

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет української й іноземної філології та журналістики
Кафедра англійської філології та прикладної лінгвістики

**РІВЕНЬ АДЕКВАТНОСТІ МАШИННОГО ПЕРЕКЛАДУ (НА
МАТЕРІАЛІ ПЕРЕКЛАДІВ ВИКОНАНИХ СИСТЕМАМИ
GOOGLE TRANSLATE, PROMT ТА MICROSOFT TRANSLATE)**

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти «магістр»

Виконав: здобувач Матвеев Сергій
Миколайович
Спеціальності: 035.041 Філологія
(германські мови та літератури (переклад
включно), перша – англійська)
Освітньо-професійної програми:
«Філологія (германські мови та літератури
(переклад включно)), перша англійська»
Керівниця: к. філол. н., доц. Хан
Олена Георгіївна
Рецензентка: стейкхолдер ОПП,
начальник центру
міжнародного співробітництва
Херсонської Торгово-промислової палати
Пономаренко Лариса Валентинівна

Херсон – 2020

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Теоретико-методологічне підґрунтя машинного перекладу	7
1.1 Машинний переклад у сучасному суспільстві.....	7
1.2 Класифікація систем машинного перекладу.....	10
1.3 Існуючі системи машинного перекладу.....	16
1.4 Методологія дослідження рівню адекватності машинного перекладу.....	21
РОЗДІЛ 2. Засоби визначення адекватності машинного перекладу	23
2.1 Спектр та специфіка застосування систем машинного перекладу в роботі перекладача.....	23
2.2 Типові помилки систем машинного перекладу.....	35
2.3 Перспективи розвитку систем машинного перекладу.....	43
ВИСНОВКИ	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТКИ	59
Додаток А.....	59
Додаток Б.....	67
Додаток В.....	76

ВСТУП

Системи машинного перекладу полегшують роботу перекладачів, надаючи їм списки термінів, пам'яті перекладів та результати машинного перекладу. В ідеалі такі інструменти автоматично передбачають, чи необхідно буде докладати зусиль для подальшого редагування вихідного тексту, ніж перекладати з нуля, та визначають, чи надавати перекладачам результати машинного перекладу чи ні. Однак, такі системи не враховують досвід перекладача, незважаючи на те що процеси перекладу для початківців відрізняються від процесів перекладу професійних перекладачів. Існує достатньо широкий систем машинного перекладу з допомогою яких можна полегшити чимало трудомісткої роботи. Системи машинного перекладу враховують морфологічні, синтаксичні та семантичні зв'язки членів речення та надають зв'язний переклад текстів. За останні роки якість перекладу помітно покращилась, в результаті чого, сьогодні системи машинного перекладу здатні надати більш вдалий варіант перекладу. Однак слід зазначити, що системи машинного перекладу все ще не можуть стовідсотково взяти до уваги всі граматичні аспекти тієї чи іншої мови, саме тому головним призначенням таких систем є переклад ділових паперів, листів з електронної пошти або сторінок з Інтернету.

Дану тему можна вважати **актуальною**, тому як на сучасному етапі системи машинного перекладу знаходяться у вільному доступі і використовуються всіма верствами населення, особливо якщо брати до уваги все глибший рівень інтеграції України в міжнародне співтовариство. В результаті чого Україна стикається не тільки з бар'єрами економічного та політичного характеру, а й мовними бар'єрами також. Не покращує ситуацію те що не так багато професійних перекладачів, в результаті чого їх послуги вкрай не дешеві. Більш того,

системи машинного перекладу також можуть слугувати допомогою професійним перекладачам.

Зв'язок роботи з науковими програмами, планами й темами.

Дослідження виконано в рамках комплексної наукової теми кафедри англійської філології та прикладної лінгвістики «Вплив лінгвальних та екстралінгвальних чинників на формування фахівця з іноземних мов у сучасному мультикультурному просторі» (номер державної реєстрації 0117U003763).

Матеріалом нашого дослідження є переклади, виконані за допомогою систем машинного перекладу Google Translate, PROMT, та Microsoft Translate, що досліджуються й аналізуються, з урахуванням алгоритмів роботи систем, їх граматики, словників та внутрішньої побудови.

Мета даного дослідження полягає у вивченні ступеню адекватності машинного перекладу у розгляді методів, за допомогою яких системи здійснюють переклад, а також в дослідженні проблем, з якими стикаються системи машинного перекладу.

Дослідження передбачає необхідність вирішення конкретних **завдань**:

- окреслити місце машинного перекладу в сучасному суспільстві;
- розглянути існуючі системи машинного перекладу, їх розвиток та типологію;
- проаналізувати сучасні системи машинного перекладу;
- з'ясувати яким чином системи машинного перекладу можуть бути використані в роботі перекладача;
- виявити проблеми досягнення адекватності машинного перекладу.

Теоретико-методологічну основу становлять теоретичні роботи А.С. Евдокимова, Д. М. Кузьменка, Ю. Кота, Ж. Г. Мацака, Р. Г.

Пиотровського, Е. А. Прохорова, А. И. Рейблата, Т. Смірної, А. А. Яфмутової, S. Bangalore, A. Borete, A. Burchardt, D. Farwell, M. C. Forcada, P. Haffner, S. Helmreich, W. J. Hutchins, S. Kanthak, Y. Willks, S. Zhou та інших вчених.

Предметом дослідження є рівень адекватності машинного перекладу та потенціал систем перекладу до покращення рівню адекватності.

Об'єктом дослідження є системи машинного перекладу, а також переклад здійснений ними.

Методика дослідження базується на загальнонаукових методах таких як: спостереження, опис, систематизація, індукція та дедукція, та емпірико-теоретичних методах таких як: аналіз, синтез, порівняння. Написання кваліфікаційної роботи за обраною темою вимагає залучення інструментарію лінгвістики, зокрема філологічного аналізу, методи якого були залучені при порівнянні уривків перекладу та у порівнянні перекладу уривків з оригіналом.

Наукова новизна дослідження полягає в тому, що на сьогоднішній день в нашій країні практично відсутні наукові роботи що присвячені вивченню систем машинного перекладу.

Теоретичне значення дослідження обумовлено тим, що були проаналізовані актуальність, класифікація та принцип роботи систем машинного перекладу.

Практична цінність отриманих результатів дослідження полягає в тому, що були окреслені недоліки систем машинного перекладу, в тому числі проблема перекладності окремих текстів. Крім того, результати проведеного дослідження можуть бути використані під час викладання курсів з теорії та практики перекладу.

Апробація результатів роботи. Результати дослідження були обговорені на засіданнях кафедри англійської філології та прикладної лінгвістики Херсонського державного університету, які були проведені у

формі попереднього захисту розпочатого дослідження на етапах опрацювання й узагальнення теоретичної й практичної частин роботи. Результати дослідження було представлено на міжнародній науково-практичній конференції "Розвиток філологічних наук: європейські практики та національні перспективи" 23-24 жовтня 2020р. м. Одеса.

РОЗДІЛ 1

ТЕОРЕТИКО-МЕТОДОЛОГІЧНЕ ПІДРУНТЯ МАШИННОГО ПЕРЕКЛАДУ

1.1 Машинний переклад у сучасному суспільстві

У ХХІ столітті машинний переклад являє собою особливий та ефективний засіб міжмовної комунікації у всіх галузях сучасної науки і техніки. Однак, за словами дослідника Т. Смірної, появу машинного перекладу можна і потрібно вважати активним помічником, а не конкурентом у перекладацькій діяльності [11, с. 1].

Сьогодні існує потреба у вирішенні проблеми швидкого та значного за обсягом перекладу, так як потреба в перекладі як виді інформаційної діяльності щорічно збільшується. Т. Смірнова вважає що, машинний переклад можна розглядати як альтернативу традиційному. Однак, не варто забувати про комунікативну еквівалентність тексту перекладу [11, с. 1].

За словами дослідника Ж. Г. Мацак, комунікативна еквівалентність нового тексту по відношенню до оригіналу забезпечується виконанням трьох основних вимог:

- вихідний текст повинен передавати зміст оригіналу в повному його обсязі;
- вихідний текст повинен відповідати нормам мови перекладу;
- вихідний текст повинен, за своїм об'ємом бути подібним до тексту оригіналу [5, с. 2].

Вказані вимоги, однак, дуже рідко вдається виконати застосовуючи машинний переклад. Причиною цього є недосконалість систем машинного перекладу. За словами дослідника П. М. Хвощ, наступним кроком є редагування вихідного тексту, при цьому важливо мати хоча б базове знання іноземної мови та добре орієнтуючись у предметній галузі [12, с. 82].

В останні роки можна побачити покращення якості машинного перекладу. Проте все одно існують проблеми з граматичними та синтаксичними зв'язками у реченні, з відтворенням термінології, вважає дослідник Д. М. Кузьменко. Беручи до уваги образність та стилістичне навантаження художньої літератури, саме автоматичний переклад таких текстів є неможливим. Найчастіше машинний переклад застосовують при перекладі, наприклад, документації. [4, с. 107].

Т. Смірнова зазначає що робота над покращенням комп'ютерного перекладу все ще ведеться багатьма науковцями, яка зумовлена практичною необхідністю у ньому. Більш того, у сучасному суспільстві спостерігається тенденція до збільшення кількості обміну інформацією [10, с. 1].

За словами дослідника А. С. Евдокимова, підраховано, що тільки з одного галузевого науково-дослідницького інституту для забезпечення доступу його співробітників до іншомовної спеціальної літератури потрібен понад 20 професійних перекладача, що цілодобово працювали б тільки над перекладами. Інформація з 2009 року свідчить що понад 1,5 млн. сторінок було перекладено системами машинного перекладу за один рік. Це всього лише 1% від усіх перекладних у світі текстів, що свідчить про те що системи машинного перекладу, тільки набирають обертів. [2, с. 86].

Говорячи про машинний переклад, слід пам'ятати, що комп'ютер, в повній мірі, невзмі розуміти мовних аспектів у текст або гри слів. Дослідник Р. Г. Пиотровський зазначив що, мислення, в нашому розумінні, не відбувається при машинному перекладі: речення ділиться на частини, в ньому виділяються конструкції, слова і словосполучення перекладаються за словниками, що знаходяться в базі даних системи перекладу, де пізніше всі частини будуть зібрані воедино в речення за правилами вихідної мови [8, с. 28].

Дослідик Е. А. Прохрова підкреслює що, комп'ютер поки багато в чому не може замінити перекладача. Однак, використання комп'ютеру для перекладу літературних текстів видає підрядний переклад, який можна відредагувати навіть якщо людина погано володіє мовою оригіналу, але вміє редагувати текст. Слід зазначити що при редагуванні перекладу технічних текстів важливо правильно вибрати словник за сферою діяльності. Загалом потреба в редагуванні комп'ютерного перекладу виникає тільки у зв'язку з проблемами що були перераховані вище. Саме для цього системи машинного перекладу мають у собі засоби редагування текстів [9, с. 94].

Т. Смірнова зазначає що, аби досягти високої якості перекладу необхідно щоб практично всі слова тексту оригіналу легко було знайти і в словнику системи. Слова в словнику відсутні будуть неперекладені, або перекладені невірно, і будуть потребувати редагуванню. Такі слова можуть вплинути на якість перекладу речень. Річ полягає у тому, що для визначення, до якої частини мови певного слова, система проводить аналіз в цілому всього речення. Навіть якщо значення одного слова в реченні не визначене, це може зруйнувати аналіз всього речення, а іноді й результати всього перекладу [11, с. 2].

За словами Т. Смірної, результати машинного перекладу, таким чином, часто вимагають редагування. Наскільки адекватними можна вважати результати перекладу на комп'ютері визначається не тільки якістю системи машинного перекладу, але і якістю подальшого редагування. Найчастіше систему машинного перекладу використовує як допомогу фахівець, якому необхідно швидко перекласти, наприклад, технічну документацію. В такому випадку проблема коректного вживання термінів вирішується сама собою [11, с. 2].

Т. Смірнова підкреслює що, на сьогоднішній день спостерігається збільшення інтересу до систем машинного перекладу у зв'язку з швидким розвитком мережі Інтернет. Мільйони людей по всьому світові говорять

різними мовами, і всі вони опинилися в єдиному інформаційному просторі. Звичайно ж, домінує в Мережі англійська мова, але існує багато користувачів, що нею не володіють, паралельно з безліччю Web-сторінок, написаних не на англійській мові [11, с. 2].

Дослідник А. І. Рейтблат зазначає що, з'явилися доповнення до браузерів задля полегшення перегляду сторінок написаних на незнайомій користувачеві мові. Вони здійснюють негайний переклад вибраних користувачем фрагментів сторінки, що переглядається користувачем. Необхідно лише виділити частину тексту і перенести її на спеціальну панель або натиснути на спеціальну кнопку меню. Прикладом такого доповнення є фірма PROMT що може бути підключена до браузеру. [10, с. 82].

Отже, системи машинного перекладу можуть надавати цілий ряд переваг серед яких: практичність, гнучкість, багатофункціональність та інші. Однак, якість перекладу залежить суттєво залежить від складності тексту, особливо при перекладі науково-технічної документації або художньої літератури.

Існує ряд типових помилок, що виникають у ході машинного перекладу таких як: невірна побудова речення, невірний підбір лексики та термінології.

Машинний переклад, на данному етапі представляє собою помічника в роботі перекладача оптимізуючи процес перекладу, обробляючи великі обсяги інформації. Однак, результати машинного перекладу все одно підлягають редагуванню для того щоб запобігти появі помилок у кінцевому тексті перекладу.

1.2 Класифікація систем машинного перекладу

На сьогоднішній день дослідження видів систем машинного перекладу набуває особливого значення, про що свідчить поява цілого ряду праць, присвячених вивченню цієї проблеми [26; 27; 32; 49].

Дослідник А. Burchardt виділяє наступні основні шляхи до побудови алгоритмів машинного перекладу:

- 1) заснований на правилах (rule-based);
- 2) заснований на прикладах (example-based);
- 3) заснований на статистиці (statistical-based).

Кожен з перелічених шляхів має свої переваги та недоліки, однак, на сьогоднішній день переважає гібридний (hybrid) підхід в межах якого розробники систем машинного перекладу хочуть максимально об'єднати переваги усіх перелічених підходів [18; с. 1].

Тим не менш, коротко розгляньмо кожен із зазначених видів систем машинного перекладу. Машинний переклад на базі правил є найстарішим підходом та охоплює найрізноманітніші технології. Системи машинного перекладу, основані на правилах (Rule-Based Machine Translation – RBTM), мають в своїй основі великі збірки правил, що розробляються експертами, які фіксують структури з вихідної мови цільовою мовою. Етапи перекладу використовують базу даних правил та лексичних елементів, на які поширюються ці правила. Правила та лексичні елементи задані однозначно, але можуть бути змінені лінгвістом. Людський фактор у системах, що основані на правилах, допомагає покращити якість машинного перекладу. Проте через значну ручну роботу система що основана на правилах, може бути досить дорогою, а її впровадження та оновлення вимагає значних часових витрат. Так як нові правила постійно додаються та оновлюються, такі системи з часом можуть видавати багато неточностей та гіршу якість перекладу [57]. Науковець W. Hutchins зазначив що, системи такого типу використовують наступні рівні аналізу для здійснення перекладу:

- морфологічний аналіз;
- семантичний аналіз;
- складовий аналіз;
- аналіз залежностей.

Перенесення структури речення буде підпорядковане правилам лексичного перетворення. Правила генерації матимуть наступний вигляд:

- прикметник, виражений словосполученням, слідує за іменником — з декількома перерахованими винятками;
- визначальне слово узгоджено за кількістю та родом з іменником, який воно модифікує;
- прикметник погоджено по числу і статі з іменником, який воно модифікує;
- дієслово узгоджено з підметом [33, с. 12].

W. Hutchins визначає одним з методів перекладу у рамках перекладу на базі правил є переклад на базі трансферу. Трансферні системи використовують з вхідним текстом правила, які ставлять у відповідні структури вихідної і цільової мов. Початковий етап роботи такої системи включає в себе аналіз тексту для створення формальної внутрішньої репрезентації тексту. Переклад робиться на основі цієї схеми з використанням словників між двома мовами та використанням їх правил граматики. Внутрішнє уявлення та представлення тексту може мати декілька рівнів абстракції звертаючи увагу на важливі фрагментах тексту [33, с. 12].

Дослідник A. L. Lagarda зазначає що, один із можливих варіантів на базі правил є інтерлінгвістичний підхід. Інтерлінгвістика як окрема наукова галузь вивчає такі системи. В такому випадку текст трансформується в абстракцію оформлену у певний вигляд що не залежить від використовуваної мови. Перекладний текст створюється на основі цієї репрезентації [38, с. 217].

Нажаль, за словами S. Bangalore, на сьогоднішній день реалізацій такого типу систем, які працювали б правильно хоча б для двох мов, не існує. Багато експертів висловили сумніви в можливості їх реалізації, принаймні на даному етапі розвитку людства [15, с. 155].

Крім того, як зазначив S. Zhou, культурні відмінності країн дуже сильно заважають створенню такої системи. Оскільки для створення такого роду систем необхідним є створення міжмовного подання, яке необхідне бути одночасно абстрактним і незалежним від кожної мови, але в той же час адекватно відображати її особливості [64, с. 6].

У своєму дослідженні А. А. Яфмутова стверджує що, системи машинного перекладу, засновані на статистиці (Statistical Machine Translation – SMT) використовують комп'ютерні алгоритми для створення перекладу. Статистичні моделі складаються з слів та словосполучень, які автоматично вивчаються та запозичуються з двомовних паралельних речень, створюючи двомовну базу даних перекладів. Привабливість статистичних систем пояснюється їхнім рівнем автоматизації з використанням можливостей машинного навчання, що призводить до швидшої роботи та нижчої вартості обчислювальної потужності, необхідної для побудови та експлуатації таких систем. Однак, головним недоліком цього типу є «ефект розбавлення даних», викликаний дефіцитом даних [13, с. 574].

За словами Р. Williams, машинний переклад на базі фраз є найпопулярнішим видом статистичного машинного перекладу. Сьогодні, як і раніше, він представляє собою інструмент що використовується в великих сервісах з перекладу. Принцип роботи такої системи представляє собою розклад вихідного тексту на фрази, переклад цих фраз, в результаті якого виникає перекладний текст з фраз. Іншими словами, це тексти що містять речення обома мовами, мовою оригіналу та мовою перекладу. [62, с. 167].

Р. Williams також зазначив що, для уникнення невідповідностей переклад фраз зазвичай відбувається за аналогією з попередніми перекладами. Наприклад, переклад російського слова *вертолёт* на українську мову як *вертолёт, гвинтокрил та гелікоптер* для громадян України, які володіють двома мовами, цілком очевидний. В свою чергу,

навіть суто переклад слова в такому випадку вводить елемент невідповідності для СМП. Якщо використовуються різноманітні синтаксичні та граматичні структури проблема набуває ще більшого масштабу. Крім того, усунення «зайвих» еквівалентних мовних структур беззаперечно приведе до збіднення мови [62, с. 168].

На думку J. Hutchins, слабкою стороною систем, заснованих на правилах (RBMT) є підбір слів у трансфері, а також відсутність стійкості у випадках помилок під час аналізу речень. Однак, в цілому їх переклад більш точний, тобто без втрат вхідної інформації. Статистичні системи хоча й не гарантують якісний переклад, вони є більш надійними і завжди дозволяють отримати переклад. Крім того, вони мають труднощі у перекладі мовних феноменів, які потребують спеціальних лінгвістичних знань, таких як морфологія, синтаксис, порядок слів. Саме ці недоліки стали причиною створення і розвитку гібридних систем [34, с. 18].

А.А Яфмутова стверджує, що на сьогоднішній день з'являється все більше нових шляхів до розробки систем машинного перекладу. Передусім вони спрямовані на розширення можливостей систем статистичного машинного перекладу а також подолання їх обмежень. Нові рішення статистичних систем машинного перекладу допомагають досягти якісних покращень, які забезпечуються створенням гібридних систем машинного перекладу, крім того, зникає необхідність використання застарілих технологій, а натомість з'являються нові мультилінгвістичні комунікаційні рішення [13, с. 577].

На думку W. Younghui, особливої уваги заслуговують новації корпорації Google у згаданій галузі, зокрема їх система нейронного машинного перекладу. Нейронний машинний переклад (NMT) – це новий алгоритм машинного перекладу, що ґрунтується на штучній нейронній мережі та дає змогу подолати недоліки традиційних пофразових систем перекладу. Однак, системи нейронного перекладу є коштовними, як у розробці, так і у виконанні перекладу. Крім того, більшість систем NMT

можуть мати труднощі з словами що рідко використовуються. Такі проблеми перешкоджають широкому використанню систем NMT у випадках, де важлива як точність, так і швидкість [63. с. 246].

W. Younghui також відмічає що, системі нейронного машинного перекладу компанії Google складається з глибокої мережі LSTM з 8 кодерами та 8 декодерами, вирішено багато проблем. Зокрема тести, під час яких здійснювався переклад текстів з англійської мови на французьку та з англійської мови на німецьку, показали, що GNMT досягає цілком задовільних результатів. Провівши детальний аналіз набору перекладених системою простих речень, було визначено, що система GNMT зменшує кількість помилок при виконанні перекладу в середньому на 60 % у порівнянні до системи Google, побудованої за по-фразовим алгоритмом дії [63. с. 246].

Отже, всі системи між собою мають свої недоліки та переваги. Системи на базі правил представляють собою найбільш поширені системи машинного перекладу. Системи такого типу надають високу якість перекладу але в той же час мають високу вартість та вимагають довготривалої розробки. Статистичний машинний переклад також може надати високу якість перекладу однак вимагає відповідні словники для створення адекватного перекладу. Системи машинного перекладу на базі нейронних мереж є вкрай популярними на сьогоднішній день, оскільки з'явилась можливість наблизити модель перекладу до моделі яку використовує людина. Перевагами такої системи є висока якість перекладу, навіть у порівнянні з іншими системами перекладу, однак такі системи потребують великий обсяг текстів для навчання і дуже багато часу.

1.3 Існуючі системи машинного перекладу

PROMT Translator

Ю. А. Кот у своєму дослідженні зазначив що, PROMT це російська компанія що є розробником системи машинного перекладу PROMT Translator. Крім того, компанія провела ряд досліджень зі штучним інтелектом. Основними напрямками роботи компанії є покращення якості перекладу, а також вдосконалення технологій для подальшого розширення функціоналу системи. [3, с. 56].

Ю. А. Кот зазначив що, на ранніх етапах існування система PROMT являла собою систему перекладу що заснована на правилах. Однак, пізніше ця технологія машинного перекладу була удосконалена і отримала назву Аналітичний машинний переклад PROMT. Нова система, у порівнянні зі старою, характеризується значно вищою швидкістю роботи і надійністю [3, с. 57].

Крім того, за словами Ю. А. Кота, одна з версій PROMT Translator являє собою систему статистичного машинного перекладу. Статистична система здійснює переклад за допомогою значних за обсягом корпусів паралельних текстів, і саме тому якість перекладу особливо залежить від якості і кількості доступних даних. Однак слід зазначити що, вихідний переклад відрізняється більшою точністю, ніж переклади що були виконані аналітичними системами машинного перекладу [3, с. 58].

Ю. А. Кот виділяє цілий ряд можливостей PROMT Translate. На відміну від деяких систем машинного перекладу що можна знайти сьогодні, система PROMT здатна на переклад не тільки окремих слів а й текстів будь-якого обсягу. Існує функція перекладу тексту завантаженого файлу, підтримуються всі популярні формати такі як: doc(x), xls(x), pdf та багато інших. Система здатна також здійснити деяке редагування тексту, а також створити спеціалізований словник чи профіль перекладу. За бажанням можна встановити спеціальне доповнення до браузеру що

дозолить миттєво здійснювати переклад сайтів, цілком або виділених фрагментів [3, с. 60].

Google Translate

Система машинного перекладу Google Translate на відміну від PROMT Translate являє собою статистичну систему яка також включає в себе елементи системи перекладу за допомогою нейронних мереж.

Ю. А. Кот стверджує що, статистичний переклад є оптимальним між спорідненими мовами. Так наприклад, переклад Google з російської на українську є найбільш інтуїтивно вірним з усіх розглянутих, найближчий до кінцевого перекладу, в ньому рідше зустрічається спотворення сенсу або правил мови, а також менше дослівного перекладу. Межою для статистичного машинного перекладу є правильна «конвертація» граматичних конструкцій однієї мови засобами іншої. Таке обмеження не відсутнє навіть на базі споріднених мов і тим самим створює більше шуму в інтерпретації [3, с. 63].

J. Sommerlat зазначає що, англійська — центральна мова в Google Translate. Переклад з англійської та на англійську в Google Translate є прямим, тобто без посередництва іншої мови. Саме це і забезпечує непогану якість при якій особливо помітно потенціал роботи сервісу: частіше правильний переклад імен та назв, термінів, фразеологізмів, рідше зустрічається дослівний переклад, а натомість використовується живі мовні звороти, нерідко також зустрічається правильний вибір лексичного значення в залежності від контексту. В 2016 році Google продовжує покращувати сервіс прибираючи повтори з перекладу; система починає використовувати краще вибудовані, зв'язні речення, іноді більш вдалий вибір слів. Однак, все ще були помітні суттєві помилки при перекладі, так наприклад іноді відбувався відкат на менш вдалий переклад (*камінь інструмент* замість *кам'яне знаряддя* в 2011 році). Таким чином, машинні переклади в 2016 році були місцями краще,

місцями гірше, ніж переклади 2011 року, але в цілому рівень залишався таким самим [52].

Ю. А. Кот підкреслює що, головним досягненням в результатах перекладу системи Google Translate в 2017 році стали більш тверде та точне розпізнавання структури речення і передача граматичних значень на цільових мовах. В англійській, наприклад, мові закінчення не грають важливої ролі в передачі граматичних значень, як в українській, німецькій, польській або російській мовах. Однак, при аналізі через нейромережу граматичні зв'язки стали губитися рідше, ніж при статистичному перекладі. Крім того, стали розпізнаватися рідковживані багатокореневі слова, тобто перекладач почав краще працювати з членуванням не тільки речення, але і слова [3, с. 64].

За словами Ю.А Кота, система нейронного перекладу Google не оперує смислами. Статистичний перекладач впевнено працював з розпізнаванням термінів, імен, фраз, часто вдало підбирав значення слів в контексті речення. Однак, проблеми починалися коли не вдавалось належним чином інтерпретувати взаємозв'язок між словами та їх граматичну роль. У перекладах 2017 року помітний значний прогрес в цьому напрямку, іншими словами, нерозпізнаних місць стало менше. Нейронний перекладач було прив'язано до структури речення. Він почав досить непогано трансформувати мовні конструкції з однієї мови на іншу [3, с. 66].

Як зазначено в статті S. Shankland, Google Translate, перекладач Google може перекладати з різних мов в різних формах текст та медіа, включаючи текст, аудіо, зображення, сайти або відео в реальному часі. Крім того, сервіс підтримує понад 100 мов на різних рівнях, а станом на травень 2017 року обслуговує понад 500 мільйонів людей щодня [50].

Довідка Google стверджує що, для деяких мов програма Google Translate може вимовляти перекладений текст, виділити відповідні слова та фрази в тексті джерела та цілі, або служити простим словником для

введення одного слова. Якщо вибрано "Визначити мову", текст може бути автоматично ідентифікований. Якщо користувач вводить URL-адресу у вихідному тексті, Google Translate створює гіперпосилання на машинний переклад веб-сайту [55]. Користувачі можуть зберігати переклади в "розмовнику" для їх подальшого використання. Для деяких мов текст можна вводити за допомогою екранної клавіатури, за допомогою розпізнавання рукописного тексту або розпізнавання мови [54].

Microsoft Translate

За словами Ю. А Кота, перша версія системи машинного перекладу Microsoft була розроблена в період з 1999 по 2000 рр. в межах Microsoft Research. Система була заснована на семантичних структурах предикатів-аргументів, відомих як логічні форми (LF). Ця система, зрештою, використовувалася для перекладу всієї бази знань Microsoft на цілий ряд мов. Як і більшість сучасних систем машинного перекладу, підхід Microsoft до машинного перекладу є "керованим даними", замість того, щоб покладатися на написання явних правил для перекладу природної мови, алгоритми навчаються розуміти та інтерпретувати перекладені паралельні тексти, дозволяючи навчитися перекладати новий текст мови самостійно [3, с. 70].

Як стверджувалось на Microsoft Translator Hub: Discussion of BLEU, Microsoft Translator використовує машинний переклад для створення миттєвих перекладів з однієї природної мови на іншу. Система заснована комп'ютерному навчанні досліджень чотирьох різних областей. Якість результатів машинного перекладу Microsoft Translator оцінюється за допомогою методу оцінка BLEU [56].

Ю. А. Кот зазначив що, BLEU (двомовна оцінка недосконалості) представляє собою алгоритм оцінки якості тексту, який було автоматично перекладено з однієї природної мови на іншу. Якість вважається відповідністю між продуктивністю машини та продуктивністю людини. BLEU була однією з перших систем для досягнення високої кореляції з

людськими судженнями про якість і залишається однією з найпопулярніших автоматизованих та недорогих показників [3, с. 71].

Ю. А. Кот також підкреслив що, так як машинний переклад заснований на статистичних алгоритмах, а не на перекладачах, створеним автоматичні переклади, не завжди є цілком точні. Microsoft Translator запровадив різні можливості зворотного зв'язку у свої продукти, такі як STF, щоб дозволити користувачам використовувати альтернативні переклади. Пізніше альтернативні переклади інтегруються в алгоритми Microsoft Translator для вдосконалення майбутніх перекладів [3, с. 71].

У листопаді 2016 року Microsoft Translator представив переклад з використанням нейронних мереж для дев'яти мов найпопулярніших у використанні мов світу, включаючи японську мову. Нейромережі забезпечують кращий переклад, ніж галузеве стандартне статистичне навчання машин [44].

Ю. А. Кот зазначив що, Microsoft Translator містить у собі низку ключових сервісів. Microsoft Translator – це хмарне API що інтегроване в численні продукти та служби Microsoft. Перекладач API може бути використаний самостійно або може бути налаштований для використання в середовищі попереднього видання чи після публікації. API, доступний через підписку та є безкоштовним для нижчих обсягів перекладу. Решта основних продуктів доступні безкоштовно [3, с. 72].

Отже, можна відмітити що всі з розглянутих вище сервісів систем машинного перекладу пішли схожим шляхом щодо у розвитку машинного перекладу. PROMT Translator та Google Translate почали своє існування як системи машинного перекладу заснованих на правилах, але з часом модернізувались та доповнювались новими функціями та в підсумку перейшли до використання машинного перекладу на базі нейромереж. Microsoft Translator також розпочав використання системи машинного перекладу на базі нейромереж, але розпочав своє існування як статистична система машинного перекладу. За роки існування цих

сервісів можна помітити стрімке покращення в якості перекладу яких вони надають, особливо якщо казати про період в останні 5 років. Однак, на даному етапі жодна з систем не може зрівнятись з перекладачем людиною, а скоріш слугують допомагають перекладачам швидше впоратись зі своєю роботою.

1.4 Методологія дослідження рівню адекватності машинного перекладу

Методика визначення рівню адекватності машинного перекладу базується на загальнонаукових методах таких як: спостереження, опис, систематизація, індукція та дедукція, та емпірико-теоретичних методах таких як: аналіз, синтез, порівняння. Написання кваліфікаційної роботи за обраною темою вимагає залучення інструментарію лінгвістики, зокрема філологічного аналізу, методи якого були залучені при порівнянні уривків перекладу та у порівнянні перекладу уривків з оригіналом.

Теоретичний розділ нашого дослідження був спрямований на аналіз фактів, окреслення суті, опанування та формулювання методів роботи систем машинного перекладу. Метод опису, спостереження, систематизації були використані при написанні першого, теоретичного, розділу кваліфікаційної роботи зокрема при аналізі систем машинного перекладу (Google Translate, PROMT, Microsoft Translate). Метод спостереження був використаний для окреслення функцій та напрямків використання, а також для визначення напрямку розвитку систем машинного перекладу у сьогоdnішній час. Описовий метод був використаний при вивченні існуючих систем машинного перекладу, описі функцій, використання та розвитку. Метод систематизації дозволив вивчити об'єкт дослідження, тобто системи машинного перекладу, як системи загалом. Було проведена оцінка об'єкта включаючи чинники що мають вплив на його функцію.

Практичний розділ нашого дослідження був націлений на більш детальний аналіз та вивчення окремих складових що становлять системи машинного перекладу, їх роль у роботі цих систем, а також вплив що вони мають на результат, тобто кінцевий переклад. Метод аналізу дозволив нам розкласти предмет дослідження у вигляді окремих складових. Кожна з складових була проаналізована окремо і були висвітлені окремі висновки що до цього. В той же час, за допомогою методу синтезу були вивчені складові елементи предмету дослідження у зв'язку один з одним. Це дало змогу роз'яснити чим саме системи машинного перекладу досягають певного рівню адекватності при перекладі, а також яким чином його можна покращити. Методи індукції та дедукції слугували задля підведення певного підсумку щодо методів покращення якості перекладу в рамках систем машинного перекладу.

Окремо слід зазначити методи порівняння, а також структурного та компонентного аналізу. Метод порівняння був використаний при вивченні результатів перекладу систем машинного перекладу та їх порівнянні з оригіналом та перекладом перекладача професіонала задля виявлення недоліків у перекладі СМП. Методи структурного та компонентного аналізу були використані при вивченні цих недоліків та виявленню слабких сторін систем машинного перекладу разом з методами їх потенційного виправлення у майбутніх версіях СМП.

РОЗДІЛ 2.

ЗАСОБИ ВИЗНАЧЕННЯ АДЕКВАТНОСТІ МАШИННОГО ПЕРЕКЛАДУ

2.1 Спектр та специфіка застосування систем машинного перекладу в діяльності перекладача

За думкою багатьох дослідників, найпоширенішим інструментом автоматичного комп'ютерного перекладу (CAT) для професійного перекладу спеціалізованих текстів є технологія пам'яті перекладу (TM), яка була вперше впроваджена в середині 1990-х і широко застосовується з того часу [16; 29; 49; 53]. TM - це в основному база даних сегментованих і парних джерел та цільових текстів, до яких перекладачі можуть отримати доступ, щоб повторно використовувати попередні роботи під час перекладу нових текстів [19; 53]. Технологія TM зараз широко використовується на робочих місцях, на більшій кількості мов, а також більшою кількістю клієнтів та перекладачів, ніж будь-коли раніше [39; 41; 42; 59].

J. Hutchins зазначив що, реалізація комерційних TM систем вперше стала можливою, коли інструменти для вирівнювання тексту зробили можливими двомовні бази даних перекладів. Виходячи з цих ідей та технічних розробок, на початку 1990-х на ринку з'явилися чотири комерційні системи TM: TranslateManager від IBM, система Transit компанії Star, а також, EuroLang Optimizer та Translator's Workbench компанії Trados. З часу появи на ринку перших TM систем розповсюдження цієї конкретної технології постійно зростає [35, с.303].

J. Hutchins підкреслює що, автоматичний комп'ютерний переклад (CAT) охоплює машинний переклад з допомогою людини (HAMT) та автоматизований переклад (MAHT). У HAMT переклад, по суті, здійснюється самою програмою, але перекладач людина повинна вирішити конкретні мовні проблеми, що виникають із вихідного тексту,

або виправити отриманий цільовий текст. У МАНТ переклад здійснюється перекладачем людиною, а допомога комп'ютера розглядається як невід'ємна частина процесу [35, с.303].

Багато дослідників стверджують що, технологія ТМ - це комп'ютерний інструмент, який найбільш широко використовується окремими перекладачами, бюро перекладів та іншими організаціями, що займаються перекладом. Хоча системи ТМ можуть відрізнятися внутрішніми процесами, які регулюють сегментацію, вирівнювання, індексацію, пошук відповідностей, всі вони мають функцію розгортання існуючих ресурсів перекладу в новому проекті перекладу. У деяких випадках це поєднується з іншими типами програмного забезпечення, такими як текстові редактори або термінологічні бази даних [29; 49; 52].

J. Sommerlad зазначає що, ТМ - це в основному база даних сегментованих і парних текстових джерел та цілей, до яких перекладач може отримати доступ, щоб повторно використовувати попередні сегменти під час перекладу. ТМ постійно надає перекладачеві так звані збіги, що представляють собою пропозиції щодо перекладу, що зберігаються в базі даних. Системи ТМ працюють із трьома видами збігів: точними, нечіткими та без збігів. Ідентифікація цих збігів спирається на автоматичне порівняння символічних ланцюгів. Точні збіги відбуваються, якщо ланцюжок символів нового сегмента вихідного тексту ідентичний ланцюгу збереженого сегмента вихідного тексту; нечіткі збіги відбуваються, якщо сегменти вихідного тексту просто ідентифіковані як подібні; а відсутність збігів відбувається, якщо жоден сегмент вихідного тексту не ідентифікований або є достатньо подібним. Параметри між цими збігами, як правило, задаються програмою ТМ, але можуть також встановлюватися окремо перекладачем [52]

У своєму дослідженні D. Gouades підкреслив що, перекладачі можуть взаємодіяти з технологією ТМ різними способами. У більшості комерційних систем перекладач працює з інтерфейсом, який

характеризується як інтерактивний. Це означає, що сегменти вихідного тексту подаються по одному, надаючи перекладачеві можливість перекладати цей сегмент, або, якщо точне або нечітке збіг був отриманий з ТМ, прийняти, переглянути чи відхилити попередній переклад цього сегмента. Перекладачі також можуть взаємодіяти з технологією ТМ, використовуючи так званий режим попереднього перекладу [31, с. 378].

В результаті нашого пошуку емпіричних досліджень ТМ, які були опубліковані в 2000 році або пізніше, ми склали наступний перелік дев'яти досліджень, представлених у хронологічному порядку.

Використовуючи шестимісячний професійний проект, здійснений компанією Ваан, С. А. Lange та W. S. Bennett досліджували недоліки та переваги поєднання МТ та ТМ. Згідно з одним із веб-сайтів Ваан, вони розробляють автоматизовані програмні рішення для виробничих та бізнес-систем, включаючи функцію багатомовного перекладу свого програмного забезпечення більш ніж на 20 мов. Метою проекту було дослідити, як комп'ютерні технології можуть допомогти в перекладі текстів Інтернеті, щоб «дати якісні результати в більш короткі терміни» з метою скорочення «часу для перекладу на 50% за рахунок використання автоматизованого перекладу» [40, с. 206].

Хоча це не зовсім зрозуміло з самої статті, дослідники схоже, використовували оціночний та описовий методи в своєму дослідженні. Підсумовуючи, автори зауважують, що продуктивність перекладачів дійсно може збільшитися, якщо комбінувати ТМ та МТ, але, можливо, їх найцікавішим спостереженням є свідчення про те що продуктивність збільшиться лише, якщо перекладачі будуть згодні з їх новою роллю редактора машинних перекладів. Один з аналізів Lange та Bennett навіть дозволяє припустити, що процес редагування перекладачів може зайняти більше часу, ніж людський переклад, якщо перекладач негативно налаштований на роботу з МТ. Ці спостереження звучать правдоподібно і є обґрунтованими, однак у роботі Lange та Bennett бракує значних

деталей щодо контексту та аналітичних методів тематичного дослідження наприклад, скільки перекладачів було задіяно [40, с. 216].

Дослідження Т. Р. Christensen про ТМ як інструмент правового перекладу базується на припущенні, що через складність та культуру, що визначає місто юридичного спілкування, технологія ТМ буде менш корисною для юридичного перекладу, ніж для технічного перекладу. Основна мета її дисертації - оцінити корисність ТМ для юридичного перекладу. Christensen вибирає та поєднує відповідні теорії та методи в рамках сфер юридичних (юриспруденція), перекладознавства, інформатики та мовознавства. На її дослідницькі питання відповідають теоретичні та емпіричні аналізи [19, с. 354].

Теоретичні висновки Т. Р. Christensen полягають у тому, що, оскільки ТМ дає можливість перекладачам отримувати доступ та повторно переробляти частини потенційно необмеженої кількості попередніх перекладів у тому ж жанрі, технологія буде безсумнівно корисною для юридичних перекладачів. Однак, вона також визнає, що технологія ТМ недостатньо підходить для вирішення тих проблем, які зазвичай виникають при юридичному перекладі. За словами Крістенсена, корисність технології ТМ для юридичного перекладу повинна значно покращитися, якщо буде поєднана з текстовою базою достовірних сегментів. Технологія ТМ також бути покращена для юридичного перекладу, якщо вона включатиме в собі пам'ять про авторство, яка стандартизує сегменти вихідного тексту шляхом їх попереднього редагування перед тим як порівняти їх з сегментами цільового тексту [19, с. 368].

Що стосується емпіричного дослідження Т. Р. Christensen, то її висновки такі: Формулювання датських статей не демонструє достатньої повторюваності. ТМ, що складається з датських статей та їхніх перекладів, не вдається виявити достатню відповідність сегментів вихідного тексту, які функціонально ідентичні. Ідентифікація таких

відповідностей дійсно може бути покращена за допомогою авторської пам'яті, як і запропоновано в Christensen теоретичній частині. Одиниці перекладу в ТМ не можна розглядати як функціонально еквівалентні, що означає, якість легальних перекладів може бути певною мірою поліпшена, якщо перекладач має доступ до функціонально однакових сегментів в аутентичному корпусі цільової мови [19, с. 370].

Як Т. Р. Christensen зазначає сама, результати можуть бути неправомірними для всіх видів юридичного перекладу, але її висновки та пропозиції, безумовно є актуальні для застосування технології ТМ загалом і, безумовно, заслуговують на подальше емпіричне дослідження [19, с. 380].

Робота В. Dragsted написана про сегментацію в системах людського перекладу та ТМ. Спираючись на дані її минулої докторської дисертації, Dragsted далі досліджує, як примусова сегментація системи ТМ на основі виразу впливає на когнітивний процес перекладачів [24, с. 264].

Ми розглянемо ці дослідження як одне ціле дослідження. Дослідження містить дані декількох експериментів із залученням шести професійних перекладачів (щонайменше двох років професійного досвіду) та шести студентів (на останньому курсі магістерських досліджень спеціалізованого перекладу в Копенгагенській бізнес-школі). Було проведено два види експериментів. В одному експерименті В. Dragsted вивчала людський переклад, який був зафіксований за допомогою реєстрації ключів (Translog). В іншому експерименті вона вивчила переклад, що виконаний системою ТМ (Trados). Крім того, В. Dragsted спирається на ретроспективні вербалізації щодо розуміння перекладачами свого вибору та тексту, який їм надали для перекладу. У своєму аналізі В. Dragsted акцентує увагу на час що був витрачений на редагування, на те наскільки змінена структура речення вихідного тексту та способи що перекладачі використовують при сегментації вихідного

тексту. Паузи, записані за допомогою реєстрації ключем, розглядаються як індикатори сегментації перекладачів [25, с. 440].

В. Dragsted дійшла наступних висновків. Речення не є центральною одиницею когнітивної сегментації перекладачів, хоча це може бути більш правдивим для професійних перекладачів, ніж для студентів. В обох групах були помітні зміни у часу виділеному на переклад та його редагування. Перекладачі професіонали були більш обізнані про вплив ТМ на процес їх перекладу. Поведінка під час редагування постраждала в обох групах, але професіонали витратили на перегляд порівняно більше часу, порівняно з тим, що вони робили під час людського перекладу [25, с. 440].

Системи ТМ можуть бути модифіковані для повторного використання перекладів фрагментів речень шляхом інтеграції лінгвістичного аналізу (синтаксичний та морфологічний синтаксичний аналіз), а також за допомогою семантично обґрунтованих прийомів. Отже, згідно з висновками В. Dragsted, системи ТМ потрібно буде налаштувати відповідно до природних процесів сегментації перекладачів.

У своїй роботі Н. Fulford та J. Granell-Zafra вивчають використання перекладачами фрілансерами програм інформаційно-комунікаційних технологій, включаючи системи ТМ. У своїй роботі вони повідомляють про основи першої фази дослідницького проекту, створеного з метою дослідження прийняття інформаційних та комунікаційних технологій перекладачами фрілансерами у Великобританії. У дослідженні застосовується підхід опитування. Дані дослідження виходять із відповідей 591 британських фрілансерів. Найцікавішим результатом є те, що технологія ТМ та інші засоби САТ менш широко використовуються, ніж очікувалось: трохи менше половини респондентів кажуть, що вони зовсім не знайомі з інструментами САТ, і лише 28% респондентів заявляють, що вони використовувати ТМ та інші засоби САТ.

Опитування також виявило, що користувачі ТМ, як правило, спеціалізуються на перекладі технічних та наукових текстів [28, с. 4].

Основний висновок полягає в тому, що використання програм із програмним забезпеченням загального призначення - таких як програмне забезпечення для обробки текстів, графічне програмне забезпечення та програмне забезпечення для друку на робочому столі - частіше зустрічається серед британських фрілансерів, ніж використання програм спеціального призначення - таких як ТМ та інструменти управління термінологією [28, с. 6].

Дослідники S. Dillon та J. Fraser наводять короткий огляд особистих поглядів професійних перекладачів із Великобританії на технологію ТМ. Дані отримані з онлайн-опитування, яке було проведено у серпні 2004 р. Автори отримали 59 корисних відповідей [23, с. 70].

Надихнувшись інструментом G. C Moore та I. Benbasat для вимірювання особистого сприйняття, S. Dillon та J. Fraser попросили перекладачів відповісти на 24 заяви. За їхніми висновками, перші дві гіпотези можуть бути правдивими. Таким чином, новокваліфіковані перекладачі та перекладачі з досвідом роботи ТМ більш позитивно сприймають технології ТМ. Третя гіпотеза виявляється неправдивою, оскільки, перекладачі з сильними ІТ-навичками також мають позитивне сприйняття технології ТМ. Ці висновки приводять авторів до думки, що недостатнє розуміння та знання технології ТМ та її можливостей - а не природи та застосування самої технології ТМ - може бути важливою причиною, чому деякі перекладачі взагалі відмовляються від таких систем [46, с. 200].

Як і H. Fulford та J. Granell-Zafra, E. Lagoudaki розглядає опитування щодо впровадження технології ТМ, як і S. Dillon та J. Fraser, вона вивчає ставлення користувачів до технології ТМ. За допомогою онлайн-анкети E. Lagoudaki отримала відповіді від 699 фахівців з перекладу

(перекладачі, термінологи, керівники проектів, рецензенти, автори субтитрів тощо) з 54 країн [39, с. 20].

На відміну від опитування проведеного Н. Fulford та J. Granell-Zafra, впровадження технології ТМ виявляється значним: відсоток що використовують систему ТМ, становить 82,5%. Відповідно до висновків Е. Lagoudaki стверджує, що ті, хто спеціалізується на технічних текстах, частіше використовують засоби ТМ, а за ними - ті, хто спеціалізується на фінансовому та маркетинговому контенті. Ті що мають юридичну спеціалізацію, також використовують інструменти ТМ, але в меншій мірі, ніж вищезгадані групи, що узгоджується з положеннями Т. Р. Christensen [38, с. 22].

Відповідно до очікувань Т. Р. Christensen, Е. Lagoudaki знайшла залежність між високим рівнем текстового повторення та високим рівнем прийняття технології ТМ. На відміну від S. Dillon та J. Fraser, Е. Lagoudaki не виявила різниці у прийнятті ТМ між різними віковими групами та між особами, які мають або не мають досвіду роботи. Однак, як і в дослідженні S. Dillon та J. Fraser, Е. Lagoudaki визнала, що високий рівень розуміння ІТ-технологій пов'язаний з високим рівнем прийняття технології ТМ. На запитання, чому вони використовують ТМ, більшість респондентів відповідають, що це економить час (86%), що система забезпечує узгодженість термінології (83%) та покращує якість (70%). Інші переваги - економія витрат (34%) та ефективний обмін ресурсами, такими як глосарії та бази даних ТМ (31%). Досить інтересним є результат, що, хоча вони володіють інструментом ТМ, деякі респонденти (16%) ще не навчилися ним користуватися [39, с. 32].

У своєму дослідженні F. Mandreoli оцінює дизайн власної ТМ, а саме - Помічника з перекладу на базі прикладів (EXTRA). У порівнянні з більшістю систем ТМ, що покладаються на штучний інтелект, пошукова система системи EXTRA заснована на передових методах пошуку інформації, що виконують два процеси: по-перше, документ, який

потрібно перекласти, проходить синтаксичний аналіз, а потім його порівнюють з даними ТМ з використанням так званої відстані редагування. Ця процедура застосовується для того, щоб забезпечити хороший компроміс між ефективністю результатів та ефективністю залучених процесів [43, с. 169].

Дані походять із статистичних експериментів моделювання, які підтверджують, що EXTRA здатний ефективно здійснити переклад текстів західними мовами. На основі тестової роботи системи EXTRA автори роблять висновок, що їхні «результати показують ефективність EXTRA у пошуку корисних пропозицій та двох процесах, що складають її, припускаючи придатність їх самостійної роботи також у контекстах, які не є строго пов'язані з перекладом». Ми можемо не цілком оцінити технічні тонкощі їх дослідження, але автори, безумовно, мають рацію, стверджуючи, що CAT загалом і зокрема ТМ можуть представляти одну з найбільш перспективних парадигм перекладу [43, с. 169].

Досліджуючи когнітивне навантаження перекладачів у зв'язку з різними типами відповідності ТМ, S. O'Brien провела одне з небагатьох досліджень ТМ щодо взаємодії перекладачів з інструментами ТМ. У своєму дослідженні 2006 року S. O'Brien ставить собі за мету дослідити, чи відстеження руху очей взагалі є ефективним методом дослідження взаємодії перекладачів з ТМ, і, щоб дослідити чи дозволяє відстеження руху очей визначати відмінності в когнітивних зусиллях з різними типами ТМ. Дослідження S. O'Brien експериментальне, за участю чотирьох професійних перекладачів, які переклали текст за допомогою ТМ SDL Trados Translator's Workbench, а потім прокоментували процес їх перекладу. Дослідження аналізує когнітивні зусилля, необхідні перекладачам для роботи з різними типами ТМ, аналізуючи кількісні дані відстеження очей, доповнені якісними даними зйомки екрана та коментарями суб'єктів щодо того, що вони виконували під час перекладацького завдання [47, с. 190].

Пізнавальне зусилля вимірюється за допомогою швидкості обробки (слова в секунду) та розширення зіниці. Аналіз проводиться як порівняння швидкості обробки для кожного типу матчу та відсоткової зміни дилатації зіниць для кожного типу ТМ. Дані свідчать про суттєвий збіг між зміною у розширенні зіниць та швидкістю обробки даних. Дослідження також показує, що точні збіги чинять найменше когнітивне навантаження на перекладачів, тоді як відсутність збігів надають найбільшого навантаження. Крім того, хоча зв'язок не є лінійним, показано, що когнітивне навантаження збільшується у міру зменшення значень нечіткої відповідності. Як загальний висновок, O'Brien припускає, що її метод відстеження руху очей у поєднанні з ретроспективними протоколами добре підходить для дослідження процесів перекладу [47, с. 202].

Надихнувшись своїми результатами 2006 року, S. O'Brien більш детально досліджує, чому взаємозв'язок між значеннями нечіткої відповідності та когнітивним зусиллям не такий лінійний, як можна було б очікувати. Вона провела експериментальне дослідження, в якому вісім учнів перекладали технічний текст за допомогою ТМ SDL Trados Translator's Workbench. Виходячи з результатів щодо швидкості обробки, дослідження показує, що зменшення нечітких збігів означає збільшення зусиль. Якщо розглядати розширення зіниць як середнє арифметичне для всіх суб'єктів, розширення збільшуються у міру зменшення значень відповідності. За словами O'Brien, це може бути пов'язано з тим, що випробувані досягли базових пізнавальних зусиль [48, с.90].

Як і В. Dragsted, С. Colominas припускає, що сегментація на основі речень є неприродною і повинна принаймні доповнюватися сегментацією словосполучень, оскільки повторення цілого речення в цілому рідко зустрічається. Тому, вона має на меті дослідити користь такої сегментації. З метою оцінки як відкриття тексту (пропозиція цільового тексту), так і точності (зручності використання) [20, с. 346].

Обидва експерименти проводяться на основі різних ТМ: багатомовна текстова база даних, витягнутий з протоколів засідань Європейського парламенту та англо-іспанська текстова база даних, побудована з документів Організації Об'єднаних Націй. Застосовується та аналізується сегментація нижче рівня речення, особливо сегментація іменникової фрази (NP). Одержані дані, є логічними та ймовірними, але експерименти обговорюються досить технічно, а концепції недостатньо визначені та функціоналізовані [20, с. 346].

Більшість сприймають як належне, що технологія ТМ прискорює роботу та покращує якість перекладу, але не існує жодних досліджень, які насправді це підтверджують. Дослідники С. А. Lange та W. S. Bennet припускають, що продуктивність перекладачів може збільшитися ще більше, якщо технологія ТМ об'єднається з машинним перекладом [40, с. 207]

S. Dillon та J. Fraser зазначають, що рівень досвіду професійних перекладачів та їх ІТ-компетентність можуть вплинути на те, як вони дивляться та використовують ТМ, хоча E. Lagoudaki цього не підтверджує [23, с. 78], [39, с. 34].

Немає сумніву, що технологія ТМ також має деякі серйозні недоліки, які розробники програмного забезпечення мають вирішити в майбутніх програмах. Так, наприклад, Т. Р. Christensen стверджує, що технологія ТМ може бути кориснішою для одних жанрів, ніж для інших: технологія ТМ, ймовірно, корисніша під час перекладу текстів, які, як правило, стандартизовані та повторювані - технічна документація, наприклад - тоді як вона може бути менш корисним, при перекладі складніших текстів - наприклад, юридичних документів. Щоб збільшити свою користь для юридичних перекладачів, Т. Р. Christensen пропонує, щоб технологія ТМ поєдналась з еталонною текстовою базою даних справжніх, функціонально еквівалентних сегментів мови цільової мови, і, можливо, також авторською пам'яттю, яка може передбачати та

стандартизувати сегменти вихідного тексту з метою полегшення знаходження відповідників [19, с. 445]

Як вказував S. O'Brien, лише точні збіги вимагають невеликих зусиль з боку перекладача, тоді як нечіткі збіги вимагають більше зусиль, хоча співвідношення зусиль і значень відповідності не є лінійним [47, с. 203], [48, с. 99].

Ще один недолік, притаманний технології ТМ, вказують С. Colominas та В. Dragsted, які стверджують, що ТМ змушує перекладачів працювати, когнітивно неприродним чином: більшість доступних у продажу систем ТМ змушують перекладача працювати на основі пропозицій цілих речень як одиниць перекладу, тоді як (досвідчені) перекладачі, як правило, прагнуть працювати з окремими словосполученнями [20, с.351], [24, с. 450].

Отже, машинний переклад дає можливість перекладачам отримувати доступ та повторно переробляти частини потенційно необмеженої кількості попередніх перекладів у тому ж жанрі, технологія буде безсумнівно корисною для юридичних перекладачів. Однак, також можна зазначити, що системи машинного перекладу недостатньо підходять для вирішення тих проблем, які зазвичай виникають при юридичному перекладі. Крім того, в разі якщо технологія ТМ об'єднається з машинним перекладом це можливо прискорить роботу та покращить якість перекладу, але не існує жодних досліджень, які насправді це підтверджують. Емпірично задокументовані знання про природу та застосування ТМ/МТ-систем та взаємодії перекладачів з ними є дефіцитними та фрагментарними. Зокрема, потрібно провести більше досліджень, як перекладачі взаємодіють із технологією ТМ та як це впливає на пізнавальні процеси перекладачів. Сама професія перекладу також вітатиме більше знань про погляд перекладачів на технологію ТМ.

2.2 Типові помилки систем машинного перекладу

Ми розглянемо стилістичні особливості уривків з книги D. Knuth «The Art of Computer Programming», а також проведемо аналіз перекладу виконаного статистичною та аналітичною системами машинного перекладу опираючись на роботу дослідника І. К. Нікітіна, що також аналізував ці уривки. Переклад аналітичною системою був виконаний за допомогою PROMT 9.0, а статистичної за допомогою Google Translate. Для більш інформативного порівняння уривки також будуть порівнянні з перекладом професіонального перекладача.

Уривки в основному написані в науковому стилі. Як і будь-який реальний текст, володіють стилістичною неоднорідністю. Безумовно, є деякі моменти, які наближають текст до художнього. Однак, з огляду на спрямованість книги, доводиться говорити виключно про науковий стиль.

Розглянутий текст має властивості:

- дохідливості;
- логічної послідовності;
- об'єктивності викладу;
- традиційності [6, с. 50].

Автори українського перекладу спробували згладити деякі традиційні наукові штампи, які використовуються в тексті, що зробило мову уривка жвавіше, хоча і трохи спотворило його стилістично. З іншого боку в деяких частинах тексту були помічені зворотні перетворення. Розглядаючи, машинний переклад уривка, звичайно, ні про жодну компенсацію не може йти й мови [37, с. 676].

Всі чотири властивості були передані, однак за ствердженням І. К. Нікітіна, деякий сумнів викликає властивість дохідливості. На деяких ділянках тексту вона страждає через помилки узгодження. Важливо відзначити, що для трансфертної системи кількість таких помилок значно нижче. У першому уривку можна знайти опис алгоритму Евкліда. Навіть без спеціального аналізу не важко сказати, що стиль алгоритму і стиль

основного тексту сильно відрізняються. Якщо уважно подивитися на англомовне опис, то можна зробити висновок (частково помилковий), про те, що виклад ведеться на формальною імперативною мовою обробки даних. У українському варіанті алгоритму стилістична різниця трохи зглажена. З одного боку, це полегшує сприйняття, але зменшує необхідну формальність опису [6, с. 52].

Логічна структура уривків, за словами І. К. Нікітіна була легко відображена в усіх варіантах перекладу. СМП змогли передати основну думку уривків. Незважаючи на те, що в перекладі аналітичної СМП третій уривок виглядає як набір слів, і вимагає значної правки, думки автора відновити не складно. Останній ефект пов'язаний з тим, що третій уривок містить велику кількість специфічних термінів, які навіть професійний перекладач не ризикнув перекладати [6, с. 52].

Стилістично машинний переклад виявився більш близьким до оригіналу. Наприклад, в наступному уривку перекладач порушив властивість традиційності. Дванадцять речення третього уривка:

“Many years have passed since the author wrote most the comments above...” [37, с. 677].

Варіант статистичної СМП:

Минуло багато років з того часу, коли автор написав більшість коментарів вище...

Варіант перекладача:

“З часу першого написання автором більшості наведених вище коментарів сплило багато води...” [2, с. 677].

Однак в ряді випадків, за словами І. К. Нікітіна, відступи від авторського стилю були продиктовані вітчизняними традиціями науково-технічної літератури, про які системам машинного перекладу нічого не відомо. Це наприклад відноситься, до того що перекладач намагався уникнути тавтології [6, с. 52].

Не завжди переклад виконаний людиною володіє об'єктивністю. При аналізі уривків ми в цьому переконалися. У ряді випадків, перекладач трохи трансформував думки автора, змінив їх власними судженнями. СМП, особливо статистичні, позбавлені цього недоліку.

Наприклад, в двадцять восьмому реченні другого уривка:

"...but it is believed..." [37, с. 659].

Перекладач передав із спотворенням сенсу:

"...але автор все ж вірить..." [2, с. 659].

Значним є той факт, що в ряді випадків переклад створений машиною практично не відрізняється від варіанту перекладача. Наприклад, друге речення другого уривка:

"«Discrete simulation» means the simulation of a system in which all changes in the state of the system may be assumed to happen at certain discrete instants of time". [37, с. 659].

Переклад аналітичної СМП:

«Дискретне моделювання» означає моделювання системи, в якій можна припустити, що всі зміни в стані системи відбуваються в певні дискретні моменти часу.

Переклад людиною:

"«Дискретне моделювання» означає моделювання системи, в якій передбачається, що всі зміни стану системи відбуваються в деякі дискретно задані моменти часу". [2, с. 659].

З першого погляду неможливо зрозуміти, що переклад зробила машина. І тільки після аналізу перекладу та оригіналу тексту можна побачити деякі стилістичні помилки. Іноді переклад машини опинявся точніше, ніж варіант перекладача.

Тепер розглянемо лексичні особливості уривків і проведемо аналіз перекладу, виконаного статистичною та аналітичною системами машинного перекладу.

У тексті часто зустрічається загальноживана лексика яку в деяких випадках машина переклала точніше ніж людина:

“The format above illustrates the style in which all of the algorithms throughout this book will be presented”. [37, с. 659].

В даному випадку, *format* виступає в значенні *формат, структура, вид*. Однак перекладач-людина це переклав це як:

“Наведене вище формулювання ілюструє стиль, в якому алгоритми будуть представлені протягом всієї книги”. [2, с. 659].

За думкою І. К. Нікітіна, слово *формулювання* може бути сприйняте читачем як твердження, що, в контексті викладу, є невірним і викликає непорозуміння. Системи перекладу в цьому випадку переклали фразу шаблонно [6, с. 52].

Переклад статистичної СМП:

Формат вище ілюструє стиль, в якому всі алгоритми в цій книзі будуть представлені.

Переклад аналітичної СМП:

Наведений вище формат ілюструє стиль, в якому будуть представлені всі алгоритми по всій книзі.

Переклад виглядає примітивно, але в цьому випадку саме він передає сенс більш повно.

За словами І. К. Нікітіна, третій уривок з точки зору термінології є найбільш складним. Важливо відзначити, що він є складним не тільки для перекладу, а й для сприйняття. Такі терміни як *зв'язуючі автомати (linking automata)*, і *машини графів (graph machines)* були переведені невірно СМП, і без оригіналу складно зрозуміти про що йде мова. Взагалі слово *linking*, протягом усього третього уривка, СМП переклали як загальноживане слово, а не як термін [6, с. 54].

З іншого боку складні фрази як *«...Turing machines having multiple tapes and read / write heads.»* [37, с. 677] були перекладені наступним чином:

Переклад статистичною СМП:

...машин Тюрінга, що мають кілька стрічок і головок читання / запису.

Переклад аналітичної СМП:

...машин Тюрінга, що мають багаторазові стрічки та комбіновані головки.

Переклад людини:

“...машин Тюрінга з безліччю стрічок і головок читання/запису.”

[2, с. 659].

І. К. Нікітін зауважив, що для аналітичної СМП виявилася відома фраза *комбіновані головки*. Це говорить про застосування варіантних трансформацій в аналітичній СМП - переклад відбувається не так примітивно, як це могло б здатися. З іншого боку застосоване узагальнення може вплинути на якість сприйняття. Однак, фраза *multitape Turing machine* перекладена досить дивно і вірно термін був переведений тільки людиною. Статистична СМП дуже цікаво переклала цей термін. І фразу *multitape Turing machine*, і *one-tape Turing machine* вона перекладає просто як *машина Тюрінга*. Розуміючи принципи роботи статистичних машин легко пояснити природу цього ефекту. З точки зору науково-технічної літератури це зовсім не вірно, але в рамках інших стилів подібний підхід міг бути виправданий [6, с. 55].

За словами І. К. Нікітіна, у першому уривку зустрічається велика кількість точних дат, які також є показником науковості. У науковому тексті, зазвичай, вся числова інформація передається цифрами, дати не є винятком, тому як для СМП так і для перекладачів «переклад» подібних слів є досить легким [6, с. 49].

Однак, стверджує І. К. Нікітін, не все так просто з географічними назвами і власними іменами. В даному тексті правильна назва міст та інших об'єктів не є критичною, але вони показують кругозір автора і його

обізнаність. При перекладі таких слів треба враховувати іменування таких об'єктів прийняте в українській літературі [6, с. 55].

В сьомому реченні першого уривка:

“*The Aral Sea in Central Asia was once known as Lake Khwarizm...*” [36, с. 650].

Переклад статистичної СМП:

Колись Аральське море в Центральній Азії було відоме як озеро Хорізм ...

Переклад аналітичної СМП:

Аральське море в Середній Азії колись було відомо як озеро Хваризм...

Переклад людини:

“*Аральське море в Центральній Азії було колись називали озеро Хорезм...*” [2, с. 650].

І. К. Нікітін підкреслив що, тут можна побачити відразу дві географічні назви, регіон Середньої Азії і озеро Хорезм. Іменування першого терміну має різну географічне та політичне трактування. Аналізуючи переклад зроблений людиною, доводиться враховувати поточні віяння і, можливо, тиск який чиниться на перекладача. У разі СМП справа йде не набагато простіше, бо ряд фраз міг бути жорстко прописаний в алгоритмі перекладу. Аналогічна ситуація стосується не тільки географічних об'єктів, але так само може проявитися при перекладі будь-яких назв і іменувань, особливо якщо об'єкти реального світу були названі в честь якого-небудь ученого або громадського діяча [6, с. 57].

Ще цікавіша ситуація складається, якщо в момент написання оригіналу об'єкти називалися інакше. Даний приклад також відображає таку ситуацію. Ми вважаємо, що перекладати термін *Central Asia* в цьому реченні слід саме як *Середня Азія*. Саме так називався цей регіон, коли Д. Кнут писав свою працю. Однак, в поточний момент регіон називається

Центральна Азія. Ця назва була закріплена за регіоном, під час роботи перекладача. Цим пояснюється його варіант перекладу.

Нижче ми розглянемо морфологічні особливості уривків і проведемо аналіз перекладу, виконаного статистичною та аналітичною системами машинного перекладу.

В третьому уривку можна побачити використання іменних конструкцій:

Оригінал:

“At the time this chapter was first written...” [37, с. 677].

Переклад статистичної СМП:

На той час, коли ця глава була вперше написана...

Переклад аналітичної СМП:

У той час ця глава була вперше написана...

Переклад людини:

“У той час, коли автор почав працювати над цією главою...” [2, с. 677].

За словами І. К. Нікітіна, і людина і СМП врахували цю особливість. Незважаючи на те, що машини переклали фразу з помилками, можна сказати, що сенс вираження був досягнутий. В цьому випадку важлива не стільки точна дата події, а те що вона проходила одночасно з іншою подією. [6, с. 60].

Широко поширені так само безособові і невизначено-особисті конструкції.

В першому уривку:

“...at first glance it may look as though someone intended to write...”

“...while in later sections it is referred to as...” [37, с. 650].

Переклад статистичної СМП:

...на перший погляд може здатися, ніби хтось мав намір писати...

...тоді як у наступних розділах це згадується як...

Переклад аналітичної СМП:

...на перший погляд це може виглядати, як ніби хтось мав намір написати...

...тоді як у наступних розділах він згадується як...

Переклад людини:

“...на перший погляд може здатися, ніби хтось збирався...”

“...посилатися на нього в наступних розділах ми будемо як...” [2, с. 650].

При машинному перекладі вирази були перекладені шаблонно. За словами І. К. Нікітіна, одного боку це зберегло їх безособову форму, яка також характерна для української мови. З іншого боку це негативно вплинуло на узгодження. Людина-перекладач в даному випадку віддав перевагу зміні порядку слів. Це трохи змінило стилістичну спрямованість тексту, але спростило його сприйняття. [6, с. 62].

В другому уривку:

“...but it is believed to be the simplest set of rules...” [37, с. 659].

Переклад статистичної СМП:

...але вважається, що це найпростіший набір правил...

Переклад аналітичної СМП:

...але вважається, що це найпростіший набір правил...

Переклад людини:

“...але автор все ж вірить, що це є простим набором правил...” [2, с. 659].

І. К. Нікітін зазначив, тут перекладач дозволив собі більше серйозний відступ від оригіналу. При прочитанні третього варіанту виникає сумнів в об'єктивності висловлювання, так як об'єктивність є одною з найважливіших особливостей науково-технічного тексту. З іншого боку, в аналізі ми розглядаємо обмежені уривки тексту вирвані з контексту, тому поведінка перекладача може бути пояснена компенсацією [7, с. 63].

У своєму дослідженні І. К. Нікітін стверджує що, згідно з однією з морфологічних особливостей науково-технічного стилю в тексті використовується майбутній час замість теперішнього. Подібна ситуація відображена і в перекладі. Для СМП використовується простий переклад фраз. Частково, це походить від того, що в українській та англійській мовах традиції для наукових текстів збігаються. Останнє має більший вплив на статистичну СМП. Частково - від того, що подібна ситуація могла бути жорстко прописана в алгоритмі перекладу. Це більше відноситься до аналітичної СМП. У варіанті перекладача-людини найбільш видно прояв перекладацької компенсації, і заміна ряду конструкцій на більш прийнятні в українській мові [6, с. 69].

Отже, кажучи про помилки при СМП перекладі можна окремо відзначити стилістичні та морфологічні. З прикладів можна побачити що незважаючи, що СМП за роки свого існування поліпшили якість перекладу все одно, помилки саме стилістичного та морфологічного характеру зустрічаються досить часто. Однак слід зазначити, що порівнюючи переклади СМП та переклад професіонального перекладачем можна помітити, що СМП більш жорстко притримуються тексту оригіналу, що в даному випадку йде на користь адже йде мова про переклад наукового тексту, а перекладач професіонал у деяких випадках дещо нехтує стилем автора з тих чи інших причин. Загалом, можна сказати що незважаючи на помилки СМП у перекладі, на нашу думку, їх варіант перекладу можна вважати вдалим, однак він все одно потребує редагуванню.

2.3 Перспективи розвитку систем машинного перекладу

Дослідження А. Kilgarriff є важливим, оскільки воно широко цитується як свідчення того, що органи чуття, не охоплюють почуття, фактично перенесені більшістю варіантів слів в текстовій базі даних. Це дійсно дуже важливо, тому що це означає, що слова, що позначають сенс

тегів у текстовій базі даних, використовуються будь-якими лексичними даними, заснованими на машиночитаному словнику чи тезаурусі. Дослідження не демонструє нічого подібного. Більше того, це відбувається за допомогою неформальної [36, с. 366].

Вихідною точкою дослідження А. Kilgarriff є банкова модель (БМ) лексичного розрізнення неоднозначності, яка встановлюється за твердженнями, а не цитатами. У БМ слова мають дискретні значення, і людський читач миттєво і без зусиль знає, яке значення цього слова застосовується, враховуючи, що словосполучення завжди стосується одного або іншого, але не обох основних значень, яким належить слово, як наприклад *банк*. Основна мета роботи А. Kilgarriff полягає в тому, щоб розрізнити ряд зв'язків між відчуттями LDOCE (Словник сучасної англійської мови Лонгмана), які не є дискретними, а потім перейти до експерименту з текстовою базою даних [36, с. 367].

За словами Y. Wilks, автори в минулому були не настільки наївними, ніж автори сьогодення, і, ймовірно, писали програми, хоч і примітивні та неефективні, що виконувати ті самі завдання як і зараз. Що ще важливіше, цитована робота, яка стала підходом, так званою семантикою переваг, що по суті була дослідженням розбіжності використання текстової бази даних від лексичних норм, і розроблена в сімдесятих роках, у набір процесів розміщення розбіжних/нестандартних/метафоричних уявлень, в які, були створені групою дослідників близько 1990 року [59, с. 222].

Дійсно у роботі Y. Wilks, були запрограмовані процедури щоб пристосувати різні вживання усталених слів в тому ж тексті, тоді як у його наступній роботі запрограмовані процедури були визначені для пристосування таких слів створюючи нові вислови. [60, с. 158], [61, с. 200].

Набагато більш вагомим упущенням, що стосується безпосередньо його основної претензії і не є лише питанням історичної коректності, є

відсутність посилання на роботу в Нью-Мексико та інших місцях наприклад роботи J. Cowie у великомасштабному розумінні сенсорних слів текстової бази даних, отриманої з MRD (Machine-readable dictionary). Це були більш масштабні експерименти, результати яких прямо суперечать попереднім. Найкраща частина роботи A. Kilgarriff - це його спроба дати інтуїтивно зрозумілий розвиток відносин між позначенням чуття словами: про це, звичайно, існує велика кількість наукової літератури [22, с. 362].

Визначити перші два типи не завжди просто, оскільки обидва базуються на взаємозв'язку між двома та більше почуттями. Але A. Kilgarriff переконливо показав, що вибір між чуттями певного слова часто зробити непросто, оскільки це залежить від їх взаємозв'язку, характеру визначень і наскільки слово специфічне.

Справжньою проблемою в аналізі сенсорних типів A. Kilgarriff є те, що він суперечить наступному:

1. Використання тексту, відмінне від показаного у цілому списку збережених почуттів для певного слова, напр. у словнику (про що піде мова в його подальшому експерименті)

2. Використання тексту, що розходяться від деякого "основного" сенсу в лексиці [36, с. 368].

Тільки пункт 2 належним чином стосується метафори/метонімії як в роботі дослідників A. Corpestone та T. Briscoe роботи групи, в яку він розміщує себе, і саме до цього явища належить його класифікація чуттєвих відмінностей. Це поняття вимагає певного уявлення про розвиток сенсу; почуттів слова, що поширюються в часі не випадково, і є лінгвістичною традицією аналізу, про яку розповідав T. Givon. Однак «опудало» БМ і експеримент, який він потім проводить з ручним позначенням почуттів у тексті, прикріплюють все до першого, не пов'язаного поняття, яке зазвичай не передбачає присутності метонімії чи метафори, а просто представляє собою неадекватний сенс. Звичайно, два

типи можуть бути історично пов'язаними, оскільки деякі можуть бути отримані метафоричними/метонімічними процесами зі слова. Ця плутанина є слабкою стороною в його роботі, оскільки важко бути впевненим, який він хоче щоб ми зробили висновок з експерименту. Однак, оскільки ми покажемо, що його результати не є дійсними, це може не мати особливого значення [21, с. 92], [30, с. 50].

Тут можна додати, що песимізм А. Kilgarriff пішов на руку з дуже цікавими опитуваннями, які він проводив через Інтернет щодо реальної потреби в розбірливості слів і сенсу NLP (Natural Language Processing) R&D (Research & Development). І слід зазначити, що є й інші, які поставили під сумнів практичну корисність даних, отриманих на багатьох сайтах із MRD. Сенс полягає в тому, що це було корисно, як роботі з позначення смислу [22, с. 363], [36, с. 370].

Експеримент А. Kilgarriff, не описаний дуже докладно. У виносці він відмовляється надати читачеві статистику, на якій ґрунтувався його результат, навіть якщо текст досить чітко містить твердження, що 87% слів у його текстовому зразку мають хоча б одне виникнення яке не може бути пов'язане з одним і лише одним визначенням LDOCE. Отже, він стверджує, що старий БМ спростовується [36, с. 378].

Але багато емпіричних доказів також стає проти твердження, яке висунув А. Kilgarriff. Наприклад, також є використання імітаційних методів J. Cowie проведених в CRL-New Mexico для присвоєння почуттів текстовій базі даних LDOCE. У цій роботі було показано, що близько 75%-80% вживання слів може бути правильно пов'язане з визначеннями LDOCE, порівняно з текстом перевіреном вручну [22, с. 364]

Два вищезазначені міркування показують, із зовсім різних джерел та методик, сумнівну природу претензії А. Kilgarriff. Y. Wilks стверджує, що слова мають лише єдине визначення почуття, і що словники/лексикони повинні виражати це єдине значення і залишати все

подальше смислове вдосконалення якомусь іншому процесу, наприклад, маніпуляції зі знанням реального світу штучним інтелектом [60, с. 160].

У. Wilks стверджує що, оскільки результат CRL припустив, що автоматичні процедури спрацьовували дуже добре, а не на переосмисленні, одним із можливих шляхів вперед для NLP було б обмеження лексичних відмінностей сенсу до рівня гомографії. Такий хід можна сприймати як заміну даних відповідно до того, чого ви можете досягти, або як відновлення штучного інтелекту та прагматики в NLP [60, с. 161].

Ця пропозиція досить відрізняється від висновку А. Kilgarriff що також є емпіричним. Він пропонує реальну основу розрізнення почуттів встановлену методами кластеризації що застосовуються до текстових баз даних. Це відмінна ідея, і робота в IBM здійснена Р. F. Brown створила вражаючі кластери використання текстових баз даних, багато з яких демонструють подібність, близьку до інтуїтивного поняття сенсу людиною [17, с. 170].

Але існують серйозні проблеми з переміщенням будь-якої лексикографії, традиційної чи обчислювальної, на будь-яку таку основу. У. Wilks стверджував, що може бути написаний словник, який повністю складається із звичок, і дослідив, як вони можуть бути кластеризовані для суто лексикографічних цілей, однак залишається незрозумілим, який саме обсяг може бути результатом такого проекту або хто буде купувати його і їм користуватися. Одним із способів сприйняття такого продукту було б зведення одномовних словників до тезаурусу. Тезауруси виконують реальну функцію як для носіїв мови, так і для тих хто не є носієм мови, але вони покладаються на те, що читач знає, що означають деякі чи всі слова в рядку, оскільки вони не дають пояснень. [60, с. 163].

Якщо казати не про словники для людського вжитку, а про лексикони NLP, пропозиція А. Kilgarriff може бути приємнішою, оскільки він пропонує, нову версію WordNet, за словами G. A. Miller, з її

синсетами, встановленими не за фактом, але за допомогою статистичні кластеризації текстової бази даних. Однак, існують складнощі: по-перше, будь-який подібний процес кластеризації створює не тільки чисті, охайні приклади, але й неминучі монстри, вироблені деякою химерністю певної текстової бази даних. Вони, звичайно, можуть бути виправлені вручну, але це не автоматичний процес [45, с. 63].

За словами Y. Wilks, які приклади ви отримуєте, а точніше, їх спільність, залежить від параметрів в алгоритмі кластеризації: тобто, отримані в різних налаштуваннях приклади, можуть або не можуть добре відповідати, скажімо, різним рівням стандартної лексичної ієрархії. Оскільки ієрархії дискретні за рівнем, а використовувані параметри є безперервними, але, навіть коли вони є, не буде жодного доданого ієрархічного терміну, подібних до WordNet (наприклад, *тварина* або *домашня тварина*). І це лише окремий випадок загальної проблеми алгоритмів кластеризації, добре відомих при пошуку інформації, що такі знайдені кластери не мають доданих імен чи функцій [60, с. 164].

Y. Wilks також стверджує що, завжди знайдеться відповідність таких емпіричних кластерів будь-яким новим текстовим проявам слова, і, настільки, сенс тегів у тексті зобов'язаний досягти успіху за такою методологією, з огляду на походження кластерів та на те, що завжди можна знайти найближчу відповідність одному із наборів кластерів. Проблема полягає в тому, як ви інтерпретуете цей результат, оскільки за цією методологією жоден обраний вручну, не буде доступний як елемент керування, оскільки незрозуміло, яке завдання можна було б попросити виконати людину. Суб'єкти можуть вважати важким визначення сенсу, але це досить зрозуміле завдання, яку виконують словники та пов'язаних з ними відчуття нашому культурному світі. Але нове завдання (приєднати тільки один з прикладів, у яких слово з'являється до його вживання в цей момент) досить менш чітко визначене. Але знову ж таки, цілий ряд оригінальних і геніальних пропозицій може зробити цю задачу набагато

простішою, і почуття, настільки вони виділені, безумовно, могли б допомогти реальним завданням що ставлять МТ, навіть якщо вони не створили повністю оригінальні словники, що можна купити [60, с. 166].

На останок Y. Wilks зазначив що, не існує суперечності між, з одного боку, пропозицією щодо ущільнення лексиконів до ядерних чи гомографічних почуттів, зроблених для оптимізації процесу міркування сенсом, а з іншого - його пропозицією на емпіричній основі для встановлення синсетів або кластерів, що складають почуття. Зважаючи на те, що існують проблеми з цілком емпірично заснованими сенсорними кластерами що були згадані вище, природним кроком було б запропонувати певну форму гібридного виведення із статистики текстової бази даних разом із деяким машиночитаним джерелом синсетів: тезаурус WordNet або навіть двомовні словники, є досить зручними. Обидва методи, мають свої позитивні сторони і свої власні систематичні помилки, тож залишається сподіватись, що розумні процедури можуть спричинити скасування, а не посилення, один одного [60, с. 176].

Як висновок, можна відзначити, що розвиток систем машинного перекладу на даному етапі представляє собою тему для досить бурхливих обговорювань. Ми розглянули лише один з можливих варіантів розвитку, що стосуються розпізнавання емоційного навантаження певного слова в залежності від контексту. І як можна побачити Y. Wilks аналізуючи роботу A. Kilgariff зазначає що існують безліч нюансів які встають на шляху до покращення систем в цьому напрямку. Однак, Y. Wilks представив свою пропозицію щодо яка включає у себе зведення словників у тезауруси містять емоційно навантажені слова і їх переклад, які будуть внесені до тестової бази даних певної системи машинного перекладу. В результаті чого, текстова база даних системи та база даних що буде внесена окремо будуть працювати разом, тим самим надавати якомога якісніший приклад перекладу тих чи інших слів та словосполучень, в залежності від контексту.

ВИСНОВКИ

У результаті нашого наукового дослідження ми розглянули рівень адекватності машинного перекладу. Для цього необхідно було розглянути ряд завдань.

Першим завданням було окреслити місце машинного перекладу в сучасному суспільстві. Нами було визначено що машинний переклад суттєво полегшує повсякденну перекладацьку роботу, прискорює її, та вдосконалює традиційний процес перекладу. Системи машинного перекладу надають наступні переваги: оперативність, практичність, доступність, гнучкість, наявність вибору, багатофункціональність. Машинний переклад, насамперед, являє собою як допоміжний інструмент в роботі перекладача, оптимізуючи процес перекладу, обробляючи великі обсяги інформації. Однак, результати машинного перекладу все одно підлягають редагуванню для того щоб запобігти появі помилок у кінцевому тексті перекладу.

Другим завданням було розглянути існуючі системи машинного перекладу, розвиток та узагальнити їх типологію. Нами було визначено наступні види систем машинного перекладу: машинний переклад на базі правил; нейронний машинний переклад; системи машинного перекладу, засновані на статистиці. Всі системи між собою мають свої недоліки та переваги. Системи на базі правил представляють собою найбільш поширені системи машинного перекладу. Системи такого типу надають високу якість перекладу але в той же час мають високу вартість та вимагають довготривалої розробки. Статистичний машинний переклад також може надати високу якість перекладу однак вимагає відповідні словники для створення адекватного перекладу. Системи машинного перекладу на базі нейронних мереж є вкрай популярними на сьогоднішній день, оскільки з'явилась можливість наблизити модель перекладу до моделі яку використовує людина. Перевагами такої системи є висока

якість перекладу, навіть у порівнянні з іншими системами перекладу, однак такі системи потребують великий обсяг текстів для навчання і дуже багато часу.

Третє завдання, зорієнтоване на аналіз сучасних систем машинного перекладу. Ми з'ясували що всі з розглянутих вище сервісів систем машинного перекладу пішли схожим шляхом щодо у розвитку машинного перекладу. PROMT Translator та Google Translate почали своє існування як системи машинного перекладу заснованих на правилах, але с часом модернізувались та доповнювались новими функціями та в підсумку перейшли до використання машинного перекладу на базі нейромереж. Microsoft Translator також розпочав використання системи машинного перекладу на базі нейромереж, але розпочав своє існування як статистична система машинного перекладу. За роки існування цих сервісів можна помітити стрімке покращення в якості перекладу яких вони надають, особливо якщо казати про період в останні 5 років. Однак, на даному етапі жодна з систем не може зрівнятись з перекладачем людиною, а скоріш слугують допомагають перекладачам швидше впоратись зі своєю роботою.

Четверте завдання передбачало з'ясування яким чином системи машинного перекладу використовуються в роботі перекладача. Машинний переклад дає можливість перекладачам отримувати доступ та повторно переробляти частини потенційно необмеженої кількості попередніх перекладів у тому ж жанрі, технологія буде безсумнівно корисною для юридичних перекладачів. Однак, також можна зазначити, що системи машинного перекладу недостатньо підходять для вирішення тих проблем, які зазвичай виникають при юридичному перекладі. Крім того, в разі якщо технологія ТМ об'єднається з машинним перекладом це можливо прискорить роботу та покращить якість перекладу, але не існує жодних досліджень, які насправді це підтверджують. Емпірично задокументовані знання про природу та застосування ТМ/МТ-систем та

взаємодії перекладачів з ними є дефіцитними та фрагментарними. Зокрема, потрібно провести більше досліджень, як перекладачі взаємодіють із технологією ТМ та як це впливає на пізнавальні процеси перекладачів. Сама професія перекладу також вітатиме більше знань про погляд перекладачів на технологію ТМ.

Останнє, п'яте завдання мало на меті виявити проблеми досягнення адекватності машинного перекладу. Кажучи про помилки при СМП перекладі можна окремо відзначити стилістичні та морфологічні. З прикладів можна побачити що незважаючи, що СМП за роки свого існування поліпшили якість перекладу все одно, помилки саме стилістичного та морфологічного характеру зустрічаються досить часто. Однак слід зазначити, що порівнюючи переклади СМП та переклад професіональним перекладачем можна помітити, що СМП більш жорстко притримуються тексту оригіналу, що в даному випадку йде на користь адже йде мова про переклад наукового тексту, а перекладач професіонал у деяких випадках дещо нехтує стилем автора з тих чи інших причин. Загалом, можна сказати що незважаючи на помилки СМП у перекладі, на нашу думку, їх варіант перекладу можна вважати вдалим, однак він все одно потребує редагуванню.

Загалом можна сказати що, системи машинного перекладу можуть надати досить вдалий та адекватний переклад. Нині жодна система не здатна повністю замінити професійного перекладача, особливо якщо мова йде про тексти художнього характеру. Натомість такі системи застосовуються перекладачами задля полегшення їх роботи та економії часу. Однак, можна сказати що у майбутньому можна чекати на ще кращі системи машинного перекладу які можливо, з часом, зможуть повністю замінити роботу перекладача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Евдокимов А.С. Искусство машинного перевода. М.: Hard'N'Soft, 2005. 86-91 с.
2. Кнут Д. Мистецтво програмування. Основні алгоритми. К: Нова Книга, 2002. 708 с.
3. Кот Ю. А. Автоматизована система двомовного перекладу за допомогою маркерів: дис. 81'322.4. Київ, 2018. 131 с.
4. Кузьменко Д.М. Комп'ютерна лінгвістика і Web 2.0. К.: Наукова думка, 2009. 250 с.
5. Мацак Ж.Г., Скоробогатова Т.І. Машинний переклад та його специфіка. Кременчуцький державний університет. Кременчук, 2009. С.1-2. URL: http://www.rusnauka.com/30_NIEK_2009/Philologia/54186.doc.htm
6. Нікітін І. К. Структурно-семантичні трансформації в науково-технічному тексті при машинному перекладі в сучасній англійській мові. К.: Національний авіаційний університет, 2011. 142 с.
7. Матвеев С. Рівень адекватності машинного перекладу (на матеріалі перекладів виконаних системами Google Translate, PROMT та Microsoft Translate) // Теорія та практика перекладу: матеріали міжнародної науково-практичної конференції: Одеса, Центр фіолологічних досліджень, 23-24 жовтня 2020 р. / С. М. Матвеев – С. 111 – 115.
8. Пиотровский Р.Г. Текст, машина, человек. М.: Наука, 1987. 160 с.
9. Прохорова Е.А. Компьютер как инструмент. К.: Випол, 2000. 198 с.
10. Рейтблат А.И. Комментарий в эпоху интернета. М.: Новое литературное обозрение, 2004. 82-90 с.
11. Смірнова Т. Машинний переклад у сучасному суспільстві. К: 2016. 4 с.

12. Хвощ П.М. Прикладна лінгвістика сьогодні. К.: Світ, 2004. 210 с.
13. Яфтумова А. А. Системи машинного перекладу як складова підготовки майбутніх перекладачів / А. А. Яфтумова // Діалог мов і культур у сучасному освітньому просторі : наук. конф, м. Суми 17 листопада 2017 р. Суми, 2017. С. 573-576.
14. Amparo A. Translation technologies. Scope, tools and resources. Valencia: Jaume I Univeristy, 2008. 79-102 с.
15. Bangalore S., Haffner P., Kanthak S. Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction. Prague: AT&T Labs – Research, 2007. 153-159 с.
16. Bowker L., Barlow M. A comparative evaluation of bilingual concordances and translation memory systems. Auckland: The University of Auckland, 2008. 1-22 с.
17. Brown P. F., Lai J. C., Mercer R. L. Aligning Sentences in Parallel Corpora. Berkeley: UC Berkeley, 1991. 169–176 с.
18. Burchardt A. Comparing Errors: Neural MT vs. Traditional Phrase-based and Rule based MT. Globalization and Localization Association. 2017. C. 1-4. URL: <https://www.gala-global.org/publications/comparing-errors-neural-mt-vs-traditional-phrase-based-and-rule-based-mt>.
19. Christensen T. P. Translation memory-systemer som vaerktoj til juridisk oversættelse.. Odense: University of Southern Denmark, 2003. 446 с.
20. Colominas C. Towards chunk-based translation memories. In Babel. 2008. 54, №4. С. 343-354. DOI: doi10.1075/babel.54.4.03col.
21. Copestake A., Briscoe T., Lexical operations in a unification-based framework. SIGLEX Workshop Berkeley. 1991. C. 88-100. DOI: 10.1007/3-540-55801-2_30.
22. Cowie J., Guthrie G., Guthrie J., Lexical disambiguation using simulated annealing. COLING92. 1992. C. 359-365. DOI: <https://doi.org/10.3115/992066.992125>.

23. Dillon S., Fraser J., Translators and TM: An investigation of translators' perceptions of translation memory adoption. In *Machine Translation*. 2006. 20, №2. C. 67-79. DOI: 10.1007/s10590-006-9004-8.
24. Dragsted B. *Segmentation in Translation and Translation Memory Systems: an empirical investigation of cognitive segmentation and effects of integrating a TM system into the translation process*. Copenhagen: Samfundslitteratur, 2004. 455 c.
25. Dragsted B. Computer-aided translation as a distributed cognitive task. In *Pragmatics & Cognition*. 2006. 14, №2. C. 443-464. DOI: 10.1075/pc.14.2.17dra.
26. Farwell D. *Pragmatics-based machine translation*. London: Routledge, 2015. 167-185 c.
27. Forcada M. L. *Open-source machine translation technology*. – London: Routledge, 2015. 152-166 c.
28. Fulford H., Granell-Zafra J. *Translation and Technology: a Study of UK Freelance Translators*. Loughborough: Loughborough University, 2005. 17 c.
29. Garcia I. Power-shifts in web-based translation memory. In *Machine Translation*. 2007. 21. C. 55-68. DOI: 10.1007/s10590-008-9033-6.
30. Givon T. *The structure of ellipsis*. Santa Monica: Systems Development Corp, 2000. 142 c.
31. Gouadec D. *Translation as a profession*. Amsterdam: John Benjamins, 2007. 396 c.
32. Helmreich S. Translation differences and pragmatics-based MT. *Machine Translation*. 1998. 13, № 1. C. 17-39. DOI: 10.1023/A:1008062303478.
33. Hutchins J. Current and potential applications of machine translation. *Hutchins Web*. 2012. C.15 URL: <http://www.hutchinsweb.me.uk/Aslib-2012.pdf>.

34. Hutchins J. Machine translation: A concise history. Theory and practice, 2007. 21 c.
35. Hutchins, J. The Origins of the Translator's Workstation. Norwich: University of East Anglia, 1998. 287-307 c.
36. Kilgarriff A. Dictionary Word-Sense Distinctions: an Enquiry into their Nature. Computers and the Humanities. 1993. 26, №5. C. 365-387. DOI: 10.1007/978-3-319-75477-2_1.
37. Knuth D. The Art Of Computer Programming. Stanford: Stanford University, 1997. 708 c.
38. Lagarda A. L. Statistical Post-Editing of a Rule-Based Machine Translation System. Valencia: University of Valencia, 2001. 217-220 c.
39. Lagoudaki E. Translation Memories Survey 2006. London: Imperial College London, 2006. 36 c.
40. Lange C. A., Bennet W. S. Combining machine translation with translation memory at Baan. In Sprung, Robert. 2000. C. 203-218. DOI: <https://doi.org/10.1075/ata.xi.21lan>.
41. LISA. LISA Translation Memory Survey 2002. Washington: LISA, 2002. 19 c.
42. LISA. LISA Translation Memory Survey 2004. Washington: LISA, 2004. 22 c.
43. Mandreoli F., Martoglia R., Tiberio P. EXTRA: a system for example-based translation assistance. In Machine Translation. 2006. 20, №3. C. 167-197. DOI: 10.1007/s10590-007-9023-0.
44. Microsoft Translator launching Neural Network based translations for all its speech languages. URL: <https://www.microsoft.com/en-us/translator/blog/2018/11/14/nextgenmt/> (дата звернення: 14.09.2020).
45. Miller G. A. WordNet: A Dictionary Browser. Waterloo: University of Waterloo, 2001. 75 c.

46. Moore, G.C., Benbasat I. Development of an instrument to measure the perceptions of adopting an information technology innovation. In *Inform Sys Res.* 2001. 2. C. 192-221. DOI: <https://doi.org/10.1287/isre.2.3.192>.

47. O'Brien S. Eye-tracking and translation memory matches. In *Perspectives: Studies in Translatology.* 2006. 14, №3. C.185-205. DOI: [10.1080/09076760708669037](https://doi.org/10.1080/09076760708669037).

48. O'Brien S. Processing fuzzy matches in translation memory tools: an eye tracking analysis. Copenhagen: Samfundslitteratur, 2008. 79-102 c.

49. O'Hagan M. Computer-aided translation (CAT). London: Routledge, 2009. 48-51 c.

50. Shankland S. Google Translate now serves 200 million people daily. CNET. 2014. URL: <https://www.cnet.com/news/google-translate-now-serves-200-million-people-daily/>.

51. Shiwen Yu. Rule-based machine translation. London: Routledge, 2015. 186-200 c.

52. Sommerlad J. Google Translate: How does the search giant's multilingual interpreter actually work? Independent. 2018. URL: <https://www.independent.co.uk/life-style/gadgets-and-tech/news/google-translate-how-work-foreign-languages-interpreter-app-search-engine-a8406131.html>.

53. Somers H. Computers and Translation. Amsterdam: John Benjamins Publishing, 2003. 349 c.

54. Translate by speech. URL: <https://support.google.com/translate/answer/6142468> (дата звернення: 17.10.2020).

55. Translate text messages, webpages, or documents. URL: <https://support.google.com/translate/answer/2534559> (дата звернення: 17.10.2020).

56. Microsoft Translator Hub: Discussion of BLEU. URL: <https://www.microsoft.com/en-us/research/video/microsoft-translator-hub-discussion-of-bleu-score/> (дата звернення: 17.10.2020).
57. What are the main types of machine translation? URL: <http://www.machinetranslation.net/quick-guide-to-machine-translation/machine-translation-technologies> (дата звернення: 20.10.2020).
58. Wheatley A. eCoLoRe (eContent Localization Resources for Translator Training) TranslationDirectory. 2003. С. 15. URL: <http://www.translationdirectory.com/article450.htm>.
59. Wilks Y. Machine Translation. Sheffield: University of Sheffield, 2009. 157-177 с.
60. Wilks Y. Grammar, Meaning and the Machine Analysis of Natural Language. London: Routledge. Bloomington: American journal of computational linguistics, 1999. 46-49 с.
61. Wilks Y. Making Preferences More Active. Mimeo Edinburgh. 1999. 11, №3. С. 197-223. DOI: [https://doi.org/10.1016/0004-3702\(78\)90001-2](https://doi.org/10.1016/0004-3702(78)90001-2).
62. Williams P., Sennrich R., Post M. Syntax-based Statistical Machine Translation. New York: Morgan & Claypool, 2016. 188 с.
63. Younghui W. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Ithaca: Cornell University, 2016. 325 с.
64. Zhou S. Has AI surpassed humans at translation? Not even close! Skynet Today. 2018. С. 10. URL: https://www.skynettoday.com/editorials/state_of_nmt.

ДОДАТКИ

Додаток А

Уривок №1 з Knuth D. The Art Of Computer Programming.

The notion of an algorithm is basic to all of computer programming, so we should begin with a careful analysis of this concept. The word "algorithm" itself is quite interesting; at first glance it may look as though someone intended to write "logarithm" but jumbled up the first four letters. The word did not appear in Webster's New World Dictionary as late as 1957; we find only the older form "algorism" with its ancient meaning, the process of doing arithmetic using Arabic numerals. During the Middle Ages, abacists computed on the abacus and algorists computed by algorism. By the time of the Renaissance, the origin of this word was in doubt, and early linguists attempted to guess at its derivation by making combinations like *algiros* [painful] + *arithmos* [number]; others said no, the word comes from "King Algor of Castile." Finally, historians of mathematics found the true origin of the word algorism: It comes from the name of a famous Persian textbook author, Abu 'Abd Allah Muhammad ibn Musa al-Khwarizmī (c. 825) — literally, "Father of Abdullah, Mohammed, son of Moses, native of Khwarizm." The Aral Sea in Central Asia was once known as Lake Khwarizm, and the Khwarizm region is located in the Amu River basin just south of that sea. Al-Khwarizm wrote the celebrated book *Kitab aljabr wa'l-muqabala* ("Rules of restoring and equating"); another word, "algebra" stems from the title of his book, which was a systematic study of the solution of linear and quadratic equations. [For notes on al-Khwarizmī's life and work, see H. Zemanek, *Lecture Notes in Computer Science* 122 A981, 1-81.] Gradually the form and meaning of algorism became corrupted; as explained by the Oxford English Dictionary, the word "passed through many pseudo-etymological perversions, including a recent algorithm, in which it is learnedly confused" with the Greek root of the word arithmetic. This change from "algorism" to "algorithm" is not hard to understand in view of the fact

that people had forgotten the original derivation of the word. An early German mathematical dictionary, *Vollst ndiges mathematisches Lexicon* (Leipzig: 1747), gave the following definition for the word *Algorithmus*: "Under this designation are combined the notions of the four types of arithmetic calculations, namely addition, multiplication, subtraction, and division." The Latin phrase *algorithmus infinitesimalis* was at that time used to denote "ways of calculation with infinitely small quantities, as invented by Leibniz."

By 1950, the word algorithm was most frequently associated with Euclid's algorithm, a process for finding the greatest common divisor of two numbers that appears in Euclid's *Elements* (Book 7, Propositions 1 and 2). It will be instructive to exhibit Euclid's algorithm here:

Algorithm E (Euclid's algorithm). Given two positive integers m and n , find their greatest common divisor, that is, the largest positive integer that evenly divides both m and n .

E1. [Find remainder.] Divide m by n and let r be the remainder. (We will have $0 \leq r < n$.)

E2. [Is it zero?] If $r=0$, the algorithm terminates; n is the answer.

E3. [Reduce.] Set $m \leftarrow n$, $n \leftarrow r$, and go back to step E1.

Of course, Euclid did not present his algorithm in just this manner. The format above illustrates the style in which all of the algorithms throughout this book will be presented. Each algorithm we consider has been given an identifying letter (E in the preceding example), and the steps of the algorithm are identified by this letter followed by a number (E1, E2, E3). The chapters are divided into numbered sections; within a section the algorithms are designated by letter only, but when algorithms are referred to in other sections, the appropriate section number is attached. For example, we are now in Section 1.1; within this section Euclid's algorithm is called Algorithm E, while in later sections it is referred to as Algorithm 1.1E.

Переклад виконаний статистичною СМП (Google Translate)

Поняття алгоритму є основою всіх комп'ютерного програмування, тому ми повинні почати з ретельного аналізу цієї концепції. Слово «алгоритм» саме по собі досить цікаво, на перший погляд може здатися, ніби хтось мав намір писати «логарифм», але неясний перших чотирьох букв. Слово не з'являлося в Новому Світовому Словнику Вебстера вже в 1957; ми знаходимо тільки стару форму «алгоритм» з його древніми сенс, процес виконання арифметичних операцій використовуються арабські цифри. У середні століття, abacists обчислюється на рахунках і algorists обчислюється алгоритм. На той час, епохи Відродження, походження цього слова було сумнівів, і на початку лінгвісти намагалися вгадати його висновок шляхом комбінації, як *algiros* [болючим] + *arithmos* [число], інші сказали, немає, слово походить від «короля Algor Кастилії». Нарешті, історики математики виявили справжнє походження слова алгоритм: Воно походить від назви відомого перського автора підручника, Абд Абу Аллаха Мухаммад ібн Муса аль-Хорезм (бл. 825), буквально, «Батько Абдулла Мухаммед, сина Мойсея, уродженець Хорезма». Колись Аральське море в Центральній Азії було відоме як озеро Хорізм, і в регіоні Хорезма розташований в басейні річки Амудар'ї на південь від цього моря. Аль-Хорезм написав знамениту книгу *Кітаб aljabr wa'l-muqabala* («Правила відновлення і прирівнюючи»); інше слово, «алгебра» походить від назви своєї книги, яка була систематичне вивчення рішення лінійних і квадратичних рівнянь. [Для замітки про життя аль-Khwarizmi і роботи, див. Н. ЗЕМАНЕК, Лекції з інформатики 122 A981), 1-81.]

Поступово форму і зміст алгоритм була пошкоджена; як пояснив Оксфордський словник англійської мови, слово «пройшла через багато псевдо-етимологічний збочень, в тому числі останні алгоритму, в якому вона знанням плутають» з грецького кореня слова арифметика. Це перехід від «алгоритм» на «Алгоритм» не важко зрозуміти, в силу того, що люди забули оригінальний висновок слова. Раннього німецького математичний

словник, *Vollst'ndiges mathematisches Lexikon* (Лейпциг: 1747 Отримати), дав наступне визна- поділ слова *Algorithmus*: «Відповідно до цього позначенням поєднуються поняття чотирьох типів арифметичних розрахунків, а саме додавання, множення, віднімання, і Відділ». Латинська *infinitesimalis algorithmus* фраза була в цей час використовується для позначення «способи розрахунку з нескінченно малими величинами, як винайшли Лейбніцем».

До 1950 року, слово алгоритм найчастіше пов'язані з алгоритмом Евкліда, процес для знаходження найбільшого загального дільника двох чисел, який з'являється в «Начал» Евкліда (книга 7, пропозицій 1 і 2). Це буде Повчально виставку алгоритм Евкліда тут:

Алгоритм E (алгоритм Евкліда). Дано два натуральних чисел m і n , знайти їх найбільший спільний дільник, т. E. Найбільший позитивний ціле число, рівномірно ділить m і n .

E1. [Знайти залишок.] Розділіть m на n і r бути залишок. (Ми будемо мати $0 \leq r < n$.)

E2. [Це нуль?] Якщо $r = 0$, алгоритм завершується, n є відповіддю.

E3. [Зменшити.] Безліч $m \leftarrow n$, $n \leftarrow r$, і поверніться до кроку E1.

Звичайно, Евклід не надасть свій алгоритм саме таким чином. Формат вище ілюструє стиль, в якому всі алгоритми в цій книзі будуть представлені.

Кожен алгоритм розглянемо було приділено виявленню лист (E і в попередньому прикладі), і кроки алгоритму позначені цим листом послідувала низка (E1, E2, E3). Глави розділений на пронумеровані розділи, в розділі алгоритми, позначені літерою тільки, але при алгоритмів, що згадуються в інших розділах, відповідних номер розділу додається. Наприклад, зараз ми знаходимося в розділі 1.1, в цьому розділі алгоритму Евкліда називається алгоритм E, тоді як у наступних розділах це згадується як алгоритм 1.1e.

Переклад виконаний аналітичною СМП (PROMT 9.0)

Поняття алгоритму є основним до всього програмування, таким чином, ми повинні почати з ретельного аналізу цього поняття. Саме слово "алгоритм" досить цікаве; на перший погляд це може виглядати, як ніби хтось мав намір написати "логарифм", але змішував перші чотири букви. Слово не з'являлося в Новому Світовому Словнику Вебстера вже в 1957; ми вважаємо тільки більш стару форму "десятеричною системою обчислення" з її стародавнім значенням, процесом виконання арифметичного використання арабських цифр. Під час Середньовіччя abacists, обчислений на абаку і algorists, обчислений десятиричною системою обчислення. До часу Ренесансу викликало сумнів джерело цього слова, і ранні лінгвісти спробували припустити його деривацію, роблячи комбінації як algiros [болючий] + arithmos [число]; інші сказали не, слово прибуває від "Короля Алгора Кастилії." Нарешті, історики математики знайшли справжнє джерело десятиричної системи обчислення слова: Це прибуває з імені відомого перського автора підручника, Абу "Абд Аллах Мухаммед ібн Муса al-Khwarizmī (с. 825) - буквально, "Батьківський елемент Абдулли, Мохаммеда, сина Мойсея, уродженця Khwarizm." Аральське море в Середній Азії колись було відомо як озеро Хваризм, і область Khwarizm розташована в Басейні річки Аму тільки на південь від того моря. Аль-Хваризм записав знаменитій книзі *Kitab aljabr wa 'l-muqabala* ("Правила відновлення і прирівнювання"); інше слово, "алгебра" походить від заголовка його книги, яка була систематичним дослідженням вирішення лінійних і квадратних рівнянь. [Для приміток щодо життя al-Khwarizmī і роботи, див. Х. Земенека, Примітки Лекції в Інформатиці 122 A981), 1-81.]

Поступово форма і значення десятиричної системи обчислення стали пошкодженими; як пояснено Оксфордським англійським Словником, слово "проходило через багато псевдоетимологічних збочень, включаючи недавній алгоритм, в якому це зі знанням справи

переплутано" з грецьким коренем арифметики слова. Ця зміна від "десятеричної системи обчислення" до "алгоритму" не важко зрозуміти через факт, що люди забули вихідну деривацію слова. Ранній німецький математичний словник, Словник Vollstk 'ndiges mathematisches (Лейпциг: 1747), дав таке визначення для слова Algorithmus: "Під цим позначенням об'єднані поняття чотирьох типів арифметичних обчислень, а саме, доповнення, множення, віднімання, і підрозділи". Латинська фраза *algorithmus infinitesimalis* тоді використовувалася, щоб позначити "способи обчислення з нескінченно невеликими кількостями, як винайдено Лейбніцем."

До 1950, алгоритм слова найбільш часто зв'язувався з алгоритмом Євкліда, процесом для того, щоб знайти найбільший спільний дільник двох чисел, який з'являється в Елементах Євкліда (Книга 7, Судження 1 і 2). Це буде повчально, щоб показати алгоритм Євкліда тут:

Алгоритм E (алгоритм Євкліда). Враховуючи два позитивних цілих числа m і n , знайдіть їх найбільший спільний дільник, тобто, найбільше позитивне ціле число, яке рівномірно ділить обох m і n .

E1. [Знайдіть останок.] Діляться m на n і r дозволяють бути залишком. (Ми будемо мати $0 \leq r < n$.)

E2. [Це - нуль?] Якщо $r=0$, алгоритм завершується, n ; відповідь.

E3. [Зменшити.] Набір $m \leftarrow n$, $n \leftarrow r$, і повертається до кроку E1.

Звичайно, Евклід не представляв свій алгоритм тільки цим способом. Наведений вище формат ілюструє стиль, в якому будуть представлені всі алгоритми по всій книзі. Кожному алгоритму, який ми розглядаємо, дали букву ідентифікації (E в попередньому прикладі), і кроки алгоритму ідентифіковані цією літерою, що супроводжується числом (ель, E2, E3). Глави розділені на пронумеровані розділи; у межах розділу алгоритми визначаються за літерами тільки, але коли алгоритми згадані в інших розділах, відповідне число розділу приєднано. Наприклад, тепер ми знаходимося в Розділі 1.1; у межах цього розділу алгоритм

Євкліда викликають Алгоритмом Е, тоді як у наступних розділах він згадується як Алгоритм 1.1Е.

Переклад виконаний професійним перекладачем (Кнут Д. Мистецтво програмування)

Поняття алгоритм є основним для всієї області комп'ютерного програмування, тому почати ми повинні з ретельного аналізу цього терміна. Слово «алгоритм» (algorithm) вже саме по собі представляє великий інтерес. На перший погляд може здатися, ніби хтось збирався написати слово «логарифм» (logarithm), але випадково переставив перші чотири букви. Цього слова ще не було в виданні словника Webster's New World Dictionary, що вийшов в 1957 році. Ми знаходимо там тільки застарілу форму «algorism» - старовинне слово, яке означає «виконання арифметичних дій за допомогою арабських цифр». В середні віки абакісти вважали на абаки (рахункових дошках), а алгоритміки використовували «algorism». В епоху Відродження походження цього слова виявилось забутиим. Одні лінгвісти того часу намагалися пояснити його значення шляхом поєднання слів *algiros* [хворий] і *arithmas* [число], інші не погоджувалися з таким тлумаченням і стверджували, що це слово походить від «King Algor of Castile». Нарешті історики математики виявили справжнє походження слова «algorism»: воно бере початок від імені автора знаменитого перського підручника з математики, Abu 'Abd Allah Muhammad ibn Musa al-Khwarizmi (Абу Абд Аллах Мухаммед ібн Муса аль-Хорезмі) (бл. 825 м), що означає буквально «Батько Абдулли, Мухаммед, син Муси, уродженець Хорезма». Аральське море в Центральній Азії було колись називали озеро Хорезм, і район Хорезма (Khwarizm) розташований в басейні річки Амудар'ї південніше цього моря. Аль-Хорезмі написав знамениту книгу *Kitab al-jabr wa'l-muqabala* (Кітаб аль-джебр валь-мукабала «Книга про відновлення та зіставлення»). Від назви цієї книги, яка була присвячена вирішенню лінійних і

квадратних рівнянь, відбулося ще одне слово - «алгебра». [Про життя і наукової діяльності аль-Хорезмі йдеться в роботі Н. Zemanek, Lecture Notes in Computer Science 122 (1981), 1-81.]

Поступово форма і значення слова *algorism* спотворилися; як пояснюється в словнику Oxford English Dictionary, це слово «зазнало безліч псевдоетимологічних спотворень, включаючи останній варіант *algorithm*, де сталася плутанина» з коренем слова грецького походження *arithmetic*. Цей перехід від «*algorism*» до «*algorithm*» здається цілком закономірним з огляду на те, що походження розглянутого слова було повністю забуте. У старовинному німецькому математичному словнику *Vollstandiges mathematisches Lexicon* (Leipzig, +1747) дається таке визначення слова *algorithmus*: «Цей термін включає в себе поняття про чотири типи арифметичних операцій, а саме: про складання, множення, віднімання і розподілі». Латинський вираз *algorithmus infinitesimalis* в той час використовувалося для визначення «способів виконання дій з нескінченно малими величинами, відкритих Лейбніцем (Leibniz)».

До 1950 року слово «алгоритм» найчастіше асоціювалося з алгоритмом Евкліда, який являє собою процес знаходження найбільшого спільного дільника двох чисел. Цей алгоритм наведено в книзі Евкліда (Euclid) Почала (книга 7, пропозиції 1 і 2). Думаю, має сенс навести тут опис цього алгоритму.

Алгоритм E (Алгоритм Евкліда). Дано два цілих позитивних числа m і n . Потрібно знайти їх найбільший спільний дільник, т. E. Найбільше ціле позитивне число, яке без остачі ділить обидва числа m і n .

E1. [Знаходження залишку.]

Розділимо m на n , і нехай залишок від ділення буде дорівнює r . (Де $0 \leq r < n$).

E2. [Порівняння з нулем.]

Якщо $r = 0$, то виконання алгоритму припиняється; n - шукане значення.

ЕЗ. [Заміщення.]

Присвоїти $t \leftarrow n$, $n \leftarrow r$, і повернутися до кроку ЕІ.

Зрозуміло, у Евкліда цей алгоритм сформульовано не зовсім так. Наведене вище формулювання ілюструє стиль, в якому алгоритми будуть представлені протягом всієї книги.

Кожного розглянутого алгоритму присвоюється ідентифікує буква (в попередньому прикладі використовувалася буква Е), а кроків алгоритму - ця ж буква в поєднанні з числом (ЕІ, Е2, ЕЗ). Глави книги поділяються на пронумеровані розділи; всередині розділу алгоритми позначаються тільки буквами. Але коли на ці алгоритми робляться посилання з інших розділів, то до букви приєднується номер відповідного розділу. Наприклад, зараз ми знаходимося в розділі 1.1; всередині цього розділу алгоритм Евкліда називається «Алгоритм Е», але посилатися на нього в наступних розділах ми будемо як на алгоритм 1.1Е.

Додаток Б

Уривок №2 з Knuth D. The Art Of Computer Programming.

As an example of the use of doubly linked lists, we will now consider the writing of a discrete simulation program. "Discrete simulation" means the simulation of a system in which all changes in the state of the system may be assumed to happen at certain discrete instants of time. The "system" being simulated is usually a set of individual activities that are largely independent although they interact with each other; examples are customers at a store, ships in a harbor, people in a corporation. In a discrete simulation, we proceed by doing whatever is to be done at a certain instant of simulated time, then advance the simulated clock to the next time when some action is scheduled to occur.

By contrast, a "continuous simulation" would be simulation of activities that are under continuous changes, such as traffic moving on a highway, spaceships traveling to other planets, etc. Continuous simulation can often be satisfactorily approximated by discrete simulation with very small time

intervals between steps; however, in such a case we usually have "synchronous" discrete simulation, in which many parts of the system are slightly altered at each discrete time interval, and such an application generally calls for a somewhat different type of program organization than the kind considered here.

The program developed below simulates the elevator system in the Mathematics building of the California Institute of Technology. The results of such a simulation will perhaps be of use only to people who make reasonably frequent visits to Caltech; and even for them, it may be simpler just to try using the elevator several times instead of writing a computer program. But, as is usual with simulation studies, the methods we will use are of much more interest than the answers given by the program. The methods to be discussed below illustrate typical implementation techniques used with discrete simulation programs.

The Mathematics building has five floors: sub-basement, basement, first, second, and third. There is a single elevator, which has automatic controls and can stop at each floor. For convenience we will renumber the floors 0, 1, 2, 3, and 4.

On each floor there are two call buttons, one for UP and one for DOWN. (Actually floor 0 has only UP and floor 4 has only DOWN, but we may ignore that anomaly since the excess buttons will never be used.) Corresponding to these buttons, there are ten variables `CALLUP [j]` and `CALLDOWN [j]`, $0 \leq j \leq 4$. There are also variables `CALLCAR[j]`, $0 \leq j \leq 4$, representing buttons within the elevator car, which direct it to a destination floor. When a person presses a button, the appropriate variable is set to 1; the elevator clears the variable to 0 after the request has been fulfilled.

So far we have described the elevator from a user's point of view; the situation is more interesting as viewed by the elevator. The elevator is in one of three states: `GOINGUP`, `GOINGDOWN`, or `NEUTRAL`. (The current state is indicated to passengers by lighted arrows inside the elevator.) If it is in

NEUTRAL state and not on floor 2, the machine will close its doors and (if no command is given by the time its doors are shut) it will change to GOINGUP or GOINGDOWN, heading for floor 2. (This is the "home floor," since most passengers get in there.) On floor 2 in NEUTRAL state, the doors will eventually close and the machine will wait silently for another command. The first command received for another floor sets the machine GOINGUP or GOINGDOWN as appropriate; it stays in this state until there are no commands waiting in the same direction, and then it switches direction or switches to NEUTRAL just before opening the doors, depending on what other commands are in the CALL variables. The elevator takes a certain amount of time to open and close its doors, to accelerate and decelerate, and to get from one floor to another. All of these quantities are indicated in the algorithm below, which is much more precise than an informal description can be. The algorithm we will now study may not reflect the elevator's true principles of operation, but it is believed to be the simplest set of rules that explain all the phenomena observed during several hours of experimentation by the author during the writing of this section.

Переклад виконаний статистичною СМП (Google Translate)

Як приклад використання двічі пов'язані списки, ми розглянемо написання окремої програми моделювання. "Дискретне моделювання" означає моделювання системи, в якій всі зміни в стані системи можна вважати статися при певних дискретні моменти часу. "Система" моделюється зазвичай набір окремих заходів, які в значній мірі незалежні, хоча вони взаємодіють один з одним; приклади клієнтів в магазині, кораблі в порту, люди в корпорації. У дискретного моделювання, ми проведемо робити те, що повинно бути зроблено в певний момент часу моделювання, то заздалегідь моделюється годинник, щоб наступного разу, коли будь-яку дію планується відбутися.

На відміну від цього, "безперервне моделювання" буде моделювання діяльності, які перебувають під постійним змінам, наприклад, трафік, що проходить по шосе, космічні кораблі подорожі на інші планети і т.д. безперервного моделювання часто може бути задовільно апроксимується дискретного моделювання з дуже малими інтервалами часу між кроками, проте в такому випадку ми, як правило, "синхронні" дискретного моделювання, в якому багато частин системи злегка змінений в кожен дискретний інтервал часу, і такі програми зазвичай вимагає дещо інший тип організації, ніж програми тут розглядається.

Програма, розроблена нижче імітує ліфт системи в будівлі математики з Каліфорнійського технологічного інституту. Результати такого моделювання, можливо, бути корисним тільки для людей, які роблять досить частих візитів в Каліфорнійському технологічному інституті, і навіть для них, це може бути простіше, просто спробуйте використовувати ліфт кілька разів замість написання комп'ютерної програми. Але, як завжди, з імітаційні дослідження, методи, які ми будемо використовувати мають набагато більший інтерес, ніж відповідей за програмою. Методи, які будуть обговорюватися нижче, ілюструють типові методи здійснення використовуватися з дискретними програм моделювання. Будівля математики має п'ять поверхів: суб-підвал, підвал, перший, другий, і третій. Існує один ліфт, який має автоматичний контроль і може зупинитися на кожному поверсі. Для зручності ми будемо змінити нумерацію поверхів 0, 1, 2, 3 і 4.

На кожному поверсі Існують дві кнопки виклику, по одному для UP і один для ВНИЗ. (Насправді слово 0 має тільки вгору і 4 поверх має тільки вниз, але ми можемо ігнорувати цю аномалію з надлишковою кнопки ніколи не буде використовуватися.) Відповідно до цих кнопками, Є десять змінних CALLUP [J] і CALLDOWN [J], $0 \leq J \leq 4$.

Є також змінні CALLCAR [J], $0 \leq J \leq 4$, що становить кнопки в ліфті, які направляють його на потрібний поверх. Коли людина натискає кнопку, відповідну змінна встановлена ?? в 1, ліфт очищає змінну в 0 після прохання було виконано.

До сих пір ми описали ліфт з точки зору користувача, ситуація цікавіша, якщо дивитися на ліфті. Ліфт знаходиться в одному з трьох станів: GOINGUP, GOINGDOWN, або нейтральним. (Поточний стан вказується на пасажирів освітлені стрілками всередині ліфта.) Якщо вона знаходиться в стані нейтральними і не на поверх 2, машина закриє свої двері і (якщо команда не дає часу свої двері закриті) він зміниться на GOINGUP або GOINGDOWN, прямуючи до поверх 2 . (Це "будинок поверсі", так як більшість пасажирів отримати там.) На поверсі 2 в нейтральному стані, двері, в кінцевому рахунку близьких і машина буде чекати мовчки для іншої команди. Перша команда отримала ще пів безлічі машин GOINGUP або GOINGDOWN в міру необхідності, вона залишається в цьому стані, поки Є немає команди очікування в тому ж напрямку, а потім відбувається перемикання напрямку або перемикається на нейтральній незадовго до відкриття дверей, в залежності від того, що інші команди в CALL змінних. Ліфт займає певну кількість часу, щоб відкривати і закривати свої двері, з метою прискорення і уповільнення, і щоб потрапити з одного поверху на інший. Всі ці величини вказані в нижче алгоритм, який є набагато більш точним, ніж неформальне опис може бути. Алгоритм Вивчимо тепер можуть не відображати дійсних принципів ліфта роботи, але вважається, що це найпростіший набір правил, які пояснюють все явища, які спостерігаються протягом декількох годин експериментів автора під час написання цього розділу.

Переклад виконаний аналітичною СМП (PROMT 9.0)

Як приклад використання двонаправлених пов'язаних списків, ми тепер розглянемо запис дискретної програми моделювання. "*Дискретне*

модельовання» означає моделювання системи, в якій можна припустити, що всі зміни в стані системи відбуваються в певні дискретні моменти часу. Модельована "система" зазвичай є низкою окремих дій, які значною мірою незалежні, хоча вони взаємодіють з один одним; приклади - клієнти в сховищі, поставках в гавані, людях в корпорації. На дискретному моделюванні ми продовжуємо, роблячи незалежно від того, що має бути зроблено в певний момент моделюваного часу, потім удосконалити моделювані годинники до наступного разу, коли деяка дія, як планують, відбудеться.

На відміну від цього, "безперервне моделювання" було б моделюванням дій, які знаходяться під безперервними змінами, такими як трафік, поглиблюючий магістраль, космічні кораблі, що переміщаються в інші планети, і т. д. Безперервне моделювання може часто задовільно наближатися дискретним моделюванням з дуже маленькими часовими інтервалами між кроками; однак, в такому випадку у нас зазвичай є "синхронне" дискретне моделювання, на якому багато частин системи трохи змінені в кожному інтервалі дискретного часу, і такий додаток зазвичай закликає до дещо різного типу організації програми ніж вид, який розглядають тут.

Програма, розроблена нижче, моделює систему ліфта в будівлі Математики Каліфорнійського Технологічного інституту. Результати такого моделювання будуть, можливо, корисні тільки для людей, які завдають розумно часті візити в Каліфорнійський технологічний інститут; і навіть для них, може бути більш просто тільки спробувати використовувати ліфт кілька разів замість того, щоб писати комп'ютерну програму. Але, як завжди з дослідженнями моделювання, методи, які ми будемо використовувати, мають набагато більше інтересу ніж відповіді, дані програмою. Методи, які будуть обговорені нижче, ілюструють типові методи реалізації, що використовуються з дискретними програмами моделювання. У будівлі Математики є п'ять поверхів: підвал,

підвал, по-перше, по-друге, і третина. Є єдиний ліфт, який має автоматичні управління і може зупинитися в кожній підлозі. Для зручності ми перенумеруємо поверхи 0, 1, 2, 3, і 4.

На кожній підлозі є дві кнопки виклику, один для і один для ВНИЗ. (Фактично підлога 0 має тільки, і старт 4 має тільки ВНИЗ, але ми можемо проігнорувати ту аномалію, оскільки надлишкові кнопки ніколи не будуть використовуватися.) Відповідність цих кнопок, є десять ЗАЛУЧЕНЬ змінних [j] і ДОГАНА [j], $0 \leq j \leq 4$. Є також змінні CALLCAR [j], $0 \leq j \leq 4$, представляючи кнопки в межах автомобіля ліфта, які направляють це до цільової статі. Коли людина натискає кнопку, відповідна змінна встановлена в 1; ліфт очищає змінну до 0 після того, як запит був виконаний.

Досі ми описали ліфт з точки зору користувача; ситуація більш цікава як ліфт, що проглядається. Ліфт знаходиться в одному з трьох станів: GOINGUP, GOINGDOWN, або НЕЙТРАЛЬНИЙ. (Поточний стан позначено пасажирам освітленими стрілками в ліфті.), Якщо це знаходиться в НЕЙТРАЛЬНОМУ стані а не на підлозі 2, машина закриє свої двері і (якщо ніяка команда не буде дана до того часу, коли її двері закриті), це зміниться на GOINGUP або GOINGDOWN, прямуючи в підлогу 2. (Це - "домашня статя", оскільки більшість пасажирів входить там.) На підлозі 2 в НЕЙТРАЛЬНОМУ стані, в кінцевому рахунку закриваються двері, і машина буде очікувати тихо іншої команди. Перша команда, отримана для іншої статі, встановлює машину GOINGUP або GOINGDOWN як відповідну; це залишається в цьому стані, поки немає ніяких команд, що очікують в тому ж самому напрямку, і потім це перемикає напрямок або перемикається на НЕЙТРАЛЬНИЙ перш, ніж відкрити двері, в залежності від того, що інші команди знаходяться в змінних ВИКЛИКУ. Ліфт бере певну кількість часу, щоб відкрити і закрити його двері, прискоритися і сповільнитися, і дістатися від однієї статі до іншої. Всі ці кількості позначені в алгоритмі нижче, який набагато

більш точний, ніж неофіційний опис може бути. Алгоритм, який ми тепер вивчимо, можливо, не відображає справжні принципи ліфта роботи, але вважається, що це найпростіший набір правил, які пояснюють всі явища, що спостерігаються протягом декількох годин експериментування автором під час запису цього розділу.

Переклад виконаний професійним перекладачем (Кнут Д. Мистецтво програмування)

Як приклад використання двічі пов'язаних списків розглянемо програму дискретного моделювання. "Дискретне моделювання" означає моделювання системи, в якій передбачається, що всі зміни стану системи відбуваються в деякі дискретно задані моменти часу. Моделювана "система" зазвичай являє собою набір окремих дійових осіб, які, хоча і можуть взаємодіяти один з одним, в основному, ведуть себе незалежно.

Наприклад, це можуть бути покупці в магазині, кораблі в гавані, співробітники деякого підприємства. При цьому процес моделювання полягає у виконанні певних дій, передбачених для даного моменту, для переходу до наступного моменту з подальшим виконанням інших дій, запланованих для нового моменту.

І навпаки, "безперервне моделювання" означає моделювання дій, які відбуваються безперервно, наприклад рух автомобілів по автостраді, польоти космічних кораблів до інших планет і т. Д. Безперервне моделювання часто можна цілком задовільно імітувати за допомогою дискретного моделювання з дуже малими інтервалами часу між сусідніми кроками. Але в такому разі вийде "синхронне" дискретне моделювання, при якому багато частин системи злегка змінюються на кожному дискретному часовому інтервалі, і такий додаток зазвичай потребує організації програми дещо іншого типу, ніж той, який розглянуто тут.

Наведена нижче програма моделює роботу ліфта в будівлі факультету математики Каліфорнійського технологічного інституту

(Калтех). Результати такого моделювання, ймовірно, будуть корисні тільки тим, кому часто доводиться відвідувати Калтех. І навіть їм, мабуть, простіше буде всього кілька разів скористатися цим ліфтом, ніж створювати спеціальну програму. Але, як завжди трапляється при вивченні методів моделювання, методи що використовуються при програмуванні набагато цікавіше, ніж результати виконання програм. Розглянуті нижче методи ілюструють типові методики, які використовуються в програмах дискретного моделювання. Будівля факультету математики має п'ять поверхів: підвальний, цокольний, перший, другий і третій. У ньому знаходиться один ліфт з автоматичним управлінням, який може зупинятися на кожному поверсі. Для зручності перенумеруємо поверхи в наступному порядку: 0, 1, 2, 3 і 4

На кожному поверсі є дві кнопки виклику ліфта: одна - для руху вгору (UP), а інша - для руху вниз (DOWN). (Насправді на поверсі 0 є тільки кнопка UP, а на поверсі 4 - тільки кнопка DOWN, але ці особливі випадки будуть ігноруватися, тому що додаткові кнопки ніколи не будуть використовуватися)

Відповідно ці кнопки будуть позначатися десятьма змінними CALLUP [j] і CALLDOWN [j], $0 \leq j \leq 4$. Крім того, змінні CALLCAR [j], $0 \leq j \leq 4$, представлятимуть кнопки всередині кабіни ліфта, які позначають поверх призначення. При натисканні кнопки відповідної змінної присвоюється значення 1, а після виконання запиту (т.е. Після того як ліфт досягне заданого поверху) змінної присвоюється значення 0.

До сих пір робота ліфта описувалася з точки зору користувача, але ситуація стане більш цікавою, якщо розглянути її з точки зору ліфта. Ліфт може перебувати в одному з трьох наступних станів: рух вниз (GOINGUP), рух вгору (GOINGDOWN) або нейтральний стан (NEUTRAL) (Для людини поточний стан позначається світлими стрілками всередині ліфта.) Якщо ліфт знаходиться в нейтральному стані (NEUTRAL) і не на поверсі 2, механізм ліфта закриє двері і (якщо до закриття дверей не дана

ніяка команда) ліфт прийде в стан руху (GOINGUP або GOINGDOWN), прямуючи до поверху 2. (Це його "базовий поверх", так як більшість людей входять в нього саме тут) Якщо ліфт знаходиться на поверсі 2 в стані NEUTRAL, двері з часом закриються і ліфт буде чекати наступної команди. Перша отримана команда для переміщення на інший поверх призведе ліфт в стан руху GOINGUP або GOINGDOWN в залежності від натиснутої кнопки. Ліфт буде перебувати в цьому стані, поки не зупиниться. Тоді, якщо надійшли нові команди, відбудеться зміна напрямку або перехід в нейтральний стан (NEUTRAL) безпосередньо перед відкриттям дверей в залежності від того, які команди знаходяться в спричиненої послідовності. Ліфту потрібен якийсь час для відкриття і закриття дверей, прискорення і уповільнення, а також для переміщення від одного поверху до іншого. Всі ці величини вказані в наведеному нижче алгоритмі, який виглядає набагато більш строго, ніж звичні нам прості правила користування ліфтом. Цей алгоритм може і не відображати істинний принцип дії ліфта, але автор все ж вірить, що це є простим набором правил, які можуть пояснити все ті явища, які автор спостерігав в ході тривалих експериментів з ліфтом під час написання цього розділу.

Додаток В

Уривок №3 з Knuth D. The Art Of Computer Programming.

Linking automata can easily simulate graph machines, taking at most a bounded number of steps per graph step. Conversely, however, it is unlikely that graph machines can simulate arbitrary linking automata without unboundedly increasing the running time, unless the definition is changed from undirected to directed graphs, in view of the restriction to vertices of bounded degree. The linking model is, of course, quite close to the operations available to programmers on real machines, while the graph model is not.

Some of the most interesting problems to solve for such devices would be to determine how fast they can solve certain problems, or how many nodes they need to solve certain problems (for example, to translate certain formal languages). At the time this chapter was first written, several interesting results of this kind had been obtained (notably by J. Hartmanis and R. E. Stearns) but only for special classes of Turing machines having multiple tapes and read/write heads. The Turing machine model is comparatively unrealistic, so these results tended to have little to do with practical problems.

We must admit that, as the number n of nodes created by a linking automaton approaches infinity, we don't know how to build such a device physically, since we want the machine operations to take the same amount of time regardless of the size of n ; if linking is represented by using addresses as in a computer memory, it is necessary to put a bound on the number of nodes, since the link fields have a fixed size. A multitape Turing machine is therefore a more realistic model when n approaches infinity. Yet it seems reasonable to believe that a linking automaton as described above leads to a more appropriate theory of the complexity of algorithms than Turing machines do, even when asymptotic formulas for large n are considered, because the theory is more likely to be relevant for practical values of n . Furthermore when n gets bigger than 1030 or so, not even a one-tape Turing machine is realistic: It could never be built. Relevance is more important than realism. Many years have passed since the author wrote most of the comments above, and everybody can be glad that substantial progress has indeed been made on the theory of linking automata (now called pointer machines). But of course much still remains to be done.

Переклад виконаний статистичною СМП (Google Translate)

Зразки автомати можуть легко імітувати графа машин, приймаючи більш обмеженим числом кроків в графі крок. З іншого боку, однак, малоімовірно, що граф машини можуть моделювати довільні Зразки

автоматів без необмежено зростаюча час роботи, якщо визначення змінюється від неорієнтованного на орієнтовані графи, в силу обмеження на вершини обмеженій мірі. Зразки модель, звичайно, зовсім близько від операцій, доступних для програмістів на реальних машинах, а граф модель не є.

Деякі з найбільш цікавих завдань для вирішення таких пристроїв буде визначити, як швидко вони можуть вирішити деякі проблеми, або, як багато вузлів їм необхідно для вирішення певних завдань (наприклад, переклад деяких формальних мов). На той час, коли ця глава була вперше написана, кілька цікавих результатів такого роду були отримані (зокрема, Дж. Хартманіс і RE Stearns), але тільки для спеціальних класів машин Тьюрінга, що мають кілька стрічок і головок читання / запису. Модель машини Тьюрінга порівняно нереально, так що ці результати як правило, мають мало спільного з практичними проблемами.

Ми повинні визнати, що, як число n вузлів, створених за Зразки нескінченності автомата підходи, ми не знаємо, як побудувати такі пристрої фізично, так як ми хочемо машинних операцій прийняти стільки ж часу, незалежно від розміру n , якщо Зразки представлені за допомогою адрес, як в пам'яті комп'ютера, необхідно поставити оцінку числа вузлів, так як посилення поля мають фіксований розмір. Тьюрінга машина Тому більш реалістичної моделі, коли n прямує до нескінченності. Однак, як видається розумним вважати, що зв'язок автомата, як описано вище, призводить до більш підходящої теорії складності алгоритмів, ніж машини Тьюрінга робити, навіть якщо асимптотические формули при великих n вважаються, тому що теорія, швидше за все, мати відношення до практичної значеннях n . Крім того, коли n стає більше, ніж 1030 або близько того, навіть не одна машина Тьюрінга є реалістичною: вона ніколи не може бути побудована.

Відповідність є більш важливим, ніж реалізм. Минуло багато років з того часу, коли автор написав більшість коментарів вище, і все може

бути радий, що істотний прогрес дійсно були зроблені на теорії Зразки автоматів (зараз він називається показчик машин). Але, звичайно, багато що ще належить зробити.

Переклад виконаний аналітичною СМП (PROMT 9.0)

З'єднання автоматів може легко моделювати машини графіка, беручи найбільше обмежене число кроків на крок графіка. Навпаки, малоймовірно, що машини графіка можуть моделювати довільні автомати з 'єднання, необмежено не збільшуючи час виконання, якщо визначення не змінено від неорієнтованого до спрямованих графіків, зважаючи на обмеження на вершини обмеженого ступеня. Модель, що з 'єднується, звичайно, цілком близько до операцій, доступних програмістам на реальних машинах, в той час як модель графіка немає.

Деякі з найцікавіших проблем вирішити для таких пристроїв полягали б у тому, щоб визначити, як швидко вони можуть вирішити певні проблеми, або скільки вузлів вони повинні вирішити певні проблеми (наприклад, щоб перетворити певні формальні мови). У той час ця глава була вперше написана, кілька цікавих результатів цього виду були отримані (особливо Дж. Хартменісом і Р. Е. Стернзом), але тільки для спеціальних класів машин Тьюрінга, що мають багаторазові стрічки та комбіновані головки. Модель Машини Тьюрінга порівняно нереалістична, таким чином, ці результати мали тенденцію мати мало спільного з практичними проблемами.

Ми повинні визнати, що, оскільки номер n вузлів, створених автоматом, наближається до нескінченності, ми не знаємо, як створити такий пристрій фізично, так як ми хочемо, щоб машинні операції зайняли ту ж саму кількість часу незалежно від розміру n ; якщо з 'єднання показано під час використання адрес як у пам'яті комп'ютера, необхідно помістити прив'язане число вузлів, оскільки у полів посилання є фіксований розмір. Машина Тьюрінга мультиленти - тому більш

реалістична модель, коли n наближається до нескінченності. Все ж здається розумним вважати, що автомат як описано вище призводить до більш відповідної теорії складності алгоритмів, ніж Машини Тьюрінга, навіть коли асимптотичні формули для великого n розглядають, тому що теорія, більш ймовірно, буде важлива для практичних значень n . Крім того, коли n стає більше ніж приблизно 1030, не, навіть Машина Тьюрінга з однією стрічкою реалістична: Це ніколи не могло створюватися.

Доречність важливіша ніж реалізм. Багато років передали, оскільки автор записав більшість коментарів вище, і всі можуть радіти, що значний прогрес був дійсно зроблений на теорії з 'єднання автоматів (тепер названим машинами покажчика). Але звичайно багато все ще залишається зробленим.

Переклад виконаний професійним перекладачем (Кнут Д. Мистецтво програмування)

Зв'язуючі автомати можуть легко моделювати машини графів, використовуючи обмежену раніше кількість кроків на один крок роботи графа. Однак, малоймовірно, щоб машини графів могли моделювати довільні зв'язуючі автомати без необмеженого збільшення часу роботи, якщо не перейти від неорієнтованих графів до орієнтованих, щоб працювати з вершинами з обмеженим ступенем. Звичайно що зв'язуюча модель набагато ближче до операцій, доступним програмісту на реальній машині, на відміну від моделі з використанням графа.

Найцікавішими проблемами, які треба буде розв'язати для такого роду пристроїв, є визначення швидкості вирішення поставлених ними завдань, наприклад, підрахунок кількості вузлів, необхідних для вирішення того чи іншого завдання (наприклад, для трансляції будь-якого формального мови). У той час, коли автор почав працювати над цією главою, були отримані цікаві результати такого роду (відзначимо роботи

Ю. Хартманіс (J. Hartmanis) і Р. Е. Стірнс (R. E. Stearns)), але тільки для спеціальних класів машин Тьюрінга з безліччю стрічок і головок читання/запису. Модель машини Тьюрінга порівняно нереальна, а тому отримані результати мають мало спільного з вирішенням практичних завдань.

Слід визнати, що при прагненні кількості створених зв'язаним автоматом вузлів n до нескінченності невідомо, як побудувати такий пристрій фізично, оскільки бажано, щоб операції машини виконувалися за один і той же час незалежно від розміру n . Якщо зв'язування представлено з використанням адрес в машинній пам'яті, необхідно визначити межу для кількості вузлів, оскільки поля зв'язків мають фіксований розмір.

Багатострічкова машина Тьюрінга тому являє собою більш реалістичну модель при прагненні n до нескінченності. Представляється також обгрунтовано впевненість в тому, що описані вище зв'язуючі автомати приведуть до створення більш прийнятної теорії складності алгоритмів, ніж машина Тьюрінга, навіть при розгляді асимптотичних формул для великих n , так як ця теорія більше підходить для практичних значень n . Крім того, коли n стає більше, ніж 1030 або близько того, навіть однострічкова машина Тьюрінга не є реалістичною: вона ніколи не буде побудована. Принципи важливіші за реалії. З часу першого написання автором більшості наведених вище коментарів сплигло багато води, і можна порадіти, що в теорії зв'язуючих автоматів (сьогодні званих машинами покажчиків (pointer machines)) досягнуто певного прогресу, хоча, звичайно ж, має бути ще чимало роботи зроблено в цій галузі.