КЛАССИФИКАЦИЯ ВАКАНСИЙ С ЦЕЛЬЮ ПОСЛЕДУЮЩЕЙ ОПТИМИЗАЦИИ ПУБЛИКАЦИИ ОБЪЯВЛЕНИЙ

Быстрова М. В., Козадаев И. А.
Факультет радиофизики и компьютерных технологий Белорусского государственного университета
Минск, Республика Беларусь
E-mail: mvb1610@gmail.com, kozadaeff@mail.ru

В работе представлен и проанализирован подход к классификации данных, базирующийся на методе обработки текстов на естественном языке с помощь инструментов ИАТ. Для непосредственной классификации был выбран предварительно размеченный набор данный, содержащий текстовые описания вакансий. Была проанализирована устойчивость и точность данного подхода.

Введение

Согласно статистическим данным, заполнение большинства рабочих мест происходит с помощью публикаций объявлений о вакансиях. Успех такого набора зависит от того, как компания преуспела в составлении соответствующего объявления. Важно знать критерии, которые способны заинтересовать потенциального сотрудника, или же, другими словами, ценность предложения для работника.

В данной работе проблема классификации вакансий рассмотрена как задача интеллектуального анализа данных, для решения которой были предложены наиболее подходящие методы предварительной обработки текста публикаций, а также выбран оптимальный алгоритм определения близости публикаций в векторном пространстве [1].

I. Основные этапы построения классифицирующей модели

В ходе построения классифицирующей модели были выполнены три нижеприведённых этапа (см. рис. 1).

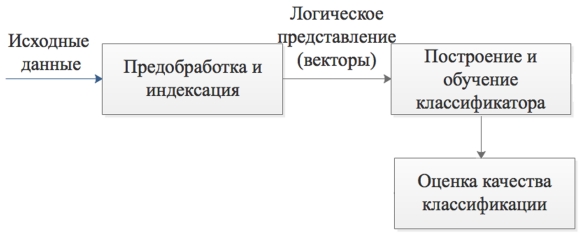

Рис. 1 – Этапы построения классифицирующей модели

II. Предварительная обработка и индексация

Опубликованные вакансии представляют собой небольшие текстовые документы различной длины. Для того, чтобы использовать эту информацию в качестве обучающей выборки, была произведена *токенизация*, то есть выделение в тексте слов, чисел, границ предложений и иных *токенов* (или *термов*), а также были удалены семантически нейтральные слова такие, как предлоги и союзы. После этого к полученной информации нами была добавлена служебная метаинформация, которая содержит обучающие метки, необходимые для работы классификатора.

Текст документа представляется в виде мультимножества термов [2] и обозначается $d_j \in D$, где D - множество всех документов, присутствующих в выборке. Множество всех термов $T = \{\tau_0, \tau_1, \ldots, \tau_{|T|}\}$. Каждому терму $\tau_i \in T$ ставится в соответствие некоторый вес w_{ij}, характеризующий встречаемость данного терма в тексте $d_j \in D$. Логическое представление принято обозначать вектором $\overline{d_j} = \{w_{0j}, w_{1j}, \ldots, w_{|T|j}\}$, где каждый w_{ij} - вес τ_i терма в документе $\overline{d_j}$.

В итоге было получено n-мерное пространство векторов, которое принято называть пространством признаком для класса данных D. Таким образом каждый документ является точкой в пространстве признаков.

$$IDF(\tau_i, D) = \left(\frac{|D|}{|(d_i \supset \tau_i)|}\right), 0 \leq i \leq |T|, \quad (1)$$

где $|D|$ - количество документов в классе, $|(d_i \subset \tau_i)|$ - количество документов, в которых встречается терм τ_i.

После этого были применены методы уменьшения размерности термов для обеспечения приемлемого времени работы алгоритма. В рамках этой работы были установлены следующие правила, при выполнении которых терм считается неинформативным:

1. Встречаемость терма в выборке меньше некоторого числа n;
2. Терм имеет большое математическое ожидание M_{f_i} и маленькую дисперсию D_{f_i}. Конкретные значения порогов задаются исходя из конкретных условий;
3. Имеет маленький информативный вес.

III. Построение и обучение классификатора

После формирования и предварительной обработки тренировочного набора документов следуют выбор и построение классифицирующей модели, архитектурными компонентами которой были выбраны методы Distributed Memory (распределенная память, DM) и Distributed Bag of Words (распределенный мешок слов, DBOW).

- DM прогнозирует слово по известным предшествующим словам и вектору абзаца;
- DBOW прогнозирует случайные группы слов в абзаце на основании вектора абзаца.

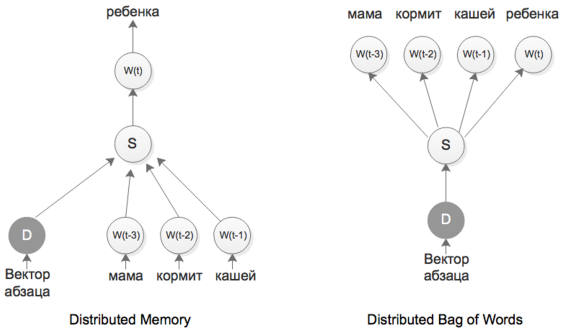

Рис. 2 – Схемы методов Distributed Memory и Distributed Bag of Words

Принцип работы этих методов заключается в нахождении связей между контекстами слов согласно предположению, что слова, находящиеся в похожих контекстах, являются семантически близкими. Формально задача представляет собой максимизацию косинусной близости между векторами слов, которые появляются в близких контекстах, и минимизация косинусной близости между векторами слов, которые не появляются рядом друг с другом: $min\{\frac{w_v \times w_c}{\sum w_{cl} \times w_v}\}$, где w – слова контекста, w_v – целевое слово, w_{cl} – другие контексты. $w_v \times w$ – близость слов контекста и целевого слова, $w_{cl} \times w_v$ – близость всех других контекстов и целевого слова.

Решить задачу такой минимизации (максимизации) можно с помощью обычной нейронной сети прямого распространения, требующей, чтобы входные векторы были фиксированной длины. Однако, если векторные представления предложений образовывать за счет склеивания соответствующих представлений слов, на выходе всегда будут получаться векторы разной длины.

В качестве решения этой проблемы был выбран свёрточный фильтр, идея которого заключается в том, что каждому нейрону подается на вход два (или более) слова, причем для каждого последующего нейрона вход сдвигается на одно слово. Например, первому нейрону на вход подается слово 1 и слово 2, второму – слово 2 и слово 3, и т.д. На выходе имеется предложение, которое в два или в N (количество нейронов входного слоя) раз короче исходного.

IV. Оценка качества работы классификатора и анализ результатов

В итоге была спроектирована и реализована система, которая позволяет классифицировать вакансии с использованием рассмотренных выше подходов. Для оценки качества работы модели была использована *F-мера*:

$$F(u) = \frac{2 * p(u) * r(u)}{p(u) + r(u)}, \qquad (2)$$

где $r(u) = \frac{|u \cap v|}{|v|}$ - *полнота (recall)* классификации по классу, то есть отношение количества документов, для которых классификатор правильно определил класс, к общему количеству документов класса, определенному без классификатора; $p(u) = \frac{|u \cap v|}{|u|}$ - *точность (precision)*, показывающая отношение количества документов, для которых классификатор правильно определил класс, к количеству документов, которые классификатор отнес к данному классу.

V. Заключение

Рассмотренные подходы были использованы для проектирования реальной системы классификации вакансий. Python использовался в качестве языка реализации. Методы DM и DBOW при сравнительно низкой вычислительной сложности позволили получить среднюю точность порядка 70%.

Несмотря на то, что нами были получены достаточно хорошие результаты, зачастую при использовании более сложных текстов на естественном языке современные технологии ИАТ не обладают высокими устойчивостью. Это связано с тем, что на текущий момент ИАТ не могут в точности определять семантику текста, а также анализировать сложные зависимости между разными его частями.

1. Быстрова, М. В. Классификация вакансий с целью последующей оптимизации публикации объявлений // Научное сообщество студентов XXI столетия: Технические науки: сб. ст. по мат. LXIV междунар. студ. науч.-практ. конф. 18 июня 2018. № 4(63)
2. Агеев, М. С. Методы автоматической рубрикации текстов, основанные на машинном обучении знаниях экспертов / М. С. Агеев // Либроком (Editorial URSS), 2004. – 106 с.