

ВЕКТОРНАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ НА ОСНОВЕ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ТЕРМОВ

© 2017 г. Д.В. Бондарчук

Уральский государственный университет путей сообщения

(620034 Екатеринбург, ул. Колмогорова, д. 66)

E-mail: dubondarchuk@gmail.com

Поступила в редакцию: 26.07.2015

Большинство методов интеллектуального анализа текстов используют векторную модель представления знаний. Векторная модель использует частоту (вес) термина, чтобы определить его важность в документе. Термы могут быть схожи семантически, но отличаться лексикографически, что, в свою очередь, приведет к тому, что классификация, основанная на частоте термов, не даст нужного результата.

Причиной ошибок является отсутствие учета таких особенностей естественного языка, как синонимия и полисемия. Неучет этих особенностей, а именно синонимии и полисемии, увеличивает размерность семантического пространства, от которой зависит быстродействие конечного программного продукта, разработанного на основе алгоритма. Кроме того, результаты работы многих алгоритмов сложно воспринимаются экспертом предметной области, который подготавливает обучающую выборку, что, в свою очередь, также сказывается на качестве выдачи алгоритма.

В работе предлагается модель, которая помимо веса термина в документе, так же использует «семантический вес термина». «Семантический вес термов» тем выше, чем они семантически ближе друг к другу.

Для вычисления семантической близости термов будем использовать адаптацию расширенного алгоритма Леска. Метод расчета семантической близости состоит в том, что для каждого значения рассматриваемого слова подсчитывается число слов упомянутых как в словарном определении данного значения (предполагается, что словарное определение содержит описание нескольких значений слова), так и в ближайшем контексте рассматриваемого слова. В качестве наиболее вероятного значения слова выбирается то, для которого такое пересечение оказалось больше. Векторная модель с учетом семантической близости термов решает проблему неоднозначности синонимов.

Ключевые слова: интеллектуальный анализ текстов, векторная модель, семантическая близость.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Бондарчук Д.В. Векторная модель представления знаний на основе семантической близости термов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. X, № Y. С. Z1–Z2. DOI: 10.14529/cmseXXXXXX.

Введение

Большинство методов подбора персональных рекомендаций используют векторную модель представления знаний. Задача подбора персональных рекомендаций заключается в следующем. Пусть имеется выборка текстовых данных (например, товары, услуги), формируемых пользователями, которые необходимо обработать и систематизировать. Пусть так же имеется выборка данных пользователей (например, покупателей, поставщиков) так же представленная в текстовом виде. Технических отличий между двумя этими выборками нет, различия скорее идеологические. Первая выборка — это то, что мы используем для построения модели, а вторая — то, для чего мы анализируем с использованием модели. Необходимо обработать вышеуказанные данные таким образом, чтобы можно было их использовать для быстрого подбора персональных рекомендаций для любого пользователя. При этом данные между категориями распределены неравномерно, и пользователи независимо от их предпочтений всегда гарантированно должны получить выборку рекомендаций

определенного объема. На практике очень часто встречаются текстовые корпуса, неравномерно распределенные между категориями. Категории в данном случае — это некие группы, число групп конечно и известно заранее, которые формируются специалистом в предметной области.

Векторная модель использует вес (частоту) термина, чтобы определить его важность в документе. Термы могут быть схожи семантически, но отличаться лексографически, что в свою очередь приведет к тому, что классификация основанная на частоте термов не даст нужного результата.

Документ в векторной модели рассматривается как неупорядоченное множество термов. Различными способами можно определить вес термина в документе — «важность» слова для идентификации данного текста. Например, можно просто подсчитать количество употреблений термина в документе, так называемую частоту термина, — чем чаще слово встречается в документе, тем больший у него будет вес. Если терм не встречается в документе, то его вес в этом документе равен нулю.

Все термы, которые встречаются в документах обрабатываемой коллекции, можно упорядочить. Если теперь для некоторого документа выписать по порядку веса всех термов, включая те, которых нет в этом документе, получится вектор, который и будет представлением данного документа в векторном пространстве. Размерность этого вектора, как и размерность пространства, равна количеству различных термов во всей коллекции, и является одинаковой для всех документов.

Более формально это утверждение можно представить в виде формулы:

$$\vec{d}_i = (w_{i1}, w_{i2}, \dots, w_{in}) \quad (1)$$

где \vec{d}_i — векторное представление i -го документа, w_{ij} — вес j -го термина в i -м документе, n — общее количество различных термов во всех документах коллекции.

Многие исследования [1–3] на тему решения проблемы лексической многозначности предлагают использовать семантическую связанность и семантические меры. Применяя базовые принципы семантического анализа можно улучшить производительность методов интеллектуального подбора персональных рекомендаций. Например, в реализации известной семантической сети WordNet [14], разработанной в Принстонском университете, используются так называемые «синсеты» — синонимические ряды, объединяющие слова со схожим значением. Каждый «синсет» содержит список синонимов или синонимичных словосочетаний и указатели, описывающие отношения между ним и другими «синсетами». Слова, имеющие несколько значений, включаются в несколько «синсетов» и могут быть причислены к различным синтаксическим и лексическим классам. Такой подход дает более точные результаты в сравнении с классической векторной моделью представления знаний.

Похожая техника представляет собой отображение термов документа в их «смысл» и составление функциональных векторов документа. В терминах СУБД это означает, что всем словам, имеющим один и тот же смысл приписывается некий идентификатор, который в свою очередь и становится термом. Конечно, качество обучения улучшается, однако опыт использования этой техники показывает, что улучшается оно незначительно [10, 15].

В статье предлагается другой подход — вычисление семантической близости термов (векторная модель семантической близости термов). Данная модель, помимо веса термина в документе, так же использует «семантический вес термина». «Семантический вес термов» тем выше, чем они семантически ближе друг к другу.

Для вычисления семантической близости термов используется адаптация расширенного алгоритма Леска [4]. Данный метод состоит в следующем. Для каждого значения рассматриваемого слова подсчитывается число слов упомянутых как в словарном определении данного значения, так и в ближайшем контексте рассматриваемого вхождения слова. В качестве наиболее вероятного значения выбирается то, для которого такое пересечение оказалось больше. Векторная модель семантической близости термов решает проблему неоднозначности синонимов. Опыт использования данной модели показывает, что эффективность интеллектуального подбора персональных рекомендаций по сравнению с использованием стандартной векторной модели значительно повышается.

Целью данной работы является разработка и исследование алгоритма вычисления семантической близости и его применения для повышения эффективности подбора персональных рекомендаций. В разделе 1 настоящей статьи производится постановка задачи устранения лексической многозначности. В разделе 2 описываются известные подходы к устранению лексической многозначности. В разделе 3 описывается алгоритм переопределения классического веса термина в векторе документа для учета семантических связей между каждой парой термов. В разделе 4 описывается способ вычисления семантической близости с помощью адаптации метода Леска, основанный на использовании семантической БД WordNet. В разделе 5 приводится пример использования алгоритма на текстах различных предметных областей, а так же оценивается его эффективность. В заключении подводятся итоги исследования, описываются недостатки предложенного алгоритма и формулируются направления для дальнейших исследований.

1. Постановка задачи устранения лексической многозначности

Проблема снятия лексической многозначности может быть переформулирована так же, как задача максимизации с использованием формализма скрытых Марковских моделей [5]. Пусть T — множество терминов, M — множество значений, соответствующих терминам. Для последовательности терминов $\tau = \{t_1, \dots, t_n\}$, где $\forall i t_i \in T$, задача состоит в нахождении наиболее вероятной последовательности значений $\mu = \{m_i, \dots, m_n\}$, где $\forall i m_i \in M$, соответствующей входным терминам.

$$\hat{\mu} = \arg_{\mu} P(\mu|\tau) = \arg_{\mu} \left(\frac{P(\mu)P(\tau|\mu)}{P(\tau)} \right) \quad (2)$$

Поскольку вероятность $P(\tau)$ для входной последовательности является величиной постоянной, то задача сводится к максимизации числителя, указанного в формуле (2). Для решения этого уравнения делается марковское предположение, что значение i -го термина зависит только от конечного числа значений дущих терминов [6]:

$$\hat{\mu} = \arg_{\mu} \left(\prod_{i=1}^n P(m_i|m_{i-1}, \dots, m_{i-k})P(t_i|m_i) \right) \quad (3)$$

где k — порядок модели.

Множители равенства (3) определяют скрытую Марковскую модель k -го порядка, где наблюдения соответствуют входным терминам, состояния соответствуют значениям терминов, $P(m_i|m_{i-1}, \dots, m_{i-k})$ — вероятность перехода между состояниями, $P(t_i|m_i)$ — вероятность появления термина t_i в каждом состоянии m_i .

Дальнейшее использование данной модели связано со значительными трудностями, в частности с разреженностью языка. Например, чтобы построить модель перехода для Марковской модели первого порядка, необходимо оценить вероятность каждой пары состояний, что для данной задачи сводится к вероятности встречи двух терминов в конкретных значениях вместе. Для задачи устранения лексической многозначности проблема оценки параметров марковской модели является нетривиальной задачей. Это связано с большими объемами обрабатываемой информации, то есть с объемами представленных знаний и с тем, что слова в тексте на естественном языке распределяются не равномерно, а по закону Ципфа [7]

Закон Ципфа — эмпирическая закономерность распределения частоты слов естественного языка: если все слова языка упорядочить по убыванию частоты их использования, то частота i -го слова в таком списке окажется приблизительно обратно пропорциональной его порядковому номеру i .

2. Подходы к устранению лексической многозначности

Большинство подходов к устранению лексической многозначности связаны с развитием огромных баз знаний, созданных вручную, таких как WordNet. Можно указать на очевидный недостаток такого подхода — ограниченность области применения данных методов, поскольку ручное поддержание баз знаний в актуальном состоянии является очень сложной и трудозатратной задачей.

Если в семантической сети, на основе которой производится анализ (например, WordNET) используются различные значения для многозначных слов, то для анализа необходимо обеспечение автоматического выбора между этими многозначными сущностями [7]. Обычно в таких случаях используется наивный метод, который выбирает наиболее часто встречающуюся сущность. Очевидно, что использование такого подхода далеко от идеала и в ряде случаев может давать необъективные результаты. Частично эта проблема решается в концепции «универсального терминологического пространства», однако пока о какой-либо реализации этого пространства неизвестно.

Другим подходом к решению проблемы лексической многозначности является использование внешних источников данных. С развитием сети Интернет появилось огромное количество документов, которые связаны между собой гиперссылками. Например, в работе [8] рассматривалась возможность использования глобальной энциклопедии «Википедия» в качестве аннотированного корпуса для обучения Марковской модели. Для методов, использующих данный подход, очень часто применяют алгоритм Леска. Алгоритм основан на предположении, что многозначное слово и его окружение относятся к одной теме [4].

Все вышеописанные методы и алгоритмы так или иначе основаны на внешних данных и имеют один общий недостаток. В основе всех этих алгоритмов явно или неявно лежит предположение, что существуют однозначные термины, на основании которых в последствии определяются многозначные термины. Это в свою очередь составляет огромную проблему, поскольку в неспециализированных текстах, таких, как объявления о поиске работы, новостных статьях, участвуют только многозначные термины, либо присутствующие однозначные термины слабо связаны с темой документа и могут быть расценены классификатором, как стоп-слова. Этот факт приводит к тому, что точность и эффективность указанных методов в значительной степени ухудшается при их применении для классификации вакансий или новостей [10].

Для избавления от лексической многозначности так же используются меры семантической связности. Отметим, что семантическая близость и семантическая связность — это разные понятия. *Семантическая близость* является частным случаем семантической связности. *Семантическая связность* — это количество связей, с помощью которых связаны два слова. Перечислим наиболее известные способы вычисления семантической связности.

1. Мера Ликока—Чодороу

Мера Ликока и Чодороу [12] основана на вычислении длины пути между терминами. Кратчайшим путем от одного термина к другому считается путь, который использует наименьшее количество соседних термов. Данную меру можно представить в виде следующей формулы:

$$related_{lch} = (t_1, t_2) = \max [-\log (L(t_1, t_2)/(2 \cdot D))] \quad (4)$$

где $L(t_1, t_2)$ — кратчайшая длинна пути (наименьшее количество узлов) между двумя терминами, D — максимальная глубина (максимальное количество узлов от корневого узла, либо количество узлов до ближайшего общего предка [12]).

2. Мера Цеша

Вычисление меры Цеша [12] подобно вычислению меры Ликока—Чодороу, основное отличие заключается в том, что поиск кратчайшего пути осуществляется между терминами по произвольным типам ссылок.

3. Мера Лина

Вычисление семантической связности с помощью меры Лина основано на теореме близости. Она гласит, что близость между двумя терминами можно вычислить с помощью коэффициента отношения количества текстов (корпусов), в которых термы встречаются вместе к частоте их встречаемости в определениях.

Отсутствие общих терминов между двумя документами еще не означает, что они являются абсолютно несхожими. Термины могут быть синтаксически различны, но в то же самое время семантически очень близки. Дальнейшее развитие метода анализа данных будет основано именно на этом утверждении.

3. Предлагаемый алгоритм

Чтобы учесть семантическую связь между терминами, вес термина в документе будем рассчитывать несколько иначе, чем в классической векторной модели представления знаний. Термином (термом) будем называть слово, обработанное с помощью стеммера Портера [9] и не содержащееся в списке стоп-слов.

Настройка весов термов производится с помощью вычисления семантической близости связанных термов. Считается, что термины связаны, если они находятся в одном документе в непосредственной близости друг к другу. Новый вес термина рассчитывается следующим образом:

$$\tilde{w}_{dt_1} = w_{dt_1} + \sum_{t_1 \neq t_2} similarity(t_1, t_2) \quad (5)$$

где w_{dt_1} — вес термина в документе d до настройки, $similarity$ — семантическая близость термов t_1 и t_2 , рассчитываемая с помощью адаптации расширенного метода Леска. Суммирование происходит по всем терминам документа d .

Этот шаг переопределяет классический вес термина в векторе документа и учитывает семантические связи между каждой парой термов. Для вычисления исходных весов термов в документе будем использовать меру $tf.idf$. Вес некоторого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции [11]. Мера $tf.idf$ термина t в документе d вычисляется следующим образом:

$$tf.idf(d, t) = \ln(tf(d, t) + 1) \ln \frac{|D|}{df(t)} \quad (6)$$

где $df(t)$ — документная частота термина, показывающая количество документов, в которых встречается терм, $tf(t, d)$ — число появлений термина t в документе d , нормализованное общим количеством термов в документе d , $|D|$ — общее количество документов.

Предлагается использовать именно эту меру, поскольку она приписывает большие веса терминам, которые редко встречаются в обучающей выборке, но часто в некоторых конкретных документах. $tf.idf$ дает примерно на 14% более точный результат, чем стандартная мера tf , основанная на частоте термина в документе [11].

В [2] показано, что каждая из категорий обычно представлена множеством «базовых» слов, а остальные слова являются слишком общими, чтобы определять категорию. Предлагается использовать общие слова для повышения значимости (весов) «базовых» слов. Данное решение в значительной степени улучшило результаты подбора персональных рекомендаций, поскольку при использовании этого подхода определение принадлежности документа некоторой категории происходит более точно.

4. Расширенный алгоритм Леска

Оригинальный алгоритм Леска [13] предусматривает использование только словарных значений анализируемого слова и же его ближайшего контекста. Это является существенным ограничением, поскольку словарные определения как правило являются очень короткими, и влияют на рассчитанную по алгоритму Леска близость слов только косвенно [13]. Если взять для примера одну из самых крупных баз знаний WordNet, то средняя длина определения слова в словаре равна всего семи словам [5].

Расширенный метод Леска расширяет определения сравниваемых слов и включает определения слов, которые связаны со сравниваемыми словами. Будем считать, что два термина похожи, если их определения содержат похожие слова. В самом простом случае расширенный метод Леска можно выразить следующей формулой:

$$\begin{aligned} similarity_{extLesk}(t_1, t_2) = & overlap(gloss(t_1), gloss(t_2)) + \\ & overlap(gloss(hyppo(t_1)), gloss(hyppo(t_2))) + \\ & overlap(gloss(hyppo(t_1)), gloss(t_2)) + \\ & overlap(gloss(t_1), gloss(hyppo(t_2))) \end{aligned} \quad (7)$$

где $overlap(t_1, t_2)$ — количество совпадений между терминами t_1 и t_2 , $gloss(t)$ — определение термина t , $hyppo(t)$ — гипероним слова, например для слова «красный» гиперонимом является слово «цвет», t_1 и t_2 — термины.

В классической версии алгоритма Леска гиперонимы не используются, однако их использование значительно улучшает качество выдачи алгоритма. В работах [3] и [6] используются синсеты и гиперонимы из английской версии WordNET, но синсет выбирается

согласно наивному методу, после чего выбирается соответствующий синсету гипероним. Эксперименты проводились на нескольких независимых общедоступных выборках. Авторы данных работ сделали вывод, что использование гиперонимов привело к улучшению качества работы классификатора на всех обучающих множествах. Кроме того, выяснилось, что применение гиперонимов почти всегда улучшает качество работы классификатора по сравнению с применением только синсетов [3].

Гиперонимы слов для первоначальной оценки можно взять, например, в российской версии WordNet, разработка которой осуществляется в Петербургском университете путей сообщения.

5. Оценка результатов

В исходный алгоритм [15] внесены изменения, позволяющие повысить эффективность метода и качество результирующей выборки. Данные изменения характеризуются иным способом вычисления весов термов на этапе построения векторных моделей.

Для оценки эффективности алгоритма классификации представления знаний, использующую семантическую близость термов, по сравнению с исходным алгоритмом, была проведена работа алгоритмов на различных множествах текстов: объявления о работе, новости, литературные аннотации.

В табл. 1 представлены сведения о выборках.

Таблица 1

Сведения о выборках

Выборка	Кол-во текстов	Кол-во кат.	Распределение
Объявления о работе	около 700 тыс.	17	неравномерное
Новости	около 1,2 млн.	10	равномерное
Литературные аннотации	около 20 тыс.	13	равномерное

В качестве мер оценок результатов использовались *F-measure* и *purity*:

$$F\text{-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \quad (8)$$

где *precision* — количество правильных результатов в выдаче алгоритма, *recall* — общее количество результатов выдачи.

$$\text{purity}(W, C) = \sum_k \max_j |w_k \cup c_j|, \quad (9)$$

где W — множество документов, w_k — k -ый документ, C — множество категорий (множество документов, отнесенных классификатором к категории k), c_j — множество документов, отнесенных к категории j экспертом.

В табл. 2 представлены результаты данных оценок. Взяты средние значения оценок 15 текстов.

Результаты экспериментов показывают, что использование векторной модели с вычислением семантической близости помогает улучшить результаты работы классификатора по сравнению с векторной моделью без учета семантической близости. Меры, описанные формулами (8) и (9), показывают, насколько результаты работы предложенного класси-

Таблица 2

Оценка результатов работы алгоритма классификации

Множество	Исходный алгоритм		Предлагаемый алгоритм	
	F-measure	Purity	F-measure	Purity
Объявления о работе	0,31	0,33	0,65	0,66
Новости	0,56	0,58	0,61	0,64
Литературные аннотации	0,56	0,57	0,63	0,67

фикатора соответствуют действительности. Чем выше значение оценки, тем качественнее работа алгоритма.

В среднем категориальная векторная модель с использованием семантической близости дает на 8–10% более точный результат. Это связано с тем, что исходная модель менее чувствительна к «шумам» за счет настройки весовых коэффициентов с помощью вычисления семантической близости. Новые весовые коэффициенты векторов документов учитывают контекст появления термов. Более высокие веса связаны с термами, которые сильнее семантически связаны с другими термами.

Эксперименты были проведены над выборками разного рода и объема, на всех из них метод отработал эффективно. Также часть выборок была распределена неравномерно, метод и на них показал хороший результат, в то время как результаты векторной модели без учета семантической близости термов оказались неудовлетворительными.

Заключение

В статье была рассмотрена задача формирования персональных рекомендаций и предложена векторная модель представления знаний, использующая семантическую близость термов, использование которой улучшает качество работы автоматического классификатора. Модель помогает решить проблему лексической неоднозначности терминов, а так же находит скрытые семантические связи между документами, сравнивая семантически близкие термины.

Проведенные эксперименты показали, что метод, основанный на использовании этой модели, показал достаточно высокую эффективность. Среди направлений дальнейших исследований можно выделить исследование возможности статистического определения семантической близости между термами, поиск альтернативного алгоритма стемминга, который лучше работает с русским языком. Кроме того, недостаточно изученным является вопрос возможности использования в алгоритмах формирования персональных рекомендаций существующих словарей, тезаурусов и баз данных интернета.

Литература

1. Budanitsky A., Hirst G. Evaluating WordNet-based Measures of Lexical Semantic Relatedness // Computational Linguistics. 2006. Vol. 32. P. 13–47.
2. Hotho A., Staab S., Stumme G. WordNet Improve Text Document Clustering // SIGIR 2003 Semantic Web Workshop (Toronto, Canada, July 28 – August 1, 2003). P. 541–544. DOI: 10.1145/959258.959263.
3. Sedding J., Dimitar K. WordNet-based Text Document Clustering // COLING 2004, 3rd Workshop on Robust Methods in Analysis of Natural Language Data (Geneva, Switzerland,

August 23 – 27, 2004). P. 104–113. DOI:10.3115/1220355.1220356.

4. Lesk M. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone // SIGDOC'86. Proceedings of the 5th Annual International Conference on Systems Documentation (Toronto, Canada, June 8 – 11, 1986). P. 24–26. DOI:10.1145/318723.318728.
5. Loupy C., El-Beze M., Marteau P.F. Word Sense Disambiguation Using HMM Tagger // Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC (Toronto, Canada, June 8 – 11, 1998). P. 1255–1258. DOI:10.3115/974235.974260.
6. Jeh G., Widom J. SimRank: a Measure of Structural-Context Similarity // Proceedings of the 8th Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining international conference on Knowledge discovery and data mining (Edmonton, Canada, July 23 – 25, 2002). P. 271–279. DOI:10.1145/775047.775049.
7. Kechedzhy K.E., Usatenko O., Yampolskii V.A. Rank Distributions of Words in Additive Many-step Markov Chains and the Zipf law // Physical Reviews E: Statistical, Nonlinear, Biological, and Soft Matter Physics. 2005. Vol. 72. P. 381–386.
8. Mihalcea R. Using Wikipedia for Automatic Word Sense Disambiguation // Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (New York, USA, April 22 – 27, 2007). P. 196–203. DOI:10.3115/1599081.1599184.
9. Willett P. The Porter Stemming Algorithm: Then and Now // Program: Electronic Library and Information Systems. 2006. Vol. 4, No. 3. P. 219–223.
10. Бондарчук Д.В. Выбор оптимального метода интеллектуального анализа данных для подбора вакансий // Информационные технологии моделирования и управления. 2013. № 6(84). С. 504–513.
11. Salton G. Improving Retrieval Performance by Relevance Feedback // Readings in Information Retrieval. 1997. Vol. 24. P. 1–5.
12. Tan P. N., Steinbach M., Kumar V. Top 10 Algorithms in Data Mining // Knowledge and Information Systems. 2008. Vol. 14, No. 1. P. 1–37. DOI:10.1007/s10115-007-0114-2.
13. Banerjee S., Pedersen T. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet // Lecture Notes In Computer Science (Canberra, Australia, February 11 – 22, 2002). Vol. 2276. P. 136–145. DOI:10.1007/3-540-46035-7 22.
14. Тезаурус WordNET. URL: <https://wordnet.princeton.edu/> (дата обращения: 05.02.2017).
15. Бондарчук Д.В. Интеллектуальный метод подбора персональных рекомендаций, гарантирующий получение непустого результата // Информационные технологии моделирования и управления. 2015. № 2(92). С. 130–138.

VECTOR SPACE MODEL OF KNOWLEDGE REPRESENTATION BASED ON SEMANTIC RELATEDNESS

© 2017 D.V. Bondarchuk

*Ural State University of Railway Transport
(st. Kolmogorova 66, Yekaterinburg, 620034 Russia)*

E-mail: [dzbondarchuk@gmail.com](mailto:dvbondarchuk@gmail.com)

Received: 26.07.2015

Most of text mining algorithms uses vector space model of knowledge representation. Vector space model uses the frequency (weight) of term to determine its importance in the document. Terms can be semantically similar but different lexicographically, which in turn will lead to the fact that the classification is based on the frequency of the terms does not give the desired result.

Analysis of a low-quality results shows that errors occur due to the characteristics of natural language, which were not taken into account. Neglect of these features, namely, synonymy and polysemy, increases the dimension of semantic space, which determines the performance of the final software product developed based on the algorithm. Furthermore, the results of many complex algorithms perceived domain expert to prepare training sample, which in turn also affects quality issue algorithm.

We propose a model that in addition to the weight of a term in a document also uses semantic weight of the term. Semantic weight terms, the higher they are semantically closer to each other.

To calculate the semantic similarity of terms we propose to use a adaptation of the extended Lesk algorithm. The method of calculating semantic similarity lies in the fact that for each value of the word in question is counted as the number of words referred to the dictionary definition of this value (assuming that the dictionary definition describes several meanings of the word), and in the immediate context of the word in question. As the most probable meaning of the word is selected such that this intersection was more. Vector model based on semantic proximity of terms solves the problem of the ambiguity of synonyms.

Keywords: text-mining, vector space model, semantic relatedness.

FOR CITATION

Bondarchuk D.V. Vector space model of knowledge representation based on semantic relatedness. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2017. vol. X, no. Y. pp. Z1–Z2. (in Russian) DOI: 10.14529/cmseXXXXXX.

References

1. Budanitsky A., Hirst G. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*. 2006. vol. 32. pp. 13–47.
2. Hotho A., Staab S., Stumme G. WordNet Improve Text Document Clustering. *SIGIR 2003 Semantic Web Workshop (Toronto, Canada, July 28 – August 1, 2003)*. pp. 541–544. DOI:10.1145/959258.959263.
3. Sedding J., Dimitar K. WordNet-based Text Document Clustering. *COLING 2004 3rd Workshop on Robust Methods in Analysis of Natural Language Data (Geneva, Switzerland, August 23 – 27, 2004)*. pp. 104–113. DOI:10.3115/1220355.1220356.
4. Lesk M. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. *SIGDOC '86: Proceedings of the 5th Annual*

- International Conference on Systems Documentation (Toronto, Canada, June 8 – 11, 1986).* pp. 24–26. DOI:10.1145/318723.318728.
5. Loupy C., El-Beze M., Marteau P.F. Word Sense Disambiguation Using HMM Tagger. *Proceedings of the 1st International Conference on Language Resources and Evaluation (Toronto, Canada, June 8 – 11, 1998).* pp. 1255–1258. DOI:10.3115/974235.974260.
 6. Jeh G., Widom J. SimRank: a Measure of Structural-context Similarity. *Proceedings of the 8th Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining international conference on Knowledge discovery and data mining (Edmonton, Canada, July 23 – 25, 2002).* pp. 271–279. DOI:10.1145/775047.775049.
 7. Kechedzhy K.E., Usatenko O., Yampolskii V.A. Rank Distributions of Words in Additive Many-step Markov Chains and the Zipf law. *Physical Reviews E: Statistical, Nonlinear, Biological, and Soft Matter Physics.* 2005. vol. 72. pp. 381–386.
 8. Mihalcea R. Using Wikipedia for Automatic Word Sense Disambiguation. *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (New York, USA, April; 22 – 27, 2007).* pp. 196–203.
 9. Willett P. The Porter Stemming Algorithm: Then and Now. *Program: Electronic Library and Information Systems.* 2006. Vol. 4., No. 3. P. 219–223.
 10. Bondarchuk D.V. Choosing the Best Method of Data Mining for the Selection of Vacancies. *Informacionnye Tehnologii Modelirovaniya i Upravleniya* [Information Technology Modeling and Management]. 2013. no. 6(84). pp. 504–513. (in Russian)
 11. Salton G. Improving Retrieval Performance by Relevance Feedback. *Readings in Information Retrieval.* 1997. Vol. 24. pp. 1–5.
 12. Tan P. N., Steinbach M., Kumar V. Top 10 Algorithms in Data Mining. *Knowledge and Information Systems.* 2008. vol. 14. no. 1. pp. 1–37. DOI:10.1007/s10115-007-0114-2.
 13. Banerjee S., Pedersen T. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. *Lecture Notes In Computer Science.* 2002. vol. 2276. pp. 136–145.
 14. Tezaurus WordNET [Thesaurus WordNET]. Available at: <https://wordnet.princeton.edu/> (accessed: 05.02.2017).
 15. Bondarchuk D.V. Intelligent Method of Selection of Personal Recommendations, Guarantees a Non-empty Result. *Informacionnye Tehnologii Modelirovaniya i Upravleniya* [Information Technology Modeling and Management]. 2015. no. 2(92). pp. 130–138. (in Russian)