Автоматизированная система распознавания и классификации резюме

© Качаева Т. В. © Южиков В. С.

Казанский Государственный Университет
cvbox@visionai.com

Аннотация

В статье описывается система для автоматического анализа, классификации и ранжирования резюме кандидатов. Рассмотрены методы и алгоритмы формирования базы кандидатов на основе автоматического анализа поступающих резюме.

1 Введение

Описание себя – сложная задача. Каждый из нас индивидуален и при общении формулирует и передает информацию по-своему. Человек, который в силу своей профессиональной деятельности просматривает и анализирует ежедневно десятки резюме, сталкивается с самыми разными стилями их написания. Ведь форматирование, шрифты и логическая структура текста порой полностью произвольны. В связи с этим возрастает роль систем, которые позволяют переложить на себя часть наиболее рутинных действий HR-менеджера по обработке резюме. Поэтому важная задача систем автоматического сбора (реферирования) информации о кандидатах - выбор вновь появившихся резюме кандидатов и определение порядка его расположения в существующей системе.

Сейчас процесс отличия нового резюме от ранее существующего ложится на плечи HR-менеджера. Также он ответственен за помещение резюме в некоторую группу. Кроме того, в данной предметной области существуют критерии, по которым отбираются резюме кандидатов, заслуживающих внимания. Чем большему количеству критериев удовлетворяет резюме, тем более вероятно, что оно будет замечено HR-менеджерами. Стоит отметить, что критерии отбора резюме субъективны и у каждого они свои. Значит, результат отбора резюме кандидатов в конечном итоге целиком зависит от HR-менеджера.

Кроме того, существуют еще и две противоположные тенденции: количество резюме кандидатов растет (причем растет количество копий резюме одного и того же кандидата), а допустимое для их обработки время сокращается. Это приводит к тому, что невозможно обработать весь поток поступающих резюме, когда их обработка производится вручную.

Вследствие этого, создание автоматизированной системы обработки резюме кандидатов является актуальной проблемой, так как позволяет ускорить процесс обработки резюме кандидата. Данная система может решать следующие задачи:

1. Анализ структуры и распознавание полей резюме для получения формализованного представления, пригодного для дальнейшей обработки.
2. Автоматический сбор резюме с определенных сайтов (например: job.ru) и добавление их в базу данных.
3. Классификация всех резюме по заданным тематикам.
4. Устранение дубликатов.
5. Гибкий поиск по запросам пользователей.
6. Ранжирование резюме внутри группы:
   - С учетом существующей иерархии предметной области.
   - По матрице навыков и умений.
7. Аннотирование резюме и групп резюме.

2 Обзор существующих решений

В доступных источниках не удалось найти описание и решение задачи анализа и распознавания документов, подобных резюме. Встречаются только отдельные подзадачи, либо общие подходы для решения задач анализа документов.

Поэтому было решено разработать собственные методы на основе имеющихся алгоритмов анализа текстов.

Описанные в работах [3,4,5,6,7] решения задач анализа текста послужили опорной точкой для создания решений задач по обработке резюме. Сейчас классификация резюме происходит используя алгоритм BP25[1, 5]. Планируется улучшить качество классификация используя методы, описанные в работах [7,8,9,10].

3 Идея исследования

Автоматическое распознавание резюме кандидата способно увеличить скорость работы с резюме кандидатов. Так как перенос информации о кандидате сейчас осуществляется вручную, то возможна потеря или искажение информации.

Труды 9ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2007, Переславль-Залесский, Россия, 2007.

Поэтому автоматическая обработка резюме берет на себя задачу «не потерять» информацию, содержащуюся в тексте. Но, надо учитывать тот факт, что автоматическое извлечение не всегда может быть корректным, поэтому полностью автоматический режим не подходит для данной задачи. Поэтому более подходящим вариантом является автоматизированный режим с ручным подтверждением. В большинстве типовых случаев автоматическая обработка дает хорошие результаты, но когда система не смогла правильно распознать некоторые фрагменты резюме, то пользователь при помощи встроенных средств выполняет разметку вручную. Таким образом, система получает в свое распоряжение еще один экземпляр обучающей выборки, который будет использоваться в дальнейшей работе.

Также система должна уметь находить ошибки и проверять непротиворечивость резюме. Например, проверка непересечения периодов работы в разных местах. Или если человек пишет, что он обладает определенным навыком, то система проверяет, насколько это понятие или его аналоги встречается в описании проектов. Также автоматизированная система берет на себя задачу определения принадлежности вновь поступающего резюме к какой-то определенной группе и обновления состояния базы резюме кандидатов.

Форматы представления резюме кандидатом нигде не зафиксированы, поэтому в общем случае можно считать, что кандидат отправляет свое резюме в произвольной форме. Подавляющее большинство резюме разбивается на фрагменты авторами, т.е. его построение имеет логику. Логические блоки обычно именуются. Это позволяет выделить их в тексте и использовать для улучшения результатов последующих методов.

Несмотря на то, что стили написания резюме сильно варьируются, во всех присутствует описание ФИО, даты рождения, описания предыдущих мест работы и т.д. Поэтому можно выделить универсальное множество признаков, которые должны (или могут) встречаться в большинстве резюме.

Для того чтобы обеспечить системе гибкость, предполагается не использовать фиксированные шаблоны и правила извлечения данных. Для этого на этапе обучения создается предполагаемая модель объекта данных (в нашем случае объект данных -это резюме). Модель должна быть максимально полной, т.е. отражать все элементы, которые могут встретиться в объекте. Основной принцип – это не создавать модель для каждого случая, а наоборот, каждое резюме «подгонять» под построенную модель.

Для каждого конечного элемента должен создаваться набор правил извлечения.

Далее система переходит в режим обработки на основе созданной модели. Если встретился вариант, который система неправильно распознала, то в режиме «дообучения» указывается дополнительное правило извлечения, которое система запоминает и заносит в свой банк знаний. Если встретился такой случай, когда в резюме присутствует новый элемент данных (скажем, информации о водительском удостоверении не было в модели данных), то в режиме редактирования модели объекта добавляется новый элемент и к нему приписывается правило извлечения. После чего во все раннее сохраненные резюме добавляется ссылка на новое свойство, чтобы вся база данных была в едином формате. После чего во все раннее сохраненные резюме добавляется ссылка на новое свойство так, чтобы вся база данных была в едином формате.

Следовательно, получается очень гибкая система, которую можно настроить на извлечение данных из практически любой области – резюме, анализ анкет, платежных поручений и т.д.

4 Описание метода

4.1 Классификация резюме

Классификация – это группирование всех резюме в классы, в соответствии с их признаками. Список классов определяется заранее и включает в себя все необходимые области деятельности: менеджмент, IT и т.д.

Каждое резюме после обработки представлено в виде схемы ключ-значение.

$R=\{ r_i \}$, где r_i - резюме.

$r_i = \{<$ ключ, значение $>\}$,

где $i=1\ldots n$, n – количество признаков.

Модель описания, как уже было сказано выше, одинакова для всех резюме.

Для каждой группы определим условия вхождения резюме в данную группу.

$F=\{f_{i1},\ldots,f_{in}\}$

Применив данные условия, получим множество пересекающихся подмножеств. $C_i \cap C_j \neq$, где $C_i = f_i(r_i)$. Это показано на *Рисунке 1*.

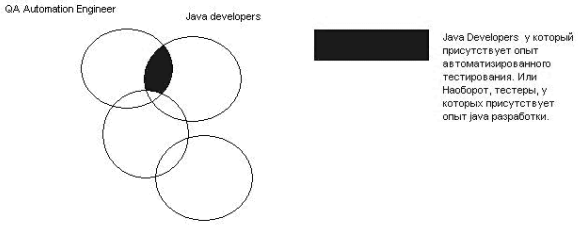

Рисунок 1. *Первоначальное разбиение множества на группы.*

Классификация производится на основе меры TF-IDF по каждому признаку. Для ее нахождения применяется стандартная BM25 формула [1,5]. Далее вычисляется суммарный TF-IDF вес, значение которого определяет класс, в который попадает резюме.

4.2 Ранжирование резюме внутри группы

Внутри каждой полученной группы определим условия, по которым множество разбивается на систему непересекающихся подмножеств. Это делается для того, чтобы сгруппировать резюме внутри каждой группы. Условия группировки

задаются пользователем в файлах настройки системы.

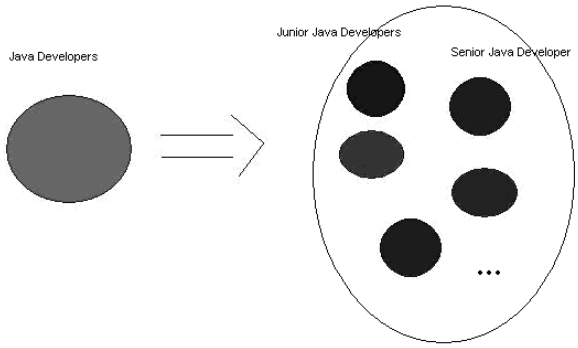

Рисунок 2. *Определение непересекающихся подгрупп с использованием матрицы навыков и умений.*

Условия принадлежности группе определяются как условия на значения свойств. Например, чтобы у человека опыт разработки был больше 2-х лет и в описании определенные навыки. После определения подгруппы, считается, что резюме распознано, соответствующим образом сохранено в системе и готово к использованию.

4.3 Аннотирование резюме кандидата

Сложной задачей является составление грамотного описания, чтобы пользователю с первого взгляда было понятно, о ком идет речь в данном резюме. В разных системах обработки текстов эта задача решается по-разному.

Наиболее распространенный способ – формирование списка ключевых слов. Этот способ прост в реализации, но недостатком его может являться недостаточная информативность получившегося текста.

Другой способ – составление реферата текста. Данный способ дает, в зависимости от системы реферирования, довольно понятный реферат, но сложен алгоритмически.

Учитывая предметную область задачи, предлагается, кроме всего прочего, аннотировать резюме кандидата методом дополнения подмножества признаков по навыкам и умениям кандидатов. Например, что у человека нет таких-то навыков.

5 Программная реализация

5.1 Обучение системы

Вручную было отобрано и размечено 200 резюме. Разметка включала в себя выделение блоков «Образование», «Опыт работы» и «Прочее». В блок «Прочее» вошли такие разделы как «Контактная информация», «Хобби» и т.д. Основная идея состоит в том, чтобы из размеченных резюме обучающей выборки выделить список ключевых слов для каждого из описанных выше блоков.

На начальном этапе обработки удаляются непечатаемые символы, стоп слова (предлоги, междометия), символы разметки, цифры, а также лишние пробелы, числа, аббревиатуры. Т.е. те части речи, которые не могут быть использованы в качестве ключевых слов. После этого файлы готовы к работе.

Вторым этапом обработки является преобразование всех слов текста в нормальную форму путем отсечения окончаний и суффиксов. Для этого используется адаптированный для русского языка алгоритм «стеммер Портера» [2]. Он прост в реализации, поскольку построен на эвристических правилах усечения слов и не требует словарной поддержки. К сожалению, в нетипичных словах он допускает ошибки, но это происходит достаточно редко и не влияет на конечный результат.

После нормализации слово помещается в список ключевых для данного блока. Для сформированных ключевых слов вычисляется также и их частота встречаемости.

Стеммер иногда ошибается и на выходе получается бессмысленные строчки длины меньше трех. Так как эти строки не относятся к смысловым носителям блока, можно их удалить. В качестве результата второго этапа получаем список основ ключевых слов с частотой их встречаемости в тексте блока.

После формирования списков основ ключевых слов для каждого блока получилось так, что одно слово может принадлежать нескольким спискам. В этом случае оно не является уникальной характеристикой блока. Для обеспечения уникальности необходимо избавится от пересечения основ ключевых слов. Сравнивались частоты слов, входящих в пересечение, и слово оставалось в той группе, где частота его вхождения больше и удалялось из группы, где частота вхождения слова меньше.

Таким образом, мы получили непересекающиеся множества основ ключевых слов с частотными характеристиками для каждой из названных групп.

5.2 Анализ заголовков

Как показали практические эксперименты, использование только основ ключевых слов не дает достаточно точного разбиения. Поэтому для определения границ блоков использовались ключевые слова и фразы заголовков блоков. На этапе ручного парсинга входных файлов отдельно выделялись заголовки, и паралельно с процессом определения ключевых слов для каждого из блоков формировался список ключевых слов и подстрок заголовков для каждого из блоков.

При попытке установить разбиение на блоки решение использовать ключевые слова заголовков работало ≈ в 80% случаев. Если граница заголовка не была распознана, то информация сохранялась в системе.

5.3 Разбиение на смысловые блоки

С помощью анализа заголовков мы определили *возможное* разбиение текста резюме на блоки. Для

того, чтобы подтвердить или скорректировать это разбиение, используется статистический подход на основе имеющейся информации о ключевых словах.

В результате получаем текст разбитый на блоки «Образование», «Опыт работы» и «Прочее».

Разбиение текста на блоки происходит следующим образом. Для каждого слова в резюме мы находим его нормальную форму, используя уже упомянутый алгоритм стеммера [2]. Далее ищем полученную нормальную форму в списках ключевых слов и если находим, то «окрашиваем» анализуемое слово в цвет группы.

После подобного анализа всех слов в резюме, мы получим картину вида:

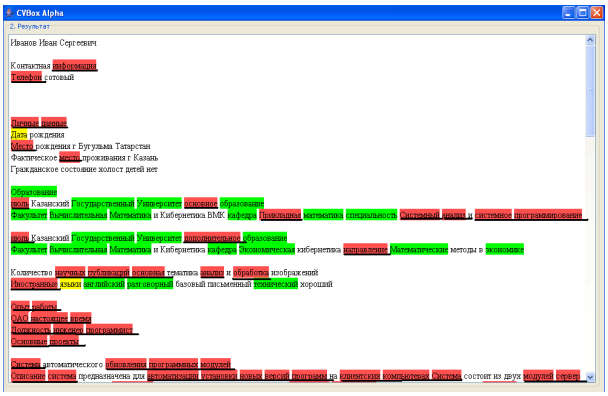

Рисунок 3. *Распознавание опыта работы и образования по ключевым словам. Подчеркнуты слова из группы «Опыт работы».*

На *Рисунке 3* можно видеть, что тип самих групп и их границы прослеживаются достаточно четко.

5.4 Выделение фактов

В информатике факт — это единичное значение данных, созданное или использованное бизнес-процессом.

Факты нашей предметной области это: дата рождения, семейное положение, дата поступления и окончания одного (нескольких) учебных заведений, навыки, гражданство, уровень владения иностранными языками и т.д.

Выделение фактов происходит по определенным правилам, построенных на основе регулярных выражений. Они формируются в процессе обучения системы, а также дообучения системы конечным пользователем.

6 Заключение

Представление резюме кандидатов в виде универсальной модели позволяет достичь хорошего качества классификации. Помимо этого, предложенный способ позволяет решать такие задачи, как аннотирование резюме кандидата и сопоставление резюме существующей вакансии.

В работе предлагается метод обнаружения групп содержательно близких резюме и метод выявления «актуальных резюме», т.е. тех, на которые существуют заявки работодателей. Также предложен метод обучения системы.

Разработан макет системы, реализующий все вышеописанные алгоритмы. Проведенный анализ системы на примерно 500-х резюме, показал, что в 87% случаев сделала правильное разбиение на блоки. И в 82% случаев были правильно выделены факты.

Был проведен анализ случаев, когда система не смогла правильно разбить текст на блоки.

Основными причинами являлись использование нетипичных стилей написания резюме, использование HTML-таблиц.

Предложенный подход может быть использован для решения задач классификации, сегментации и выделения фактов в других областях, связанных с документооборотом.

Литература

1. S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In TREC-3, 1994.
2. Russian stemming algorithm, 2005: http://snowball.tartarus.org/algorithms/russian/stemmer.html
3. Зевайкин А.И., Корнеев В.В. Формирование выпуска новостей на основе автоматического анализа новостных сообщений. Интернет-математика 2005, Москва.
4. А. Гулин, М. Маслов, И. Серголович Алгоритм текстового ранжирования Яндекса на РОМИП-2006. РОМИП-2006, Суздаль.
5. Р.Ф. Кузнецов Классификатор веб-страниц на базе SVM-Multiclass. РОМИП-2006, Суздаль.
6. А. Максаков Оценка эффективности масштабируемых алгоритмов классификации текстов. РОМИП-2006, Суздаль.
7. Борисова П.В., Мышков П.С., Незлобин А.А., Петров А.Д. Классификация веб-страниц на основе алгоритма машинного обучения. Интернет-математика 2005, Москва
8. T. Joachims “A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization.” , In Proc. Of the ICML’97, 143-151, 1997.
9. T. Joachims “Making large-scale CVM learning practical”, Advances in Kernel Methods Support Vector Learning, MIT Press, 1999.
10. K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, Bernhard Scholkopf “An Introduction to Kernel-Based Learning Algorithms”, IEEE Transactions on Neural Networks, Vol. 12, NO 2, 2001.

CV recognition and grading automated system (MS Word document)

Kachaeva T. Yuzhikov V.

Article contains the description of CV recognition and grading automated system. Methods and algorithms of CV database forming based on automated analysis of incoming resumes were considered.