Применение методов анализа текста для рекомендаций выбора обучающегося

П. В. Корытов
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
thexcloud@gmail.com

И. И. Холод
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
iiholod@etu.ru

***Аннотация*. Доклад посвящен использованию методов обработки естественного текста для рекомендаций в выборе обучающегося. Рассмотрен процесс составления профиля студента на основе пройденных рабочих программ. Профиль использован для составления рейтинга проектов, наиболее соответствующих истории обучения студента. Произведена оценка качества рекомендаций на основе данных системы «Индивидуальные образовательные траектории».**

Ключевые слова: обработка естественного текста; ключевые слова; образовательные траектории

I. Введение

Траектория обучения студента в существенной части определяется документами на естественном языке. Так, структура дисциплин студента задается рабочими программами (РП) этих дисциплин; такими документами являются и текстовые описания различных проектов, вакансий, мероприятий и т. п. Это порождает определенные трудности.

Между описанием проекта и рабочей программой есть определенное отношение; например, проект, касающийся web-разработки должен больше соотноситься с РП “web-технологии”, чем с РП “философия”. Аналогичные отношения должны существовать и внутри РП – цели и задачи должны определенным образом совпадать с прочим содержимым дисциплины и т. п. Но структура таких отношений не поддается прямой формализации [1].

В данной работе для оценки этих отношений используются средства анализа текста. Основной сценарий использования связан с “цифровым профилем” студента – из пройденных студентом рабочих программ и прочих данных формируется характеристика обучения, по которой определяются подходящие студенту проекты и вакансии.

II. Методы решения

Для реализации этого сценария необходимо извлечь из рабочих программ, пройденных проектов и т. п. набор характеризующих их признаков, чтобы составить из них профиль студента. Затем, надо определить соответствие этого профиля с другими объектами, например, с доступными проектами.

В данном докладе в качестве “признаков” выступают ключевые слова, т. е. подмножества исходного текста, а определение соответствия является нечетким поиском подстроки в целевом тексте. Преимуществом такого подхода являются независимость от предметной области, языка, отсутствие требований к корпусу документов и более низкие системные требования. Недостатки также очевидны – не учитывается семантика текста.

Таким образом, решаются две основные задачи:

- (Задача 1) Определение соответствия между текстовым описанием проекта, дисциплины (РП) и т. п. и другим произвольным текстом, существенно меньшего размера (ключевым словом).
- (Задача 2) “Обратная задача” – извлечение из текста таких подстрок (ключевых слов), которые будут определены как «соответствующие» в рамках задачи 1.

Общий процесс и место вышеописанных задач в процессе представлены на рис. 1.

A. Задача определения соответствия

Для задачи определения соответствия использовано расстояние редактирования (Edit Distance). Из-за существенной разницы в длине сравниваемых строк, метрики вроде расстояния Левенштейна [2] или сходства Джаро-Винклера [3] в чистом виде неприменимы. Характер используемых вычислений требует, чтобы строки совпадали в длине, и несовпадение длины увеличивает значение метрик. Аналогичная ситуация наблюдается для алгоритма определения максимальной общей подстроки [4], и для алгоритмов на основе N-грамм [5] и [6].

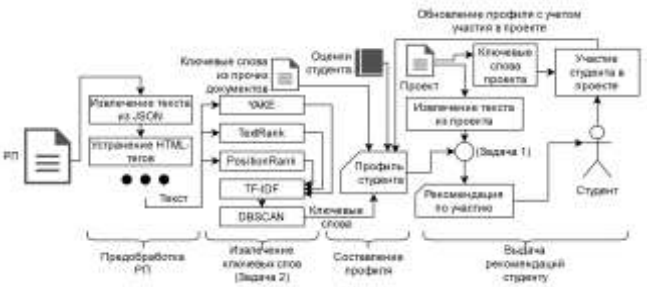

Рис. 1. Процесс работы с профилем студента

Однако, есть модификации описанных алгоритмов, которые также используют расстояние редактирования, но работают с подстроками. Субъективно лучшие результаты в данной работе показала комбинация реализаций [7] и [8]. В обоих случаях вычисляется минимальное расстояние редактирования до подмножества строк более длинной строки. При этом в варианте [7] алгоритм может находить неполные

совпадения, но периодически ошибается. Реализация [8], напротив, может находить только очень близкие совпадения, но практически безошибочно.

С учётом вышесказанного, итоговый алгоритм представлен на рис. 2.

α – минимальная длина строки, при которой считается расстояние $dist_2$. Поскольку алгоритм [8] срабатывает на близкие попадания, и поскольку в данной реализации расстояния считаются по немного другому методу, $dist_2$ масштабируется указанным образом.

B. Задача выделения ключевых слов

Как и при решении прошлой задачи, рассмотрены методы, которые выделяют подможества исходного текста, т.е. ключевые слова и фразы.

На нескольких рабочих программах проведено испытание различных методов:

- статистические модели YAKE [9] (текущий State-of-theArt) и KPMiner [10];
- графовые модели TextRank [11], SingleRank [12], TopicRank [13], PositionRank [14], MultipartiteRank [15];

Сложности с применением данных подходов связаны с характером рассматриваемых текстов. В рабочих программах какие-то части достаточно развернуты и похожи на естественный текст (примеры контрольных, лабораторных), какие-то имеют тезисный характер (содержание). Аналогичная ситуация наблюдается и в описании проектов.

На данной выборке субъективно осмысленные результаты показали YAKE, TextRank и PositionRank. YAKE оказался способен лучше всего вытаскивать короткие ключевые слова. Например, из РП “Базы данных” это “access”, “субд”, “субд access”. TextRank и PositionRank, напротив, хорошо формируют более полные фразы, например “основные понятия баз данных”, “проектирование баз данных”, “реляционной модели данных”. Однако, периодически встречаются излишне длинные фразы. Поэтому разработан алгоритм, приведенный на рис. 3, объединяющий результаты работы YAKE, TextRank и PositionRank с помощью кластеризации.

Input: a: string, b: string; len(a) « len(b)
$dist_1 \leftarrow$Расстояние из реализации [7];
if *len(a) > 10* **then**
  $dist_2 \leftarrow$ Расстояние из реализации [8];
  $dist_2 \leftarrow (25 - dist_2)/25 \cdot 100$;
else
  $dist_2 \leftarrow \infty$;
end
Result: $\max(dist_1, dist_2)$

Рис. 2. Алгоритм определения соответствия

Ключевые слова длиной больше max_len = 95 отфильтровываются. Кластеризация ключевых слов (cluster) осуществляется с помощью алгоритма DBSCAN [16] следующим образом:

- производится стемминг ключевых слов с помощью Snowball Stemmer;
- ключевые слова преобразуются в векторы с помощью TF-IDF для N-грамм с N = 1, 2, 3 [17];
- векторы кластеризуются DBSCAN с параметрами min_samples = 1 и ε = 1.125.

Параметр min_samples для DBSCAN имеет такое значение, поскольку здесь устраивает ситуация, когда в кластере находится один элемент. Параметр ε подобран экспериментально, чтобы показать субъективно хорошие результаты.

Input: $text$ - предобработанный текст
begin parallel
  $kws_{\text{TextRank}} \leftarrow \text{TextRank}(text)$;
  $kws_{\text{PositionRank}} \leftarrow \text{PositionRank}(text)$;
  $kws_{\text{YAKE}} \leftarrow \text{YAKE}(text)$;
end
/* Все ключевые слова. Кортежи вида (<значение>, <уверенность>) */
$kws \leftarrow [*kws_{\text{TextRank}}, *kws_{\text{PositionRank}}, *kws_{\text{YAKE}}]$;
/* Отфильтровать длинные */
$kws \leftarrow \{kw \mid kw \in kws, \text{len}(kw) < max\_len\}$;
/* Кластеры. i-й элемент - индекс кластера i-го ключевого слова */
$clusters \leftarrow \text{cluster}(kws)$;
$kws\_clusters \leftarrow \text{groupby}(kws, clusters)$;
$result \leftarrow []$;
/* Цикл по кластерам. В одном элементе итерации выбирается представитель кластера и итоговая уверенность в кластере */
for $kws\_in\_cluster \in kws\_cluster$ **do**
  $kws\_in\_cluster \leftarrow \text{sort}(kws\_in\_cluster,$ key=$(kw \rightarrow kw[1])$);
  $max\_score \leftarrow \max(\{kw[1] \mid kw \in kws\_in\_cluster\})$;
  $top\_33 \leftarrow \{kw[0] \mid kw \in kws\_in\_cluster, \text{kw}[1] > max\_score \cdot 0.66\}$;
  $top\_33 \leftarrow \text{sort}(kws\_in\_cluster,$ key=$(kw \rightarrow \text{len}(kw))$);
  $value \leftarrow top\_33[0]$;
  $score \leftarrow \min(\{\text{sum}(\{kw[1] | kw \in kws\_in\_cluster\}), 1\})$;
  $result.\text{push}((value, score))$
end
Output: result

Рис. 3. Алгоритм выделения ключевых слов

Представители кластеров выбираются особым образом, поскольку в алгоритмах TextRank и PositionRank более длинные ключевые фразы, как правило, имеют более высокую степень уверенности. Поэтому в качестве представителя берется самая короткая ключевая фраза, у которой степень уверенности превышает 0.66 от максимальной уверенности по кластеру. Итоговая уверенность в кластере считается как суммарная уверенность во всех представителях кластера, но с верхней границей в 1.

III. Составление профиля студента

A. Профиль одного студента

Для составления профиля студента использованы данные из АИС “Учебный процесс” по успеваемости

студентов и ИС “Индивидуальные образовательные траектории” для получения описания РП и проектов. Все текстовые данные предобработаны следующим образом: убраны HTML-теги из полей со свободным вводом, списки объединены в предложения, убраны спецсимволы. Все поля одного документа объединены в одну строчку.

Из каждой рабочей программы, по предмету которой у студента есть оценка, способом, описанным в п. 2A извлекаются ключевые слова. Также, РП дополнительно классифицируются по виду компетенций – УК, ПК, ОПК и т. п. В итоге профиль одного студента содержит записи вида, представленного в табл. 1.

ТАБЛИЦА I ПРИМЕР ЗАПИСЕЙ В ПРОФИЛЕ

Значение	Оценка	Предмет	Класс РП
web-приложения	5	Web-технологии	ОПК
simultaneous localization	5	SLAM-алгоритмы	ПК
немецкая классическая философия	5	Философия	УК

B. Взвешивание записей в профиле

Проблема с таким профилем заключается, во-первых, в его размере. Для автора данного доклада за 6 лет обучения набралось 1025 ключевых слов. Во-вторых, такой профиль не делает явного различия между общими предметами (вроде философии) и предметами специальности (вроде web-технологий).

Чтобы снизить влияние данной проблемы, из каждой группы, которая сейчас учится в ЛЭТИ, выбрано по одному студенту с наибольшим количеством оценок. Для всех этих студентов посчитан профиль способом, указанным в предыдущем пункте. Полученные профили объединены так, чтобы множество ключевых слов в одном профиле не могло быть подмножеством ключевых слов в другом.

Таким образом, всего набралось 817 студентов, из которых получилось 277 репрезентативных профилей. Из каждого профиля взято множество ключевых слов, и ключевые слова взвешены по формуле:

$$\text{вес слова} = 1 - \left(\frac{\text{вхождений } k \text{ во все профили}}{\text{вхождений } k_{\max} \text{ во все профили}} \right)^{0.25}, \quad (1)$$

где k_{max} – слово, входящее во все профили наибольшее количество раз. Полученные веса для записей из табл. 1 приведены в 2.

ТАБЛИЦА II ВЕСА ЗАПИСЕЙ В ПРОФИЛЕ

Значение	Вес
web-приложения	0.66
simultaneous localization	0.8
немецкая классическая философия	0.37

В целом, результаты отражают распределение предметов по группам – предмет “философия” в этом году находится в 179 учебных планах, web-технологии в 12, а SLAM-алгоритмы в 3-х. Также, если посчитать средний вес ключевого слова по классу РП, видно, что средний вес ключевого слова из ПК существенно выше, чем ключевого слова из УК. Эти результаты приведены в табл. 3.

ТАБЛИЦА III СРЕДНИЕ ВЕСА КЛЮЧЕВЫХ СЛОВ ПО КЛАССУ РП

Класс РП	Вес
ПК	0.73
ОПК	0.58
УК	0.39

Это объясняется тем, что УК (универсальные компетенции) должны отражать предметы “общего” направления, тогда как ПК (профессиональные компетенции) – наиболее специализированные предметы.

Таким образом, исходя из имеющихся данных, выбранное взвешивание имеет смысл.

C. Учет остальных параметров

Помимо веса ключевого слова на основе его частоты, представляется разумным дополнительно учесть класс РП и данные по оценке.

Для классов РП экспериментально подобраны значения: УК – 0.1, ОПК – 0.8, ПК, СПК – 1. Для промежутков оценок: [0,4) – 0.1, [4, 5) – 0.5, [5, 10] – 1.

Для итогового веса выбрана формула:

$$w_k = (\text{вес слова} + \text{вес класса}) \cdot \text{вес оценки} \quad (2)$$

Таким образом, в итоговом профиле студента наименьший вес имеют общие дисциплины и те дисциплины, по которым у студента наименьшие оценки. Напротив, наибольший вес имеют наиболее специализированные дисциплины с лучшими оценками.

D. Выдача рекомендаций студенту

В качестве данных для предоставления рекомендаций выбраны данные проектов из ИС “Личный кабинет партнера”. С помощью метода, описанного в п. 2A, произведено сопоставление текстового описания проекта и профиля студента.

Поскольку профиль студента содержит множество ключевых слов, итоговый “рейтинг” проекта представлен как сумма:

$$\text{рейтинг} = \sum_{k \in \text{ совпадения}} (w_k \cdot \text{вес совпадения}) \quad (3)$$

Чем выше рейтинг, тем больше данный проект соответствует истории обучения данного студента. Также, в ходе данных вычислений можно сохранить информацию о структуре данного рейтинга.

Так, для автора доклада у проекта “Разработка ИС ‘Расписание’ / ‘Деканат’” получается рейтинг 2910 со структурой, представленной в табл. 4.

ТАБЛИЦА IV СТРУКТУРА РЕЙТИНГА

Значение	Оценка	Вес
Web-технологии	5	21.6
Инструменты визуализации данных	5	9.2
Операционные системы	5	7.1

Т. е. большая часть рейтинга формируется за счет предмета “web-технологии”, что верно, т. к. этот проект действительно связан с web-разработкой.

Кроме того, для человека с направления “Лингвистика” с профилем похожего размера максимальный проекта рейтинг составляет 288, что в 14 раз ниже, чем максимальный рейтинг по профилю автора. Это также отражает реальную ситуацию, поскольку проектов, связанных с лингвистикой, в ИС на данный момент нет.

IV. Заключение

В статье рассмотрены задачи, связанные с анализом текста в траектории обучения студента. Задача определения соответствия между текстом и ключевым словом решена на основе расстояния редактирования; задача извлечения ключевых слов решена с помощью алгоритмов YAKE, TextRank и PositionRank и последующей кластеризации с помощью TF-IDF и DBSCAN.

Предложенные методы применены к данным информационных систем ЛЭТИ, чтобы составить студенту “индивидуальный рейтинг” наиболее интересных проектов. Полученные результаты субъективно имеют смысл. Дальнейшая работа в данном направлении включает в себя поиск более объективного критерия оценки результатов.

Недостатком приведенных решений является их относительная простота. Все методы работают с подмножествами текстов без учета их семантики, что может привести к неоптимальным результатам. Поэтому далее возможна оценка использования методов, учитывающих особенности русского языка [18], других способов анализа данных [19] и методов глубокого обучения для решения этой проблемы.

Последнее также может изменить «статический» характер модели, т.к. в текущем виде решение не может быть автоматически скорректировано, если выбор студента сильно отличается от предоставленной рекомендации.

Список литературы

[1] Bengfort B., Bilbro R., Ojeda T. Applied Text Analysis with Python. Enabling Language Aware Data Products. O’Reilly Media Inc., 2018. 332 с.

[2] Crochemore M., Lecroq T. Pattern-matching and text-compression algorithms // ACM Computing Surveys (CSUR). 1996. Т. 28, № 1. С. 39-41.

[3] Winkler W.E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. 1990.

[4] Larsen K.S. Length of maximal common subsequences // DAIMI Report Series. 1992. № 426.

[5] Kondrak G. N-gram similarity and distance // International symposium on string processing and information retrieval. Springer. 2005. С. 115-126.

[6] Ukkonen E. Approximate string-matching with q-grams and maximal matches // Theoretical computer science. 1992. Т. 92, № 1. С. 191-211.

[7] seatgeek/thefuzz: Fuzzy String Matching in Python. URL: https://github.com/seatgeek/thefuzz (дата обр. 03.04.2022).

[8] taleinat/fuzzysearch: Find parts of long text or data, allowing for some changes/typos. URL: https://github.com/taleinat/fuzzysearch (дата обр. 03.04.2022).

[9] YAKE! Keyword extraction from single documents using multiple local features / Campos R., Mangaravite V., Pasquali A., Jorge A., Nunes C., Jatowt A. // Information Sciences. 2020. Т. 509. С. 257-289.

[10] El-Beltagy S. R., Rafea A. Kp-miner: Participation in semeval-2 // Proceedings of the 5th international workshop on semantic evaluation. 2010. С. 190-193.

[11] Mihalcea R., Tarau P. Textrank: Bringing order into text // Proceedings of the 2004 conference on empirical methods in natural language processing. 2004. С. 404-411.

[12] Wan X., Xiao J. CollabRank: towards a collaborative approach to single-document keyphrase extraction // Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008). 2008. С. 969-976.

[13] Bougouin A., Boudin F., Daille B. Topicrank: Graph-based topic ranking for keyphrase extraction // International joint conference on natural language processing (IJCNLP). 2013. С. 543-551.

[14] Florescu C., Caragea C. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017. Т. 1. С. 1105-1115.

[15] Boudin F. Unsupervised Keyphrase Extraction with Multipartite Graphs // arXiv preprint arXiv:1803.08721. 2018.

[16] A density-based algorithm for discovering clusters in large spatial databases with noise. / Ester M., Kriegel H.P., Sander J., Xu X. // Proceedings of Second International Conference on Knowledge Discovery and Data Mining, Т. 96. 1996. С. 226-231.

[17] Mohammed J., Wagner M. Data Mining and Analysis. Fundamental Concepts and Algorithms. Cambridge University Press, 2019.

[18] S.A. Belyaev, A.S. Kuleshov, I.I. Kholod. Solution of the answer formulation problem in the question-answering system in Russian // 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). 2017. С. 360-365.

[19] Вайнтрауб А.И., Беляев С.А., Жукова Н.А. Применение интеллектуальных методов анализа на примере обработки информации, получаемой с ракет-носителей типа «Союз» в процессе пуска // I-methods. 2007. Т. 9. № 4. С. 11-25.