Софийски университет „Св.Климент Охридски”

Факултет по математика и информатика

Курсов проект на тема:

„Инструмент за извличане и съвпадение на обяви за работа от LinkedIn“

„Извличане на информация“

Зимен семестър 2024/25г.

Изготвен от: Ивана Дончевска фн.0MI3400614

Преподавател: **проф. Иван Койчев** Консултант: **Димитър Димитров**

Февруари 2025,
София

==========

Декларация за липса плагиатство:

- Плагиатство е да използваш, идеи, мнение или работа на друг, като претендираш, че са твои. Това е форма на преписване.
- Тази курсова работа е моя, като всички изречения, илюстрации и програми от други хора са изрично цитирани.
- Тази курсова работа или нейна версия не са представени в друг университет или друга учебна институция.
- Разбирам, че ако се установи плагиатство в работата ми ще получа оценка “Слаб”.

14.02.2025 г. Подпис на студента:

Ивана Д.

Съдържание

1. Увод

Процесът на търсене на работа често може да бъде времеотнемащ, неефективен и труден. Кандидатите прекарват часове в преглеждане на нерелевантни обяви, като съществува риск да пропуснат подходящи възможности. Затова целта на този проект е да автоматизира процеса, като позволява на търсещия работа да разглежда само обяви, които съответстват на неговите умения, вместо да се губи сред всички налични обяви.

2. Преглед на областта на обяви за работа в LinkedIn

LinkedIn е една от най-големите професионални мрежи в света, предоставяща платформа както за работодатели, така и за търсещи работа. Обявите за работа в LinkedIn съдържат информация за позицията, изискваните умения, местоположението, типа заетост и други детайли, които помагат на кандидатите да намерят подходящи възможности.

Основни характеристики на обявите за работа в LinkedIn:

- **Филтриране и препоръки** – Платформата предлага различни филтри (локация, индустрия, ниво на опит и др.), както и персонализирани препоръки въз основа на профила на потребителя.
- **Изисквания и умения** – Всяка обява съдържа списък с ключови умения и квалификации, които кандидатът трябва да притежава.
- **Автоматизирани процеси** – LinkedIn използва алгоритми за машинно обучение, за да съпоставя профилите на потребителите с подходящи обяви.
- **Конкуренция и динамика** – Търсенето на работа в LinkedIn е динамично, като популярните обяви често привличат голям брой кандидати за кратко време.

Тези особености правят LinkedIn ценен източник за обяви, но също така създават предизвикателства при намирането на най-релевантните предложения. Автоматизираният подход, разработен в този проект, цели да оптимизира този процес, като извлича, анализира и класифицира обявите според уменията на потребителя.

Съществуват различни методи и технологии, които могат да помогнат в автоматизирането на процеса. Един от основните подходи е **уеб скрейпинг**, който позволява извличането на обяви от уебсайтове като LinkedIn чрез автоматизирани скриптове. След като се съберат обявите, могат да бъдат приложени **методи за обработка на текст** (например с използване на NLP технологии), за да се извлекат ключови умения и изисквания от обявите и да се сравнят с уменията на кандидатите. Друг подход включва **системи за препоръки**, които използват алгоритми за машинно обучение, за да предоставят персонализирани предложения на потребителите въз основа на техните профили.

Съществуват няколко основни решения, които се използват за автоматизирано търсене на работа, като:

- **Вградените алгоритми за препоръки на LinkedIn**: Платформата използва данни за потребителите и техните професионални профили, за да предлага персонализирани обяви, които се очаква да са релевантни за потребителите.
- **Applicant Tracking Systems (ATS)**: Това са автоматизирани системи, които анализират автобиографии и документи на кандидати, съпоставяйки ги с изискванията на обяви за работа.
- **Приложения за скрейпинг и ботове**: Външни инструменти и ботове, които следят обявите за работа на различни платформи, събирайки нови предложения въз основа на зададени критерии.

3. Проектиране

За система/приложение: На кратко: Анализ на изискванията, Обща архитектура – напр. слоеве, модули, блокове, компоненти...; Модел на данните; Схема за представяне на знанията. Диаграми; Потребителски интерфейс (ако има); Ресурсни;…

Изискванията за проекта могат да бъдат разделени на функционални и нефункционални.

Функционални изисквания:

- **Събиране на данни:** Изисква се събиране на данни и формиране на корпус на база на извлечените информации от обявите за работа в LinkedIn чрез web scraping.
- **Обработка на текст:** След извличането на обявите, системата трябва да използва обработка на естествен език(NLP), за да извлече ключови умения и квалификации, изисквани за всяка обява.
- **Съпоставяне на умения и намиране на процент на съвпадение:** Системата трябва да сравнява уменията на потребителя с изискваните умения в обявите за работа чрез алгоритми за съвпадение и да показва резултати въз основа на процента на съвпадение.
- **Хранилище на данни**: системата трябва да има място, където да се съхраняват извлечените данни.

Нефункционални изисквания:

- **Използваемост:** Интерфейсът трябва да бъде интуитивен, лесен за използване и да предоставя визуална обратна връзка на потребителя.
- **Сигурност:** Защита на данните и системата от неоторизиран достъп и злоупотреби.
- **Ефективност:** Системата трябва да бъде ефективна и да осигурява бързи и точни резултати от търсенията.
- **Скалируемост**: Системата трябва да бъде скалируема, за да може да се справя с растящия обем на данните.
- **Поддръжка**: Възможност за поддръжка и обновяване на системата, включително актуализации на визуализацията (интерфейса), след като бъде внесена нова информация.

Архитектурата на системата се състои от няколко компонента:

Frontend (Потребителски интерфейс):

Интерфейсът съдържа три основни ендпойнта:

- **`/dashboard`** – Тук се показват визуализациите, като:
  - облак от най-често срещаните думи;
  - **pie chart** диаграма, показваща нивото (Junior, Mid, Senior и др.);
  - диаграма, илюстрираща кога са публикувани обявите (преди колко дни).
- **`/resume_analyzer`** – Тази функционалност позволява на потребителя да качи автобиография в **PDF** формат за анализ. Анализът извлича уменията на потребителя и ги визуализира:
  - уменията, които съвпадат с изискванията в обявите, се маркират в **синьо**;
  - липсващите умения се маркират в **червено**.
    След анализа се показва списък с най-релевантните обяви, сортирани по процент на съвпадение. Потребителят може да разгледа обявата или да отвори директно линка в **LinkedIn**, тъй като се съхраняват **ID-тата** на всяка обява.
- **`/resume_vs_job`** – Подобно на предишната функционалност, тук потребителят може да качи автобиография и да предостави текстово описание на конкретна обява. Системата изчислява процента на съвпадение между дадената обява и автобиографията.

Backend (Логиката на приложението):

Сървърната част включва:

- извличане на данни от **LinkedIn**;
- обработка на текста и анализ на обявите;
- извличане на умения от автобиографията на потребителя;
- намиране на процента на съвпадение между обявите и потребителя.

Първоначалните стъпки от създаването на функциите, тестването и експериментите върху данните главно са представени в четири Jupyter Notebook-а, както следва:

- **linkedIn_jobs_scraper.ipynb**
  В тази тетрадка са представени началните стъпки от проекта. Тук с помощта на библиотеки като Selenium, Requests и BeautifulSoup се извършва скрейпването на данните, като се търси по ключови думи и локация, събират се ID-тата на обявите в един .csv файл и след което всяка една обява се зарежда, извлича се детайлното ѝ описание и се започва изграждане на dataset-а. От него се извличат особености като локация, компания, позиция и други, които са ясни и лесни за извличане на база на скрейпнатия HTML.
- **jobs_analysis.ipynb**
  В тази тетрадка главно е представена обработката на естествен език и извличането на уменията от самия текст на обявите, тъй като за тях няма отделен HTML елемент, който директно да посочва уменията. За целта се използва разпознаване на обекти със SpaCy и моделът ‘en_core_web_lg’, като съответно се правят негови подобрения и се следи как той успява да разпознае уменията. Извършват се различни визуализации на най-често срещаните думи, диаграми на обявите, разделени по нива, и т.н. Тук също се смята и процентът на съвпадение между обявите и уменията на търсещия работа, като съответно се добавя нова колона в корпуса. Също така се прави оценка за това как различните функции се справят със смятането на съвпадението, която се измерва с precision, recall и F1-score.
- **pdf_reader.ipynb**
  Главната задача, която се извършва в тази тетрадка, е извличането на уменията от автобиографията на търсещия работа, като се прилагат три основни метода. Единият е с използване на основните техники за обработка на естествен език, а другите два са с използване на библиотеки за работа с .pdf файлове като PyPDF2.PdfReader и pdfMiner. Представено е и как всеки подход се справя със задачата.

- **Models_prediction.ipynb**
  В тази тетрадка са използвани два модела за обработка на естествен език:
  - BART (facebook/bart-large-mnli)
  - MiniLM (sentence-transformers/all-MiniLM-L6-v2)
    Представени са резултатите от това как всеки модел се справя.

Функциите от тетрадките впоследствие се интегрират в приложението, където резултатите се визуализират и се извършват различни анализи, както беше споменато по-горе. За самото приложение е използван Flask, със Bootstrap за стилизиране и подобряване на външния вид.

4. Реализация, тестване/експерименти

4.1. Използвани технологии, платформи и библиотеки

Проектът е реализиран с **Python** като основен език за програмиране. Уеб приложението е създадено с **Flask**, като за изграждането на интерфейса се използват **HTML темплейти**, **JavaScript** и **CSS** за персонализирани стилове и интерактивни елементи. Допълнително, **Bootstrap** е използван за стилизиране и по-добър потребителски опит.

За обработка и анализ на данни се използват **pandas** и **numpy**, които осигуряват ефективно манипулиране на таблични и числови данни. Обработката на естествен език (NLP) се извършва с **SpaCy**, който се използва за разпознаване на обекти, извличане на умения от обявите за работа.

За работа с **PDF файлове** и извличане на текст от тях са използвани **PyPDF2** и **pdfMiner**, което позволява анализ на автобиографиите на потребителите.

Визуализацията на данни се осъществява с помощта на **Matplotlib** и **Plotly**, които предоставят разнообразни графики за анализ на резултатите. Освен това, **displacy** се използва за графично представяне на зависимостите между думите, а **wordCloud** – за генериране на облаци от най-често срещаните думи.

4.2. Реализация

Освен оценките като **precision**, **recall** и **F-1 score**, направени са сравнения между моделите **BART** и **MiniLM** за обработка на текст, както и между различни методи за извличане на умения от автобиографиите. На всяка стъпка са извършвани сравнения и анализи на поведението на различните подходи.

5. Заключение

Проектът успешно реализира система за анализ на автобиографии и обяви за работа, използвайки обработка на естествен език и алгоритми за съвпадение на уменията. Уеб приложението предоставя визуализации и резултати, базирани на процент на съвпадение, и осигурява интуитивен интерфейс за потребителите.

Идеи за бъдещо развитие

- **Подобряване на изчисляването на процента на съвпадение** чрез интегриране на по-точни модели за оценка на уменията.
- **По-ефективно извличане на умения** с напреднали NLP техники и дообучаване на модели.
- **Автоматизиране на обновяването на данните** с регулярни актуализации на скрейпнатите обяви.

6. Използвана литература

[1] https://spacy.io/usage/rule-based-matching

[2] https://www.scrapingdog.com/blog/scrape-linkedin-jobs/

[3] https://selenium-python.readthedocs.io/

[4] https://huggingface.co/facebook/bart-large-mnli

[5] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2