

XVIII республиканский научный конкурс
молодых исследователей «Шаг в будущее Осетии»
Секция «Информатика»

Разработка автоматической интеллек- туальной системы тестирования (АИСТ)

Автор работы: Макоев Турмец Заурович

Школа: 11 класс МБОУ СОШ №7, ВЦНМО

Научный руководитель: Макаренко Мария Дмитриевна, преподаватель
программирования и информатики ВЦНМО

Республика Северная Осетия - Алания, г. Владикавказ

2016 год

Аннотация к работе

Целью моей работы является разработка системы компьютерной обработки текстовой версии материалов курса с возможностью автоматического создания тестов и проведения тестирования. Пользование этой системы не требуется каких-либо дополнительных навыков от преподавателя или учащегося. При обработке текста курса будет строиться семантическая сеть содержания, на основе которой генерируются вопросы теста. Система позволяет генерировать большое количество случайных тематических вопросов и проходить тестирование многократно. Во время работы над проектом я столкнулся с проблемами: обработки текста, предоставления пользователю инструмента визуализирующего исследуемый материал, создания GUI приложения под PC и Mac для проведения тестирования (с использованием библиотеки tkinter на языке Python 3), хранения обработанной информации для генерации тестов (выбран формат XML), а также проблемами реализации. На данный момент создана alpha-версия тестирующей системы, которая способна генерировать тестовые материалы для преподавателей и учащихся по готовой семантической сети уже сейчас. Ведется работа над созданием семантической сети по предложенному пользователем тексту.

Введение

В настоящее время большинство школьных классов и других образовательных помещений обеспечены компьютерами, интерактивными досками, проекторами и т.д.. Вся эта периферия помогает преподавателю лучше обучать своих учеников. Эти технологии помогают повысить качество образования в целом, но преподаватель порой затрачивает слишком много усилий для создания качественных материалов, которые будут соответствовать уровню компьютеризации современных школ и вузов. Во всех учебных учреждениях и образовательных интернет-сервисах практикуются мониторинги и тесты. Их вы можете встретить на таких сайтах как foxford, stepic, coursera и так далее. Например национальный открытый университет «ИНТУИТ» платит за создание тестовых вопросов. Преподаватель тратит на создание те-

стов немалые усилия и время, поскольку надо создать уникальные варианты мониторинга для каждого учащегося, оптимальное количество правильных и неправильных ответов к каждому вопросу, чтобы ученик не набрал случайным образом высокий результат. Только так обучение и проверка знаний будут эффективными. Также учителю приходится создавать материалы для тестирования по одной и той же теме при обновлении курса. Кроме того, количество созданных вопросов конечно и при повторном тестировании происходит заучивание ответов.

Наша система способна без лишних и усилий и каких-то дополнительных знаний создавать большое количество уникальных вариантов тестирования по заданному материалу.

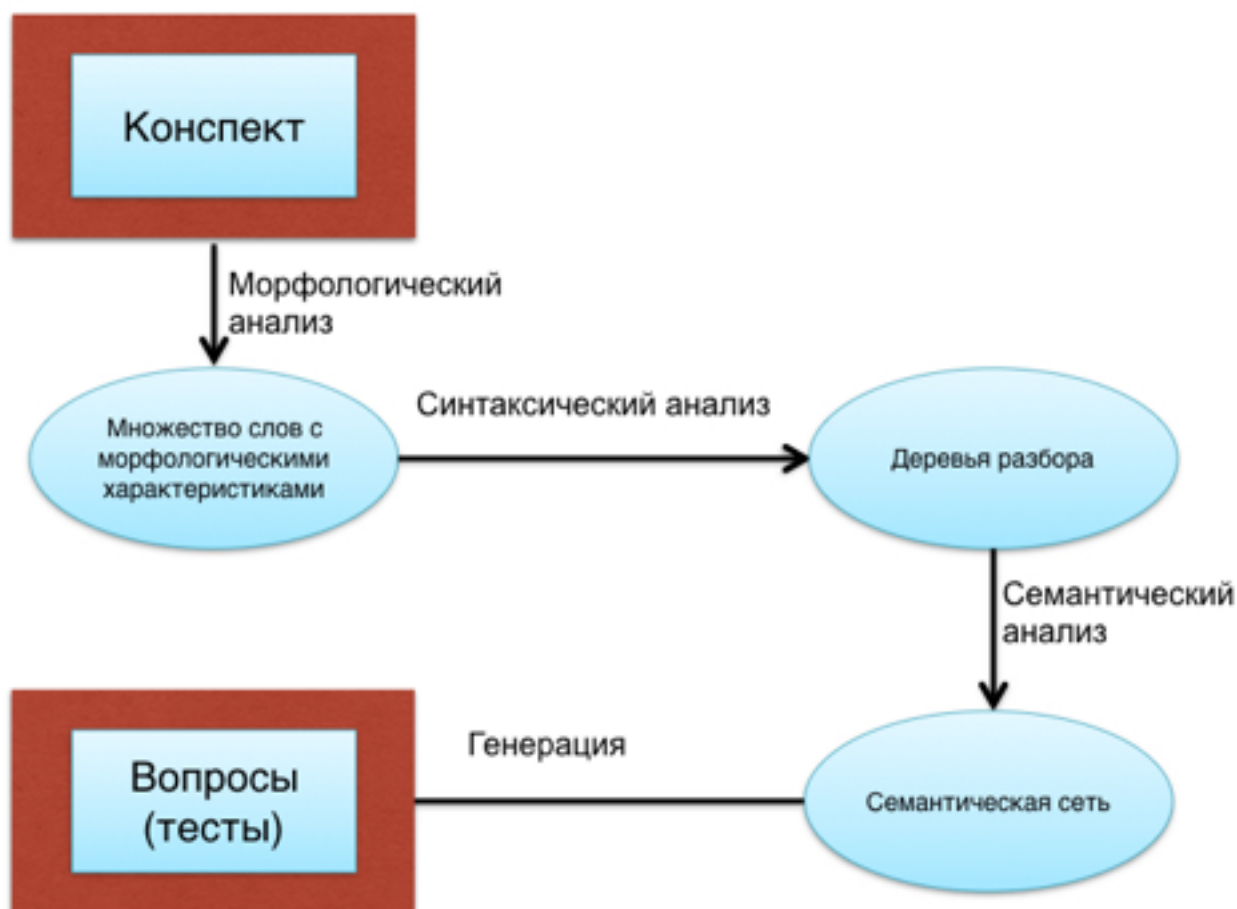
Для достижения результата, а именно, создание автоматической интеллектуальной системы тестирования, мы поставили перед собой следующие задачи:

1. Разработать архитектуру системы.
2. Выбрать нужный формат хранения всей информации.
3. Создать базовые виды вопросов и разработать логику их генерации.
4. Изучить существующие методы обработки теста и создать на основе этих материалов способ вычленения смысла из предложенного материала.
5. Создать механизм интерпретации изучаемой информации в семантическую сеть.
6. Разработать технологию визуализации данных.
7. Создать GUI приложения для ОС Windows и Mac.

Архитектура системы

Глобально система состоит из пяти этапов (см. рис. 1). Первый этап - получение базы знаний, на основе которой производится тестирование. То есть преподаватель или ученик открывает приложения для компьютера и загружает в систему изучаемый материал. Он может быть загружен в двух видах: текстовом (конспект или параграф учебника) или уже готовой семантической сети в формате xml.

рис. 1



Если была загружена готовая семантическая сеть, то сразу переходим к генерации вопросов теста. Если же на вход программе был дан текст, то производится несколько анализирующих процессов: морфологический анализ, синтаксический анализ, семантический анализ. После всех трех видов анализа из исходного текста получается семантическая сеть, которая также может быть представлена в формате xml.

Пример семантической сети (интеллект-карты) по теме «робототехника и устройство компьютера» (см. рис. 2)

Объекты на данной карте: процессор, пк, вычислительная машина и т.д.

Связи изображены стрелками. Каждой стрелки соответствует определенная надпись, к примеру, стрелки, связывающая «процессор» и «intel» имеет надпись «isa». Все эти надписи характеризуют тип связи между объектами, ок которых подробнее раскатывается ниже.

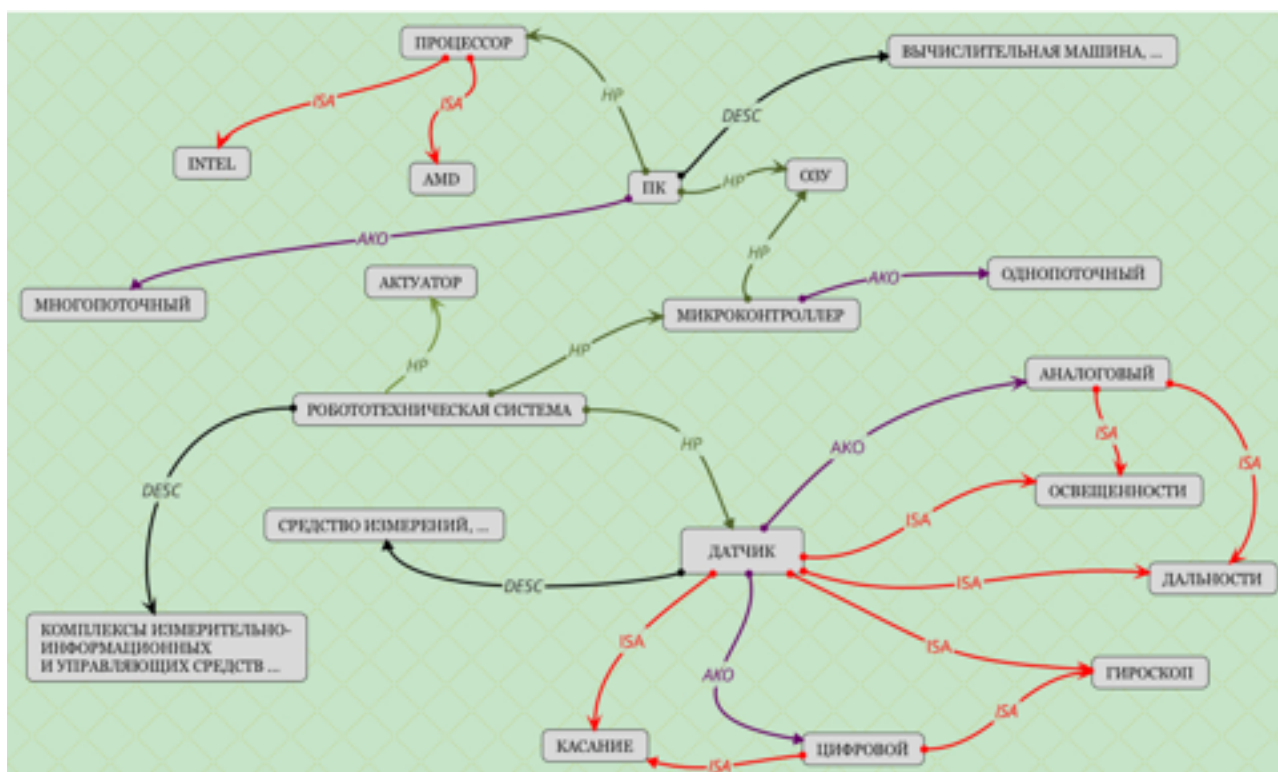


рис. 2

Формат хранения информации

В процессе погружения в решаемую проблему встречались различные существующие структуры данных и форматы их хранения. Для того, чтобы выбрать подходящее решение для решаемой задачи мы сформулировали конкретные требования:

1. Возможность хранить структуру данных «граф» (интеллект-карту, семантическую сеть), вершинами которого являются объектами, а ребра связями (видами вопросов)
2. Формат должен быть легко обрабатываемым программно

Согласно этим критериям был выбран формат xml, который способен хранить любые структуры данных. Для него существует множество библиотек на языке Python 3.

Виды вопросов и логика их генерации

Для генерации тестовых материалов в системе были придуманы несколько типов вопросов (связей объектов). В нашей системе существует шесть типов связей (стрелочек) (см. рис. 2):

1. Термин - определение (DECS - description). Пример - «датчик» является «средством измерения». С помощью этого вида вопросов генерируются вопросы, требующие ответа в виде определения
2. Определение - термин (DECS - description). Пример - «средством измерения» является «датчиком». С помощью этого вида вопросов генерируются вопросы, требующие в качестве ответа объект, определение которого было предложено.
3. Состав (HP). Пример - «робототехническая система» состоит из «актуатор», «микроконтроллер», «датчик». Вопросы этого типа используются чтобы спросить о составе какого-либо объекта.
4. Свойства (АКО). Пример - «датчик» может быть «аналоговый», может быть «цифровой». Вопросы этого вида используются для побуждения к перечислению всех свойств или видов определенного объекта.
5. Пример (ISA). Пример - пример «датчик» - это «гироскоп». Вопрос такого типа используется, когда необходимо привести пример какого-либо объекта.
6. Класс, тип (ISA). Пример - «гироскоп» - это «датчик». Этот тип вопроса является обратным пятому типу. Используется тогда, когда нужно по примеру объекта определить его принадлежность к какому-либо классу.

Генерирование самих вопросов происходит по следующей логике:

(Для примера разобран алгоритм генерации вопросов тип DESC)

Выбирается пять случайных связей типа DESC. После из выбранных пять (возможно другое число) случайно выделяется одна, которая будет содержать вопрос и правильны ответ.

Остальные две будут являться неправильными ответами. Такое отношение кол-ва правильных ответов к вол-ву неправильных (а именно 0,5) уменьшает вероятность случайного выбора правильного ответа до 0,33, что делает обучение эффективным. Разберем частный случай.

Выберем все три связи DESC на рис. 2. Это связи:

1. «робототехническая система» - «комплексы измерительно-информационных и управляющих средств»
2. «ПК» - «вычислительная машина»
3. «датчик» - «средство измерений»

Теперь выберем случайным образом одну из трех связей, к примеру возьмем вторую связь («ПК» - «вычислительная машина»). Она будет содержать вопрос и правильный ответ. Соответственно первая и вторая связи будут неправильными ответами. В результате сформируется следующий вопрос:

«ПК это?»

- «комплексы измерительно-информационных и управляющих средств»
- «вычислительная машина»
- «средство измерений»

В прил. 1 схематично описаны алгоритмы генерации всех шести видов вопросов.

Изучение способов анализа текста, вычленение смысла

Рассмотрим логику обработки текста подробнее. Происходит три вида анализированная: морфологический анализ, синтаксический анализ, семантический анализ. В процессе обработки из текста убираются не несущие смысловой нагрузки знаки препинания (кроме тире, поскольку после тире часто следуют определения различных понятий, что очень важно при тестировании), убираются междометия, большинство союзов и предлогов. Выделяются и программно помечаются редкие слова русского языка (если слово редко используется в русском языке, но относительно часто в исходном тексте, то оно несет важную смысловую нагрузку), помечаются слова, связывающие два подряд идущих предложения (поскольку это помогает вычленять смысл не отдельного предложения, а части текста в целом).

Синтаксический анализ описывает связь между словами, что является основополагающим фактором в дальнейшем анализе текста.

Морфологический анализ обрабатывает каждое слово. Задача этого этапа определить часть речи и форму каждого слова. В процессе этого анализа каждому слову, часть речи которого удалось определить, присваивается массив возможных типов вопросов, в которых это слово может участвовать. К примеру, существительное «трансмиссия» может иметь свойство «автоматическая», либо «механическая», которые выражены именами прилагательными, эти объекты могут использоваться в вопросах вида «ISA». То есть определенные части речи используются в определенных видах вопросов.

С помощью семантического анализа мы узнаем некоторые статистические данные о тексте такие как: частота использования определенных слов, частота использования определенных частей речи. и т.д.. Слова, которые использовались в исходном тексте чаще других несут более значимую смысловую нагрузку, то есть про них в тексте имеется больше информации и следовательно про них должно быть больше вопросов для качественной экзаменации.

После выполнения алгоритмов обработки из исходного текста получается семантическая сеть, которая представлена в программе в виде графа, который хранится списком ребер. Вершинами графа являются слова, а ребрами типы связей (вопросы).

Визуализация материала с использованием сторонних приложений

Для визуализации и проверки корректности семантической сети нужно было разработать или найти существующий редактор семантической сети. Во время решения этой подзадачи нами были изучены разные программы разных компаний:

1. ConceptDraw MINDMAP Pro
2. XMind
3. MindManager
4. EdrawMindMap
5. MindMeister

6. MindOmo

Все эти продукты помогают визуализировать знания, но есть и отличия, которые побудили нас выбрать в качестве инструмента приложение XMind. Критерии сравнения приведены в таблице ниже.

Название	визу- а л и- зация	редак- ти ро- вание	доб.объ- е к т (п о д- пись)	фильтр	сохр\от кр	Э к с- порт в откры- т ы й формат	W e b - сервис	С в о- бодное ПО
ConceptDraw MINDMAP Pro	+	+	+	+	+	+	-	-
XMind	+	+	+	-	+	+	-	+
MINDMANAG ER	+	+	+	+	+	+	-	-
EDRAWMIND MAP	+ -	+ -	+	-	+	+	-	-
MINDMEISTE R	+	+ -	+	-	+ -	+	+	-
MINDOMO.C OM	+	+	+	-	+	+	+	-

Также XMind создает xml файл по интеллект-карте, что является большим плюсом этого решения. Преподаватель просто перетягивает мышкой геометрические фигуры и связывает их стрелками, в результате получая семантическую сеть, про которой уже наша система генерирует варианты для тестирования.

Разработка GUI приложения

Для реализации был выбран язык Python 3. Поскольку разработка на нем ведется быстро, приложения на этом языке исполняются любой ОС, для него существует множества библиотек, которые были использованы в процессе разработки (библиотеки: xml, tkinter).

При запуске приложения пользователь видит окно, в которых необходимо указать нужный файл интеллект карты, свое имя и количество вопросов, которое нужно сгенерировать. Если все данные корректны, то происходит переход к тестированию. Пользователь, отвечая на вопросы, может отмечать один или несколько ответов (в зависимости от вопроса). Тестирование не будет закончено пока пользователь не прочтет (если не может ответить на всё) все вопросы. После завершения тестирования пользователю показывается оценка и процент выполнения теста.

Рассмотрим техническую часть реализации. Первым этапом происходит считывание xml файла, содержащего описание семантической сети. Создаются ассоциативные массивы, содержащие вершины графа (объекты) и ребра (типы связей). После запускается цикл генерации вопросов. В каждой итерации генерируется случайное число, определяющее номер шаблона (см. прил. 1) и в зависимости от него создается вопрос определенного типа. Итерация заканчивается записываемым вопросом в список (list) вывода. Каждый элемент списка вывода хранит в себе: номер типа вопроса, вопрос, варианты ответов, номер(а) правильного(ых) ответа(ов). Затем происходит последовательное отображение вопросов на экран. При выборе пользователем варианта ответа он сохраняется в массиве, который обрабатывается после завершения тестирования, что позволяет пользователю возвращаться к предыдущим вопросам повторно.

Заключение

Нашей задачей стояла разработка автоматической интеллектуальной системы тестирования. На данный момент разработано приложение обладающее следующим функционалом:

генерация тестов по уже существующей интеллект карте, проведение тестирования по сгенерированным вопросам, предоставление оценки и процента выполнения теста пользователем. На данный момент ведется работа над изучением технологий обработки текстов на естественном языке.

Эта система будет полезна преподавателям при проведении мониторингов, для учащихся, которые хотят проверить свои знания. Пользователь всего лишь должен создать интеллект карту в приложении XMind и загрузить полученный файл в систему «АИСТ». Приложение проведет тестирование и предоставит статистические данные о нем, что будет очень полезно как преподавателям, так и учащимся.

Используемая литература

- 1) tkinter - Python interface to Tcl/Tk - <https://docs.python.org/3.4/library/tkinter.html>
- 2) Работа в программе XMind. - <http://comp-user.ru/using-xmind.htm>
- 3) Справка по базовым фичам интерфейса XMind, сделанная в самой программе - <https://habrahabr.ru/post/123627/>
- 4) Классификация предложений с помощью нейронных сетей без предварительной обработки - <https://habrahabr.ru/company/meanotek/blog/256593/>
- 5) Прохоренок Н.А. Python. БХВ-Петербург, 2011.