Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Навчально-науковий інститут філології
кафедра української мови та прикладної лінгвістики

Автоматичне визначення хибних друзів перекладача для української та польської мов

Кваліфікаційна робота бакалавра
студента 4 курсу
освітньої програми
«Прикладна (комп'ютерна) лінгвістика
та англійська мова»
спеціальності – 035.10 Філологія (прикладна лінгвістика)
галузі знань – 03 гуманітарні науки
Кирила КЛИЧЛІЄВА
Науковий керівник:
к.т.н., доц. Микола КОСТІКОВ

«Допущено до захисту»
Протокол засідання
кафедри української мови та прикладної лінгвістики
протокол № 15 **від «** 6 **»** 06 **2024 року**

завідувач кафедри _____(підпис)
к.філол.н., доц. Сергій РІЗНИК

КИЇВ – 2024

АНОТАЦІЯ

Це дослідження присвячене створенню алгоритму автоматичного виявлення хибних друзів перекладача для української та польської мов. Актуальність роботи зумовлена відсутністю напрацювань у цьому напрямку для аналізованої пари мов, а також можливістю застосування такої системи, зокрема, для автоматизації укладання словників міжмовних омонімів і генерації вправ для вивчення таких слів. Об’єктом дослідження є українська та польська лексичні системи, а предметом – явище міжмовної омонімії, притаманне цим мовам. Створення алгоритму, здатного автоматично ідентифікувати пари хибних друзів перекладача, є метою й основним завданням цієї роботи. Для досягнення цієї мети були поставлені наступні завдання: аналіз українсько-польської міжмовної омонімії, виокремлення морфологічних і фонетичних особливостей цих мов, створення тестувальної вибірки українсько-польських лексичних пар, аналіз підходів і бібліотек, програмування алгоритму та демонстраційного вебзастосунку. У першому розділі здійснено теоретичний огляд наявних досліджень у галузі міжмовної омонімії, проаналізовано українську та польську лексичні системи в контексті слов'янських мов. Другий розділ присвячено створенню алгоритму автоматичного виявлення фальшивих друзів перекладача, а також вебзастосунку, що дозволяє користувачеві протестувати алгоритм на власних текстах. Результати дослідження показали, що створений класифікатор є ефективним інструментом для автоматичного виявлення фальшивих друзів перекладача в українській і польській мовах. Алгоритм поєднує традиційні NLP підходи із сучасними, зокрема, векторними репрезентаціями слів і мовними моделями. Майбутні напрямки дослідження охоплюють покращення точності класифікаторів, розширення датасету та лексичних категорій слів, а також подальші тестування й інтеграцію мовних моделей.

Ключові слова: хибні друзі перекладача, міжмовна омонімія, класифікація, українська мова, польська мова, обробка природної мови, машинне навчання.

ABSTRACT

This study is devoted to the creation of an algorithm for the automatic detection of false friends for the Ukrainian and Polish languages. The relevance of the work is due to the lack of research in this direction for the analyzed pair of languages, as well as the possibility of using such a system, in particular, to automate the compilation of dictionaries of interlingual homonyms and the generation of exercises for learning such words. The object of research is the Ukrainian and Polish lexical systems, and the subject is the phenomenon of interlingual homonymy inherent in these languages. The creation of an algorithm capable of automatically identifying pairs of false friends is the goal and main task of this work. To achieve this goal, the following tasks were set: analysis of Ukrainian-Polish interlingual homonymy, identification of morphological and phonetic features of these languages, creation of a test sample of Ukrainian-Polish lexical pairs, analysis of approaches and libraries, programming of an algorithm and a demonstration web application. In the first chapter, a theoretical review of available research in the field of interlingual homonymy was carried out, Ukrainian and Polish lexical systems were analyzed in the context of Slavic languages. The second section is devoted to the creation of an algorithm for automatic detection of false friends, as well as a web application that allows the user to test the algorithm on their own texts. The results of the study showed that the created classifier is an effective tool for automatic detection of false friends in the Ukrainian and Polish languages. The algorithm combines traditional NLP approaches with modern, in particular, word embeddings and language models. Future research areas include improving the accuracy of the classifiers, expanding the dataset and lexical word categories, as well as further testing and integration of language models.

Keywords: false friends, interlingual homonymy, classification, Ukrainian, Polish, natural language processing, machine learning.

ЗМІСТ

ВСТУП

При вивченні нової іноземної мови людині можуть значно допомогти знання мов, які вона знає або раніше опановувала. Таким чином, людина може розуміти окремі слова, цілі речення та тексти, написані чи озвучені, здавалося б, невідомою мовою. Причиною цього феномену є наявність такого пласта лексем як когнати. Когнати – слова, що мають схоже написання та значення у двох чи більше мовах і при цьому мають спільне походження. З іншого боку, існують також пари слів, які здаються подібними, але мають різне значення в деяких або всіх контекстах – це так звані хибні друзі перекладача або міжмовні омоніми [5]. Часто хибні друзі вводять в оману людей, що вивчають нові мови, особливо близькоспоріднені, як-от українська та польська, що належать до слов'янської гілки індоєвропейських мов.

Словники, що фіксують міжмовні омоніми, вже тривалий час укладаються, зокрема, такий словник існує й для мовної пари українська-польська [4]. Такі словники створюються переважно мануально, і ця робота може тривати роками. Автоматизація виявлення фальшивих друзів перекладача – це малодосліджений напрямок в NLP. Автоматичне визначення міжмовних омонімів для двох і більше мов визначається як проблема класифікації, тип якої залежить від обраної кількості лексичних груп для класифікації:

1. Бінарна класифікація передбачає групування вхідних даних за двома класами, наприклад, хибні друзі та когнати [22];
2. Завданням мультикласової класифікації є розподіл вхідних пар слів за трьома або більше класами: когнати, часткові когнати, фальшиві друзі, непов'язані пари слів тощо;
3. Мультилейблова класифікація є подібної до мультикласової, за винятком того, що вона передбачає можливість призначення вхідному прикладу даних більше ніж одного класу [39]. Оскільки мова є ймовірнісною системою, багатозначною та невизначеною, значення слова X в мові A може збігатися зі значенням лексеми Y в мові B в одному

контексті, а в іншому – відрізнятися. Таким чином у першому випадку слова X та Y будуть когнатами, а в другому випадку – міжмовними омонімами, тож таким чином можливе присвоєння більше одного класу цій парі слів.

Дослідження міжмовних омонімів і когнатів має велике **практичне значення** в NLP. Автоматичне виявлення фальшивих друзів може віднайти застосування, зокрема, у вирівнюванні речень (англ. sentence alignment), покращенні статистичних моделей машинного перекладу, розведенні омонімії для різних лінгвістичних задач, укладанні вправ для завчання хибних друзів перекладача, автоматизації укладання словників міжмовних омонімів тощо.

Більшість досліджень у напрямку автоматичного виявлення хибних друзів перекладача стосувалися західноєвропейських мов, зокрема, англійсько-французької та іспансько-португальської мовних пар. Зважаючи на це, **актуальність** теми визначається відсутністю напрацювань щодо автоматичної ідентифікації хибних друзів перекладача між українською та польською мовами.

Наукова новизна та мета цього дослідження полягає у створенні алгоритму, що зможе класифікувати українсько-польські пари слів на три категорії: хибні друзі, когнати та непов’язані слова (такі, що не мають орфографічної подібності та, відповідно, не можуть бути ані когнатами, ані фальшивими друзями). Важливою частиною роботи є створення датасету, що міститиме пари українських і польських слів, кожна з яких належатиме до однієї з трьох вищевказаних категорій.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

1. Опрацювати відповідну наукову літературу та термінологію, визначити категорії слів, якими ми оперуватимемо в ході дослідження;
2. Проаналізувати лексичний склад української та польської мов, а також специфіку їхньої міжмовної омонімії;
3. Провести огляд підходів, що використовуються в напрямку автоматичного виявлення хибних друзів перекладача;

4. Створити датасет, що міститиме пари українських і польських слів, згрупованих за трьома категоріями: фальшиві друзі, когнати, непов’язані слова;
5. Виробити метод уніфікації українських і польських слів, оскільки порівняння орфографічних характеристик для слів обох мов є складною задачею через використання різних алфавітів носіями цих мов;
6. Протестувати метрики визначення орфографічної та семантичної подібності, створити пайплайн з використанням цих метрик для виявлення хибних друзів перекладача;
7. Проаналізувати результати роботи протестованого алгоритму;
8. Створити вебзастосунок, що імплементує можливості мовних моделей для виявлення фальшивих друзів перекладача, для демонстрації результатів дослідження.

Об’єктом дослідження є українська та польська лексичні системи.

Предмет дослідження – явище міжмовної омонімії в українській і польській лексиці.

Матеріал дослідження – датасет з парами українських і польських слів, зібрані з різноманітних вебресурсів, а також українські та польські тексти для пошуку міжмовних омонімів у них.

Методи дослідження охоплюють класифікаційні методи в NLP, метрики орфографічної та семантичної подібності слів, великі мовні моделі, методи машинного навчання.

Практичне значення роботи полягає в можливості використання створеного алгоритму для виявлення хибних друзів серед інших мовних пар. Крім того, результати роботи програми можуть бути використані для автоматизованого укладання словників міжмовних омонімів та генерації вправ для вивчення таких слів.

Теоретичне значення: отримані результати дослідження можуть надалі використовуватися в компаративному аналізі української та польської мов на

різних рівнях завдяки детальному аналізу та порівнянню орфографічних, семантичних і морфологічних характеристик цих мов.

Інформаційна база дослідження: великі мовні моделі, попередньо натреновані моделі spacy, дані формату HTML, JSON, DataFrame, мова програмування Python і її бібліотеки fasttext, requests, beautifulsoup4, pandas, difflib, jellyfish, nltk, math, polyglot, re, spacy, scikit-learn, openai, streamlit.

Структура роботи. Кваліфікаційна робота бакалавра складається зі вступу, двох розділів з підрозділами, висновків до кожного розділу, загальних висновків, переліку використаних джерел (50) і додатків.

Апробація дослідження: доповідь на тему «Автоматичне виявлення хибних друзів перекладача для української та польської мов» на X Міжнародній науково-технічній Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» [2].

РОЗДІЛ 1. ОГЛЯД НАЯВНИХ ДОСЛІДЖЕНЬ МІЖМОВНОЇ ОМОНІМІЇ

1.1. Розмежування понять хибних друзів перекладача, когнатів і часткових когнатів.

Лінгвістична спільнота зробила значний внесок у напрямку вивчення міжмовної омонімії, зокрема й українсько-польської, а також автоматичного виявлення хибних друзів перекладача. Однак автори робіт на цю тематику використовують різні категорії слів і відмінну термінологію, тому нам важливо чітко визначити поняття, якими ми будемо оперувати в ході нашого дослідження та керуватися при створенні набору даних.

Міжмовна омонімія – це лінгвістичне явище, що охоплює лексико-семантичний рівень мови, а саме пласт слів у двох або більше мовах, які співвідносяться у плані вираження і розрізняються в плані змісту [5]. Міжмовні омоніми є більш науковим терміном порівняно з хибними або фальшивими друзями перекладача, хоча вони означають абсолютно тотожне й будуть використовуватися в ході цього дослідження синонімічно. Така більш народна назва міжмовних омонімів зародилася у французькому мовознавстві (фр. faux amis du traducteur) і закріпилася в лінгвістичних школах інших країн, замінивши місцеві терміни, наприклад, німецький «irreführende Fremdwörter» та англійський «misleading words of foreign origin» [5]. Такі лексеми можуть ввести в оману перекладача, змушуючи його вважати, що він знає значення слова тоді, коли воно насправді означає щось інше. Термін «хибні друзі» (англ. false friends, калька з фр. faux amis) символізує цю оманливість, адже як і друзі, ці слова виглядають знайомо і безпечно, але насправді вони легко можуть стати джерелом помилок у перекладі, зокрема й машинному. Інші терміни на позначення цієї лексичної категорії охоплюють «псевдоінтернаціоналізми», «міжмовні пароніми», «помилкові/неповні лексичні паралелі» й «оманливі когнати» (англ. deceptive cognates) [1]. В українсько-польській мовній парі таких слів є сотні, деякі відомі приклади подано в таблиці 1.1.1.

Таблиця 1.1.1. Приклади хибних друзів перекладача для української та польської мов

Українське слово	Польське слово	Переклад польського слова
Завод	Zawód	Професія
Світ	Świt	Світанок
Склеп	Sklep	Магазин
Овоч	Owoc	Плід
Мотлох	Motłoch	Юрба

Натомість візуально й аудіально схожі пари слів, що мають однакову семантику, називаються *когнатами* (англ. cognates) або справжніми друзями перекладача (англ. true friends від фр. vrais amis). Термін «когнат» походить від латинського «cognatus», що означає «кровно пов'язаний» [24], себто «родич», що вказує на спільне походження слів у різних мовах. Відомий британський лінгвіст Девід Крістал у своїй роботі «A Dictionary of Linguistics and Phonetics» зазначає, що когнати – це слова, що мають спільне етимологічне походження [9]. Як наслідок, такі слова переважно мають схоже написання та звучання.

Українська та польська мови належать до різних груп слов'янської гілки (східні та західна відповідно) індоєвропейської мовної сім'ї, і наявність великого спільного пласта слів між цими мовами пояснюється тим, що ще півтори тисячі років тому існувала праслов'янська мова, яка дала початок усім сучасним мовам цієї групи. Приклади українських когнатів та їхніх слів-предків подано у таблиці 1.1.2.

Таблиця 1.1.2. Приклади українсько-польських когнатів та їхніх слів-предків.

Українське слово	Польське слово	Праслов'янське слово-предок
Кров	Krew	*Krv̂
Ніс	Nos	*Nosъ
Долоня	Dłoń	*Dolnь

Деякі лінгвісти відносять до когнатів тільки ті пари слів, які є стовідсотково орфографічно ідентичними, а ті, що мають близьку, проте не ідентичну фонетико-морфологічну будову, виокремлюють в іншу категорію – «близькоспоріднені слова» (англ. near-cognates). Так само можна натрапити на думку, що хибними друзями є лише слова, які збігаються повністю в плані вираження. Керуючись цим правилом, можна стверджувати, що українській і польській мові фактично не властива міжмовна омонімія через послуговування цими мовами різними алфавітами, а також через значні морфологічні й фонетичні відмінності цих мов. До прикладу, візьмемо до уваги пару слів *зв'язківець* і *związkowiec*. Українцю, що вивчає польську мову, чи поляку, що опановує українську, слова відповідних мов найімовірніше здаватимуться знайомими навіть попри фонетичні чи морфологічні відмінності. Це спричинено контрастивними явищами між українською та польською мовами та відповідними лінгвістичними закономірностями. Наприклад, зазвичай український суфікс *-ець* відповідає польському *-iec* з непалаталізованим кінцевим *[c],* як-от у парах слів *кравець – krawiec, кінець – koniec, мешканець – mieszkaniec*. Праслов'янські етимологічні *[e]* та *[o]* в українській мові внаслідок явища редукції голосних перейшли в *[i]*, натомість усі інші слов'янські, зокрема й польська, мови зберегли етимологічні голосні *[o]* та *[e]* (див. таблицю 1.1.3), як і в словах *зв'язк**і**вець – związk**o**wiec*. Український апостроф найчастіше в польській мові має відповідну літеру – *i*; слова типу «м**'**ясо» у польській пишуться як «m**i**ęso». Таким чином, подані українське та польське слово попри

візуальні чи звукові відмінності мають очевидно спільне походження, що стає ще більш очевидно після знаходження мовних закономірностей.

Таблиця 1.1.3. Порівняння українських слів (зі зредукованими голосними) з іншими слов'янськими мовами.

Українська	Білоруська	Польська	Чеська	Болгарська
Кіт	Кот	Kot	Kočka	Котка
Шість	Шэсць	Sześć	Šest	Шест
Лід	Лёд	Lód	Led	Лед
дім	Дом	Dom	Dům	Дом
Кінь	Конь	Koń	Kůň	Кон

Генетичні когнати (англ. genetic cognates) – це пари слів у споріднених мовах, які походять безпосередньо від одного й того ж слова в протомові [10]. Як уже зазначалося, в контексті української та польської такою мовою є праслов'янська, яка не була писемною і може бути лише реконструйована на основі живих і мертвих писемних слов'янських мов. Через поступові фонетичні та семантичні зміни протягом тривалого часу генетично споріднені слова часто відрізняються за формою та/або значенням, наприклад: *мати – matka, четвер – czwartek, ніч – noc* тощо. Таких слів можна віднайти безліч між аналізованими мовами, оскільки відсоток спільної лексики між українською та польською сягає 70. Категорія генетичних когнатів не включає лексичні запозичення, тобто слова, адаптовані з інших мов, зокрема, численні польські запозичення в українській мові й навпаки, а також запозичення з третіх мов, перш за все німецької (таблиця 1.1.4) та французької (таблиця 1.1.5).

Таблиця 1.1.4. Німецькі запозичення в українській і польській мовах

Німецьке слово-джерело	Польське запозичення	Українське запозичення
Farbe	Farba	Фарба
Gattung	Gatunek	Ґатунок
Danke	Dziękuję	Дякую
Geschmack	Smak	Смак
Schinken	Szynka	Шинка

Таблиця 1.1.5. Французькі запозичення в українській і польській мовах

Французьке слово-джерело	Польське запозичення	Українське запозичення
Restaurant	Restauracja	Ресторан
Boulevard	Bulwar	Бульвар
Salon	Salon	Салон
Parasol	Parasolka	Парасолька
Shampooing	Szampon	Шампунь

Наразі українська та польська мови, як загалом і всі інші мови світу, активно насичуються англізмами. Щоправда, через порівняно нещодавнє становлення англійської як міжнародної мови та мови інтернету позиція англізмів у цих мовах не така закріплена і їхнє вживання часто не є внормованим порівняно з давніми запозиченнями з впливових раніше німецької та французької мов. Приклади англійських запозичень в українські та польській мовах подано в таблиці 1.1.6.

Таблиця 1.1.6. Сучасні англійські запозичення в українській і польській мовах

Англійське слово-джерело	Польське запозичення	Українське запозичення
Hacker	Haker	Хакер
Email	Email	Емейл
Smartphone	Smartfon	Смартфон
Startup	Startup	Стартап
Podcast	Podcast	Подкаст

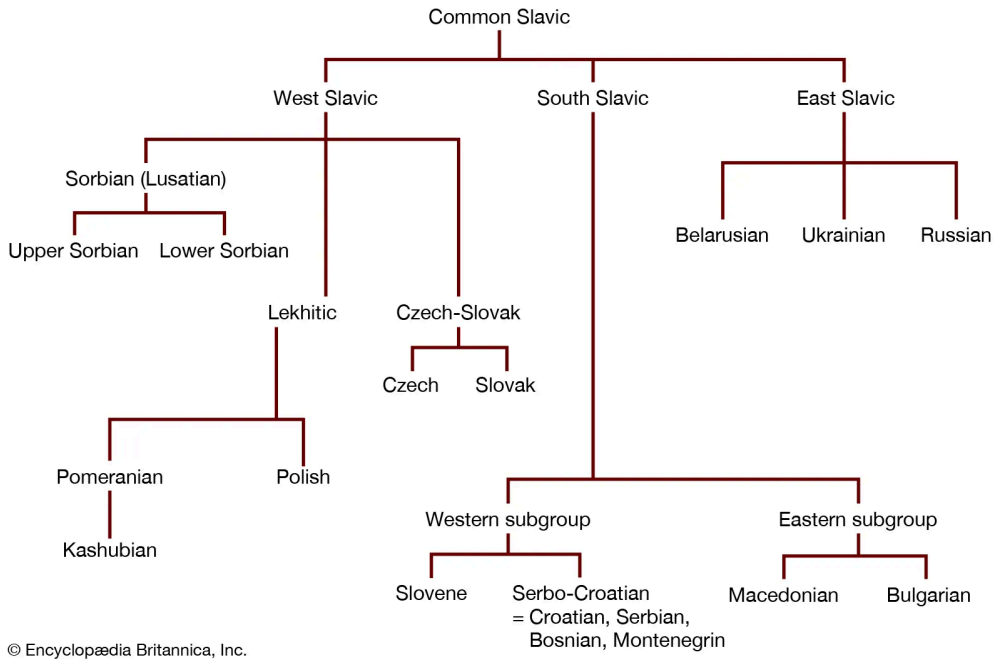
Часткові когнати (англ. partial cognates) – це пари слів, які мають однакове значення в обох мовах, але не в усіх контекстах. Такі лексеми можуть поводитися як когнати або ж як хибні друзі перекладача залежно від того, яке значення використовується в кожному контексті [10].

Непов'язані пари (англ. unrelated words) – це слова, що не мають орфографічної подібності (шлюб – rozwód), хоча можуть мати семантичну (шлюб – małżeństwo). Цю категорію слів ми протиставляємо когнатам і хибним друзям, оскільки ті є орфографічно подібними категоріями слів

1.2. Українська та польська лексичні системи в контексті слов'янських мов. Міжмовна омонімія.

Філогенетика (також лінгвістична філогенетика) – напрямок історичної лінгвістики, метою якої є групування мов зважаючи на їхню відстань в деревоподібному графі (англ. rooted tree), що відображає їхню історичну еволюцію. Метою цієї мовознавчої дисципліни є оцінка еволюційної історії мов, які зазвичай представлені у формі дерева, де коренем є спільний предок мовної групи чи сім'ї, а листя відображає мови, що розвинулися з мови-предка [38]. Таким чином, у контексті слов'янських мов, коренем дерева є праслов'янська мова, а листками – сучасні слов'янські мови, яких налічується близько 20 і в

сучасному мовознавстві поділяються найчастіше на три підгрупи, як зазначено на малюнку 1.2.1.



Малюнок. 1.2.1. Сучасна загальноприйнята трихотомічна класифікація слов'янських мов [23]

Крім трихотомічної класифікації слов'янських мов раніше популярними серед лінгвістів були різні дихотомічні, які переважно виокремлювали західну гілку (ті ж мови, що й в сучасній західній підгрупі), але об'єднували східну та південну в одну – південно-східну, рідше – північну (сучасні мови західної та східної підгруп) та південну [8].

У філогенетиці найчастіше використовуються методи лексикостатистики, що базуються на попередньо визначеній сукупності лінгвістичних концептів і їхньому лексичному прояві в мовах, що підлягають класифікації. У 1950-х роках був розроблений американським лінгвістом і антропологом Моррісом Сводешем один із найпоширеніших і дотепер лексикостатистичних методів [38]. Алгоритм Сводеша виглядає наступним чином:

1. Визначення стандартного списку лінгвістичних концептів. Зазвичай такий список включає базові категорії слів, які є найбільш

стабільними та найкраще зберігаються в споріднених мовах попри століття мовної еволюції: частини тіла, базові дієслова, числа, матеріали, їжа, явища природи тощо. Кількість слів у списку не є сталою, але найчастіше можна зустріти 207 або 215 позицій;

2. Оцінка на основі створеного списку того, чи подібним є те, чи інше слово в аналізованих мовах (тобто когнат чи ні);
3. Обрахування відношення когнатів для кожної мовної пари;
4. Створення матриці подібності та генерація на її основі графіка (зазвичай у вигляді дерева) для наочного відображення близькості та віддаленості мов. На такому графіку чим ближче мови знаходяться одна до одної – тим подібніше є їхній лексико-семантичний склад.

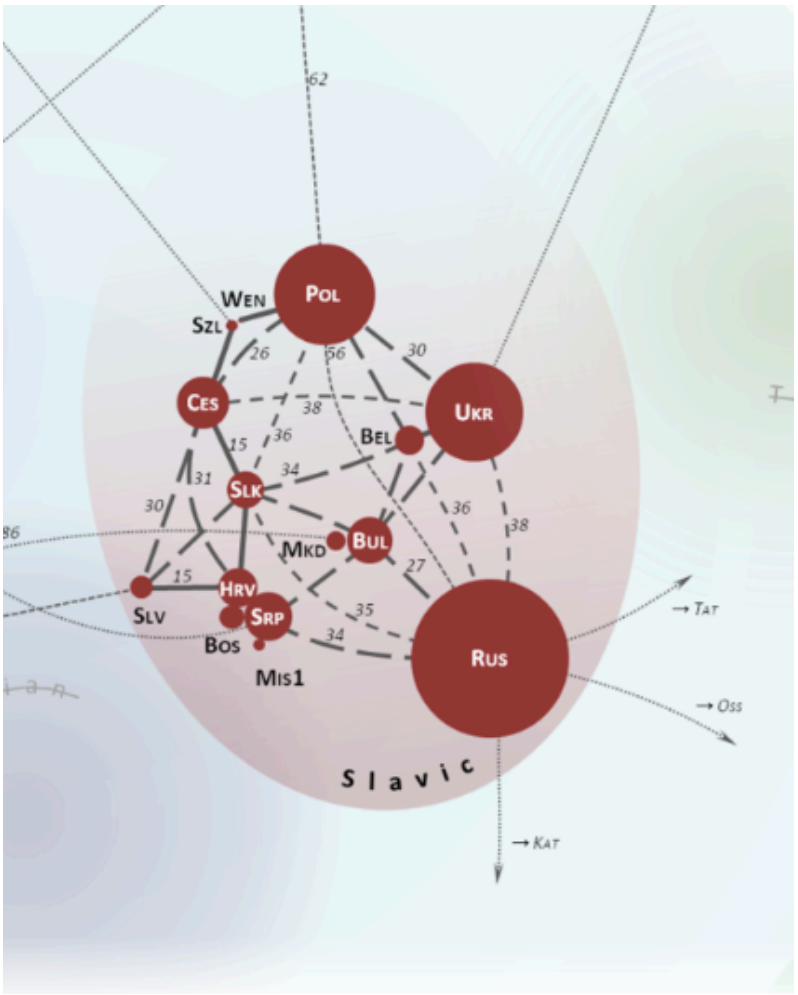
Така тактика мала значний вплив на філогенетику й історичну лінгвістику загалом, оскільки заведено вважати, що саме лексичний компонент якнайяскравіше відображає спорідненість мов. На основі вищеописаного лексикостатистичного дослідження визначається міра лексичної подібності двох мов. У мовознавстві лексична подібність – це міра подібності загальних вокабулярів двох мов. Лексична подібність, що дорівнює 1 (або 100%), означає повний збіг між словниками, тоді як 0 вказує на те, що в них немає спільних слів [29]. До прикладу, якщо порівнювати лексичну подібність слов'янських мов за списками Сводеша, то найближчою до української мови є білоруська (92% спільних слів), російська (86%), польська та словацька (по 76%). Найменш подібними до української мови є македонська та словенська (по 71%) [15].

	лат.	пр.	рос.	укр.	білор.	пол.	чеськ.	сл.	н-л.	в-л.	сл.	сер.	мак.	бол.
литовська	68	49	47	47	47	43	44	46	46	46	44	44	45	46
латиська		44	45	40	44	40	41	42	43	45	40	41	41	42
пруська			41	41	40	39	42	42	42	42	40	41	39	40
російська				86	86	77	74	74	73	74	74	71	70	74
українська					92	76	73	76	74	74	71	73	71	72
білоруська						80	77	80	78	78	76	77	74	77
польська							81	85	83	80	79	75	71	74
чеська								92	87	87	84	79	75	74
словацька									87	86	80	80	76	75
н.-лужицька										94	78	74	73	71
в.-лужицька											78	77	76	73
словенська												85	75	76
сербська													84	80
македонська														86
болгарська														

Мал. 1.2.2. Подібність балто-слов’янських мов за списками Сводеша [15]

Для слов’янських мов, які загалом мають вищі показники лексичної подібності ніж навіть романські та германські через відносну молодість гілки, за допомогою методу Сводеша було визначено, що в списку 97 слів (47 %) є ідентичними для всіх без винятку слов’янських мов [16].

Відомою у філологічних колах є діаграма авторства українського мовознавця Тищенка Костянтина Миколайовича, що відображає лексичну відстань між усіма індоєвропейськими мовами (див. малюнок 1.2.2.). Відстань між мовами базується на різниці словників цих мов: чим більше лексичних збігів – тим ближче мови та навпаки. Наприклад, хорватська та сербська, болгарська та македонська згідно з графіком мають лексичну подібність понад 90-95%, при тому, що лінгвістичною спільнотою заведено вважати діалектами однієї мови говірки, подібність яких складає понад 85%. Говорячи про українську мову, то як і відповідно до попереднього дослідження, згідно з К. Тищенком найнаближенішою до української є білоруська мова (84% спільної лексики), далі йде польська (70%), словацька (66%) та російська (62%).



Мал. 1.2.2. Дерево лексичних відстаней слов'янських мов (фрагмент дерева лексичних відстаней індоєвропейських мов) [25]

Лексична подібність – це лише одна з ознак взаємозрозумілості (англ. mutual intelligibility) двох мов, оскільки вона також залежить від ступеня фонетичної, морфологічної та синтаксичної подібності. Варіації через відмінності у словниках впливають на цей показник. Наприклад, лексична подібність між французькою й англійською мовами є значною в лексичних галузях пов'язаних з культурою, політикою, кухнею, тоді як їхня подібність значно падає, коли йдеться про базові або ж функціональна слова. Порівнюючи показники лексичної подібності для української та російської мов легко помітити, що подібність цих мов складає 86% за списками Сводеша і всього 62% за методом Тищенка. Така значна різниця пояснюється зокрема тим, що українська та російська базова лексика, яку до уваги бере метод Сводеша, має переважно спільне східнослов'янське походження, а от книжна та наукова

українська та російська мови надто відрізняються. Отже, метод Тищенка є більш об'єктивним при порівнянні загальної лексичної подібності мов, оскільки метод Сводеша не є репрезентативним для всієї лексичної системи, бо охоплює лише найбазовішу лексику, яка є дуже подібною серед споріднених мов. Натомість алгоритм Сводеша простим способом наочно відображає зв'язки між мовами в межах групи чи сім'ї.

Основні ознаки, характерні слов'янській протомові, значною мірою збереглися в усіх (в окремих випадках – у більшості) сучасних слов'янських мовах. Зокрема, на фонетичному рівні це наявність випадних голосних (укр. *взяти – візьму*, пол. *wziąć – wezmę*), чергування *[г], [к], [х]* з *[ж], [ч], [ш] (рік – річний, rok – roczny*). У сучасних слов'янських мовах збереглося, зокрема, протиставлення категорій недоконаного та доконаного виду, зближення особових форм теперішнього часу тощо. Водночас кожна зі слов'янських мов мала свій шлях розвитку, внаслідок чого у їхній структурі з'явилися своєрідні риси. Наприклад, у фонетичних системах української та польської мов знайшли відображення різні рефлекси звукосполучень типу **tort, tolt, tert, telt*, наприклад, *голова – głowa, берег – brzeg*; українська мова розвинула повноголосі форми цих звукосполучень, а польська – зберегла неповноголосі. Праслов'янські **tj, dj, kt'* в українській мові перейшли в *[ч], [ж],* у польській – у *[с], [dz]*, наприклад, *чудо – cudo, свіча – świeca, межа – miedza*. Сучасні українська та польська мови мають як подібні риси, що пов'язані зі спільним походженням і взаємовпливами, так і своєрідні ознаки, які виявляють свою специфіку не тільки при зіставленні цих двох мов, але й у порівнянні з іншими слов'янськими та неслов'янськими мовами [3].

Таблиця 1.2.1. Автентичні граматичні риси української та польської мов [3]

Українська мова	Польська мова
Тверді приголосні перед [е]: *вітер, земля, оселя*	Фіксований і нерухомий наголос на передостанньому складі
М'які приголосні [с'], [з'], [ц'] на місці [х], [г], [к] унаслідок другої палаталізації: *льосі, нозі, щуці*	Збереження носових голосних [ą], [ę]: *jętka, wąsy, dziękuję*
[і] на місці новозакритих [о], [е]: *біб, ніс, ніч* Див. табл. 1.1.3	Наявність двох рядів шиплячих типу [š] і [ś]
Фрикативний [г] на місці зімкнено-проривного праслов'янського [g]: *горб, гора, горе*	Відсутність якісної редукції ненаголошених голосних.
Іканізм – поява початкового [і] перед групою приголосних (*імла, іржа*), або втрата [і] в аналогічній позиції (*гра, голка*)	Неомонімічність творення прикметникових і прислівникових форм ступенів порівняння
Синтетичні форми майбутнього часу недоконаного виду на *-му, -меш, -ме, -муть*: *любитиму, сидітимуть, мріятиме*	Наявність особливого типу відмінювання числівників

Питання міжмовної омонімії у близькоспоріднених мовах розглядали у своїх працях українські філологи Акуленко М., Бублейник Л., Заславська Н., Кочерган М., Паламарчук О., зокрема, на прикладі українсько-чеської, українсько-російської та українсько-польської мовних пар [18]. Українська мовознавиця Кононенко Ірина Віталіївна у своїй праці «Українська та польська мови: контрастивне дослідження» [3] виокремлює наступні причини утворення груп українсько-польських міжмовних омонімів:

1. Українські та польські слова можуть мати спільне індоєвропейське чи праслов'янське походження, проте в процесі розвитку кожної мови вони набули різних лексичних значень зберігши подібне звучання. Вважається, що найбільше хибних друзів виникло саме у зв'язку з розходженням в семантиці слів споріднених мов у ході мовної еволюції. Приклади таких слів:

марення – marzenie

хребет – grzbiet

казати – kazać

країна – kraina

крісло – krzesło

2. Міжмовні омоніми можуть виникати внаслідок фонетичного зближення етимологічно різних слів:

опал – opal

вудка – wódka

3. Окремі пари міжмовних омонімів утворилися внаслідок розходжень у значеннях запозичених з інших мов слів під впливом внутрішньомовних процесів:

адрес – adres (з французької)

акорд – akord (з італійської)

панель – panel (з німецької)

Буває, що українські та польські слова іншомовного походження далеко відходять від значень у мові-першоджерелі, наприклад:

диня – dynia (з грецької мови, у якій відповідне слово означає «олива»)

гарбуз – arbuz (з перської мови, у якій *χarbuza* – це «ослячий огірок»)

4. Явище міжмовної омонімії може виникати як наслідок процесу переходу слів з польської в українську або навпаки:

байка ← bajka

бавитися ← bawić się

застава ← zastawa

czupurny ← чепурний

5. Різні за значенням слова можуть утворюватися завдяки подібним словотвірним процесам від коренів спільного походження, наприклад:

бігун – biegun

камінка – kamionka

риболов – rybołów

Аналізуючи міжмовні омоніми, варто враховувати, що українська та польська належать до споріднених підгруп – східнослов'янської та західнослов'янської – слов’янської мовної підгрупи. Це зумовлює наявність між ними закономірних контрастивних явищ. Лише невелика частина українсько-польських фільшивих друзів повністю збігається у плані вираження, наприклад:

афера – afera

бар – bar

череп – czerep

дека – deka

натура – natura

Левова частка українсько-польських міжмовних омонімів розходиться також за наголосом, який у польській мові, як уже зазначалося, є фіксованим, наприклад (наголоси в польських словах виділено жирним шрифтом):

*виказати – wyk**a**zać*

*гарнітур – garn**i**tur*

*маяк – m**a**jak*

*бархан – b**a**rchan*

*аромат – ar**o**mat*

Міжмовні пароніми можуть мати відмінності, пов’язані зі специфікою вираження морфем в обох мовах. Приклади:

голубок – gołąbek

духовий – duchowy

спаяти – spajać

Внаслідок звукових і морфемних змін, викликаних як закономірними, так і випадковими формальними розходженнями в основному зовні подібних слів, виникає явище міжмовної паронімії, яка найбільш поширена у граматичних класах прикметників і дієслів [3]. Порівняйте приклади подібних сходжень/розходжень:

актуальний – aktualny

ангельський – angielski

бронзовий – brązowy

вигідний – wygodny

видатний – wydatny

істотний – istotny

корисний – korzystny [3]

У нашій роботі ми будемо вважати фальшивими друзями перекладача як ідентичні у двох мовах слова, так і ті, що незначно відрізняються, зважаючи на те, що українська та польська мови мають значні морфологічні та фонетичні відмінності та послуговуються різними системами письма: українська – кириличним, польська – латиницею. Тому поняття міжмовних омонімів і паронімів будуть об’єднані в одне – міжмовні омоніми або ж хибні друзі перекладача. Також при створенні датасету буде частково враховуватися суржик, оскільки явища мовної інтерференції в Україні дуже поширені й можуть стати причиною плутанини при опануванні польської мови. Наприклад, українцю може здатися, що «jutro» – це українське «ранок», але насправді зазначене польське слово означає «завтра» [2].

1.3. Автоматичне виявлення хибних друзів перекладача

У попередніх роботах для автоматичного виявлення хибних друзів перекладача використовувалося поєднання орфографічних, синтаксичних, семантичних і частотних ознак слів.

Зокрема, була спроба розв'язати проблему автоматизації виявлення фальшивих друзів для англійської та французької мов лише враховуючи орфографічні характеристики слів. Головний недолік і неефективність цього методу полягає в ігноруванні семантики слів [3].

В іншому дослідженні [12] автоматичне розрізнення когнатів та хибних друзів відбувається у два етапи.

Перший – виявлення пар-кандидатів у непаралельному білінгвальному корпусі. Цей метод базується на орфографічній близькості двох слів подібно попередньому дослідженню. Орфографічна подібність була обрахована за формулою LCSR (англ. Longest common sub-sequence):

$$LCSR(w_1, w_2) = \frac{|LCS(w_1, w_2)|}{\max(|w_1|, |w_2|)}$$

Формула 1.3.1. LCSR

У формулі 1.3.1:

- знаменник – найдовша спільна послідовність літер
- чисельник – довжина довшого слова.

Приклади застосування формули для обчислення орфографічної подібності українського і польського слів:

1. *LCSR(склеп, sklep=склеп) = 5/5 = 1;*
2. *LCSR(актуальний, aktualny=актуальни) = 9/10 = 0.9;*

Другий етап – класифікація виявлених попередньо лексичних пар на когнати, хибні друзі та непов’язані слова залежно від семантичної близькості

слів у цій парі за допомогою семантичного словника EuroWordNet. Семантична подібність розраховувалася методами lch (Leachcock&Chodorow) і wup (Wu&Palmer) [6].

Деякі науковці ще у 2000-х використовували векторні репрезентації слів (англ. word embeddings) для класифікації справжніх і хибних друзів перекладача, однак цей підхід набув широкої популярності у 2013 році, коли була представлена некерована техніка (англ. unsupervised learning), відома як word2vec, що використовує неглибоку нейронну мережу (англ. shallow neural network) для представлення слів з великого немаркованого корпусу у вигляді векторів [11, 42].

Вектори слів полегшують обробку природної мови через їхню високу здатність встановлювати зв’язки між словами та відповідно моделювати людську мову. Така техніка в NLP базується на обчисленні векторного простору, в якому вектори є близькими, якщо їхні відповідні слова часто з'являються в одному і тому ж контексті в корпусі, що використовується для навчання [47]. Вектори дозволяють з легкістю виявляти взаємозв’язки між словами, наприклад, результат обчислення вектора *vector(«Варшава»)* - *vector(«Польща»)* + *vector(«Україна»)* ближчий до *vector(«Київ»)*, ніж до будь-якого іншого слова-вектора. Такі обрахунки для визначення подібності слів можна провести за допомогою методу *get_analogies* бібілотеки *fasttext* [31]:

model_uk.get_analogies(‘Варшава’, ‘Польща’, ‘Україна’), де:

- *model_uk* – модель з попередньо натренованими векторами для української мови (джерело текстів – Вікіпедія);
- *get_analogies* – метод для отримання слів-аналогій. У цьому випадку модель намагається визначити, яке слово відноситься до слова «Україна» так, як відноситься «Варшава» до лексеми «Польща».

  Результат роботи коду:

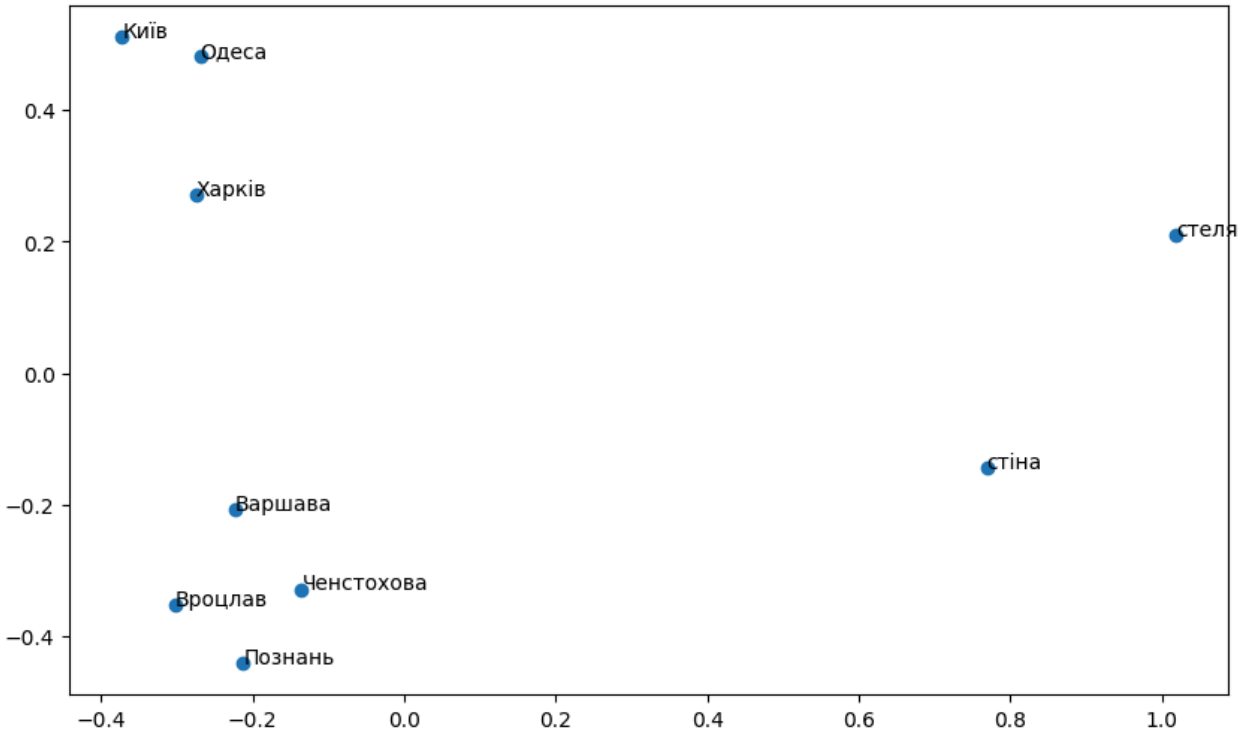
```
[(0.6110314130783081, 'Київ'),
 (0.6055029630661011, 'Москва'),
 (0.5506971478462219, 'Одеса'),
```

```
 (0.5502026081085205, 'Київ-Варшава'),
 (0.539129912853241, 'Харків-Варшава'),
 (0.5382523536682129, 'Київ-Нью-Йорк'),
 (0.5381291508674622, 'Львів-Варшава'),
 (0.5311408638954163, 'Європа'),
 (0.529330849647522, 'Україна2'),
 (0.5263713598251343, 'Київ-Прага')]
```

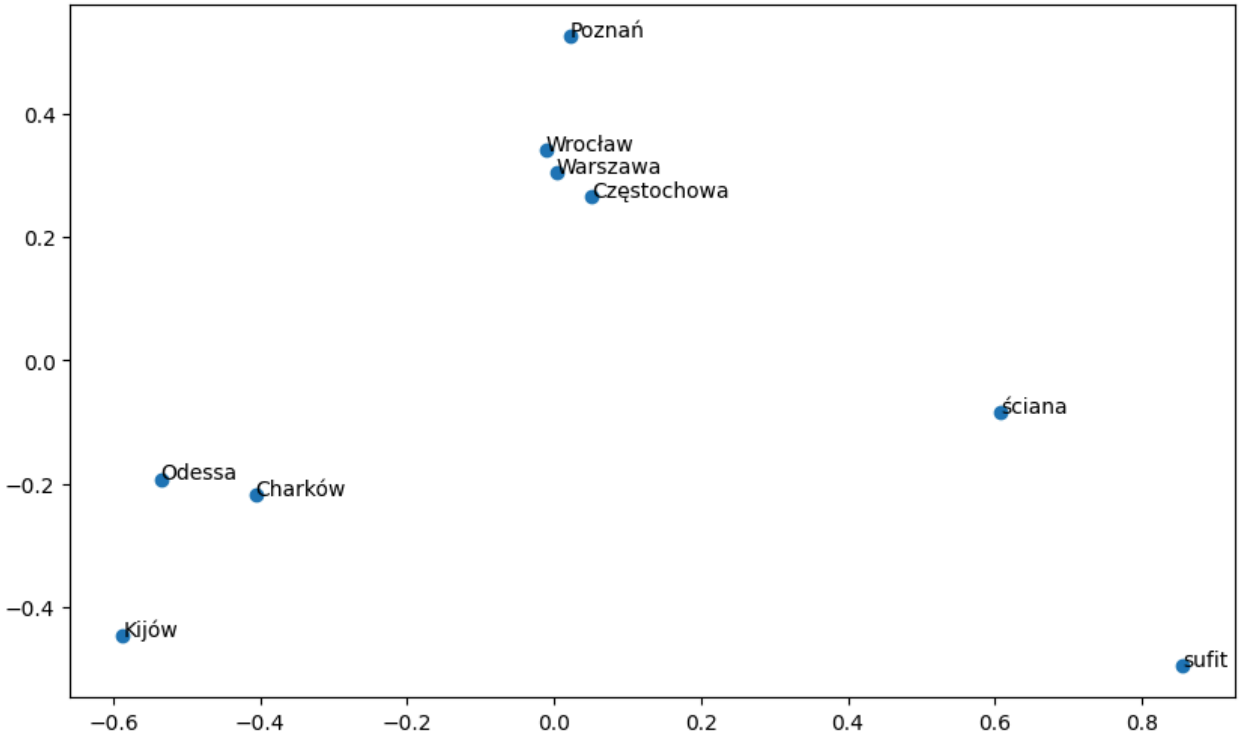
Отже, відношення між словами «Україна» та «Київ» такі самі, як між «Польща» та «Варшава»

Для використання багатомовних можливостей векторів було розроблено метод автоматичної генерації словників з невеликих наборів двомовних даних (перекладених пар слів), що ґрунтується на обчисленні лінійних перетворень між векторними просторами, побудованими за допомогою word2vec. Це подається як задача оптимізації, яка намагається мінімізувати суму евклідових відстаней між векторами вхідного та вихідного слова кожної пари, а матриця суміжності (англ. translation matrix) отримується за допомогою стохастичного градієнтного спуску.

Основні концепції word2vec наочно зображено на Мал. 1.3.1. і 1.3.2. У прикладах наведено візуалізації векторів для 9 слів, які можна розділити на три семантичні групи: міста України, міста Польщі та складові будинку. Як можемо спостерігати, споріднені слова на графіку знаходяться ближче один до одного, умовно утворюючи кластер, оскільки мають схожі вектори, тоді як вектори семантично віддалених слів знаходяться на більшій відстані [7, 11].



Мал. 1.3.1. Візуалізація векторів для українських слів



Мал. 1.3.2. Візуалізація векторів для польських слів

Висновки до розділу 1

У розділі 1 було здійснено огляд наявних досліджень у напрямку вивчення українсько-польської міжмовної омонімії, а також автоматизації виявлення хибних друзів перекладача.

Першочергово нам необхідно було дослідити термінологію, вживану лінгвістами-дослідниками явища міжмовної омонімії, оскільки існують різні підходи до визначення понять цієї лексикологічної галузі. Таким чином було визначено, що міжмовна омонімія є явищем, за якого слова в різних мовах мають однакове або подібне написання, але різне значення. Це може спричинити помилки в перекладі, особливо серед близькоспоріднених мов, таких як українська та польська. Хибні друзі перекладача (або міжмовні омоніми) часто вводять в оману людей, які вивчають нову мову, що наразі, в період підвищеної зацікавленості українцями польською культурою, і поляками — українською, є особливо актуально. Когнати — це слова, які мають спільне походження і схоже значення та написання у двох або більше мовах. Часткові когнати можуть мати однакове значення, але не в усіх контекстах, що робить їх важливими для детального лінгвістичного аналізу. Проаналізувавши термінологію, ми вирішили продовжити дослідження оперуючи трьома лексичними категоріями: когнати, хибні друзі та непов’язані пари слів. Остання категорія була виокремлена з огляду на те, що когнати та хибні друзі в будь-якому випадку – орфографічно подібні слова, і для того, аби мати змогу шукати їх у тексті для подальшої класифікації нам потрібно їх протиставляти всім іншим словам, які було виокремлено в цю категорію. Тобто третя категорія включає будь-яку пару слів, що не є ані когнатами, ані фальшивими друзями.

У ході аналізу української та польської лексичних систем в контексті слов’янської мовної групи ми виявили, що лексична подібність польської мови з українською становить аж 70%, що робить її найближчою до нашої мови після білоруської. Попри такий високий відсоток, українська та польська значно відрізняються морфологічно та фонетично, однак такі відмінності переважно контрастивні та закономірні. Було вирішено, що міжмовними омонімами ми

вважатимемо не лише абсолютно тотожні лексеми орфографічно, а й ті, що мають незначні відмінності зважаючи на ці відмінності та закономірності. Однак варто зазначити, що відмінності у фонетичних та морфологічних системах цих мов створюють певні труднощі в автоматичному виявленні хибних друзів перекладача, так само як і використання різних алфавітів носіями цих мов.

Огляд наявних підходів до автоматичного виявлення хибних друзів перекладача свідчить, що більшість методів базується на аналізі орфографічної та семантичної подібності слів. В сучасних NLP дослідженнях остання обраховується переважно за допомогою векторних репрезентацій слів. Новітні технології, зокрема, великі мовні моделі, мають великий потенціал для підвищення точності та ефективності виявлення фальшивих друзів. Одна з проблем, що постає при використанні векторних репрезентацій слів — складність обчислення подібності векторів одночасно для декількох мов.

Цей розділ закладає основу для подальшого дослідження і розробки систем автоматичного виявлення хибних друзів перекладача, що буде детально розглянуто в наступному розділі роботи.

РОЗДІЛ 2. СТВОРЕННЯ СИСТЕМИ АВТОМАТИЧНОГО ВИЯВЛЕННЯ ХИБНИХ ДРУЗІВ ПЕРЕКЛАДАЧА ДЛЯ УКРАЇНСЬКОЇ ТА ПОЛЬСЬКОЇ МОВ

2.1. Створення навчально-тестувальної вибірки

Для нашого дослідження за допомогою технології вебскрейпінгу було зібрано набір даних, що містить українські та польські пари міжмовних омонімів. Вебскрейпінг (англ. Web scraping) – процес перетворення інформації з вебресурсів у структуровані дані [43]. Джерелом даних стали декілька вебсторінок, зокрема, Вікіпедія [17]. Структуру вебсторінки, що містить пари фальшивих друзів, подано на малюнку 2.1.1. Пари польських і українських слів, як можемо бачити, розділені між собою тире. Така структура даних має назву невпорядкований список (англ. unordered list) і в розмітці HTML позначається тегом <ul>, а кожен елемент такого списку – тегом <li> [48]. Фрагмент розмітки подано на малюнку 2.1.2.

- angielski — не «ангельський» (*anieli, anielski*), а «англійський»
- arbuz — не «гарбуз» (*dynia*), а «кавун»
- bajka — не тільки «байка», але й «казка» (*baśń, bajka magiczna*)
- baretka — не «маленький бар», а «орденська планка»
- bęben — не «бубон» (*bębenek baskijski, tamburyn, tamburyno*), а «барабан»
- bidło — не «бидло» (*bydło*), а «ляда ткацького верстата»
- błąd — не «блуд» (*cudzołóstwo*), а «помилка»

Мал. 2.1.1. Фрагмент списку українсько-польських хибних друзів перекладача (Вікіпедія) [16]

Як бачимо на малюнку 2.1.1, українське слово супроводжується поясненням, що ускладнює процес витягування самих хибних друзів, тому після вивантаження даних необхідно було також провести очищення даних

Мал. 2.1.2. HTML розмітка вебсторінки «Фальшиві друзі перекладача» [17]

Для подальшого розширення вибірки необхідно використовувати словники міжмовних омонімів або ж додаткові вебресурси, однак у нас наразі немає потреби у великому датасеті чи корпусі.

Для взаємодії з вебсайтами було використано функціонал бібліотеки *requests*, що дозволяє надсилати HTTP запити певному вебсайту для подальшого вилучення інформації з нього [44]. Безпосередньо вилучення даних з вебсайту відбувається за допомогою модуля *BeautifulSoup*, що дозволяє отримувати потрібну користувачеві інформацію з файлів форматів HTML і XML [20]. Для структурування вилучених даних використовувалася бібліотека *Pandas*, що дозволяє працювати з файлами різних форматів (JSON, XLS, XLSX, CSV тощо), перетворюючи вхідний файл на особливий табулярний тип даних (DataFrame), інформацією в якому дуже легко та зручно маніпулювати [42].

Після отримання доступу до вебсайту наш код вилучає дані, що нас цікавлять, а саме пари українських і польських хибних друзів, і зберігає їх у словник, де польське слово – ключ, українське – значення. Вбудовані методи *pandas* дозволяють конвертувати такий словник у *DataFrame*.

Пари українсько-польських когнатів знайти значно легше – для цього достатньо відкрити будь-який розмовник для цих мов, оскільки такого роду словники містять переважно базову лексику, яка є в більшості випадків спільною для близькоспоріднених мов. Такі лексичні групи охоплюють числа, назви днів тижня й місяців, овочів і фруктів, базові дієслова й прикметники тощо. Порівняйте: *сніданок – śniadanie, кава – kawa, пиво – piwo, час – czas, вівторок – wtorek, чотири – cztery, шість – sześć, ходити – chodzić, малий – mały.*

Пари непов'язаних слів не вилучалися зі сторонніх ресурсів шляхом вебскрейпінгу, натомість було випадковим чином перемішано між собою лексеми в колонці польських слів *pl* (метод *sample* бібліотеки *pandas*), а колонка з українськими словами *uk* залишилася незмінною. Таким чином було отримано пари орфографічно неподібних слів. Вибірка з орфографічно непов'язаними словами є необхідною для тестування метрик орфографічної подібності двох слів.

	pl	uk	false_friends
0	przystojny	сніданок	2
1	adidasy	вегетаріанський	2
2	zabieg	напій	2
3	uczyć się	кава	2
4	dynia	сік	2

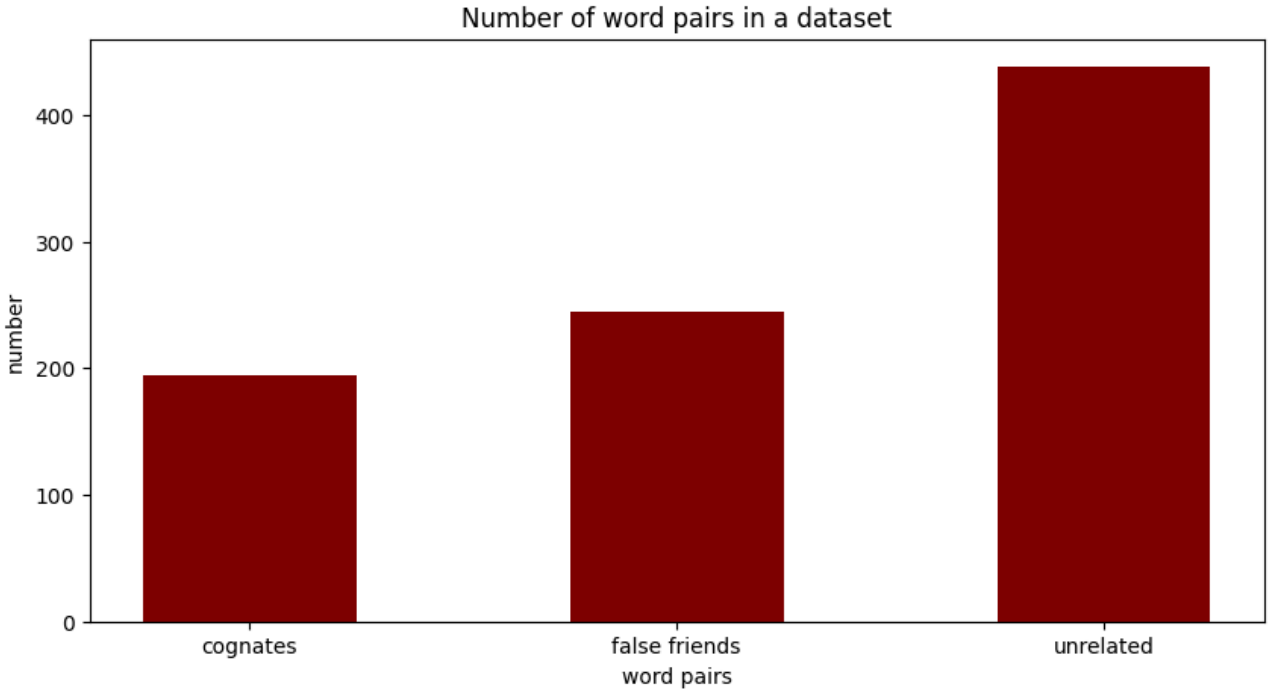
Мал. 2.1.3. Перші 5 рядків датасету непов'язаних слів

Загалом створений датасет містить 876 записів, згрупованих за трьома класами:

0 – пари когнатів;

1 – пари хибних друзів перекладача;

2 – пари непов’язаних слів.



Мал. 2.1.4. Розподіл пар слів за класами в датасеті

2.2. Автоматичне визначення орфографічної подібності слів

2.2.1. Створення класифікатора для виявлення омографів

Пайплайн у машинному навчанні – серія взаємопов’язаних етапів обробки

даних і їх моделювання, призначених для автоматизації процесу створення, навчання, оцінки та розгортання моделей і алгоритмів машинного навчання [49]. Залежно від характеру та послідовності виконання дій в пайплайні їх класифікують на серійні та паралельні. У серійному пайплайні (англ. serial pipeline) кожен етап обробки даних виконується один за одним, послідовно. Вивідні дані (англ. output) кожного етапу – вхідні дані (англ. input) наступного. Натомість в паралельному пайплайні (англ. parallel pipeline) декілька дій виконуються одночасно.

У нашому випадку для класифікації вхідних українсько-польських лексичних пар на 3 класи може стати в пригоді саме серійний пайплайн, який

матиме двоступеневу структуру. Актуальність застосування саме цього типу пайплайну пояснюється тим, що ми використовуватимемо два бінарні класифікатори, а вхідними даними другого класифікатора виступають вихідні дані першого, тобто ці два етапи мають відбуватися суворо послідовно.

Архітектура нашого пайплайну матиме наступний вигляд:

1) Бінарна класифікація пар слів на орфографічно подібні, які можуть бути своєю чергою або когнатами, або фальшивими друзями, та орфографічно неподібні. Далі ми ці класи називатимемо омографи та неомографи відповідно.
2) Подальша бінарна класифікація виявлених на першому етапі омографів на когнати та хибні друзі з використанням векторної репрезентації слів для виявлення семантичних зв’язків між словами.

У цьому розділі описано шляхи розв'язання проблеми бінарної класифікації вхідних пар слів на омографи та неомографи. Для цього були протестовані наступні метрики для визначення орфографічної подібності українських і польських слів:

1. Подібність Джаро (англ. Jaro similarity)
2. Відстань Левенштейна (англ. Levenshtein distance)
3. Індекс Тверськи (англ. Tversky index)
4. Метод зіставлення шаблонів (англ. Gestalt pattern matching)
5. Коефіцієнт узгодження (англ. Simple matching coefficient)
6. Коефіцієнт Дайса (англ. Dice’s coefficient)
7. Коефіцієнт перетину (англ. Overlap coefficient)

Застосування вищеописаних метрик для виявлення потенційних пар омографів може бути здійснено трьома способами в мові програмування Python:

1. Вбудована бібліотека *difflib*. У ній міститься клас SequenceMatcher, який є абстракцією для застосування методу зіставлення шаблонів.

```
from difflib import SequenceMatcher

def similar(a: str, b: str): -> float
```

```
    return SequenceMatcher(None, a, b).ratio()
```

2. Сторонні бібліотеки (англ. third-party libraries). На відміну від *difflib*, такі модулі не є частиною стандартного пакета мови програмування Python, тому потребують додаткового встановлення, наприклад, через консольну команду *pip install [назва бібліотеки]*. Прикладами таких бібліотек для обрахування орфографічної подібності слів можуть слугувати *jellyfish* та *nltk*.

    - Модуль *jellyfish* [35] містить імплементації для багатьох метрик, зокрема, відстань Левенштейна, подібність Джаро, відстань Геммінга тощо. Для використання цієї бібліотеки варто її викликати та вибрати необхідну метрику, що представлена методом бібліотеки, тобто викликається за наступним синтаксисом:

    ```
    import jellyfish

    jellyfish.levenshtein_distance(u'Чернігів', u'Чернівці')
    ```

    - Відома бібліотека для обробки природної мови *nltk* [40] містить у собі підмодуль *metrics* з деякими імплементованими в ньому метриками, як-от відстань Левенштейна:

    ```
    from nltk.metrics.distance  import edit_distance

    edit_distance("Чернігів", "Чернівці")
    ```

3. Написання коду власноруч для метрик, імплементація яких відсутня в бібліотеках Python. Приклад такої функції для коефіцієнта перетину:

```
def overlap_coefficient(word1: str, word2: str) -> float:
    """
    Функція для обрахування подібності слів за коефіцієнтом перетину.
      Приймає на вхід два об'єкти типу string, які є українським та
польським словом,
       для яких буде обраховано показник подібності за метрикою
"коефіцієнт перетину".

    Args:
```

```
        word1 (str): українська слово;
        word2 (str): польське слово.

    Returns:
        float: обрахований коефіцієнт подібності для поданих слів.
    """

    set1 = set(word1)
    set2 = set(word2)

    intersection_size = len(set1.intersection(set2))

    min_size = min(len(set1), len(set2))

    if min_size == 0:
        return 0

    return intersection_size / min_size
```

Загалом усі метрики показали хороші показники виявлення омографів у створеному нами датасеті; середня точність класифікатора становить 89.7%.

Таблиця 2.2.1.1. Зведена таблиця точності метрик орфографічної подібності [32]

Метрика	Точність на початкових вхідних даних	Точність на даних після обробки
Метод зіставлення шаблонів	0.92	0.95
Подібність Джаро	0.92	0.94
Індекс Тверськи	0.9	0.91
Коефіцієнт узгодження	0.9	0.91
Коефіцієнт Дайса	0.9	0.91

Відстань Левенштейна	0.87	0.89
Коефіцієнт перетину	0.87	0.89

Детально буде розглянуто два алгоритми з найвищою точністю – подібність Джаро і метод зіставлення шаблонів.

Подібність Джаро (англ. Jaro Similarity) – це метрика, що застосовується в статистиці та комп'ютерних науках для визначення подібності між двома послідовностями символів, в NLP – словами чи їх сукупністю (фрази, речення) [28] Значення такої метрики може коливатися в межах 0 і 1, де 0 означає, що слова не мають жодної орфографічної подібності, а 1 – що слова ідентичні.

Цей статистичний показник обраховується за наступною формулою:

$$Jaro\ similarity = \begin{cases} 0, \text{ if m=0} \\ \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right), \text{ for m!=0} \end{cases},$$

Формула 2.2.1.1. Подібність Джаро

У формулі 2.2.1.1:

- m – кількість символів, що збігаються;
- t – половина кількості транспозицій;
- |s1| та |s2| – довжини першого та другого слова відповідно [37].

Наприклад, обрахуймо подібність між словами «скеля» (s1) та «келія» (s2) за формулою Джаро враховуючи наступні показники:

1) |s1| = |s2| = 5;
2) m = 4, оскільки в обох словах є літери *к, е, л, я;*
3) t=2, оскільки число символів, що мають різний порядок у двох словах становить 4 (тільки літера *я* в обох випадках стоїть останньою), а кількість транспозицій – це половина від цього значення.

Отже, показник подібності Джаро для слів «скеля» та «келія» становить:

$$1/3 * (4/5 + 4/5 + 4\text{-}2/4) = 0.7 = 70\%$$

Тепер розгляньмо детально метод зіставлення шаблонів (англ. Gestalt pattern matching). Ця метрика була розроблена в 1983 році американськими програмістами Джоном Ратклiффом та Джоном Обершелпом для покращення програмного забезпечення, що використовується в освітній царині [27]. Як і подібність Джаро, значення цієї метрики може коливатися в межах [0,1], де 1 вказує на те, що аналізовані слова – тотожні, а 0 – що слова не мають спільних підмножин (англ. substrings).

Метрика обраховується за формулою:

$$D_{ro} = \frac{2K_m}{|S_1| + |S_2|}$$

Формула 2.2.1.2. Метод зіставлення шаблонів

У формулі 2.2.1.2:

- |S1| - довжина першого слова (в літерах);
- |S2| - довжина другого слова;
- 2Km - кількість збігів між двома словами помножена на два.

Використовуючи цю формулу обрахуймо орфографічну подібність між тими самими словами (S1=«скеля», S2=«келія») за наступним алгоритмом:

1) |S1| + |S2| = 5 + 5 = 10;
2) Km = |«КЕЛ»| + |«Я»| = 3 + 1 = 4;
3) (2 * 4) / 10 = 0.8.

Отже, показник методу зіставлення шаблонів для слів «келія» та «скеля» складає 0.8 або ж 80%, що на 10% більше ніж показник подібності Джаро для цих же слів.

Для бінарної класифікації вхідних слів на омографи та неомографи було додано нову колонку в наш датасет, що містить тільки два значення – 0 (омографи) та 1 (неомографи). Одиницею маркувалися об’єднані в одну

надкатегорію когнати та хибні друзі (загалом 438 пар слів), нулем – всі інші пари орфографічно непов'язаних слів (також 438). Таким чином, наш датасет є ідеально збалансованим, тому що кожному класу притаманна однакова кількість даних.

Оскільки українська та польська мови використовують різні абетки, нам необхідно нормалізувати слова, тобто звести їх до однієї системи письма, якою був обраний стандартний латинський скрипт (англійський алфавіт). Для такої транслітерації була використана бібліотека *Polyglot* [33] що підтримує транслітерацію для 69 мов зокрема, й української. Для використання цієї бібліотеки для транслітерації українських і польських слів англійською латиницею необхідно встановити пакети для цих трьох мов за допомогою наступної команди:

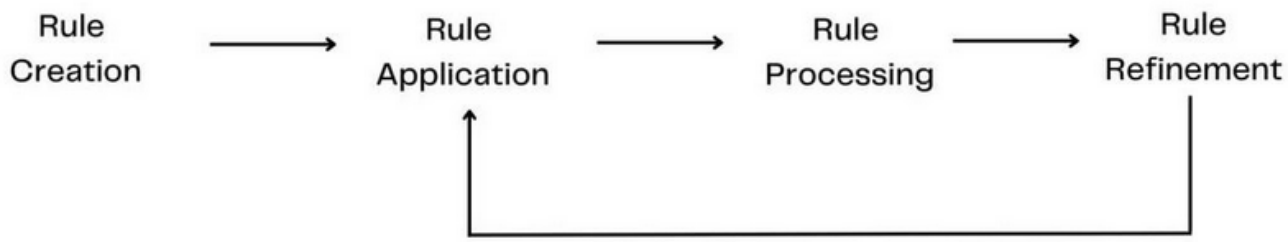
```
polyglot download transliteration2.uk
```

Аби завантажити пакети для обробки польської та англійської мов треба замінити атрибут *uk* на *pl* та *en* відповідно.

Після зведення слів до однієї системи письма було перевірено класифікатор на раніше створеному датасеті. Для кожної метрики було встановлено порогове значення, за якого класифікатор найкраще розрізняє омографи та неомографи. Для подібності Джаро та методу зіставлення шаблонів такий поріг становить 0.6 та 0.4 відповідно, оскільки при таких значеннях загальна точність класифікатора є найвищою – 0.92 або 92% для обох метрик. Отже, пари слів, що мають показник подібності Джаро вищий за порогове значення 0.6, класифікуються як омографи, натомість ті, що мають подібність Джаро 0.59 або менше визначаються як неомографи. Так само метод зіставлення шаблонів визначає слова зі значенням метрики понад 0.4 як омографи, а 0.39 і нижче – неомографи.

Для покращення точності метрик було написано та ітеративно протестовано правила для подальшої уніфікації українських і польських слів. Такий підхід в NLP називається rule-based, себто спосіб обробки текстових

даних, що спирається на лінгвістичні правила та закономірності. Ілюстрація нижче демонструє сутність цього класичного підходу:



Мал. 2.2.1.1. Rule-based підхід

Правила, що покращили роботу класифікатора, охоплюють наступні морфологічні та фонетичні особливості й закономірності української та польської мов:

1. Українські суфікси *-ок, -ук, -ік* відповідають найчастіше одному польському суфіксу *-ek:*

Українське слово	**Польське слово**
понеділок	poniedzialek
чоловік	czlowiek
чубчик	czubek

2. Українська повноголоса буквосполука *-ере-* відповідає польській неповноголосій *-rze-*:

Українське слово	**Польське слово**
дерево	drzewo
перепрошую	przepraszam
вересень	wrzesien

3. Українському префіксу *від-* завжди відповідає польський *-od:*

Українське слово	**Польське слово**
відповідати	odpowiadać
відвідувати	odwiedzać
відділ	oddział

4. Український дієслівний суфікс *-ува-* відповідає польському *-owa-*:

Українське слово	**Польське слово**
купувати	kupować
подорожувати	podróżować
готувати	gotować

Також при транслітерації українських і польських слів звук *ш* передається як через буквосполуку *sz*, так і *sh*. Так само і звук ч, який може передаватися двома шляхами: через *cz* і *ch*. Було вирішено позбутися дублетів та лишити по одній формі – *sh* та *ch*.

Уніфікація українських і польських слів відповідно до описаних правил, що охоплюють перш за все морфологічні риси українських і польських слів, а також виправлення помилок автоматичної транслітерації, дозволила підвищити точність класифікатора загалом на 1.7% (середня точність усіх метрик до нормалізації становила 89.7%, після - 91.4%). Точність методу зіставлення шаблонів підвищилася з 92% до 95%, а точність метрики подібність Джаро – з 92% до 94%.

Загалом було створено більше правил, однак деякі з них не зіграли помітної ролі в покращенні точності класифікатора, а деякі навпаки – погіршили. Такі правила охоплювали наступні особливості української та польської мов:

1. Польська префіксальна буквосполука *-je* часто відповідає українському *-о*:

Українське слово	**Польське слово**
осінь	jesień
один	jeden
одинадцять	jedenaście

2. Українська буквосполука *-не-* зазвичай співвідноситься з палаталізованою польською *-nie-*:

Українське слово	**Польське слово**
неділя	niedziela
невідомий	niewiadomy
понеділок	poniedziałek

3. На письмі українська літера *д* зазвичай відповідає палаталізованій польській буквосполуці *dz*:

Українське слово	**Польське слово**
день	dzień
тиждень	tydzień
грудень	grudzień

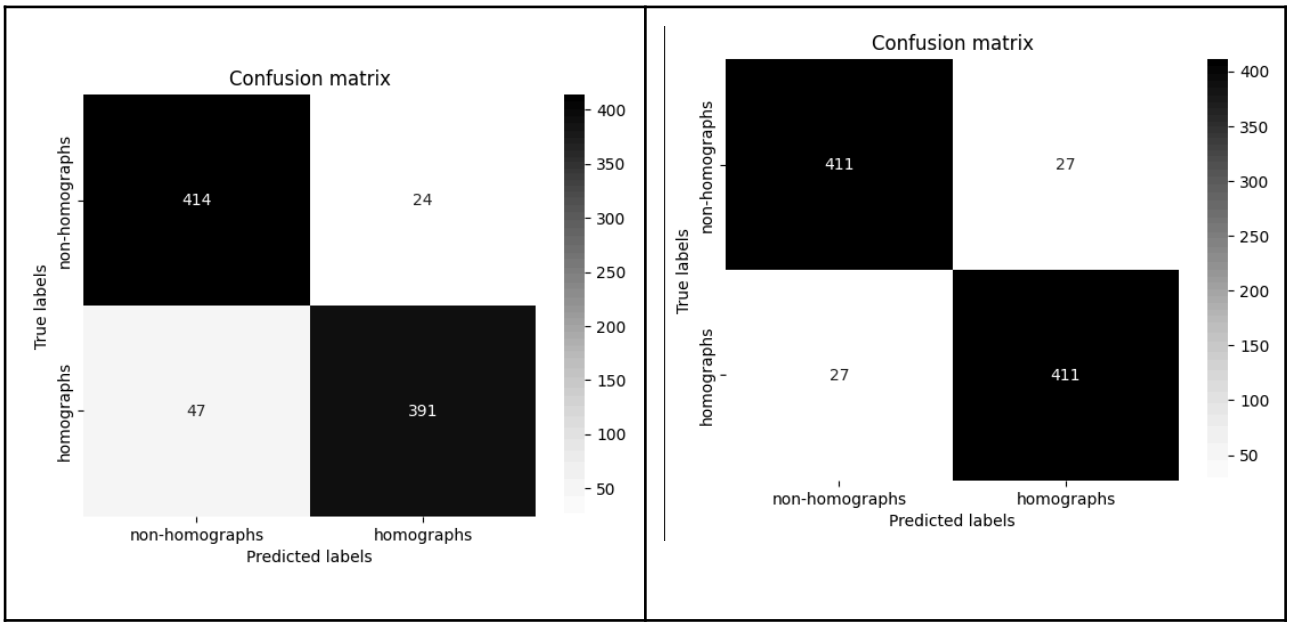
4. Українські дієслова в інфінітиві мають суфікс *-ти*, а польські - *ć*:

Українське слово	**Польське слово**
купувати	kupować
робити	robić
літати	latać

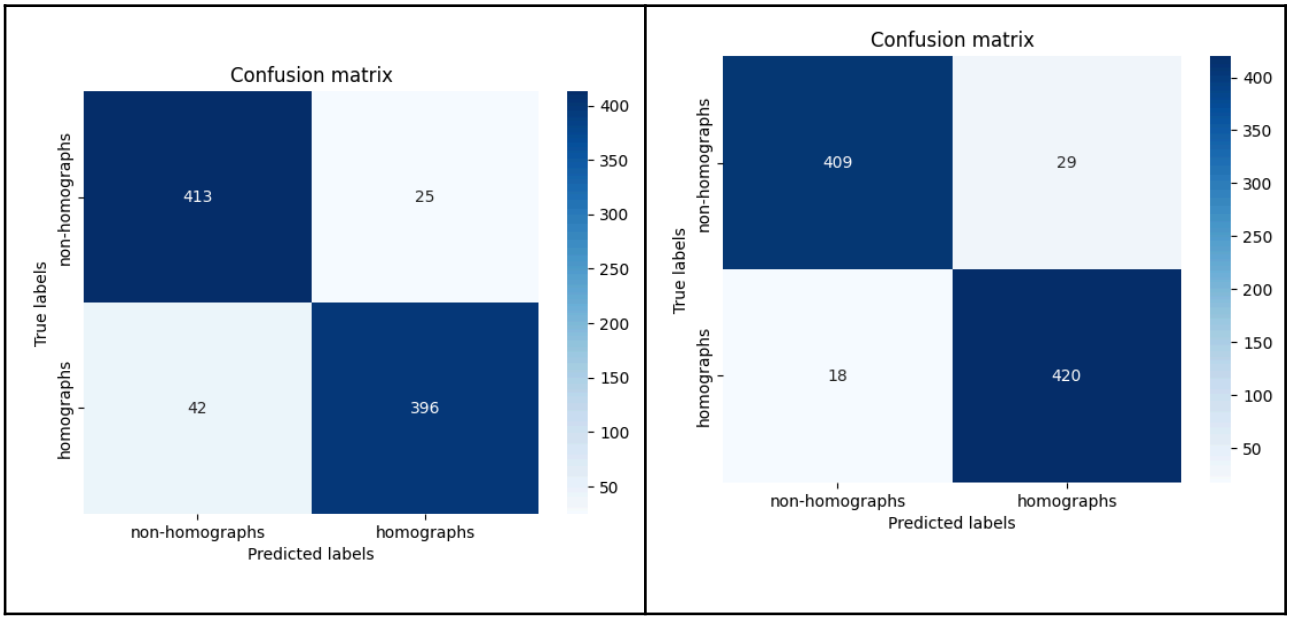
2.2.2. Аналіз результатів роботи першого класифікатора

Для опису отриманих результатів роботи першого класифікатора було використано бібліотеку класичного машинного навчання *Scikit-learn*, а саме такі її методи: матриця невідповідностей та звіт про класифікацію [19].

Матриця невідповідностей (англ. confusion matrix) – це інструмент, який використовується в машинному навчанні та статистиці для оцінки ефективності алгоритму класифікації. Така матриця подає візуальний звіт про число хибно істинно позитивних (англ. true positives), істинно негативних (англ. true negatives), хибно позитивних (англ. false positives), хибно негативних (англ. false negatives) випадків класифікації [21, 26].



Мал. 2.2.2.1. Матриці невідповідностей для подібності Джаро. Зліва – на необроблених даних, справ – на покращених за допомогою правил.



Мал. 2.2.2.2. Матриці невідповідностей для методу зіставлення шаблонів. Зліва – на необроблених даних, справ – на покращених за допомогою правил.

Звіт про класифікацію дозволяє нам зрозуміти, наскільки точним в цілому є класифікатор, а також статистику класифікації окремих класів (0 і 1). Результат роботи програмного коду для даних, покращених правилами можна побачити на ілюстраціях 2.2.2.3. і 2.2.2.4.

	precision	recall	f1-score	support
0	0.94	0.94	0.94	438
1	0.94	0.94	0.94	438
accuracy			0.94	876
macro avg	0.94	0.94	0.94	876
weighted avg	0.94	0.94	0.94	876

Мал. 2.2.2.3. Звіт про класифікацію для подібності Джаро

	precision	recall	f1-score	support
0	0.96	0.93	0.95	438
1	0.94	0.96	0.95	438
accuracy			0.95	876
macro avg	0.95	0.95	0.95	876
weighted avg	0.95	0.95	0.95	876

Мал. 2.2.2.4. Звіт про класифікацію для методу зіставлення шаблонів

Звіт містить інформацію про точність, повноту, оцінку f1 і загальну точність моделі [19].

1) Точність (англ. precision) відображає відсоток об'єктів, які були класифіковані як позитивні та дійсно є позитивними;
2) Повнота (англ. recall) відображає відсоток об'єктів позитивного класу, які були правильно визначені класифікатором з усіх можливих об'єктів позитивного класу;
3) Оцінка F1 (англ. F1-score) є середнім гармонійним між точністю та повнотою, що забезпечує баланс між цими двома метриками.
4) Загальна точність (англ. accuracy) вказує на відсоток правильно класифікованих об'єктів серед усіх об'єктів, що є показником загальної ефективності методу.

При застосуванні подібності Джаро як метрики для виявлення омографів класифікатор має точність 0.94, тобто коли він класифікує пару слів як омографи, він це робить правильно в 94% випадків. Повнота становить 0.89, що означає, що модель правильно класифікує 89% пар омографів. Для неомографів вищеописані характеристики дорівнюють 0.9 та 0.95 відповідно. Загальна точність класифікатора становить 92%. Після покращення транслітерації слів точність підвищилася до 94%

При застосуванні методу зіставлення шаблонів класифікатор демонструє подібні показники до подібності Джаро, але після імплементації правил його загальна точність підвищилася до 95%, що на один відсоток більше за попередню метрику.

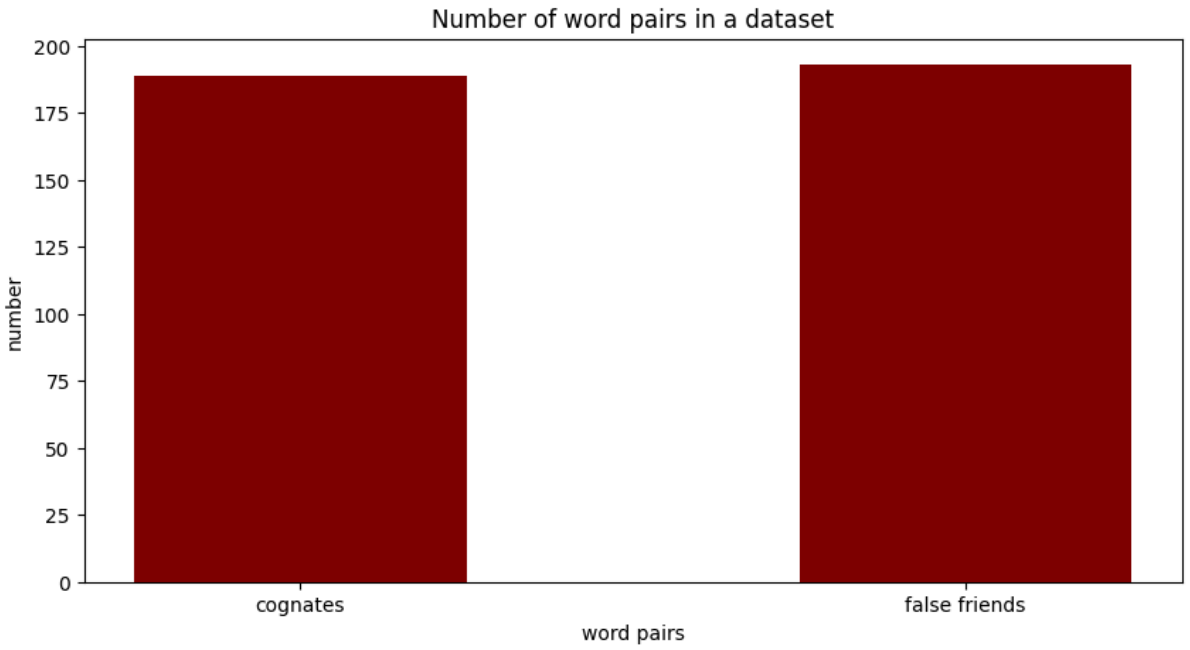
2.3. Автоматичне виявлення хибних друзів перекладача

2.3.1. Набір даних

Унаслідок роботи попереднього класифікатора було прокласифіковано вхідні пари слів з нашого датасету на омографи та неомографи, яким були присвоєні лейбли 1 та 0 відповідно. Ці результати було автоматично

вивантажено в наш датасет в новостворену колонку «predictions». Для тестування нового класифікатора ми будемо використовувати тільки ті рядки, де predictions=1, тобто рядки з омографами, які можуть бути або когнатами або хибними друзями. Отже, перед нами знову постає задача бінарної класифікації.

Співвідношення між когнатами й хибними друзями в датасеті, який ми будемо використовувати при тестуванні нашого алгоритму, зображене на малюнку 2.3.1.1.



Мал. 2.3.1.1. Дистрибуція пар слів за класами в датасеті

Як бачимо, кількість когнатів і фальшивих друзів у датасеті майже однакова, таким чином, наші дані є добре збалансованими.

2.3.2. Специфіка класифікатора

Для автоматичної класифікації когнатів і хибних друзів ми послуговувалися багатомовними векторами, що використовують сингулярний розклад (англ. singular-value decomposition, SVD) для лінійного перетворення матриці, що вирівнює одномовні вектори для двох різних мов у єдиному векторному просторі [34, 13] Єдиний двомовний векторний простір для розв'язання поставленої задачі є необхідними, оскільки схожі слова в межах

однієї мови хоч і мають подібні вектори, вектор певного слова, перекладеного іншою мовою, не матиме зв’язку зі словом у мові-джерелі і його значення буде сильно відрізнятися. Як наслідок, порівняння таких векторів слів двох різних мов не матиме сенсу, оскільки між ними немає зв’язку.

Щоб упевнитися в цьому, можемо порівняти вектори слів двох мов в окремих одномовних просторах і спільному двомовному просторі за допомогою косинусної подібності.

Проаналізуймо вектори українського слова «кіт» та польського «kot» в одномовних векторних просторах:

```
uk_vector = uk_dictionary["кіт"]

pl_vector = pl_dictionary["kot"]

print(FastVector.cosine_similarity(uk_vector, pl_vector))
```

Результат роботи коду:

```
0.09863913206925096
```

Значення косинусної подібності може коливатися в межах від -1 до 1, тому такий результат можемо оцінити як нейтральний, оскільки він наближений до 0 і нам явно не каже, чи є ці слова семантично подібними.

Тепер порівняймо вектори цих слів у єдиному багатомовному векторному просторі:

```
uk_dictionary.apply_transform('uk.txt')

pl_dictionary.apply_transform('pl.txt')

print(FastVector.cosine_similarity(uk_dictionary["кіт"], pl_dictionary["kot"]))
```

Результат роботи коду:

```
0.48332476480896835
```

Бачимо, що в другому випадку ми отримали значно вище значення, ніж у першому, що є свідченням вищої ефективності використання багатомовних векторних просторів при роботі з декількома мовами.

Для нашого класифікатора ми використовували попередньо натреновані вектори *fasttext* [50].

Таблиця 2.3.2.1. Зведена таблиця точності метрик семантичної подібності

Метрика	Точність
Косинусна подібність	0.82
Евклідова відстань	0.78
Манхеттенська відстань	0.78
Точковий добуток векторів	0.72
Відстань Джаккарда	0.79
Кореляція Пірсона	0.81
Відстань Хаммінга	0.79

Подібність між векторами українських і польських слів визначалася за допомогою різних метрик, зазначених у таблиці 2.3.2.1. Найкращі показники виявлення хибних друзів перекладача були отримані при обчисленні подібності векторів з використанням косинусної подібності, математична сутність якої полягає в тому, що якщо відстань між векторами мала (близька до 0 градусів), то це означає, що ці вектори мають високий рівень подібності та навпаки (чим більше кут – тим більше слова є семантично віддаленими) [30]. Подібно до попереднього класифікатора ми встановили поріг значення косинусної подібності, при якому модель найкраще класифікує когнати та хибні друзі перекладача. Такий поріг становить 0.5; при такому значенні загальна точність нашої моделі становить 82%. Отже, пари слів, вектори яких мають косинусну подібність 0.5 або вище, класифікуються як когнати, і їм присвоюється лейбл 0. Натомість лексичні пари, що мають косинусну подібність нижчу за 0.49, класифікуються як хибні друзі перекладача, і їм присвоюється лейбл 1.

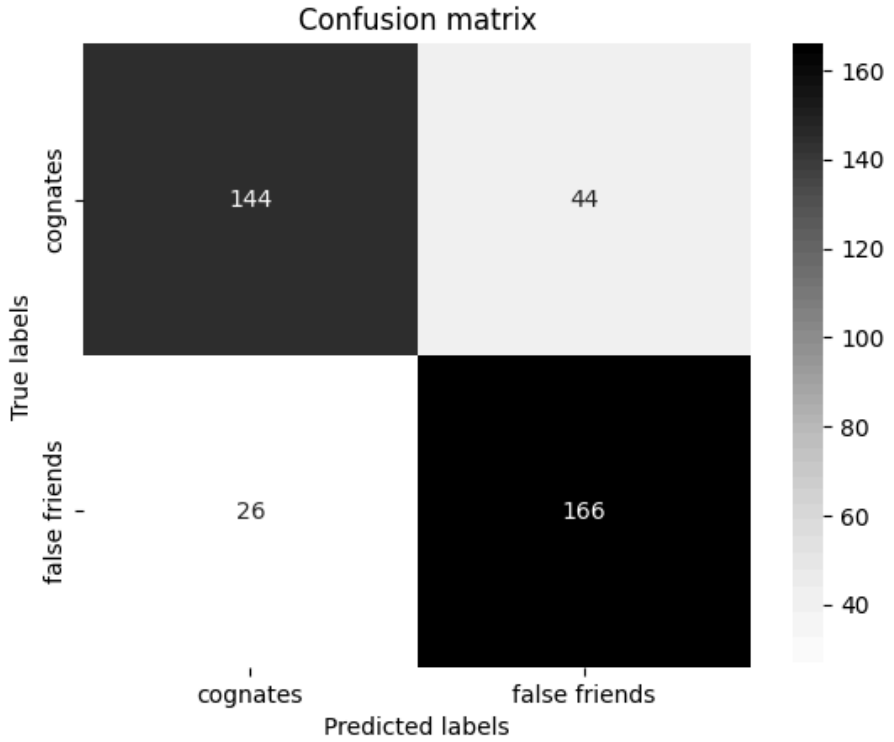
2.3.3. Аналіз роботи другого класифікатора та моделі в цілому

Результати роботи класифікатора було отримано аналогічним способом до описаного в розділі «2.2.2. Аналіз результатів роботи першого класифікатора»

	precision	recall	f1-score	support
0	0.85	0.77	0.81	189
1	0.79	0.87	0.83	193
accuracy			0.82	382
macro avg	0.82	0.82	0.82	382
weighted avg	0.82	0.82	0.82	382

Мал. 2.3.3.1. Звіт про класифікацію

Реалізований нами другий класифікатор має точність 0.79, тобто коли він класифікує пару слів як хибні друзі, він це робить правильно в 79% випадків. Повнота складає 0.87, що означає, що модель правильно класифікує 87% усіх фальшивих друзів. Для когнатів вищезазначені характеристики дорівнюють 0.85 і 0.77 відповідно. Загальна точність класифікатора становить 82%, тобто загалом було прокласифіковано правильно 82% усіх вхідних пар слів.



Мал. 2.3.3.2. Матриця невідповідностей

Оскільки наша трикласова модель, що розрізняє хибні друзі, когнати та непов’язані пари слів, є двоступеневою, тобто складається з двох класифікаторів, окремі загальні точності яких становлять 82% і 95%, її загальна точність становить 78%.

2.4. Створення демонстраційного вебзастосунку для автоматичного виявлення хибних друзів з використанням мовних моделей

2.4.1. Великі мовні моделі та N-shot learning

Великі мовні моделі наразі відіграють надважливу роль не лише в NLP, а й програмуванні та штучному інтелекті загалом. Вони є досить універсальними та можуть використовуватися фактично для будь-якої NLP задачі. Для деяких завдань достатньо звичайного налаштування запитів до мовної моделі (англ. prompt engineering), однак часто, якщо точність мовної моделі, наприклад, GPT-4, Claude 3 Opus чи Llama 2 для виконання певної задачі занизька, може стати в пригоді файнтюнінг моделі (англ. fine-tuning).

Справжній бум розвитку мовних моделей спричинила публікація статті у 2017 році під назвою «Attention is all you need», що запропонувала трансформерну архітектуру нейромереж, яка значно краще почала справлятися з генерацією даних порівняно з популярними на той час згортковими (англ. Convolutional neural networks – CNN) та рекурентними (англ. Recurrent neural networks – RNN) нейромережами [14]. Пізніше новим, не менш потужним поштовхом до розвитку мовних моделей і генеративного штучного інтелекту став реліз ChatGPT, що базується на мовних моделях сім’ї GPT, оскільки тоді мовні моделі стали доступними кожному.

Наразі в машинному навчанні провідне становище займає кероване навчання, яке може бути ефективним лише за наявності великої кількості розмічених даних, що часто є обмеженням. Одним зі шляхів уникнення необхідності збирати та розмічати великий датасет в NLP є використання великих мовних моделей, зокрема, техніки N-shot learning.

N-shot learning (NSL) – техніка в NLP, підвид трансферного навчання (англ. transfer learning), що полягає в передачі у вигляді запиту (англ. prompt) мовній моделі опису задачі, а також певної кількості прикладів успішного її виконання. N-shot learning підходить найкраще для розв'язання задач з класифікації, як у нашому випадку. Параметр *N* у назві техніки означає кількість прикладів, що надаються моделі [36]. Залежно від цього показника, N-shot learning поділяють на три типи:

1. Zero-shot (ZSL) – варіація NSL, при використанні якої немає потреби в передачі прикладів у мовну модель. У такому випадку покладаються лише на знання моделі, здобуті при її початковому тренуванні.
2. One-shot (OSL) – підвид NSL, що дозволяє моделі вчитися на одному прикладі даних.
3. Few-shot (FSL) або Low-shot learning – варіація NSL, що полягає в донавчанні моделі на декількох прикладах (зазвичай до 10).

2.4.2. Створення вебзастосунку

Для наочності нашого дослідження було створено вебзастосунок для демонстрації алгоритму виявлення хибних друзів перекладача. Ідея застосунку полягає в тому, що користувач має можливість завантажити два тексти, українською та польською мовами, і програма поверне об’єкт формату JSON зі знайденими в текстах фальшивими друзями з контекстами, в яких ті зустрічаються. У користувача є можливість обрати, чи має містити остаточний об’єкт JSON крім міжмовних омонімів ще когнати. Також користувач може обрати один з двох форматів виводу результатів виявлення хибних друзів:

1) Сирий об’єкт JSON, даними в якому можна легко маніпулювати і який також можна завантажити у вигляді файлу із розширенням .json;
2) Зручний для читання формат, що відображає хибні друзі та за потреби когнати з відповідними контекстами.

Обробка вхідних текстових файлів відбувається в класі FileManager, в конструкторі якого ініціалізуються необхідні інструменти для роботи з

лінгвістичними даними, в нашому випадку – текстами, написаними української та польською мовами. Загалом, ми ініціалізуємо три змінні:

```
def __init__(self, api_key: str):
    self.uk_nlp = spacy.load('uk_core_news_sm')
    self.pl_nlp = spacy.load('pl_core_news_sm')
    self.client = OpenAI(
        api_key=api_key
    )
```

- Змінні self.uk_nlp та self.pl_nlp підвантажують при ініціалізації класу мовні моделі бібліотеки spacy [45] для української та польської мов відповідно. Моделі типу *_core_news_sm – це найменші за розміром моделі spacy, однак їхнього функціонала та точності на цьому етапі нам достатньо. Ми використовуватимемо лише токенізацію, лематизацію, а також вбудовану перевірку на стоп-слова та пунктуацію. Такі моделі не є вбудованими в бібліотеці spacy, тому потребують додаткового встановлення:
- За допомогою іншої змінної (self.client) ми підключаємо API мовних моделей GPT для роботи з наданими користувачем лінгвістичними даними. OpenAI – клас бібліотеки openai [41] яка надає доступ до OpenAI API для програм, написаних мовою програмування Python версії 3.7+. Цей клас приймає в нашому випадку один параметр – api_key (API ключ для взаємодії з мовними моделями GPT через Python код), який необхідно згенерувати на офіційному сайті компанії.

  У класі FileManager містяться такі методи обробки вхідних текстів:

- **jaro_distance**

  Метод, що містить імплементацію подібності Джаро для двох слів. Приймає на вхід два об’єкти типу string (українське та польське слово), повертає значення float у діапазоні від 0 до 1, де 1 означає, що слова тотожні.

- **process_text**

  Цей метод приймає на вхід український та польський текст, які були завантажені користувачем. На цьому етапі відбувається обробка тексту за допомогою spacy та її моделей, а саме токенізація, лематизація та перевірка на наявність стоп-слів. Якщо вхідне слово є стоп-словом – воно ігнорується.

- **create_candidate_pairs**

  Приймає на вхід словник українських і польських лем, де ключ – лема слова, а значення - словоформа. На цьому етапі відбувається уніфікація українських польських слів, себто зведення до спільної абетки (стандартна латинка), потім за допомогою попередньо визначеної метрики орфографічної подібності визначаються омографи, які надалі будуть прокласифіковані як хибні друзі перекладача, або як когнати. Також на цьому етапі ми зберігаємо для кожного слова контекст, який дорівнює одному реченню, в якому те чи інше слово зустрічається. Функція повертає список подібних слів і контексти

- **find_false_friends**

  Останній метод, які підключає мовну модель для класифікації хибних друзів перекладача і когнатів серед визначених попередньо орфографічно подібних слів. Приймає пари слів і також контексти. Повертає словником усю зібрану інформацію про слова. Цей метод містить також системний промпт, що передає OpenAI API інформацію про наше завдання, а також пари-кандидати для класифікації.

Сам вебзастосунок було створено за допомогою srtreamlit. Streamlit – фреймворк з відкритим вихідним кодом, створений мовою програмування Python, і насамперед використовується спеціалістами та аматорами у сфері машинного навчання та штучного інтелекту, тому має розширений функціонал для опрацювання та візуалізації даних [46]. Бібліотека поєднує в собі як бекенд, так і фронтенд концепти, надаючи зручний Python API для побудови вебзастосунків без необхідності заглиблення в концепти веброзробки.

Бібліотека надає можливості створювати вебзастосунки за допомогою звичних GUI елементів, таких, як:

- кнопки (streamlit.button);
- текстові поля (streamlit.text_area);
- бокове меню (streamlit);
- таблиця (streamlit.table).

Як зазначалося, в streamlit є безліч утиліт для роботи з даними, зокрема, в ньому є підтримка типів DataFrame, JSON, а також різних метрик.

Для запуску застосунку необхідно виконати в терміналі наступні команди:

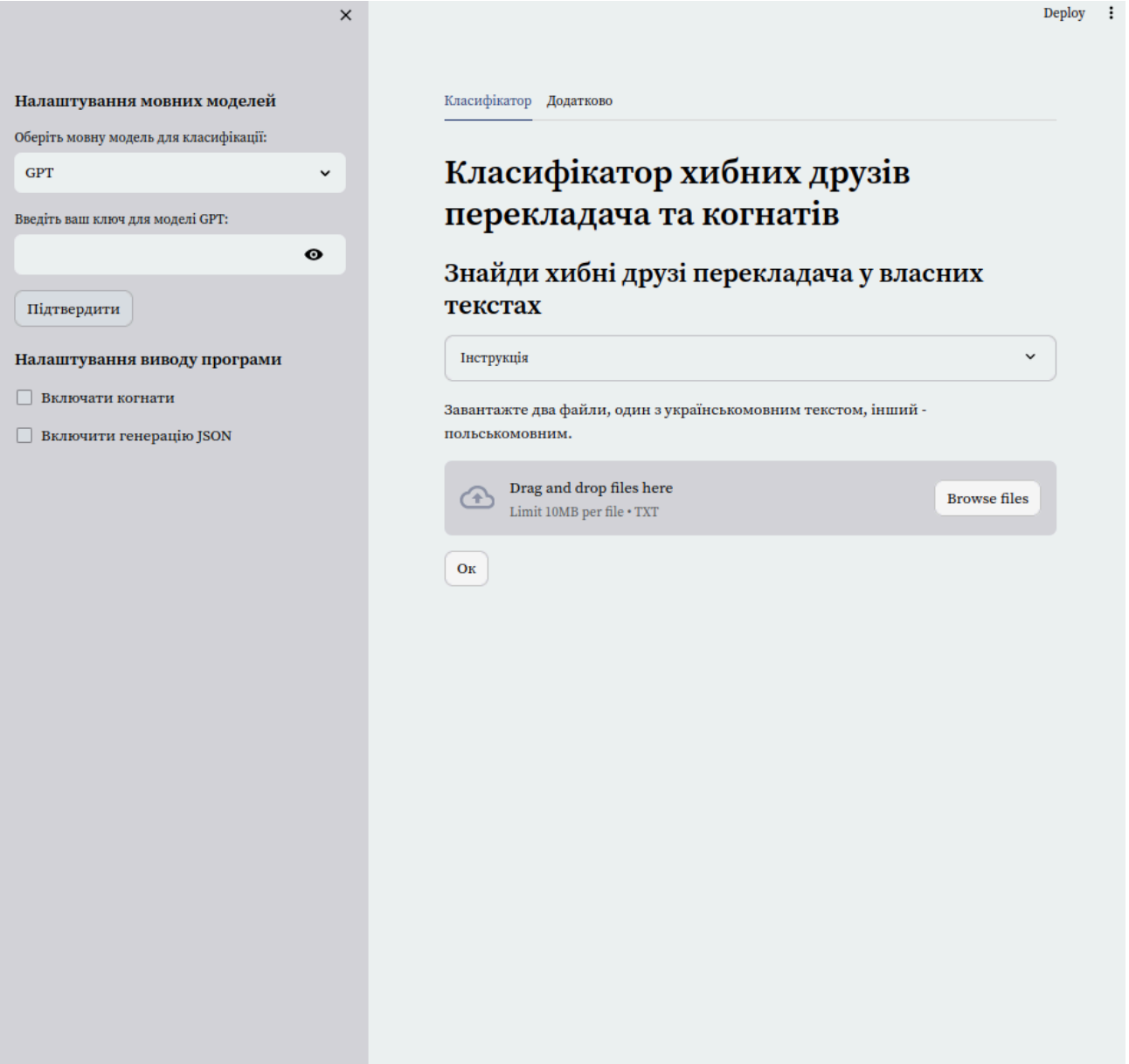
```
$ python3 -m venv false-friends

$ source false-friends/bin/activate

$ pip install -r requirements.txt

$ streamlit run streamlit_app/app.py
```

Перша команда створює віртуальне середовище venv, що дозволяє уникнути конфліктів вже встановлених локально пакетів з модулями, які необхідно встановити для нашої програми. Друга команда активує створене середовище, а третя – встановлює бібліотеки та моделі попередньо записані у файлі requirements.txt. Четверта команда власне виконує основний код нашої streamlit програми, що містить у файлі app.py, та запускає вебзастосунок, з яким тепер можна взаємодіяти у браузері. Графічний інтерфейс вебзастосунку подано на малюнку 2.4.2.1.

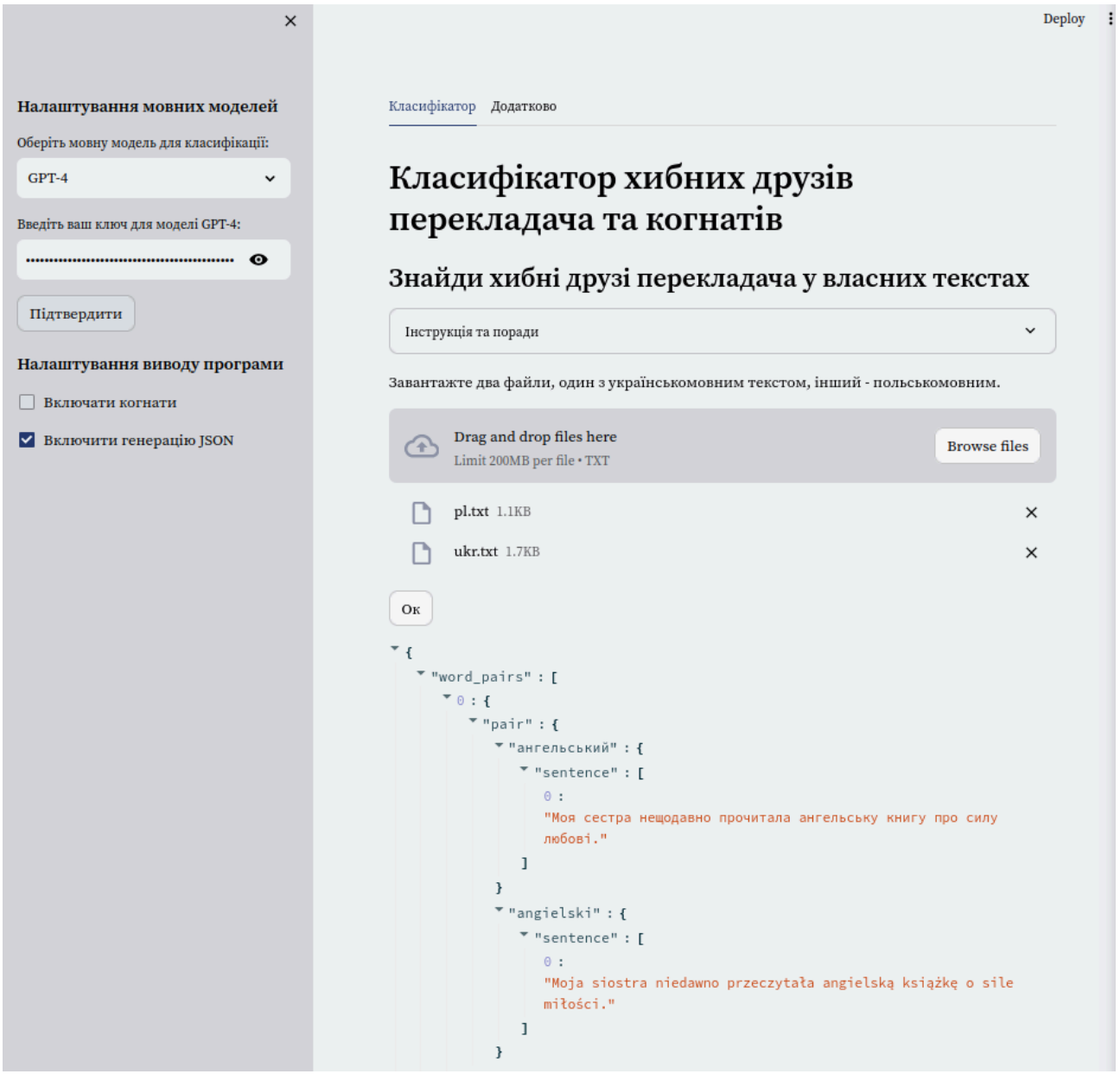


Мал. 2.4.2.1. Головна сторінка створеного вебзастосунку

Для його використання початково потрібно обрати мовну модель для zero-shot класифікації хибних друзів перекладача та когнатів, а також внести API ключ для обраної мовної моделі. Наразі імплементовано лише одну модель – GPT-4. Після натиснення кнопки «Підтвердити» в боковому меню, ключ буде передано в конструктор класу FileManager для виконання API запитів до мовної моделі.

Тепер користувачу слід завантажити два текстові файли – один українською мовою, інший – польською та натиснути кнопку «Ок», після чого почнеться обробка завантажених користувачем текстових файлів, виявлення

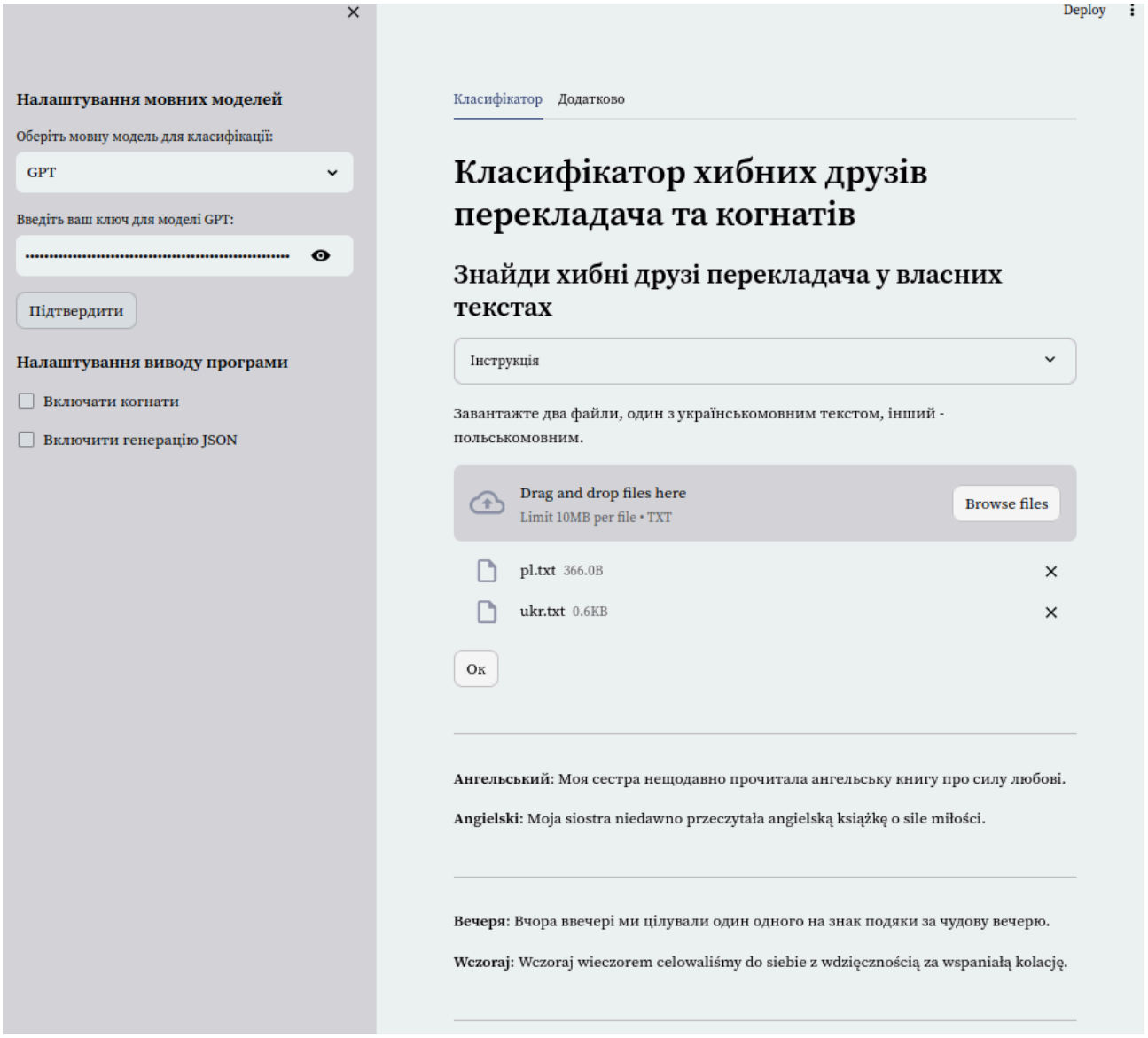
орфографічно подібних слів та власне класифікація їх на когнати та міжмовні омоніми. Після виконання всіх кроків головна сторінка вебзастосунку матиме наступний вигляд:



Мал. 2.4.2.2. Вигляд вебзастосунку після завантаження текстових файлів та класифікації хибних друзів і когнатів.

Вивід визначених пар хибних друзів перекладача у форматі JSON можна відключити, прибравши прапорець у боковому меню, і тоді користучач отримає просто список міжмовних омонімів (і когнатів, якщо вибрати відповідний

прапорецт у боковому меню) у зручному для читання форматі, як зображено на малюнку 2.4.2.3.



Мал. 2.4.2.3. Вигляд вебзастосунку при відключенні виводу даних у форматі JSON

Лінгвістичні дані про українську-польську мовну пару збираються гібридним шляхом: за допомогою подібності Джаро визначаються орфографічно подібні пари-кандидати, які можуть бути або когнатами, або хибними друзями; а використання мовних моделей відбувається класифікація хибних друзів перекладача і когнатів, а за допомогою інструментів бібліотеки spacy проводиться базова обробка текстів (лематизація, токенізація, фільтрація стоп-слів).

Приклад згенерованого JSON об’єкта для пари слів, яка була правильно визначена мовною моделлю GPT-4 як хибні друзі перекладача:

```
{
  "ангельський": {
        "sentence": [
        "Моя сестра нещодавно прочитала ангельську книгу про силу любові."
        ]
  },
  "angielski": {
        "sentence": [
        "Moja siostra niedawno przeczytała angielską książkę o sile miłości."
        ]
  }
}
```

Створений нами вебзастосунок було розміщено на хмарі (Streamlit Cloud), тож тепер кожен може його протестувати за посиланням: https://ua-pl-false-friends-detector.streamlit.app/

Висновки до розділу 2

У цьому розділі було розглянуто процес створення системи автоматичного виявлення хибних друзів перекладача для української та польської мов. Цей розділ охоплює етапи створення тестувальної вибірки, тестування методів визначення орфографічної та семантичної подібності слів, аналіз роботи класифікатора, а також опис архітектуру створеного вебзастосунку, що надає можливість користувачу знайти пари хибних друзів у власних текстових файлах.

Перший підрозділ присвячений створенню навчально-тестувальної вибірки. Для цього було зібрано пари українських і польських слів,

використовуючи технологію вебскрейпінгу. Вибірка включає три категорії слів: когнати, хибні друзі перекладача та непов'язані слова. Останній клас було створено шляхом перемішування у вибірці когнатів та хибних друзів відносно одне одного. Вибірка є невеликою та містить усього 876 записів, але цього достатньо для тестування нашого алгоритму на цьому етапі. Надалі датасет можна буде розширити використовуючи онлайн ресурси та словники.

Далі ми описуємо класифікатор для виявлення омографічно подібних пар слів. Було протестовано декілька метрик, зокрема, подібність Джаро, відстань Левенштейна, індекс Тверськи. Результати показали, що деякі з цих метрик є більш ефективними для розрізнення омографів та неомографів, наприклад, подібність Джаро демонструє точність 94%, а метод зіставлення шаблонів – 95%.

Третій підрозділ зосереджений на автоматичному виявленні хибних друзів перекладача серед визначних на попередньому етапі пар-кандидатів омографів. Було розроблено алгоритм, що використовує векторні репрезентації слів для визначення семантичної подібності. Векторні моделі, такі як word2vec, дозволяють визначати відношення між словами на основі їхнього контексту в корпусі текстів, що значно покращує точність класифікації. Подібність векторів word2vec обраховувалася так само за допомогою декількох метрик, найкращою з яких виявилася косинусна подібність, яка правильно визначила 82% хибних друзів. Таким чином, загальна точність алгоритму, що класифікує вхідні пари слів на когнати, хибні друзі та непов’язані пари слів, становить 78%.

У заключному підрозділі розглянуто можливості великих мовних моделей для задачі класифікації фальшивих друзів і когнатів. Було створено демонстраційний вебзастосунок, що дозволяє користувачу завантажити власні тексти – українською та польською мовами – і як результат програма поверне віднайдені пари хибних друзів з контекстами, в яких їх вжито.

Таким чином, ми використали здобуті теоретичні знання в напрямку міжмовної омонімії для створення алгоритму виявлення хибних друзів перекладача для української та польської мов. Сучасні методи виявлення

семантично подібних слів на міжмовному рівні, зокрема, векторні репрезентації слів, демонструють високу точність у розв’язанні поставленої задачі, а мовні моделі пропонують спрощений механізм класифікації завдяки методу N-shot learning, який не потребує великої кількості тренувальних даних та часу на тренування моделі.

ВИСНОВКИ

Здійснене дослідження ґрунтується на проблемі автоматичного виявлення хибних друзів перекладача для української та польської мов. Для розв’язання цієї задачі ми поєднуємо тривіальні методи та метрики, зокрема, для визначення орфографічної та семантичної подібності вхідних пар слів, із сучасними, такими, як мовні моделі та векторні репрезентації слів.

У першій частині роботи було розглянуто, в чому полягає актуальність дослідження міжмовних омонімів, а також імовірні застосування системи автоматичного виявлення хибних друзів перекладача. Ми проаналізували термінологію, що використовується в такого роду дослідженнях, зокрема, поняття когнатів, хибних друзів перекладача, часткових когнатів тощо, і визначили поняття, якими ми оперуватимемо в нашій роботі. Було визначено, що міжмовна омонімія охоплює лексико-семантичний рівень мови, де слова у двох або більше мовах мають схоже написання, але різне значення. Таким чином, ми відрізняємо їх від когнатів, що мають спільне етимологічне походження, схоже написання та значення. Було проведено короткий контрастивний аналіз лексичних систем польської та української мов в контексті слов’янської підгрупи індоєвропейської мовної сім’ї, а також ми розглянули та проаналізували причини утворення міжмовних омонімів у цих мовах. Останнім завданням на цьому етапі був огляд можливих підходів для розв'язання питання автоматичного виявлення хибних друзів перекладача. Методи охоплюють аналіз орфографічних і семантичних характеристик слів, використання векторних репрезентацій слів, а також мовних моделей.

Другий розділ охоплює практичні аспекти задачі автоматичного виявлення фальшивих друзів, а саме укладання тестувальної вибірки з українсько-польськими лексичними парами, створення та тестування алгоритму виявлення міжмовних омонімів, а також розробку демонстраційного вебзастосунку, що дозволяє користувачеві завантажити власні тексти, у яких будуть знайдені пари когнатів і хибних друзів. Новостворений тестувальний датасет містить 876 лексичних польсько-українських пар, згрупованих за

трьома категоріями – когнати, хибні друзі, непов’язані пари. На цьому наборі даних було протестовано наш алгоритм за допомогою 14 метрик: 7 для визначення орфографічної подібності, і 7 – семантичної. Початкова точність класифікатора для виявлення орфографічно подібних слів становила 92% для двох найкращих метрик – подібність Джаро та метод зіставлення шаблонів. Після написання та застосування правил для уніфікації українських і польських лексичних пар на основі виокремлених попередньо морфологічних і фонетичних особливостей цих мов метод зіставлення шаблонів показав точність 95%, а метрика подібності Джаро – 94%, натомість для виявлення семантичної подібності найкращі результати показала міра косинусної подібності – 82%. Загальна точність алгоритму виявлення хибних друзів перекладача склала 78%. На фінальному етапі ми створили вебзастосунок, що використовує метрику подібності Джаро для виявлення орфографічно подібних пар слів у двох вхідних текстах – українською та польською мовами. Після цього за допомогою мовної моделі GPT-4 відбувається класифікація визначених за допомогою попередньої метрики пар слів на хибні друзі перекладача та когнати. Результат виводиться або у форматі JSON, або звичайним списком, який є більш зручним для інтерпретації пересічним користувачем.

Подальші дослідження в цьому напрямку можуть охоплювати тестування інших метрик або покращення вже наявних, наприклад, завдяки доопрацюванню правил уніфікації українських і польських слів, або створенню якісного алгоритму українсько-польської транслітерації. Також враховуючи стрімкий розвиток мовних моделей пріоритетним є їхнє використання, зокрема, тестування нових, таких як Claude, Llama, Mistral. Крім того, варто доповнити створений нами датасет і оцінити квантитативно точність роботи інших мовних моделей з ним. Корисним і цікавим рішенням було б виокремлення нових лексичних категорій для класифкації, наприклад, часткових когнатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Бурлака О. ПРОБЛЕМА ПЕРЕКЛАДУ „ХИБНИХ ДРУЗІВ ПЕРЕКЛАДАЧА» У НАУКОВО-ПОПУЛЯРНІЙ ЛІТЕРАТУРІ. ЄВРОПЕЙСЬКІ МОВИ-2022: ІННОВАЦІЇ ТА РОЗВИТОК : Зб. наук. робіт, м. Дніпро. Дніпро, 2022. С. 12–14.
2) Кличлієв К.С. Автоматичне виявлення хибних друзів перекладача для української та польської мов // Матер. X Міжнар. наук.-техн. Internet-конф. «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 24 листоп. 2023 р. – К.: НУХТ, 2023. – С. 94–95.
3) Кононенко І. Українська та польська мови: контрастивне дослідження / Ірина Кононенко. – Warszawa: Wydawnictwa Uniwersytetu Warszawskiego, 2012. – 809 с. – (1).
4) Кононенко І., Співак О. Українсько-польський словник міжмовних омонімів. Київ, 2008. 348 с.
5) Лексична міжмовна омонімія у слов’янських мовах / А. А. Турчина // Компаративні дослідження слов’янських мов і літератур. Пам’яті академіка Леоніда Булаховського. - 2012. - Вип. 17. - С. 121-126.
6) Angryk R. Measuring semantic similarity using WordNet-based Context Vectors / R. Angryk, S. Wan, 2007.
7) Castro S. A High Coverage Method for Automatic False Friends Detection for Spanish and Portuguese / S. Castro, J. Bonanata, A. Rosá // Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018) / S. Castro, J. Bonanata, A. Rosá. – Santa Fe: Association for Computational Linguistics, 2018. – С. 29–36.
8) Classification of Slavic languages: evolution of developmental models. Slavia Occidentalis 77, 2020, 33-64.
9) Crystal D. Dictionary of Linguistics and Phonetics. Wiley & Sons, Incorporated, John, 2009. 560 p.

10) Frunza O. M. Automatic Identification of Cognates, False Friends, and Partial Cognates. Ottawa, 2006. 137 p.
11) L. Savytska, N. Vnukova, I. Bezugla, V. Pyvovarov, M. Sübay, Using Word2vec Technique toDetermine Semantic and Morphologic Similarity in Embedded Words of the Ukrainian Language,in: Proceefings of the International Conference on Computational Linguistics and IntelligentSystems, 2021, pp. 235–248.
12) Methods for extracting and classifying pairs of cognates and false friends / R.Mitkov, V. Pekar, D. Blagoev, A. Mulloni, 2008.
13) Offline bilingual word vectors, orthogonal transformations and the inverted softmax Samuel L. Smith, David H. P. Turban, Steven Hamblin and Nils Y. HammerlaICLR 2017 (conference track)
14) Vaswani, Ashish, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. «Attention is All you Need.» Neural Information Processing Systems (2017).

ЕЛЕКТРОННІ ДЖЕРЕЛА

15) Відсотки спільності основної лексики. *Діаріуш або тиск слова*. URL: https://maksymus.wordpress.com/2007/02/22/48232/ (дата звернення: 12.06.2024).
16) Костянтин Тищенко. Прадавність української мови, віддзеркалена в мовах сусідів. 1. *Мовознавство: науково-теоретичний журнал Інституту мовознавства імені О. О. Потебні НАН України та Українського мовно-інформаційного фонду НАН України*. URL: https://movoznavstvo.org.ua/vsi-nomera-zhurnalu/statti/2023-1-sichen-liutyi/kostiantyn-tyshchenko-pradavnist-ukrainskoi-movy-viddzerkalena-v-movakh-susidiv-1 (дата звернення: 14.06.2024).
17) Учасники проектів Вікімедіа. Фальшиві друзі перекладача – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Фальшиві_друзі_перекладача (дата звернення: 12.06.2024).

18) Чуба Г. МІЖМОВНА ОМОНІМІЯ ЯК ДЖЕРЕЛО ПОМИЛОК ПРИ ВИВЧЕННІ УКРАЇНСЬКОЇ МОВИ СТУДЕНТАМИ-ПОЛЯКАМИ. *Repozytorium Uniwersytetu Jagiellońskiego(RUJ) :: Home*. URL: https://ruj.uj.edu.pl/server/api/core/bitstreams/7e80e464-bc67-44d8-b119-ee7391b47cdd/content (дата звернення: 14.06.2024).

19) 3.4. Metrics and scoring: quantifying the quality of predictions. *scikit-learn*. URL: https://scikit-learn.org/stable/modules/model_evaluation.html (date of access: 12.06.2024).

20) Beautiful Soup Documentation – Beautiful Soup 4.12.0 documentation. *Swear not by the wiki, the fickle wiki, the inconstant wiki*. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (date of access: 12.06.2024).

21) B H. N. Confusion matrix, accuracy, precision, recall, F1 score. *Medium*. URL: https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd (date of access: 12.06.2024).

22) Binary Classification NLP - Best simple and efficient model. *Inside Machine Learning*. URL: https://inside-machinelearning.com/en/a-simple-and-efficient-model-for-binary-classification-in-nlp/ (date of access: 12.06.2024).

23) Browne W., Ivanov V. V. Slavic languages | list, definition, origin, map, tree, history, & number of speakers. *Encyclopedia Britannica*. URL: https://www.britannica.com/topic/Slavic-languages (date of access: 12.06.2024).

24) Cognate - Wiktionary, the free dictionary. *Wiktionary*. URL: https://en.wiktionary.org/wiki/cognate (date of access: 12.06.2024).

25) Complex Networks / ed. by L. da F. Costa et al. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011. URL: https://doi.org/10.1007/978-3-642-25501-4 (date of access: 14.06.2024).

26) Contributors to Wikimedia projects. Confusion matrix - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Confusion_matrix (date of access: 12.06.2024).
27) Contributors to Wikimedia projects. Gestalt pattern matching - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Gestalt_pattern_matching (date of access: 12.06.2024).
28) Contributors to Wikimedia projects. Jaro–Winkler distance - wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Jaro–Winkler_distance (date of access: 12.06.2024).
29) Contributors to Wikimedia projects. Lexical similarity - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Lexical_similarity (date of access: 12.06.2024).
30) Cosine Similarity - GeeksforGeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/cosine-similarity/ (date of access: 12.06.2024).
31) FastText/docs/unsupervised-tutorials.md at master · facebookresearch/fastText. *GitHub*. URL: https://github.com/facebookresearch/fastText/blob/master/docs/unsupervised-tutorials.md (date of access: 12.06.2024).
32) Find the similarity metric between two strings. *Stack Overflow*. URL: https://stackoverflow.com/questions/17388213/find-the-similarity-metric-between-two-strings (date of access: 12.06.2024).
33) GitHub - aboSamoor/polyglot: Multilingual text (NLP) processing toolkit. *GitHub*. URL: https://github.com/aboSamoor/polyglot (date of access: 12.06.2024).
34) GitHub - babylonhealth/fastText_multilingual: multilingual word vectors in 78 languages. *GitHub*. URL: https://github.com/babylonhealth/fastText_multilingual (date of access: 12.06.2024).

35) GitHub - jamesturk/jellyfish: 🪼 a python library for doing approximate and phonetic matching of strings. *GitHub*. URL: https://github.com/jamesturk/jellyfish (date of access: 12.06.2024).

36) Gopalani P. Zero shot, Few shot, One shot Learning in NLP. *Medium*. URL: https://prachi-gopalani.medium.com/zero-shot-few-shot-one-shot-learning-in-nlp-341aa684cdb2 (date of access: 12.06.2024).

37) Jaro and Jaro-Winkler similarity - GeeksforGeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/jaro-and-jaro-winkler-similarity (date of access: 12.06.2024).

38) Measuring Language Distance of Isolated European Languages. *MDPI*. URL: https://www.mdpi.com/2078-2489/11/4/181 (date of access: 12.06.2024)

39) Multiclass classification vs multi-label classification - geeksforgeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/multiclass-classification-vs-multi-label-classification/ (date of access: 12.06.2024).

40) NLTK :: nltk.metrics.distance module. *NLTK :: Natural Language Toolkit*. URL: https://www.nltk.org/api/nltk.metrics.distance.html (date of access: 12.06.2024).

41) Openai. *PyPI*. URL: https://pypi.org/project/openai/ (date of access: 12.06.2024).

42) Pandas - python data analysis library. *pandas - Python Data Analysis Library*. URL: https://pandas.pydata.org/ (date of access: 12.06.2024).

43) Perez M. What is web scraping and what is it used for? | parsehub. *Web Scraping Blog (Tips, Guides + Tutorials) | ParseHub*. URL: https://www.parsehub.com/blog/what-is-web-scraping/ (date of access: 12.06.2024).

44) Requests: HTTP for Humans™ – Requests 2.32.3 documentation. *Requests: HTTP for Humans™ – Requests 2.32.3 documentation*. URL: https://requests.readthedocs.io/en/latest/ (date of access: 12.06.2024).

45) SpaCy · industrial-strength natural language processing in python. *spaCy · Industrial-strength Natural Language Processing in Python*. URL: https://spacy.io/ (date of access: 12.06.2024).

46) Streamlit docs. *Streamlit documentation*. URL: https://docs.streamlit.io/ (date of access: 12.06.2024).

47) Turing. Word embeddings in NLP: a complete guide. *AI-Powered Engineering Services, LLM Training, Teams | Turing*. URL: https://www.turing.com/kb/guide-on-word-embeddings-in-nlp (date of access: 12.06.2024).

48) W3Schools.com. *W3Schools Online Web Tutorials*. URL: https://www.w3schools.com/tags/tag_ul.asp (date of access: 12.06.2024).

49) What is a machine learning pipeline? | IBM. *IBM - United States*. URL: https://www.ibm.com/topics/machine-learning-pipeline (date of access: 12.06.2024).

50) Wiki word vectors · fastText. *fastText*. URL: https://fasttext.cc/docs/en/pretrained-vectors.html (date of access: 12.06.2024)

ДОДАТКИ

Додаток А. Програмне забезпечення

Посилання на вебзастосунок:

https://ua-pl-false-friends-detector.streamlit.app/

Посилання на GitHub проєкту:

https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages

Структура проєкту:

- alignment_matrices – директорій з матрицями для вирівнювання векторів українських (uk.txt) і польських (pl.txt) слів.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/tree/main/alignment_matrices
- datasets – директорій зі створеними та використаними в ході дослідження датасетами.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/tree/main/datasets
- false_friends_detection – директорій з блокнотами, що містять код для виявлення семантичної подібності векторів, а також класифікації хибних друзів і когнатів.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/tree/main/false_friends_detection
- homographs_classification – директорій з блокнотами, що містять код для класифікації омографів і неомографів.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/tree/main/homographs_classification

- streamlit_app – репозиторій з код вебзастосунку.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/tree/main/streamlit_app
- requirements.txt – використані у цьому проєкті бібліотеки.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/blob/main/requirements.txt
- texts – директорій з текстами-прикладами для тестування вебзастосунку.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/tree/main/texts
- fast_vector.py – код для створення багатомовного векторного простору.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/blob/main/fast_vector.py
- alignment_matrices – директорій з матрицями для вирівнювання векторів українських (uk.txt) і польських (pl.txt) слів.

  Посилання:

  https://github.com/klychliiev/Automatic_false_friends_detection_for_Ukrainian_and_Polish_languages/tree/main/alignment_matrices