**Разработка системы рекомендации специальности абитуриенту на основании анализа профиля социальной сети**

**Автор – Андрющенко М. В. (941 группа)**

**Дипломный руководитель: доц. Швец О. М.**

**Очень часто связи нет даже на явных специальностях типа мех-мат или даже программирование (взять некоторых одногруппников). Это беда.**

**В таком случае, нужно обучающую выборку формировать сложнее, подтверждая заинтересованность в направлении другими способами:**

* **с помощью словарей находить множество относящихся к направлению терминов (Линкедин?) и искать их в постах (или в topics из LDA)**
* **можно выявлять «типичные» профили (образуют кластер) и их отбрасывать, обучаясь только на «нетипичных», содержащих тематический уклон**

Программное обеспечение систем – 89%  
Филология и перевод – 11%  
Управление процессами перевозок – 5%  
**(использовать скорее не специальность, а направление)**

IT

математика

физика

менеджмент

филология

экономика

дизайн

**менее явные:**

химия

биология

**Обучаться на бывших/настоящих студентах и их специальностях!  
Профиль1 ->** Программное обеспечение систем  
**Профиль2 ->** Филология и перевод  
…

Здесь можно применить целое множество разных методов и провести их сравнительный анализ.   
Выводим признаки (featuers) для классификации. Подключаем и анализ текста, и анализ метаинформации, анализ школы.

Веб-приложение, без базы.

**В какие школы направлять агит-бригады.**

Какие факторы оказались наиболее весомыми?

**Предикторы:**

**какой вуз закончили родители? (если указаны)**

**Технологии**: Python, Scala. Все методы реализовать с нуля, проверять их с помощью готовых библиотек (Gensim).

Применять LDA к названиям/описаниям **интересных страниц** и **групп**.   
**word2vec** по ключевым словосочетаниям.  
Использовать корпус текстов русского языка: <http://www.ruscorpora.ru/>

**Как валидировать?**

1. Человек интересуется айти, если он подписан на одну из {DOU, HabraHabr, Типичный программист, Data Science, IT КПИ, …} – подобрать вручную список из 40 самых популярных айти-групп. Возможно, кто-то уже ведёт такой список, поискать.  
   **Обоснование**: группы для программистов – тематика очень специфическая, это не котики и искусство. Будет и определённый процент ошибки при таком подходе.  
   **Проблема**: это можно сделать для конкретной темы, но не для всех возможных тем сразу. Так мы можем провалидировать определение конкретной темы.
2. Брать людей, у которых есть вуз и специальность. Если человек учится на Архитектуре, то это его интерес.

Следовательно, можно применять и **supervised learning** подход. Text[i] –> 1/0.

Плюс можно анализировать:

* метаинформация профиля (плохо структурированные данные);
* название/описание интересных групп/страниц;

**Нужна ли БД вообще? Нет.**

**Применение латентного размещения Дирихле для анализа публикаций из наукометрических баз данных**. Целью работы является определение наиболее подходящей тематической модели для классификации научных публикаций по авторам-однофамильцам. Проанализированы вероятностные тематические модели и предложено использование модели латентного размещения Дирихле — лидирующей среди вероятностных тематических моделей благодаря многочисленным обобщениям и приложениям к анализу коллекций текстовых документов. Для сравнения выбрана модель латентно семантического анализа, недостатки которой решаются при помощи рассматриваемой модели. Модель применена в проекте по извлечению публикаций из наукометрических баз данных. В этом проекте применение тематического моделирования позволяет решить проблему разделения публикаций авторов-однофамильцев, где в качестве коллекции документов выбраны названия публикаций. Результаты показали что модель латентного размещения Дирихле уступает латентно-семантическому анализу, когда используется малый обьем содержимого документов. Поэтому для коллекций документов малого обьема предпочтительным является и

На сегодняшний день очень важно эффективно планировать процесс разработки в информационно-технологической компании. Скорость и качество разработки – основные критерии для успешного выполнения любых проектов.

Основную задачу, по автоматизации организации процесса разработки, берут на себя системы управления проектами. Благодаря внедрению определенных проверенных методик управления проектами (SCRUM, KANBAN и т.д.), обеспечивается выполнение поставленных задач в срок. Ключевым фактором успеха проектного управления является наличие чёткого заранее определённого плана, минимизации рисков и отклонений от плана, эффективного управления изменениями. Одни из самых известных и популярных систем управления проектами: VisualStudio Online, JIRA, Asana, Wrike, Worksection, Redmine, Битрикс24 и другие.

В основном, все доступные системы управления проектами имеют бесплатный ограниченный набор функциональных инструментов, предназначенных для ознакомления. Данный фактор является ключевым в возможности внедрения таких систем для небольших командах или групп студентов (для обучения коллективной разработки)

Основной целью разработки системы планирования производственного процесса информационно-технологической компании является предоставления широкого инструментария всем желающим для организации собственного структурированного процесса разработки ПО. Для создания исключительно новой отличительной черты данного вида ПО, необходимо позаботиться об автоматическом прогнозировании сроков выполнения работы (на основе предыдущих, похожих проектов). Также необходимо отслеживать эффективность работы (вести статистику) каждого члена команды, и на основе полученных данных, при помощи специальных алгоритмов прогнозировать вероятность успешного выполнения, поставленной перед разработчиком, задачи.

Преследуемые цели:

* Доступность данного вида ПО для всех желающих (особенно студентов);
* Увеличение степени организации производственного процесса путем разбиения на подзадачи;
* Налаживание процесса продуктивной коммуникации между членами команды;
* Сбор, анализ и прогнозирование сроков разработки и эффективности разработчика для данного проекта.

Название системы: KBPM (Kite business process management)

Технические особенности: серверная часть ПО – PHP 5.6, MySQL, Sphinx; клиентская часть ПО – HTML5, CSS3, ECMAScript 6; Принцип клиент-серверного взаимодействия: RESTful API