Н.Н. Дацун, А.А. Стольникова

Пермский государственный национальный исследовательский университет, г. Пермь

УДК 004.031.42

ОБРАБОТКА ЗАПРОСА НА ЕСТЕСТВЕННОМ ЯЗЫКЕ ДЛЯ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ МООС НА ОСНОВЕ КОНТЕНТНОЙ ФИЛЬТРАЦИИ

Аннотация. В статье описаналгоритм для реализации рекомендательной системы курсов для интегратора МООС рунета. Предложены решения проблемы приведения запросов на естественном языке к виду, с которым могут работать методы построения рекомендаций, и проблемы сравнения запросов, включающих в себя синонимы. Описан подход, с помощью которого можно получить наиболее подходящие рекомендации курсов.

Ключевые слова: МООС, интегратор, рекомендательная система, фильтрация, взвешивание слов, степень схожести, синонимы.

Введение

Прототип интегратора [1] умеет находить в своей базе данных массовые открытые онлайн курсы (МООС) с выбранными характеристиками и/или по слову, присутствующему в названии [1]. Поиск нужного курсаначинается с формулировки того, чему хочет научиться пользователь. Потенциальный обучающийсячасто не может точноуказать предметную область курса, название и другие его характеристики. Чтобы найти нужный курс, пользователю нужно перебирать сочетания характеристик и запросов для поисковой строки, т.к. в названии могут присутствовать синонимы. Это займёт много времени. Можно сделать вывод, что для поиска МООС необходима рекомендательная система.

Рекомендательная система – это система, предоставляющая пользователям рекомендации по объектам, которые могли бы их заинтересовать. Используя возможности рекомендательных систем, пользователи тратят значительно меньше времени на поиск нужного

объекта [2]. Была поставлена задача создания рекомендательной системы для существующего МООС интегратора.

1. Обзор существующих решений

Существует несколько проектов рекомендательных систем для МООС. Были проанализированы методы, предложенные в этих проектах [3-5].

Один из интересных методов – персонализация рекомендаций на основе данных из социальной сети LinkedIn [3]. Это сеть для установления деловых контактов, в ней создаются группы по интересам, размещаются вакансии и резюме, есть возможность публиковать данные о конференциях, читаемых книгах и т.п. Эти данные можно использовать для рекомендации курсов, соответствующих профессиональным интересам пользователя. Предлагается следующий подход: рекомендательная система основывается на схожести предпочтений пользователей. Каждому пользователю присваиваются наборы профессиональных сфер, и чем больше у пользователей одинаковых интересов, тем вероятнее, что их интересуют одинаковые курсы [3].

Аналогичный подход предложен и в работе [4], но основная идея метода деления на кластеры заключается в таком подборе хеш-функций для некоторых измерений, чтобы похожие объекты с высокой степенью вероятности попадали в один кластер. В качестве векторов для пользователей используются оценки, которые были даны ими определённым курсам [4].

Альтернативный метод предложен в работе [5]. Система рекомендует не курсы, а показывает список других, «похожих» на пользователя студентов. Системой предоставляется возможность переписки с этими студентами для обсуждения интересующей темы и взаимной рекомендации курсов [5].

Во всехрассмотренных научных работах предложены методы, основанные на сходстве информации о пользователях – коллаборативной фильтрации. Существуют различные варианты реализаций этих методов и выдачи рекомендаций, но для всех реализаций рекомендации формируются на основе уже собранных данных о пользователях и о том, как они

оценивают курсы. В нашем случае, в новой системе никаких данных о пользователях ещё нет, поэтому при таком подходе качество рекомендаций будет низким.

В работе [1] исследовались русскоязычные МООС-провайдеры, выбирались характеристики, которые использует большинство из них. Рейтинг МООС в итоговый список характеристик курсов не вошёл, т.к. он присутствует у небольшого числа МООС-провайдеров. Следовательно, мы не сможем получать оценки курсов от пользователей с провайдеров, поэтому концепция подхода на основе коллаборативной фильтрации не подходит для решения поставленной задачи. В нашем случае с русскоязычными провайдерами наиболее подходящим является метод, основанный на анализе содержимого. В этом заключается подход контентной фильтрации для рекомендаций – анализ метаинформации рекомендуемых объектов. Формирование рекомендаций для пользователя основано на анализе этой информации и нахождении похожих объектов [6].Основной сложностью при этом подходе является получение этой метаинформации, но в нашем случае она уже есть – это список критериев поиска, предложенный в работе [1].

2. Описание выбранных алгоритмов

Входными данными для рекомендательной системы будут являться характеристики курса и текст поискового запроса для названия МООС.

Результатом работы системы должен стать небольшой список курсов с похожими важными характеристиками и/или содержащих в названии близкие по смыслу слова. Важными характеристиками можно считать предметную область и целевую аудиторию, так как именно они должны влиять на содержание МООС. Остальные характеристики можно считать вспомогательными.

В предполагаемый список рекомендаций должны попасть курсы, содержащие в названии синонимы слов из текста запроса. Рекомендательная система должна выбрать из этого списка некоторое количество наиболее близких и предложить их пользователю. Чтобы определить, какие курсы

наиболее близки, нужно найти их схожесть с запросом. У каждого МООС и запроса есть характеристики и строка названия (или текст поисковой строки). Система должна рассчитать как схожесть векторов характеристик, так и названий, т.к. если учитывать только названия, то в выдачу могут попасть неподходящие курсы из других предметных областей, которые по какой-то причине имеют похожие названия. На рис. 1 показан алгоритм работы рекомендательной системы и её взаимодействие с пользователем и интегратором МООС, в который она будет встроена.

Существует несколько методов расчёта степени схожести. Все они практически равноценны, т.к. в случае рекомендаций курсов на точность влияет числовое значение слова или характеристики, а не метод. При имеющихся входных данных удобнее использовать косинусную меру. Считается она по формуле косинуса между векторами [7]:

$$d1 = \sqrt{(\text{вес слова 1 из названия 1})^2 + (\text{вес слова 2 из названия 1})^2 + \cdots}$$

$$d2 = \sqrt{(\text{вес слова 1 из названия 2})^2 + (\text{вес слова 2 из названия 2})^2 + \cdots}$$

$$dp = (\text{вес слова 1 из названия 1}) * (\text{вес слова 1 из названия 2})$$

$$+(\text{вес слова 2 из названия 1}) * (\text{вес слова 2 из названия 2}) + \cdots$$

$$+(\text{вес слова } n \text{ из названия 1}) * (\text{вес слова } n \text{ из названия 2})$$

$$\text{степень схожести двух названий} = \frac{dp}{d1 * d2}$$

Проблема реализации расчёта степени схожести курса с запросом заключается в том, что все алгоритмы работают с числовыми векторами. Поэтому необходимо преобразовать имеющиеся данные к числовым.

В работе [1] были выбраны возможные значения характеристик MOOCs. Значит, каждому значению каждой характеристики можно на этапе построения базы данных присвоить числовое значение. Так решается проблема построения числового вектора для характеристик курсов.

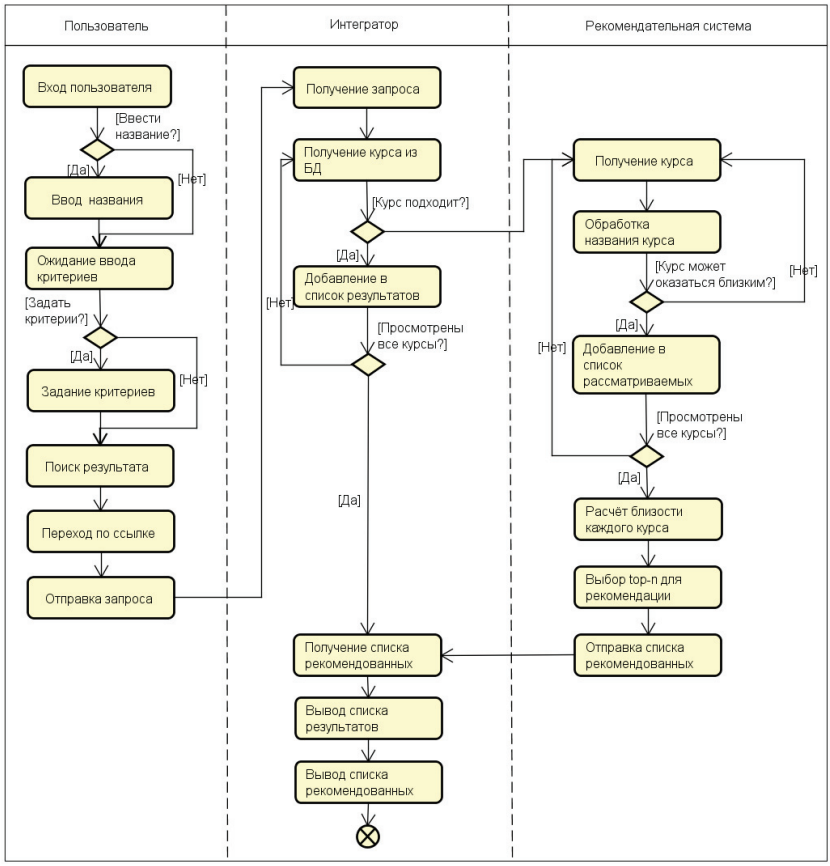

Рис. 1. Алгоритм работы системы

Строку названия курса и текст запроса тоже необходимо преобразовать в числовой вектор. В рекомендательных системах числовым значением слова можно считать его вес. Он указывает на степень связи слова с документом. Для нахождения веса слова из названия удаляются все стоп-слова (слова, которые не несут смысловой нагрузки, например, предлоги), оставшиеся слова приводятся к их основам. Так решается проблема падежных окончаний и однокоренных слов. Вес в нашей системе вычисляется по алгоритму

«сигнал-шум». Алгоритм основывается на вычислении соотношения «сигнал-шум», по аналогии с теорией передачи информации Шеннона [8]:

$$\text{шум} = \frac{\text{число названий, в которых хоть раз было это слово}}{\text{общее число названий}} * log \frac{\text{общее число названий}}{\text{число названий, в которых хоть раз было это слово}}$$

$$\text{сигнал} = \log(\text{общее число названий}) - \text{шум слова}$$

$$\text{вес слова} = \frac{\text{сигнал}}{\text{шум}}$$

3. Решение проблемы приведения слов к одному виду

Слова в названиях и запросе могут присутствовать в различных формах, а значит, чтобы их сравнивать, необходимо привести их к основе. Стемминг – процесс нахождения основы слова для данного исходного слова. Для стемминга был использован алгоритм Портера [9]. Основная идея его заключается в том, что существует ограниченное количество формо- и словообразующих суффиксов, поэтому основа слова преобразуется без использования словарейоснов. Используется множество существующих суффиксов и вручную заданные правила.Преимущество алгоритма в том, что ему не требуются громоздкие базы и словари, и этоповышает его быстродействие [10].

4. Поиск синонимов

Для того чтобы в список предполагаемых рекомендаций попали только те курсы, которые действительно могут быть подходящими, их название должно быть похоже на запрос – должны присутствовать однокоренные слова и/или синонимы. Следовательно, алгоритму нужно получать списки синонимов для слов, использующихся в запросе. Существует несколько возможностей получения синонимов слова.

Основной и открытый сервис, предоставляющий возможности получения синонимов – Яндекс. Словарь. Проблема в работе с его API заключается в ограниченности запросов. Лимит запросов к словарю достаточно велик, но если интегратор МООС создается как сайт для

открытого использования, то ограничения могут привести к нестабильной работе при большом количестве запросов.

Существует много словарей синонимов для английского языка, самый распространенный – WordNet [11]. Были попытки создать аналоги для русского языка, но на данный момент завершенных проектов нет. Поэтому было решено взять обычный словарь синонимов [12] и адаптировать его для обработки программных запросов. Для этого из словаря были удалены слова, которые не будут нести смысловой нагрузки при поиске MOOCs. Название курса – формализованный параметр, и искать в нём редко употребляемые, специфические слова или выражающие эмоции, нет смысла. Множество слов исходного словаря было разделено на 27 отдельных файлов в соответствии с алфавитом, чтобы не искать каждый раз по большому файлу, а проверять тольков отдельном. Словарь не пополнялся специфическими словами из узких предметных областей, хотя необходимость дополнений может возникать. Упорядочивать дополнения будет удобно при алфавитном разбиении словаря. Слова в файлах словаря также были приведены к своим основам. Для оптимизации поиска и простоты реализации файлы словаря были преобразованы в формат xml.

5. Формирование списка для расчёта степени схожести

Для того чтобы считать степень схожести не для всех курсов базы данных, было решено «откладывать» некоторый список курсов, которые предположительно могут оказаться подходящими. Один из критериев для попадания в этот список – наличие синонимов. Нужно решить, сколько синонимов должно включать название, чтобы быть «отложенным».

Было произведено несколько расчетов для запроса «Современные веб-технологии». Будут выбираться для расчета схожести курсы, имеющие в названии синонимы и однокоренные слова. Так, если будут «откладываться» курсы при наличии одного синонима, то в выдачу попадет большое количество неподходящих, потому что слова «современные» и «технологии» употребляются часто, но не всегда относятся к веб-разработке.

Количество синонимов должно быть пропорционально длине запроса. Например, если установить, что должно подходить 2/3 от количества слов в запросе, то это значение критерия будет верно работать для длинных строк. Однако для текста из двух слов список рекомендаций будет очень коротким.

Чтобы рекомендации формировались равномерно для всех запросов, было решено формировать не общий список «отложенных» курсов, а несколько подсписков по количеству входящих в название синонимов. Если курсов с наибольшим количеством оказывается мало, то расчет будет производиться для следующего подсписка. Внем степень схожести будет браться с меньшим коэффициентом, т.е. курсы должны оказаться ниже в списке выдачи.

Был произведен расчет для выбранного запроса, результаты расчетов степени схожести для разных подсписков представлены на рис. 2. Степень схожести для курсов с двумя синонимами бралась с коэффициентом 1, а с одним – 0.8. На графике по оси X – номер позиции в отсортированном подсписке (чем выше номер, тем ближе курс); по оси Y – степень схожести курса с запросом. Во втором подсписке было 110 курсов, поэтому на графике показан Top-10.

По результатам видно, что, хотя курсы из подсписка с одним синонимом брались с коэффициентом, первые среди них оказались даже выше, чем курсы первого подсписка. Этот результат объясняется тем, что алгоритмы вычисления веса слов и схожести лучше работают на больших объемах данных.

При полученных результатах список рекомендаций не будет полезен пользователю, так как в Top-N общего списка и в выдачу попадают не близкие к запросу результаты. Для решения проблемы должен учитываться вектор характеристик, тогда часть курсов будет отброшена по несоответствию, например, предметной области.

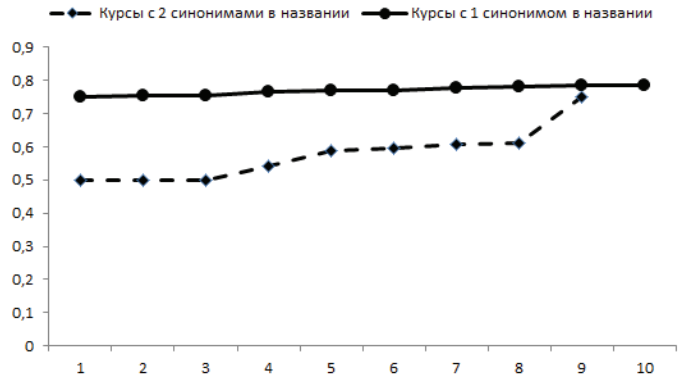

*Рис.2.*График степеней схожести курсов с разным кол-вом синонимов

Но для поиска пользователь не обязан задавать характеристики. Следовательно, необходимо улучшить алгоритм выбора курсов, близкихпо названию. Проблему можно решить достаточно просто: список результатов с посчитанными значениями степени схожести нужно также делить на подсписки, сортировать по убыванию значений и при выдаче результатов брать оттуда только первые Top-N, которых не хватает до какого-то определенного количества рекомендаций. В этом случае, курсы с большим количеством синонимов точно окажутся выше в списке результатов, и нет необходимости считать степени схожести с коэффициентом.

СПИСОК ЛИТЕРАТУРЫ

1. Стольникова А.А., Дацун Н.Н. Проектирование МООС-агрегатора для рунета. Математическое и информационное моделирование сборник научных трудов. Тюмень, 2017. С. 440-449.

2. Прохоров И.В., Мысев А.Э. Подходы к построению мульти-критериальных рекомендательных систем, использующих неявные оценки. Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2014. № 1. С. 33-36.

3. A New MOOCs' Recommendation Framework based on LinkedIn Data / K. Dai1[et al.] 2017 Lecture Notes in Educational Technology. (9789811024184), p. 19-22.

4. Collaborative Filtering Recommendation for MOOC Application/ Y. Pang [et al.] Computer Applications in Engineering Education 25(1), p. 120-128.

5. Who Wants to Chat on a MOOC? Lessons from a Peer Recommender System / F. Bouchet [et al.] Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 10254 LNCS, p. 150-159.

6. Латкин И.И. Коллаборативная фильтрация в рекомендательных системах. SVD-разложение. Молодежный научно-технический вестник. 2015. № 7. С. 15-23.

7. Селивёрстов Е.В. Повышение качества рекомендательных систем за счет учета структуры документов. Nauka-Rastudent.ru. 2014. № 4 (04). С. 13-27.

8. Селяев А.Г. Взвешивание терминов в процессах индексирования электронных информационных ресурсов. Автоматизация процессов управления. 2007. № 2. С. 92-96.

9. Porter M.F. An algorithm for suffix stripping. Program electronic library and information systems.1980. 14(3).

10. Модель определения нормальной формы слова для казахского языка. Вестник Новосибирского государственного университета. Серия: Информационные технологии. / А.М. Федотов [и др.] 2015. Т. 13. № 1. С. 107-116.

11. George A. Miller. WordNet: A Lexical Database for English.Communications of the ACM.Vol. 38, No. 11.1995, p. 39-41.

12. Александрова З.Е. Словарь синонимов русского языка: Практический справочник: Ок. 11 000 синоним. рядов. М.: Рус. яз., 2001. 568 с.