

АЛГОРИТМ СРАВНЕНИЯ СТРУКТУРИРОВАННОЙ ИНФОРМАЦИИ В ЗАДАЧЕ ОПТИМИЗАЦИИ УЧЕБНЫХ ПЛАНОВ

Деменёва О.С.,

научный руководитель канд. техн. наук Якунин Ю.Ю.

Сибирский Федеральный Университет

Несогласованность требований к компетенциям специалистов на рынке труда с процессом подготовки студентов в ВУЗах, а также развитие системы высшего образования России и необходимость соответствия с европейской системой предъявляют новые требования к организации учебного процесса, отражённые, в том числе, в государственных образовательных стандартах третьего поколения. Учебное планирование является основополагающим элементом в организации всего учебного процесса. В новых условиях индивидуальные учебные планы становятся правилом, а значит и процесс их создания должен обеспечить каждому студенту возможность обучаться по своей собственной траектории в рамках направления подготовки и базового учебного плана, но с учётом специфики будущего рабочего места на конкретном предприятии или в организации. Комплексное решение указанной проблемы возможно путём создания мультиагентной системы управления процессом учебного планирования, которая позволит повысить эффективность и качество учебного процесса, и, следовательно, качество образования. Одной из важных задач создания мультиагентной системы является создание методов и алгоритмов анализа текстовой и структурированной информации для анализа рабочих программ дисциплин и их аннотаций с целью генерации предложений по объединению дисциплин в поток внутри факультетов и институтов.

Для сравнения двух дисциплин представим содержимое их рабочих программ в виде структурированной аннотации. Анализ рабочих программ дисциплин позволил выявить оптимальную структуру аннотации, которая позволит максимально точно провести сравнение. Ниже представлен пример структуры аннотации.

Наименование. Основы объектно ориентированного программирования.

Семестр. 3	Связанные дисциплины. Информатика. Информатика и программирование.
-------------------	---

Ключевые слова. Основы объектного программирования, классы и объекты, наследование, полиморфизм, инкапсуляция, шаблоны проектирования.

Форма контроля. Экзамен.	Часы. 216	Курсовая работа. Нет.
---------------------------------	------------------	------------------------------

Аудиторные занятия. Лекции.	Часы. 36
------------------------------------	-----------------

Разделы дисциплин, дидактические единицы, план работ. Объектно-ориентированное программирование. Введение в объектно-ориентированный анализ и проектирование. Классы. Наследование. Обработка исключительных ситуаций. Шаблоны проектирования. Введение в шаблоны проектирования. Шаблоны для распределения обязанностей (GRASP). Порождающие шаблоны. Шаблоны поведения.

Аудиторные занятия. Лабораторные работы.

Часы. 36

Наименования лабораторных работ. Классы и объекты. Наследование и виртуальные функции. Иерархия классов. Коллекции. Модель вариантов использования. Шаблоны для распределения обязанностей. Порождающие шаблоны. Шаблоны поведения.

Самостоятельные работы. Изучение теоретического курса

Часы. 72

Литература.

1. Гамма Э., Хелм., Джонсон Р., Влиссидес Дж. Приёмы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2003. – 368 с.
2. Ларман Крэг. Применение UML и шаблонов проектирования. 2-е издание. – М.: Издательский дом «Вильямс», 2002. – 624 с.
3. Леонников А.В. Самоучитель UML. – 2-е изд. – СПб: БХВ-Петербург, 2004. – 432 с.
4. Страуструп Б. Язык программирования C++/ Пер. с англ. С.Анисимова, М.Кононова под ред. Ф.Андреева, А.Ушакова. – 2001

Определение схожести дисциплин возможно путём введения критерия (расстояния) схожести и попарного сравнения. Однако каждое из полей структуры аннотации может оказать существенное влияние на схожесть двух дисциплин. Тогда задачу поиска множества схожих дисциплин можно свести к задаче многокритериальной оптимизации

Текстовые элементы аннотации можно сравнивать, используя алгоритм Шинглов. Он состоит из четырёх шагов.

1. Канонизация текста.
2. Разбиение текста на шинглы.
3. Получение контрольных сумм.
4. Нахождение одинаковых подпоследовательностей.

Шаг 1. Для сравнения двух текстов требуется их привести к единому виду. Под канонизацией подразумевается исключение из текста знаков препинания и слов, которые не несут смысла при сравнении, так называемые «Стоп-слова».

В нашем случае, в стандартный набор «Стоп-слов»: союзов, предлогов, частиц и местоимений, включены также слова, выделенные методом экспертных оценок из предложенного для сравнения материала так, как тексты для сравнения весьма специфичны. Так же на этапе канонизации можно привести существительные к именительному падежу, единственного числа, либо оставить для сравнения только корни.

Рассмотрим для примера ключевые слова двух дисциплин «Основы объектного проектирования и программирования» и «Объектно ориентированное программирование».

1. Основы объектного программирования. Классы и объекты. Наследование, полиморфизм, инкапсуляция. Шаблоны.
2. Объектно ориентированное программирование. Классы и объекты. Наследование классов. Иерархия классов.

После канонизации:

1. Объектно ориентированное программирование Классы объекты Наследование, полиморфизм инкапсуляция Шаблоны
2. Основы объектного программирования Классы объекты Наследование классов Иерархия классов

Шаг 2. Шингл (англ.) – чешуйки, выделение из текста подпоследовательности слов. Из сравниваемых текстов выделяются подпоследовательности слов. В нашем случае тексты не превышают в среднем 3000 знаков. Поэтому выделение из текста подпоследовательности слов приведет к увеличению ошибки. Было принято решение сравнивать только знаковые последовательности, оптимальное количество знаков определяется экспериментальным путем.

Пример:

1. Шингл1 (Объектно ориентированное программирование), Шингл 2(Классы объекты Наследование), Шингл 3 (полиморфизм инкапсуляция Шаблоны)
2. Шингл 1 (Основы объектного программирования), Шингл 2 (Классы объекты Наследование) Шингл 3 (классов Иерархия классов)

Шаг 3. После разбиения текста на шинглы, используя определенную функцию и алгоритм хэширования, на выходе получаем некоторый набор контрольных сумм.

Пример:

Хеш 1:

a7bdbcb13968a694f626a5682b7f2dfd | c0c522529b0e810f73b210cc972e9966 |
95ed3beeb9bc9ff61affd4421a24c44f |

Хеш 2:

de5790caa3ee48c73f62e49000121c6f | c0c522529b0e810f73b210cc972e9966 |
7172b4096aa49236a2f7edd298a47de2 |

Затем проводим сравнения двух множеств контрольных сумм, находим их пересечение и рассчитываем процент схожести двух текстов.

Пример:

В результате, получилось одно совпадение – второе число (c0c522529b0e810f73b210cc972e9966). Это совпадение показывает то, что между двумя текстами схожесть составляет не менее 33,3%. Конечно для такого маленького текста, можно было уменьшить шаг, но и при таких начальных параметрах это является хорошим примером.

Решение задачи формирования оптимальных планов в рамках институтов и факультетов требует последовательного решения задач не только сравнения структурированных текстов, но и решения оптимизационных задач и исследования результатов на реальных данных.