Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

NLP: СТВОРЕННЯ ПАРСЕРА ДЛЯ СТРУКТУРУВАННЯ ТЕКСТУ ВАКАНСІЙ

Текстова частина до курсової роботи
за спеціальністю „Комп'ютерні Науки" 122

Керівник курсової роботи
Магістр, асистент
Смиш О.Р.

_____
(підпис)
"____" _____ 2022 р.

Виконала студентка
Золотаревич О.В.
"____" _____ 2022 р.

Київ 2022

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
проф., д.ф.-м.н.
_____ М. М. Глибовець
(підпис)
“____”_____2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студентці Золотаревич О.В. факультету інформатики 3-го курсу
ТЕМА Розробка парсера для структуризації даних з природної мови
Зміст ТЧ до курсової роботи:

Індивідуальне завдання
Вступ
1 Частотний аналіз
2 Розробка схеми алгоритму
3 Розробка програми
Висновки
Список літератури
Додатки

Дата видачі “___” _____ 2022 р. Керівник _____
(підпис)
Завдання отримала _____
(підпис)

Тема: Розробка парсера для структуризації даних з природної мови

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проєкту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	20.10.2021	
2.	Огляд технічної літератури за темою роботи.	08.11.2021	
3.	Опрацювання матеріалів	26.11.2021	
3.	Проведення аналізу	30.12.2021	
4.	Проєктування методів	20.01.2022	
5.	Розробка застосунку	10.05.2022	
6.	Тестування застосунку	17.05.2022	
7.	Написання пояснювальної роботи.	21.05.2022	
8.	Аналіз отриманих результатів з керівником, написання доповіді та попередній захист курсової роботи.	23.05.2022	
9.	Корегування роботи за результатами попереднього захисту.	03.06.2022	
10.	Остаточне оформлення пояснювальної роботи та слайдів.	05.06.2022	
11.	Захист курсової роботи		

Студентка Золотаревич О.В.

Керівник Смиш О.Р.

“____” _____ 2022 р.

ЗМІСТ

АНОТАЦІЯ

У роботі описано основні етапи розробки програми, яка здійснює структуризацію вхідних даних природною мовою в обраній галузі. Обґрунтовано методи та інструменти аналізу, на базі яких було створено та оптимізовано парсер, як результат цієї роботи.

Парсер на вхід отримує мало структуровану інформацію і перетворює її у структуровану.

ВСТУП

Обробка природної мови (надалі - NLP) є доволі актуальною темою на сьогодні. Наразі можемо спостерігати стрімке зростання досліджень у цій галузі, проте робіт, які специфікуються та працюють з українською мовою все ще недостатньо.

Важливо зазначити, що з погляду лінгвістики застосування для української мови чинних моделей обробки, наприклад для англійської мови, не є коректним, оскільки різняться мовні сім'ї, до яких ці мови належать, що ще раз підкреслює користь цієї роботи.

У цій роботі порушено проблему обробки текстів природною мовою Нині вакансії, написані природною мовою, не є структурованою інформацію, з цієї причини з'являється потреба у їхньому програмуванні. Для роботи з вакансіями було обрано галузь дизайну як одну з найбільш неструктурованих. Задля вичленення потрібної інформації необхідні певні методи та інструменти.

Метою є реалізація готового продукту – програми, яка згодом може бути імплементована у вебзастосунки. Науковий внесок до галузі NLP саме української мови.

Завданням цього дослідження є окреслення предметної області, здійснення лематизації вакансій природною мовою та виконання частотного аналізу. З урахуванням отриманих результатів диференціювати групи слів для подальшої роботи з ними – написання методів задля вичленення їх з тексту. Залучення технік та інструментів з комп'ютерної лінгвістики, філології, Computational social science.

У цій роботі оглянуто що таке галузь NLP, пропонується аналіз того, що доступно в цій сфері саме для української мови, наведені частотний та порівняльний аналізи задля розуміння подальшої архітектури парсера

детально описаний процес дизайну та розробки програми, а також продемонстровано фінальний продукт.

ОБРОБКА ПРИРОДНОЇ МОВИ

1.1 ГАЛУЗЬ ОБРОБКИ ПРИРОДНОЇ МОВИ

Люди мають багато способів комунікації, і текст лишається вагомою її частиною. Зі швидкістю розвитку та руху сучасного життя доступ до структурованої інформації стає не привілеєм, а необхідністю. Такі потреби сьогочасної людини дали поштовх для розвитку галузі обробки природної мови - NLP – Natural Language Processing. Метою NLP методів є перетворення мало структурованої інформації у структуровану шляхом обробки та вичленення необхідного.

У минулому сторіччі в галузі обробки природної мови покладалися на шаблони, тобто на їх збіги, згодом, стали використовувати рекурсивні розбори – спеціальні таблиці, схожі до синтаксичних аналізаторів, і нарешті, одним з останніх і досі еволюційним рішенням став парсер. Саме він і є одним з методів комп’ютерної лінгвістики.

Варто відзначити вагому роль галузі прикладної лінгвістики, а саме – комп’ютерної лінгвістики, яка працює з обробленням даних, представлених природною мовою. Однією з нагальних проблем комп’ютерної лінгвістики наразі є “усвідомлення тексту” – тобто, розуміння сенсу, що також є значущим питанням у розробці цього продукту.

Межа між інтерпретацією граматичних зв’язків у тексті та власне розумінням самого тексту є доволі довільною. Деякі з методів можна відносити до другої когорти, особливо ті, які своєю чергою дуже добре володіють роботою з контекстом, проте є методи, які змушують вагатися у їх розумінні тексту, наприклад – стемінг або токенізація, оскільки їх варто

відносити до першої когорти, тому що інтерпретація побудована винятково на з’ясуванні логічних зв’язків.

Мова – це ознака інтелекту та здатності мислення, якою наділені виключно люди, наданням комп’ютерів такої унікальної компетентності як розуміння сенсу природного мовлення неймовірно полегшило б взаємодію людини та технологій. Такі можливості комп’ютерних систем не тільки зможуть задовольнити наші денні потреби, але й нададуть змогу ефективніше працювати з великими обсягами інформації.

1.2 ОСНОВНІ МЕТОДИ NLP

Розглянемо основні методи NLP:

1. Токенізація

Процес розбиття рядка на символи, розділові знаки, цифри – токени. Токени це складові, які будують природну мову. Зазвичай робота з оригінальним текстом починається саме звідси. Отримуючи рядок, ми розбиваємо його на складові, де кожна складова – це токен. Яким би звичним методом токенізація не була, все одно вона має певні мінуси. Найважливішим є те, що цей метод не працює зі словами, які не містяться у словнику, їх ще називають “OOV”(Out Of Vocabulary) – поза словником. Також сюди відносять нові слова, з якими зіткнуться під час тестування.

2. Стемінг

Стемінг – це процес позбавлення слів закінчення, таким чином приводячи їх до інфінітивної форми. Цей метод не враховує контекст або словники, а просто видаляє закінчення, натомість часто приводячи до помилок, неправильної форми або написання слова.

3. Лематизація

Цей метод враховує контекст та приводить слова до початкової форми. Внаслідок врахування контексту цей метод є ефективнішим за більшість інших(наприклад стемінг) та призводить до меншої кількості помилок. Приклад: Світлина 1.

4. Теґінг

Також “розмічування частин мови”. Цей процес відповідальний за встановлення позначки для кожного слова в тексті, що вказує до якої частини мови воно відноситься, залежно від визначення та контексту.

1.3 РЕСУРСИ ДОСТУПНІ ДЛЯ УКРАЇНСЬКОЇ МОВИ

Візьмемо до уваги UDPipe – засіб для токенізації, тегування, лематизації та парсингу залежностей. На зображенні(Світлина 2) наведено навчені моделі для різних мов. Більшість моделей сфокусовані на англійську, італійську та французьку мову, але маємо лише одну модель, що зосереджена саме на українській мові.

```
# text = У зв'язку з розширенням мережі та збільшенням об'ємів робіт, запрошуємо в команду відділу маркетингу та реклами
1	У	у	ADP	Spsl	Case=Loc	2	case	_	_
2	зв'язку	зв’язок	NOUN	Ncmsln	Animacy=Inan|Case=Loc|Gender=Masc|Number=Sing	11	obl	_	_
3	з	з	ADP	Spsi	Case=Ins	4	case	_	_
4	розширенням	розширення	NOUN	Ncnsin	Animacy=Inan|Case=Ins|Gender=Neut|Number=Sing	2	nmod	_	_
5	мережі	мережа	NOUN	Ncfsgn	Animacy=Inan|Case=Gen|Gender=Fem|Number=Sing	4	nmod	_	_
6	та	та	CCONJ	Ccs	_	7	cc	_	_
7	збільшенням	збільшення	NOUN	Ncnsin	Animacy=Inan|Case=Ins|Gender=Neut|Number=Sing	4	conj	_	_
8	об'ємів	об'єм	NOUN	Ncmpgn	Animacy=Inan|Case=Gen|Gender=Masc|Number=Plur	7	nmod	_	_
9	робіт	робота	NOUN	Ncfpgn	Animacy=Inan|Case=Gen|Gender=Fem|Number=Plur	8	nmod	_	SpaceAfter=No
10	,	,	PUNCT	U	_	2	punct	_	_
11	запрошуємо	запрошувати	VERB	Vmpip1p	Aspect=Imp|Mood=Ind|Number=Plur|Person=1|Tense=Pres|VerbForm=Fin	0	root
12	в	в	ADP	Spsa	Case=Acc	13	case	_	_
13	команду	команда	NOUN	Ncfsan	Animacy=Inan|Case=Acc|Gender=Fem|Number=Sing	11	obl	_	_
14	відділу	відділ	NOUN	Ncmsgn	Animacy=Inan|Case=Gen|Gender=Masc|Number=Sing	13	nmod	_	_
15	маркетингу	маркетинг	NOUN	Ncmsgn	Animacy=Inan|Case=Gen|Gender=Masc|Number=Sing	14	nmod	_	_
16	та	та	CCONJ	Ccs	_	17	cc	_	_
17	реклами	реклама	NOUN	Ncfsgn	Animacy=Inan|Case=Gen|Gender=Fem|Number=Sing	15	conj	_	SpacesAfter=\
```

Світлина 1

UDPipe

dutch-lassysmall-ud-2.6-200830
english-ewt-ud-2.6-200830
english-gum-ud-2.6-200830
english-lines-ud-2.6-200830
english-partut-ud-2.6-200830
estonian-edt-ud-2.6-200830
estonian-ewt-ud-2.6-200830
finnish-tdt-ud-2.6-200830
finnish-ftb-ud-2.6-200830
french-gsd-ud-2.6-200830
french-sequoia-ud-2.6-200830
french-partut-ud-2.6-200830
french-spoken-ud-2.6-200830
galician-ctg-ud-2.6-200830
galician-treegal-ud-2.6-200830
german-hdt-ud-2.6-200830
german-gsd-ud-2.6-200830
gothic-proiel-ud-2.6-200830
greek-gdt-ud-2.6-200830
hebrew-htb-ud-2.6-200830
hindi-hdtb-ud-2.6-200830
hungarian-szeged-ud-2.6-200830
indonesian-gsd-ud-2.6-200830
irish-idt-ud-2.6-200830
italian-isdt-ud-2.6-200830
italian-partut-ud-2.6-200830
italian-postwita-ud-2.6-200830
italian-twittiro-ud-2.6-200830
italian-vit-ud-2.6-200830
japanese-gsd-ud-2.6-200830
korean-kaist-ud-2.6-200830
korean-gsd-ud-2.6-200830
latin-ittb-ud-2.6-200830
latin-llct-ud-2.6-200830
latin-proiel-ud-2.6-200830
latin-perseus-ud-2.6-200830
latvian-lvtb-ud-2.6-200830
lithuanian-alksnis-ud-2.6-200830
lithuanian-hse-ud-2.6-200830
maltese-mudt-ud-2.6-200830
marathi-ufal-ud-2.6-200830
naija-nsc-ud-2.6-200830
north_sami-giella-ud-2.6-200830
norwegian-bokmaal-ud-2.6-200830
norwegian-nynorsk-ud-2.6-200830
norwegian-nynorsklia-ud-2.6-200830
old_church_slavonic-proiel-ud-2.6-200830
old_french-srcmf-ud-2.6-200830
old_russian-torot-ud-2.6-200830
old_russian-rnc-ud-2.6-200830
persian-seraji-ud-2.6-200830
polish-pdb-ud-2.6-200830
polish-lfg-ud-2.6-200830
portuguese-gsd-ud-2.6-200830
portuguese-bosque-ud-2.6-200830
romanian-rrt-ud-2.6-200830
romanian-nonstandard-ud-2.6-200830
russian-syntagrus-ud-2.6-200830
russian-gsd-ud-2.6-200830
russian-taiga-ud-2.6-200830
afrikaans-afribooms-ud-2.6-200830
ancient_greek-perseus-ud-2.6-200830
ancient_greek-proiel-ud-2.6-200830
arabic-padt-ud-2.6-200830
armenian-armtdp-ud-2.6-200830
basque-bdt-ud-2.6-200830
belarusian-hse-ud-2.6-200830
bulgarian-btb-ud-2.6-200830
catalan-ancora-ud-2.6-200830
chinese-gsdsimp-ud-2.6-200830
chinese-gsd-ud-2.6-200830
classical_chinese-kyoto-ud-2.6-200830
coptic-scriptorium-ud-2.6-200830
croatian-set-ud-2.6-200830
czech-pdt-ud-2.6-200830
czech-cac-ud-2.6-200830
czech-fictree-ud-2.6-200830
czech-cltt-ud-2.6-200830
danish-ddt-ud-2.6-200830
dutch-alpino-ud-2.6-200830
sanskrit-vedic-ud-2.6-200830
scottish_gaelic-arcosg-ud-2.6-200830
serbian-set-ud-2.6-200830
slovak-snk-ud-2.6-200830
slovenian-ssj-ud-2.6-200830
slovenian-sst-ud-2.6-200830
spanish-ancora-ud-2.6-200830
spanish-gsd-ud-2.6-200830
swedish-talbanken-ud-2.6-200830
swedish-lines-ud-2.6-200830
tamil-ttb-ud-2.6-200830
telugu-mtg-ud-2.6-200830
turkish-imst-ud-2.6-200830
ukrainian-iu-ud-2.6-200830
urdu-udtb-ud-2.6-200830
uyghur-udt-ud-2.6-200830
vietnamese-vtb-ud-2.6-200830
welsh-ccg-ud-2.6-200830
wolof-wtb-ud-2.6-200830

Світлина 2

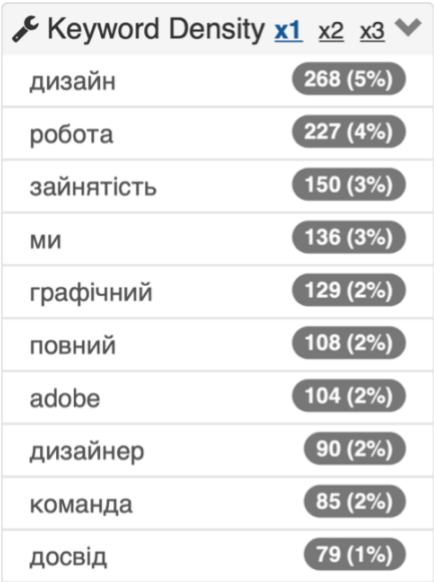
АНАЛІЗ ДЛЯ АРХІТЕКТУРИ ПАРСЕРА

Перед тим як розпочати порівняльний та частотний аналізи варто здійснити лематизацію задля отримання коректних результатів. На Світлині 1 зображена лематизація українського тексту вакансії. Як зазначено раніше, цей процес полягає в розбитті цілого рядка (string) на слова в їх інфінітивній формі, і що важливо, цей метод є чутливим до контексту.

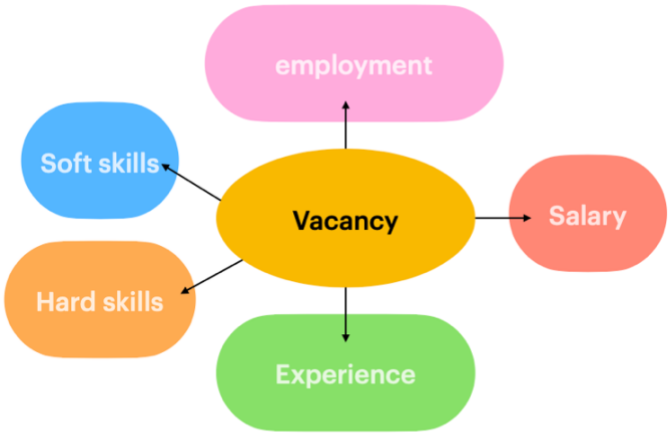
Цей крок є важливим, адже допомагає отримати коректні результати. У вакансії використовуються різні форми одного й того ж слова, наприклад “дизайн”, “дизайну”, “дизайном” – усі ці слова різні для нашого алгоритму, і тільки людина розуміє, що це вони ідентичні за значенням. Задля того, щоб уникнути появу дублів та не дати цим словам формувати свої окремі групи, що своєю чергою призведе до некоректних результатів, використання методу лематизації є беззаперечним у цьому процесі.

2.1 ЧАСТОТНИЙ АНАЛІЗ

Задля проведення частотного аналізу використано 50 різних текстів вакансій, написаних українською природною мовою. Усі ці тексти було пролематизовано та перевірено на частотність появи різних слів. Найпопулярніші слова формували до ключових груп, які можна побачити на Світлині 3. На цих ключових групах буде згодом зосереджено більшість методів у парсері.



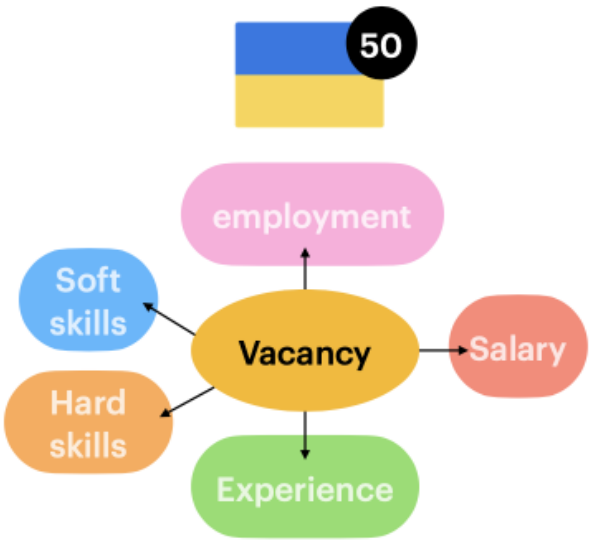
Keyword Density x1 x2 x3	
дизайн	268 (5%)
робота	227 (4%)
зайнятість	150 (3%)
ми	136 (3%)
графічний	129 (2%)
повний	108 (2%)
adobe	104 (2%)
дизайнер	90 (2%)
команда	85 (2%)
досвід	79 (1%)



Світлина 3

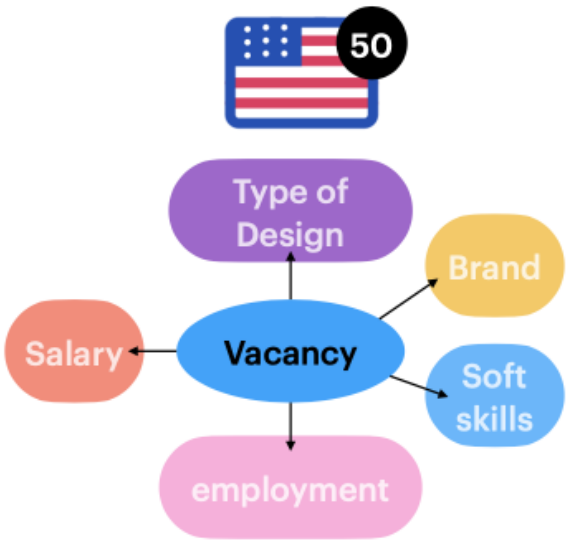
2.2 ПОРІВНЯЛЬНИЙ АНАЛІЗ

Задля проведення порівняльного аналізу було використано 50 різних текстів вакансій написаних двома мовами – природною українською та природною англійською. Для англійської мови була використана ідентична техніка – попередня лематизація та частотний аналіз. Результати можна побачити на Світлині 4. Варто зазначити, що деякі групи, сформовані ключовими словами в обох мовах, збігаються: “employment”, “salary”, “soft skills”, але інші відрізняються.



Keyword Density x1 x2 x3

дизайн	268 (5%)
робота	227 (4%)
зайнятість	150 (3%)
ми	136 (3%)
графічний	129 (2%)
повний	108 (2%)
adobe	104 (2%)
дизайнер	90 (2%)
команда	85 (2%)
досвід	79 (1%)



Keyword Density x1 x2 x3

design	366 (4%)
graphic	227 (3%)
job	205 (2%)
designer	168 (2%)
work	166 (2%)
creative	143 (2%)
full	140 (2%)
time	136 (2%)
digital	125 (2%)
brand	110 (1%)

Світлина 4

ДИЗАЙН ТА РОЗРОБКА

Процес дизайну та розробки починається з визначення основних методів. Використовуючи частотний аналіз було отримано певні групи, на яких буде орієнтовано парсер, а саме: локація, інструменти, досвід, заробітна плата, зайнятість та гнучкі навички.

Ідея роботи продукту полягає в отриманні текстового файлу з текстом вакансії, написаного природною українською мовою, надалі цей текст обробляється, інформація структурується та вичленяється інформація, яка потрібна, вона ж і повертається до користувача як результат.

3.1 РОБОТА З ВХІДНИМ ФАЙЛОМ

Робота з вхідним файлом є критичним кроком в розробці парсера. Саме цей етап є одним з найважливіших, що забезпечує коректне відпрацювання програми.

Перш за все, на вхід отримується сирий текст вакансії, написаний українською природною мовою.

Наступним кроком є теґування та лематизація з використанням UDPipe (Лістинг 1).

```
os.system(
"curl -F model=ukrainian-iu-ud-2.6-200830 -F data='@" +
vacancy_file + "' -F tokenizer= -F tagger= -F parser=
http://lindat.mff.cuni.cz/services/udpipe/api/process > " +
"uv_res_nlp.txt")
```

Лістинг 1

Після його відпрацювання отримано слова в їх інфінітивній формі та розбиті рядки на частини мови.

Третім кроком є очищення файлу. Файл позбавлено даних, які не є важливими і немає потреби в їх подальшому опрацюванні (Лістинг 2).

```
 flines = file.readlines()
   for_deletion = ['PUNCT',
                  'SYM',
                  'CCONJ',
                  'ADP',
                  'DET',
                   'AUX',
                   'NUM', '#', 'model', 'ukrainian-iu-ud-2.6-200830',
'acknowledgements', '{', '}', '[', ']']
   for line in flines:
      if not any(f_words in line for f_words in for_deletion):
         file_r2.write(line)
```

Лістинг 2

Фінальним кроком є формування списку з інфінітивних форм слів, які містяться у вакансії. Такий список формується кожного разу, коли обробляється текст вакансії (Лістинг 3).

```
list_words = []
list_lines = []

# розбивається вхідний текст на списки рядків
for line in file:
    strip_lines = line.strip()
    list_line_x = strip_lines.split()
    #print(list_line_x)
```

```
    if list_line_x != []:
        # тепер list_lines - тепер список списків НЕ містить пусті списки
        m = list_lines.append(list_line_x)
        #print(list_line_x)
        #  створено список, який складається з кожного слова вакансії
        for word in list_line_x:
            list_words += word.split(" ")
print(list_words)
```

Лістинг 3

4.2 РОЗРОБКА МЕТОДІВ

Для деяких методів використовуватимуться однакові техніки задля структуризації та вичленення потрібної інформації.

Метод з визначення досвіду та зайнятості.

Перш за все, обробляється вхідний текст вакансії, процес якого детально розглянутий вище.

Наступним кроком буде робота зі списком усіх слів вакансії, який сформовано як результат роботи з вхідним файлом.

Процес полягає у тому, що отримується список усіх слів вакансії та перевіряється на збіг ключового слова. Наприклад для методу досвід ключовим словом буде: “досвід”. Коли знайдено збіг між ключовим словом та словами вакансії – повертаємо це як результат відпрацювання методу. Приклад методу Досвіду – Лістинг 4.

```
x = "досвід"
y = "Досвід"
```

```
print("🔎 ДОСВІД:")
for line in file_n:
    if (x in line or y in line):
        print(line)
```

Лістинг 4

Метод з визначення Локації.

Робота починається з отримання списку слів вакансії в інфінітивній формі. Проте, для реалізації методу Локації цього недостатньо. Додатково створено список з усіма містами України.

Надалі отримується на вхід два списки – список слів вакансії та список усіх міст України, відбувається пошук збігу – цей збіг і буде результатом відпрацювання програми (Лістинг 5).

```
a_set = set(l1)
b_set = set(l2)
# перевірка на збіг
if (a_set & b_set):
    city_match = str(a_set & b_set)
    # позбуваємося ', {}  в str
    city_clear = [char for char in city_match if char.isalnum()]
    city = "".join(city_clear)
    # результат
    print("📍ЛОКАЦІЯ:", city)
else:
    print("📍ЛОКАЦІЯ: Локацію не вказано")
```

Лістинг 5

Метод з визначення Інструментів та Гнучких навичок.

Отримується список слів вакансій в інфінітивній формі. Надалі сформовано словник для кожного з методів. Оскільки гнучкі навички

можуть бути різними, але означати те саме, обрано саме словник для подальшої роботи. Результатом відпрацювання програми буде збіг серед списку слів вакансії та значеннями у словнику. Ключ, який відповідатиме значенню, що Збіглося, буде повернуто як результат відпрацювання методу (Лістинг 6).

```
def check_soft_match(l1):
   # словник значень
   dict_soft = {
      'Командний гравець': ['команда', 'робота в команда'],
      'Комунікативний': ['спілкування', 'комунікація', 'порозуміння', 'обговорення',
'відкритість', 'комунікабельність'],
      'Креативний': ['креативний', 'креативність', 'фантазія', 'творчий'],
      'Пунктуальний, відповідальний': ['тайм менеджмент', 'пунктуальний',
'відповідальність', 'відповідальний', 'самостійність', 'самостійний'],
      'Вмотивований': ['мотивація', 'самомотивація'],
      'Наявність смаку': ['смак'],
      'Стресостійкість': ['стресостійкість']
 }
   result_soft = [k for k, v in dict_soft.items() if any(x in v for x in l1)]
   print("🤔 ГНУЧКІ НАВИЧКИ:", result_soft)
```

Лістинг 6

Метод з визначення заробітної плати.

Задля отримання більш точних результатів метод було диверсифіковано. Надалі буде продемонстровано підходи, які застосовані у цьому методі.

На вхід отримується файл вакансії, який потім буде розбито на окремі рядки, а в деяких випадках – і на окремі символи.

Пошук цифрових значень.

Додатково створено список валют. Пошук цифрових значень починається з нумерації символів рядка – поточний, попередній та наступний. Таким чином здійснюється якісний перебір. (Лістинг 7).

Кожен символ перевіряється на таке:

1) Чи є поточний символ цифрою
2) Якщо так, то чи є наступним символом валюта, або позначення тисяч “k”.
3) Якщо так – ці символи(поточний та наступний) виводяться як результат.
4) Додатково перевіряється чи не є поточний символ нулями – у разі, якщо позначення тисяч написано окремо. Якщо так, і наступним символом є валюта або “k”, то в результат виводиться додатково попереднє число.

```
for index, item in enumerate(list_words):
    # пошук поточного, попереднього і наступного елементів в рядку
    if (index + 1 < len(list_words) and index - 1 >= 0):
        # попередній
        prev_item = str(list_words[index - 1])
        # поточний
        curr_item = str(item)
        # наступний
        next_item = str(list_words[index + 1])
        # перевірка на тип
        if curr_item.isdigit():
            # перевірка на наступний знак (чи є валютою або "k")
            if (next_item in val_list_cur or next_item == "k"):
                # якщо тисячі зазначені окремо - виводиться попереднє число
                if curr_item == "000":
                    print("💰 ЗАРОБІТНА ПЛАТА:", prev_item, curr_item, next_item)
```

```
        # якщо ні виводиться результат
        else:
            print("💰 ЗАРОБІТНА ПЛАТА:", curr_item, next_item)
        # збільшення лічильника
        check += 1
```

Лістинг 7

Пошук спеціальних сценаріїв.

У методі заробітної плати розглянуто 3 спеціальні сценарії:

1) Міститься “виделка” заробітної плати
2) Позначення тисяч написано разом з сумою заробітної плати
3) Заробітна плата договірна або схожі випадки.

(1) Задля пошуку “виделки” перевіряється вхідний символ - чи наявні в ньому цифри, чи міститься дефіс, та закінчення – чи наявні сотні (Лістинг 8). Поява в кінці “k” передбачена вже іншими методами.

```
for item in list_words:
  if cont_num(item) == True and '-' in item and item.endswith('00'):
    print("💰 ЗАРОБІТНА ПЛАТА:", item)
    # збільшення лічильника
    check += 1
```

Лістинг 8

(2) Задля пошуку “k” перевіряється вхідний символ - чи наявні в ньому цифри та закінчення на “k” (Лістинг 9).

```
for item in list_words:
  if cont_num(item) == True and item.endswith('k'):
    print("💰 ЗАРОБІТНА ПЛАТА:", item)
```

```
    # збільшення лічильника
    check += 1
```

Лістинг 9

(3) Спеціальні сценарії занесено до словника, оскільки написання “ за результатами співбесіди” може різнитися у різних вакансіях. Кожен рядок вакансії перевіряється на наявність значень зі словника, якщо знайдено збіг результатом повертається ключ словника (Лістинг 10).

```
 # словник значень для заробітної плати
dict_salary = {
   'за результатами співбесіди': [['за', 'результатами', 'співбесіди'], ['по', 'результатам',
'співбесіди']],
   'договірна' : [['заробітня', 'плата',  'договірна'], ['заробітна', 'плата',  'договірна']]
}

# перевірка чи не міститься у вакансії values зі словника значень заробітної плати
result_salary = [k for k, v in dict_salary.items() if any(x in v for x in vacancy_lines)]
if (result_salary != []):
   print("💰ЗАРОБІТНА ПЛАТА:", result_salary)
```

Лістинг 10

Пошук валюти.

Задля роботи з валютою у вакансії створено валютний словник – де значення словника – різні варіанти написання валют, а ключі – їхні відповідники.

Перевірка відбувається наступним чином:

Перевіряємо кожен символ вакансії на збіг зі значеннями валютного словника, у разі збігу – виводиться його ключ як результат (Лістинг 11).

```
 for item in dict_cur.items() and list_words:
    # якщо item збігається з values в словнику - виводиться key
    res_cur = [k for k, v in dict_cur.items() if any(x in v for x in list_words)]
    # якщо ні - то валюта не зазначена
    if (res_cur == []):
        res_cur = "не зазначено"
print("          Валюта:", res_cur)
```

Лістинг 11

Лічильник.

Додатково в методі створено лічильник checker, який рівний 0, з метою ідентифікації результатів пошуку заробітної плати. При позитивному відпрацюванні різних алгоритмів пошуку лічильнику додаються значення, це продемонстровано у лістингах вище. У разі безрезультатного пошуку усіма алгоритмами лічильник лишається з початковим значенням, що свідчить про те, що інформації про заробітну плату не міститься, що і виводиться користувачу.

3.3 ТЕСТУВАННЯ

На Світлині 5 продемонстрована робота створених методів. Ліворуч зображено повний текст вакансії написаний природною українською мовою, праворуч – структуровану та вичленену інформацію, отриману з цієї вакансії як результат роботи програми.

25.11.2021 10:49 | Київ
Middle designer
EVO
Команда маркетингу проєкту Prom.ua групи компаній EVO знаходиться у пошуку Middle designer.

Prom.ua — найбільший український маркетплейс, де продається більше за 100 млн товарів від 60 тис. продавців.

Разом з Prom.ua:

кожен підприємець може створити інтернет-магазин і почати продавати в інтернеті по всій країні;

кожен покупець може знайти все, що потрібно, по найкращій ціні: від зубної щітки до комбайну.

Наша команда:

Ти станеш частиною команди MRD (Merchant Research & Development) найбільшого маркетплейсу країни. Твоїми колегами будуть комунікаціонщики і піарщики, продуктові і трейд-маркетологи, які інформують продавців, навчають продукту й інтернет-продажам, описують і створюють цінність продуктів для продавців, запускають акції.
Наша спільна задача: щоб інтернет – підприємцям було зручно продавати через Prom.

Наш кандидат:

На "ти" з інструментами для графічного дизайну [Photoshop, Illustrator, Sketch/Figma];

Має досвід роботи з бренд-гайдами;

Знає все о композиції і сітках, типографіці, кольорознавстві;

Готовий презентувати свої роботи, ідеї й аргументувати їх цінність і важливість для проєкту;

Знайомий з тайм-менеджментом і вміє фокусуватися на поточних задачах;

Націлений на результат [важливо не просто закрити задачу, а зрозуміти і проаналізувати, яку проблему вирішує дизайн];

Відкритий до спілкування і доброзичливий [комунікації буде багато, так як ми команда, яка вирішує загальні задачі проєкту];

Знайомий з основами копірайтингу;

Має аналітичне і креативне мислення;

Малює від руки [це не ключове, але буде перевагою];

Відчуває, що робота в команді це його формат.

Буде плюсом:

Досвід роботи з розробниками

Досвід роботи з символами і стилями

Досвід роботи з mobile apps і web інтерфейсами

Ділимося нашим командним портфоліо EVO: https://www.behance.net/evocompany. У відповідь чекаємо приклади ваших робіт ;)

Що потрібно буде робити:

В пріоритеті:

Розробляти лендінги

Підтримувати рекламні компанії Prom.ua для продавців [на зовнішніх ресурсах]

Розробляти концепти розпродажів [банери і ресайзи, кастоми, e-mail – розсилки]

Брати участь в брейнштормах, презентувати особисті роботи, повноцінно вести особисті проєкти і комунікацію із замовниками

Розвиватись у бік продуктового дизайну:

Брати участь у розробці інтерфейсу для мобільного додатку [Android / iOS] і особистого кабінету продавця [web / tablet / mobile]

Брати участь в створенні і розвитку дизайну системи

І таке буває:

Розробляти дизайн поліграфії

Розробляти фірмовий стиль

Розробляти анімації, ілюстрації [колаж, векторна ілюстрація, іконки], інфографіка для pr-задач

Що ми можемо запропонувати:

Допомогу і підтримку колеги/наставника, обмін досвідом з усією командою дизайнерів EVO;

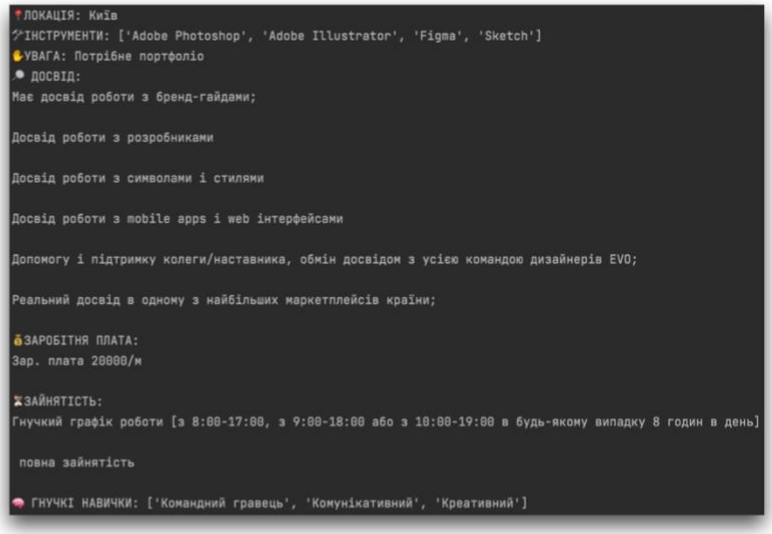
Реальний досвід в одному з найбільших маркетплейсів країни;

Розвиток навичок і професійне зростання;

Гнучкий графік роботи [з 8:00-17:00, з 9:00-18:00 або з 10:00-19:00 в будь-якому випадку 8 годин в день] – зараз працюємо віддалено. Наш маніфест віддаленої роботи в EVO

Як у нас проходить відбір на вакансію: відбір по резюме з портфоліо (це важлива умова), після цього коротке телефонне інтерв'ю з рекрутером [15-20 хвилин]. Наступний етап тестове завдання. Після запрошення на онлайн співбесіду з представниками команди, можливі дві зустрічі [обидві по годині].

Графік роботи:
повна зайнятість



Світлина 5

ФІНАЛЬНИЙ ПРОДУКТ

4.1 ДЕМОНСТРАЦІЯ РОБОТИ ФІНАЛЬНОГО ПРОДУКТУ

Текст вакансії природною українською мовою :

```
26.11.2021 03:01 | Київ
Graphic Designer
Genesis
 Привіт!

Ми, Lift: Story Maker – амбітний та швидкозростаючий проєкт у категорії Photo&Video. Наш однойменний
додаток дозволяє тисячам людей створювати креативний та естетичний контент за лічені секунди. Ми
регулярно входимо у ТОП-5 найкращих Graphic Design додатків у більше ніж 100 країнах світу і ти
матимеш змогу впливати на те, яким наш продукт бачитимуть мільйони людей.

Зараз ми шукаємо Graphic Designer, який/яка не уявляє свого життя без щоденного креативу та готовий/
ва пропонувати нові рішення для реклами нашого додатку. Тебе чекають нестандартні та круті задачі!

Твоїми завданнями будуть:

створювати дизайн рекламних банерів для Facebook, Instagram, Google та інших соціальних мереж;

брати участь у створення шаблонів для нашого додатку;

активно генерувати нові креативні підходи та ідеї для реклами у співпраці з командою маркетингу.

Які знання і навички тобі потрібні на цій посаді:

досвід роботи на аналогічній позиції;

відмінне володіння Adobe Photoshop, Adobe Illustrator, Figma;

розуміння законів композиції, типографіки та кольору;

креативне і нестандартне мислення;

уважність до деталей, високий рівень самоорганізації та тайм-менеджменту;

любов до реклами, знання трендів соціальних мереж.

Буде перевагою:

базові навички створення анімації.

Genesis – це унікальне місце для роботи, розвитку і зростання:

експертиза в розвитку високотехнологічних продуктів на міжнародному ринку. Робота з найкращими
професіоналами в Україні;
чудові можливості для навчання: внутрішні ком'юніті frontend та backend розробників, семінари, доступ
до корисної літератури, курси англійської та участь у ключових заходах IT-індустрії по всьому світу;
умови для роботи: відмінний офіс в 5 хвилинах від станції метро Тараса Шевченка, безкоштовна їжа в
офісі, безкоштовне медичне страхування, заняття бігом, плаванням, футболом, баскетболом та іншими
видами спорту.
Genesis визнано найкращим IT-роботодавцем в Україні в категорії понад 1500 співробітників за
результатами щорічного опитування DOU. Ми отримали високі оцінки за такими критеріями як професійне
зростання, умови і оплата праці, спілкування з керівництвом і колегами тощо.

Графік роботи:
 повна зайнятість

Роботодавець Genesis
```

Світлина 6

На цьому прикладі продемонстровано роботу методів. У вакансії йшлося про володіння інструментами Adobe Photoshop, Adobe Illustrator, Figma – що і зобразив парсер. Також додатково виведено “прапорець”, який ідентифікує чи потрібне портфоліо чи ні. Відображується також інформація щодо заробітної плати та зайнятості, які вичленені з тексту за допомогою розроблених методів. Отримано список гнучких навичок відповідно до створеного словника. Результат роботи програми можна побачити на Світлині 7.

```
📍ЛОКАЦІЯ: Київ
🛠️ІНСТРУМЕНТИ: ['Adobe Photoshop', 'Adobe Illustrator', 'Figma']
✋УВАГА: Потрібне портфоліо
🔎 ДОСВІД:
досвід роботи на аналогічній позиції;

💰 ЗАРОБІТНА ПЛАТА: 27000 UAH
          Валюта: ['UAH']
⏳ЗАЙНЯТІСТЬ:
 повна зайнятість

🧠 ГНУЧКІ НАВИЧКИ: ['Командний гравець', 'Креативний']
```

Світлина 7

```
📍ЛОКАЦІЯ: Львів
🛠️ІНСТРУМЕНТИ: ['Adobe Photoshop', 'Adobe Illustrator', 'Adobe InDesign']
✋УВАГА: Потрібне портфоліо
🔎 ДОСВІД:
💰 ЗАРОБІТНЯ ПЛАТА: 34 000 $
          Валюта: ['USD']
⏳ЗАЙНЯТІСТЬ:
 повна зайнятість

🧠 ГНУЧКІ НАВИЧКИ: ['Комунікативний', 'Пунктуальний, відповідальний', 'Наявність смаку', 'Стресостійкість']
```

Світлина 8

Демонстрація різних підходів у методі заробітної плати:

На Світлині 8 продемонстровано, відпрацювання пошуку заробітної плати з розділеними тисячами, позначенням валюти, та ідентифікація символу за допомогою розробленого алгоритму по розпізнаванню валюти. Успішно відпрацював метод ідентифікації гнучких навичок – виведені всі важливі вміння, що зазначені у вакансії.

```
📍ЛОКАЦІЯ: Львів
🛠️ІНСТРУМЕНТИ: ['Adobe Photoshop', 'Adobe Illustrator']
✋УВАГА: Потрібне портфоліо
🔎 ДОСВІД:
Досвід створення логотипів та фірмового стилю.

💰 ЗАРОБІТНА ПЛАТА: не зазначено
          Валюта: ['EUR']
⏳ЗАЙНЯТІСТЬ:
 повна зайнятість

🧠 ГНУЧКІ НАВИЧКИ: ['Командний гравець', 'Креативний']
```

Світлина 9

На Світлині 9 продемонстровано результат роботи методу, у разі сценарію, коли суми не зазначено, проте зазначена валюта (напр.: “дохід в євро”) .

```
📍ЛОКАЦІЯ: Локацію не вказано
🛠️ІНСТРУМЕНТИ: ['Adobe Photoshop', 'Adobe Illustrator']
✋УВАГА: Потрібне портфоліо
🔎 ДОСВІД:
-має досвід на аналогічній посаді від 1 року;

💰 ЗАРОБІТНА ПЛАТА: 29k
          Валюта: ['UAH']
⏳ЗАЙНЯТІСТЬ:
 повна зайнятість

🧠 ГНУЧКІ НАВИЧКИ: ['Командний гравець', 'Креативний']
```

Світлина 10

Світлина 10 зображає випадок, коли міститься позначка тисяч “k”. Метод успішно відпрацьовує та повертає результат разом з ідентифікацією валюти.

ВИСНОВКИ

Описано основні етапи розробки програми, яка структурує вхідні дані природною мовою у сфері графічного дизайну. Наразі вакансія, написана природною мовою, не є структурованою інформацією, тому виникає необхідність у знаходженні NLP рішень.

В результаті цієї роботи було досліджено сферу NLP, розроблено аналітичні методи та інструменти, на основі яких був створений та оптимізований парсер. За мету було взято реалізацію готового продукту – програми, яку згодом можна вбудувати у застосунок, його ж і представлено у розділі “фінальний продукт”.

ЛІТЕРАТУРА

1) О. Р. Жежерун, О.Р. Смиш, “Автоматизація розв’язування задач з планіметрії, записаних природною українською мовою”, 2020. http://ekmair.ukma.edu.ua/handle/123456789/19355

2) R.Kibble “Introduction to natural language processing”, Undegraduate study in Computing and related programmes, University of London, International Programs, 2013. https://london.ac.uk/sites/default/files/study-guides/introduction-to-natural-language-processing.pdf

3) Когут В.С., О.Тарасова, “Комп’ютерна лінгвістика як галузь мовознавства”, Хмельницький Національний університет, 2020. http://elar.khmnu.edu.ua/jspui/bitstream/123456789/9691/4/182-709-PB-83-85.pdf

4) Lenhart Schubert, “Computational Linguistics”, Stanford University, 2014 https://plato.stanford.edu/entries/computational-linguistics/

5) Вакансія ст. 26, Світлина 6: https://jobs.dou.ua/companies/genesis-technology-partners/vacancies/150332/

6) Тексти вакансій: https://www.work.ua/jobs-графічний+дизайнер/