DOI: 10.36910/6775-2524-0560-2019-36-3
УДК 004.91:331.5
Єрошенко О. С., Степаніщева В. С., Єременко О. С.
Національнальний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

ВИЗНАЧЕННЯ ТРЕНДІВ НА РИНКУ ПРАЦІ. АНАЛІЗ ТА СТРУКТУРУВАННЯ ОТРИМАНОЇ ІНФОРМАЦІЇ

Єрошенко О. С. Степаніщева В. С. Єременко О. С. Визначення трендів на ринку праці. Аналіз та структурування отриманої інформації. У даній роботі розглянуто способи визначення трендів на ринку праці. Проведено огляд провідних веб сервісів з надання інформації про вимоги роботодавців до майбутніх працівників. Представлено алгоритми парсингу HTML коду сайтів вакансій. Запропоновані методи пост обробки та аналізу результатів парсингу. Наведені моделі зберігання результатів парсингу та критерії подальшого використання результатів в системі дистанційного навчання. Для парсингу та аналізу використовувались можливості мови Python.
Ключові слова: парсинг веб-сайтів, API, інтелектуальний аналіз текстових даних, класифікація ключових слів в текстах.

Ерошенко А. С., Степанищева В. С., Еременко Е. С. Определение трендов на рынке труда. Анализ и структурирования полученной информации
В данной работе рассмотрены способы определения трендов на рынке труда. Проведен обзор ведущих веб сервисов по предоставлению информации о требованиях работодателей к будущим работникам. Представлены алгоритмы парсинга HTML кода сайтов вакансий. Предложенные методы пост обработки и анализа результатов парсинга. Приведенные модели хранения результатов парсинга и критерии дальнейшего использования результатов в системе дистанционного обучения. Для парсинга и анализа использовались возможности языка Python.
Ключевые слова: парсинг сайтов, API, интеллектуальный анализ текстовых данных, классификация ключевых слов в текстах

Yeroshenko O., Stepanishcheva V. Yeremenko O. Determining trends in the labour market. analysis and structuring of obtained information. This research paper discusses ways to identify trends in the labour market. An overview of the leading web services to provide information on employers' requirements for future employees has been conducted. The algorithms for parsing HTML code for job sites are presented. Methods of post-processing and analysis of parsing results are offered. Models of storage of parsing results and criteria of further use of results in the distance learning system are presented. Python opportunities were used for parsing and analysis.
Keywords: website parsing, API, textual data mining, keyword classification in texts.

Постановка проблеми. Зараз, коли цифрові технології зазнають дуже швидкого та глобального розвитку, дуже важливо йти в ногу з часом. Це стосується не тільки розваг та буденних послуг, а й такої галузі як дистанційне навчання.

Чи потрібні сьогодні технікуми, університети, очні курси підвищення кваліфікації? В частині випадків ці заклади просто необхідні. Але ж є галузі, викладання яких дещо відстає від розвитку технологій. Адже інформаційні технології розвиваються настільки швидко, що не кожен викладач встигає ознайомитись зі всіма новинками, опанувати їх та викладати студентам. Наприклад, розглянемо викладання веб-дизайну. В рамках дисципліни треба розглянути багато технологій: дизайн (загальна теорія, користування графічними редакторами тощо), розмітка та стилізування (HTML, CSS та різноманітні препроцесори, на кшталт Jade) та фреймворки для різних мов програмування.

У дистанційному навчанні все набагато простіше, адже там на кожну технологію є багато спеціалістів, для яких простіше робити курс з однієї технології.

Отже, в дистанційному навчанні є сенс та користь. Але з'являється нова проблема. Яким чином людина може визначити, що вивчати?

Відповідь треба шукати серед тих варіантів, які популярні на ринку. Якщо людина не знає, що вивчати, треба звернути увагу на те, що актуально і затребувано серед роботодавців сьогодні і зараз.

Саме темі визначення трендів на ринку праці присвячене дане дослідження.

Мета статті. Головним питанням є визначення вимог серед роботодавців для працевлаштування за різними напрямками. Яким чином можна дізнатись навички, якими треба володіти? У цій справі також варто відійти від особистого спілкування з людьми: опитування представників IT компаній та експертів в певних предметних областях є неефективними. Але як тоді отримати відомості про потреби ринку без відкритих баз даних та однозначних переліків знань?

Коли постає питання про сервіс, в якому можна знайти вимоги до людей, які шукають роботу, відповість виникає сама по собі: сайти з вакансіями.

© Єрошенко О. С. Степаніщева В. С. Єременко О. С.

На меті стоїть отримання відомостей про навички, які необхідні для опанування професії. Також їх треба проградіювати за важливістю.

Отже, система складатиметься з парсера, який збиратиме інформацію про вакансії, та з аналізаторів, які будуть підраховувати усі необхідні для роботи критерії: приналежність слова до галузі, категорії та зв'язок між ключовими словами.

Парсер. Для початку необхідно зібрати описи вакансій. Щоб результат був максимально об'єктивним і таким, що відображає реальні потреби ринку, варто збирати інформацію з кількох сайтів: Head Hunter, Rabota.ua.

Такий перелік сайтів обумовлений тим, що Head Hunter - це дуже відомий міжнародний сайт для пошуку роботи, а Rabota.ua надзвичайно поширений сервіс в Україні.

Отже, розглянемо процес отримання даних з усіх перерахованих джерел.

Head Hunter. Парсинг інформації з даного сервісу є найлегшим, адже сайт надає безкоштовний API, отримати доступ до якого дуже легко: подача та розгляд заявки зайняли не більше тижня.

В рамках роботи нас цікавить лише один запит: https://api.hh.ru/vacancies. Для спрощення роботи доступні різні параметри фільтрації: пошуковий запит, область пошуку, досвід роботи, тип зайнятості, графік роботи, регіон, спеціалізація, індустрія компанії, розмір заробітної плати, тощо.

Під час парсингу вакансій нас цікавитимуть лише наступні аргументи:

- text — пошуковий запит. Наприклад: веб-дизайнер.
- experience — досвід. За допомогою цього поля є можливість обирати різні вакансії для користувачів з різним рівнем знань: початківців, досвідчених, професіоналів. Доступні наступні градації: немає досвіду, від 1 до 3 років, від 3 до 6 років, більше 6 років
- area — регіон. Таким чином, ми можемо визначати потреби на ринку в різних країнах або містах.
- specialization — код спеціалізації. Доступно дуже багато спеціалізацій для пошуку: адміністрування баз даних, тестування, програмування тощо.

Пропонується використовувати постійні та змінні комбінації параметрів.
Постійні: пошуковий запит, кількість вакансій на сторінці, номер сторінки та регіон, значення якого залежатиме від вказаного регіону користувачем в профілі системи.

Змінні ж залежатимуть від потреб пошуку. Загальними критеріями потреби внесення додаткових фільтрацій є:

- Потреба уточнення. Якщо запит користувача: «дизайнер» в сфері IT, то є сенс уточнити спеціалізацію та індустрію, в якій працює компанія.
- Недостатність результатів. Наприклад, людина з рівнем досвіду N хоче перейти на рівень N + 1 (з без досвіду - на 1-3 роки досвіду або з 1-3 роки на 3-5 років досвіду). Задаючи необхідну градацію за досвідом, система може отримати мало даних за запитом. В такій ситуації доцільно прибрати даний параметр, щоб отримати більше результатів.
- Потреба в локалізації. Обернений випадок до попереднього. Іноді система містить достатньо даних для того, щоб звузити пошук до певного рівня досвіду, регіону (наприклад, перейти з масштабів країни до міста або району).

В результаті роботи будь-якого успішного запиту ми отримаємо відповідь у вигляді об'єкта json, який має наступну структуру:

```
{
   "items": [
                  Об'єкти вакансій
   ],
   "found": Кількість вакансій за запитом,
   "pages": Кількість сторінок,
   "per_page": Кількість вакансій на сторінці,
   "page": Номер поточної сторінки,
   "clusters": Кластери,
   "arguments": Додаткові аргументи,
```

```
  "alternate_url": Альтернативний запит для пошуку
}
```

Об'єкт кожної вакансії має наступну структуру (наведені лише найважливіші аргументи, адже кожна вакансія містить дані, які в даній роботі не несуть інформаційної цінності):

```
{
        "id": ID вакансії,
        "name": Назва вакансії,
        "area": регіон,
        "salary": рівень заробітньої плати,
        "type": Тип: "відкрита"/"закрита",
        "employer": Інформація про компанію,
        "url": URL до повноцінної сторінки вакансії,
        "relations": Певні зв'язки,
        "snippet": {
                "requirement": Вимоги,
                "responsibility": Основні задачі
        },
        "contacts": Контакти,
        "specializations": [
                 {
        "profarea_id": ID галузі,
        "profarea_name": назва галузі,
        "id": ID спеціалізації,
        "name": спеціалізація
                },
                ...
        ],
}
```

З усіх наданих даних найбільше нас цікавить поле "snippet", в якому є властивість "requirement" - це і є опис вакансії, який буде використовуватись для аналізу.

Також, варто зберегти значення "specializations", адже це поле містить назву галузі, до якої відноситься дана вакансія.

Rabota.ua. Парсинг даного сервісу вже трохи ускладнюється, адже в нього немає API у відкритому доступі, тож доведеться отримувати інформацію більш складним способом.

Парсинг буде проводитись в декілька етапів:

1. Формування пошукового URL запиту
2. Визначення кількості сторінок з результатами пошуку
3. Діставання посилань на вакансії з кожної сторінки
4. Парсинг кожної вакансії

Формування пошукового запиту відбувається за наступною схемою: до базового URL "https://rabota.ua/zapros/" потрібно додати те, що необхідно знайти. Тобто якщо потрібні вакансії за позицією веб-дизайнера, потрібно додати назву професії до базового пошукового запиту, замінивши всі небуквенні символи на "-". У результаті отримаємо наступний запит: "https://rabota.ua/zapros/веб-дизайнер".

Визначити кількість сторінок з результатами пошуку можна двома способами.

Необхідно отримати HTML код першої сторінки, знайти в ньому блок, в якому зберігаються посилання на всі інші сторінки, та просто забрати текст з останнього посилання (Рис. 1).

Рисунок 1 - Спосіб визначення кількості сторінок

© Єрошенко О. С. Степаніщева В. С. Єременко О. С.

В другому способі потрібно отримати не першу сторінку результатів пошуку, а другу. Зробити це можна додавши до пошукового запиту "pg2". Після цього ми з коду сторінки і витягаємо елемент TITLE, в якому міститься номер другої сторінки та номер останньої сторінки: <title>Стор 2 з 13: Робота веб-дизайнер у Україні, пошук вакансій веб-дизайнер у Україні | Robota.ua</title> (Рис. 2).

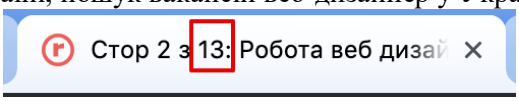

Рисунок 2 - Title сторінки з номером останньої сторінки результатів пошуку

Після того, як ми дізнались кількість сторінок видачі, ми можемо переходити до збирання посилань на вакансії. На кожній сторінці результатів пошуку забираємо посилання зі списку елементів з класом ga_listing.

Кінцевий етап: опис вакансій. Збираючи HTML код сторінки кожної вакансії, виділяємо блок з класом d_des, який і містить необхідну нам інформацію.

Аналіз описів вакансій. Процес аналізу розпочнемо з виділення ключових слів. Алгоритм роботи доволі простий: спочатку необхідно прибрати все зайве, що залишилось після парсингу, як от HTML теги <p>, <b>, <highlighttext> та знаки пунктуації [3].

На даному етапі розробки, система розрахована тільки на пошук вакансій в IT сфері, тож доцільним є рішення прибрати усі не латинські слова, адже більшість технологій та навичок мають англійські назви. Хоча, багато вакансій вже описуються англійською, звичайні слова не дуже впливатимуть на результати, бо в подальшому вони будуть видалятися з використанням відповідних алгоритмів.

Фінальним етапом парсингу буде визначення того, скільки разів зустрічається кожне слово.

Критерії. З метою якісного збереження результатів та використання їх у подальшому аналізі введемо три початкових критерії [2]:

1. частота повторювання у вакансії. Позначимо цей критерій V;
2. частота повторювання у загальній пошуковій видачі. Позначимо цей критерій S;
3. зв'язок ключового слова з іншими словами у вакансії. Позначимо цей критерій K.

Це пропонується зробити, по-перше, для того, щоб не було ситуацій, коли певна навичка в одній вакансії зустрічається стільки ж разів, скільки якась інша зустрічається в усіх результатах. По-друге, ця операція виступатиме в ролі нормування і вагування: отримані значення будуть вагами приналежності до галузі, категорії та зв'язку одне з одним [1].

Частота повторювання у вакансії визначатиметься наступною формулою (1):

$$V_{keyword} = n_{keyword} / N_i \quad (1),$$

де $n_{keyword}$ - це кількість повторювань ключового слова keyword в описі вакансії, N_i - загальна кількість ключових слів у вакансії.

Таким чином, сума вагів усіх ключових слів дорівнюватиме одиниці.

Обрахувавши суму усіх значень повторювань для кожного слова та поділивши на кількість вакансій (виходячи з того, що сума частот повторювання усіх ключових слів в межах однієї вакансії дорівнює одиниці, стає очивидним той факт, що сума частот повторювання усіх ключових слів дорівнює кількості вакансій помножених на одиницю), отримаємо частоту повторення ключового слова в результатах даного пошукового запиту (2) [4].

$$S_{keyword} = N_{keyword} / N_{vacancies} \quad (2)$$

Зв'язок ключового слова з іншими словами у вакансії обраховуватиметься як (формула 3):

$$K_{i\,j} = V_{keyword\,i} / V_{keyword\,j} \quad (3),$$

де $V_{keyword\,i}$ - це частота повторювання у вакансії слова i, а $V_{keyword\,j}$ - це частота повторювання слова j.

Метою обрахування цієї величини є не знаходження схожості. Ми намагаємось виразити величину, яка могла б характеризувати грубу і приблизну вірогідність $K_{i\,j}$ того, що якщо у вакансії є слово keyword 1, то і keyword 2 може тут бути.

В результаті, ми отримаємо ваги зв'язаності для ключових слів в межах окремих вакансій для даної галузі, яка визначається пошуковим запитом та значенням категорії з описів вакансій, що надає можливість мати найбільш точні дані для подальшого використання [6, 5].

Описані критерії в прикладному вигляді - у вигляді сутностей бази знань - представлені нижче.

На Рис. 3 ми бачимо зв'язок між галуззю, яку ми отримуємо з опису вакансії, та ключовими словами, які були знайдені в даній вакансії. Вагою зв'язку є критерій $V_{keyword}$.

© Єрошенко О. С. Степаніщева В. С. Єременко О. С.

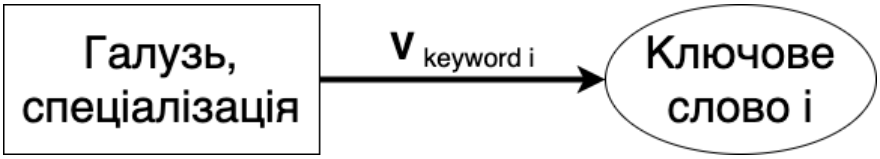

Рисунок 3 - Зв'язок галузі (спеціалізації) з ключовим словом і.

На Рис. 4 продемонстрований аналогічний зв'язок між пошуковим запитом та ключовими словами, які були отримані в результаті аналізу пошукової видачі. Вагою зв'язку є $S_{keyword}$.

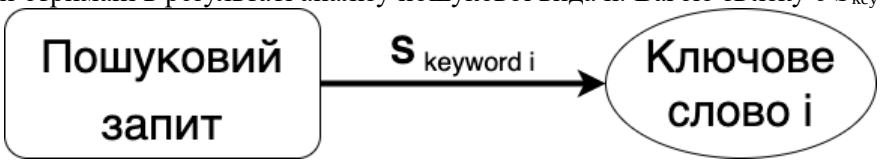

Рисунок 4 - Зв'язок тексту з пошукового запиту з ключовим словом і.

Останнім відношенням є зв'язок між ключовими словами, який характеризується вагою зі значенням $K_{i\,j}$ (Рис. 5).

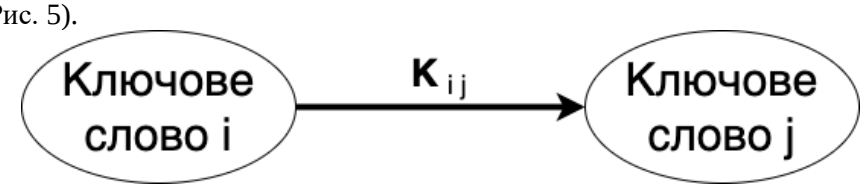

Рисунок 5 - Зв'язок ключових слів і та j.

Висновок. Як результат були створені парсери двох сайтів з вакансіями та аналізатори описів цих вакансій. За допомогою аналізаторів дані з вказаних сервісів структуруються та зв'язуються між собою. Утворюються зв'язки з критеріальними вагами між такими сутностями, як: ключове слово - галузь, ключове слово - запит/категорія та ключове слово - ключове слово. Ваги, що пов'язують сутності, обраховуються за максимально оптимальними виразами, що не ускладнюють структуру, але в той же час достатньо якісно описують моделі.

Перспективи подальших досліджень. В подальшому планується розширити функціонал, додавши парсинг нових сервісів, а саме: другого за популярністю сайта з вакансіями Work.ua та найпопулярнішого сайту в IT сфері України dou.ua. Аналізатори будуть покращені (як і структура моделей бази знань) з метою охоплення нового функціоналу. Наробки будуть використовуватись в таких сервісах, як: автоматизація створення профілю користувача, підготовка даних для тестування знань та створення інформаційно-пошукової системи в рамках системи дистанційного навчання.

References

1. Modelling and Detecting Changes in User Satisfaction. Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM'14) / J.Kiseleva, E. Crestan, R. Brigo, R. Dittel. – New York: ACM, 2014.

2. Multimedia Content Recommendation Engine with Automatic Inference of User Preferences. Proceedings of the IEEE International Conference on Image Processing / A. M. Ferman, P. Van Beek, J. H. Errico, M. I. Sezan – 2003.

3. An adaptive user profile for filtering news based on a user interest hierarchy [Електронний ресурс] / S. Singh, M. Shepherd, J. Duffy, C. Watters // ASIS&T Digital Library. – 2007. – Режим доступу до ресурсу: https://doi.org/10.1002/meet.1450430154.

4. Salton G. Automatic Information Organization and Retrieval / Gerard Salton. – New York: McGraw-Hill, 1968.

5. Ahn J. Personalized Search: Reconsidering the Value of Open User Models, Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI '15) / J. Ahn, P. Brusilovsky, S. Han. – New York: ACM, 2015.

6. Kim H. R. Learning Implicit User Interest Hierarchy for Context in Personalization IUI'03 / H. R. Kim, P. K. Chan. – Miami, 2003.

© Єрошенко О. С. Степаніщева В. С. Єременко О. С.