких сайтов требует использования такой технологии, как распознавание копий. Попытка масштабирования метода определения копий, например, описанного в работах [7, 8], была сделана в работе [23]. Однако требуется проделать больше работы, прежде чем эти методы смогут быть применены к анализу данных, извлеченных из миллиардов ресурсов.

---

[3] https://answers.yahoo.com/question/index?qid=20070206090808AAC54nH
[4] http://en.wikipedia.org/wiki/Catherine Zeta-Jones

6. СВЯЗАННЫЕ РАБОТЫ

Существует множество работ по оценке качества веб-ресурсов. Метод PageRank [4] и анализ первоисточника-посредника (Authority-hub analysis) [19] анализируют сигналы, полученные при анализе ссылок (изучено в работе [3]). Алгоритмы EigenTrust [18] и TrustMe [28] анализируют сигналы, полученные на основании поведения ресурса в сети P2P. Алгоритмы Web topology [5], TrustRank [17], и AntiTrust [20] обнаруживают веб-спам. Оценка достоверности ресурса, основанная на знаниях, которую мы предложили в данной работе, анализирует важный *внутренний* сигнал, а именно корректность фактической информации, предоставляемой веб-ресурсом.

Наша работа связана с основной частью работы в *области интеграции данных* (вопрос рассмотрен в работах [1, 12, 23]), цель которой заключалась в разрешении конфликтов между данными, предоставленными множеством ресурсов, и в нахождении истин, которые соответствовали бы реальной информации. Большая часть последних работ в этой области анализировала достоверность ресурсов, измеренную показателями, основанными на ссылках [24, 25]; показателями, основанными на методе информационного поиска [29]; показателями, основанными на надежности [8, 9, 13, 21, 27, 30] и на анализе графической модели [26, 31, 33, 32]. Однако эти работы не содержат концепции экстрактора и, таким образом, не могут отличать ненадежный ресурс от ненадежного экстрактора.

Для решения проблемы интеграции данных были предложены графические модели [26, 31, 32, 33]. Эти модели в большей или меньшей степени похожи на предложенную нами в Разделе 2.2 однослойную модель. В частности, в работе [26] рассматривается идея «единственной истины», в работе [32] рассматриваются числовые значения, работа [33] допускает наличие множества истин, а в работе [31] анализируются связи между ресурсами. Однако все эти более ранние работы не содержат концепции экстрактора и, таким образом, не отражают того факта, что ресурсы и экстракторы привносят качественно разные виды помех. К тому же, массивы данных, использованные для экспериментов в этих работах, как правило на 5-6 порядков меньше по объему, чем использованные нами массивы данных, а представленные в них алгоритмы, основанные на логическом выводе, изначально слабее нашего алгоритма. Нашу работу от многих других также отличает наличие многослойной модели и объем экспериментальных данных.

В конечном итоге, с нашей текущей работой в большей степени оказалась связана наша предыдущая работа в области интеграции знаний [11]. В Разделе 2.3 мы представили детальное сравнение, а также эмпирическое сравнение в Разделе 5, показав, что МНОГОСЛОЙНАЯ модель выигрывает у ОДНОСЛОЙНОЙ в том, что касается ее применения в области интеграции данных, и дает возможность использовать оценку KBT для определения качества веб-ресурса.

7. ЗАКЛЮЧЕНИЯ

В данной работе предложена новая система показателей для оценки качества ресурса – доверия, основанного на знаниях. Мы предложили современную вероятностную модель, которая позволяет совместно оценить правильность извлечения данных, правильность исходных данных, и достоверность ресурсов. К тому же, мы представили алгоритм,

который динамически определяет степень разбиения для каждого ресурса. Экспериментальные результаты показали, как потенциал оценки качества веб-ресурса, так и превосходство данного метода на существующими, используемыми для интеграции знаний.

8. ССЫЛКИ

[1] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys*, 41(1):1–41, 2008.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[3] A. Borodin, G. Roberts, J. Rosenthal, and P. Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *TOIT*, 5:231–297, 2005.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[5] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *SIGIR*, 2007.

[6] C. Chambers, A. Raniwala, F. Perry, S. Adams, R. R. Henry, R. Bradshaw, and N. Weizenbaum. Flumejava: Easy, efcient data-parallel pipelines. In *PLDI*, pages 363–375, 2010.

[7] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *PVLDB*, 2010.

[8] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *PVLDB*, 2(1), 2009.

[9] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *PVLDB*, 2(1), 2009.

[10] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *SIGKDD*, 2014.

[11] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From data fusion to knowledge fusion. *PVLDB*, 2014.

[12] X. L. Dong and F. Naumann. Data fusion–resolving data conflicts for integration. *PVLDB*, 2009.

[13] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. *PVLDB*, 6, 2013.

[14] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open information extraction: the second generation. In *IJCAI*, 2011.

[15] L. A. Galarraga, C. Teflioudi, K. Hose, and F. Suchanek. Amie:´ association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, pages 413–422, 2013.

[16] Top 15 most popular celebrity gossip websites. http://www.ebizmba.com/articles/gossip-websites, 2014.

[17] Z. Gyngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *VLDB*, pages 576–587, 2014.

[18] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *WWW*, 2003.

[19] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, 1998.

[20] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *AIRWeb*, 2006.

[21] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *SIGMOD*, pages 1187–1198, 2014.

[22] X. Li, X. L. Dong, K. B. Lyons, W. Meng, and D. Srivastava. Truth finding on the Deep Web: Is the problem solved? *PVLDB*, 6(2), 2013.

[23] X. Li, X. L. Dong, K. B. Lyons, W. Meng, and D. Srivastava. Scaling up copy detection. In *ICDE*, 2015.

[24] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In *COLING*, pages 877–885, 2010.

[25] J. Pasternack and D. Roth. Making better informed trust decisions with generalized fact-finding. In *IJCAI*, pages 2324–2329, 2011.

[26] J. Pasternack and D. Roth. Latent credibility analysis. In *WWW*, 2013.

[27] R. Pochampally, A. D. Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing data with correlations. In *Sigmod*, 2014.

[28] A. Singh and L. Liu. TrustMe: anonymous management of trust relationshiops in decentralized P2P systems. In *IEEE Intl. Conf. on Peer-to-Peer Computing*, 2003.

[29] M. Wu and A. Marian. Corroborating answers from multiple web sources. In *Proc. of the WebDB Workshop*, 2007.

[30] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proc. of SIGKDD*, 2007.

[31] X. Yin and W. Tan. Semi-supervised truth discovery. In *WWW*, pages 217–226, 2011.

[32] B. Zhao and J. Han. A probabilistic model for estimating real-valued truth from conflicting sources. In *QDB*, 2012.

[33] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A Bayesian approach to discovering truth from conflicting sources for data integration. *PVLDB*, 5(6):550–561, 2012.