

Київський національний університет імені Тараса Шевченка  
Кафедра математичної інформатики

**Тарануха В.Ю.**

# **ІНТЕЛЕКТУАЛЬНА ОБРОБКА ТЕКСТІВ**

Частина 1

*Навчальний посібник*

Київ-2014

УДК 004.912:81'32  
ББК 81.1 я73  
Т19

**Тарануха В.Ю.**

Інтелектуальна обробка текстів: [навчальний посібник] / В. Ю.Тарануха. – Київ: електронна публікація на сайті факультету, 2014. – 80 с.

У навчальному посібнику розглянуто задачі інтелектуальної обробки текстів. Виклад зосереджено на питаннях автоматичного реферування, автоматичного та автоматизованого перекладу, діалогових та довідкових систем.

Призначений для студентів фізико-математичних та технічних спеціальностей вищих навчальних закладів.

## ПЕРЕДМОВА

Лінгвістична обробка природномовних текстів є однією з центральних проблем інтелектуалізації інформаційних технологій. Цій проблемі приділяється значна увага в розвинутих країнах Європи та США, свідченням чого є виділення величезних коштів на розробку лінгвістичного програмного забезпечення. Велику кількість науково-дослідних програм спрямовано на розвиток лінгвістичних інформаційних систем. У зв'язку з бурхливим розвитком Інтернету, інших комп'ютерно-комунікаційних технологій ця проблема набуває ще більшої значущості.

Ще з середини 50-х років минулого століття значні зусилля науковців були спрямовані на розробку математичних алгоритмів та комп'ютерних програм обробки текстів природною мовою. Для автоматизації аналізу та синтезу текстів створювалися різноманітні моделі процесів обробки тексту, а також відповідні алгоритми та структури представлення даних. Традиційно аналіз природномовних текстів представлявся як послідовність процесів – морфологічний аналіз, синтаксичний аналіз, семантичний аналіз. Для кожного з цих етапів було створено відповідні моделі та алгоритми. Для семантики тексту - класичні семантичні мережі та фреймові моделі Мінського, для синтаксису речення - граматики Хомського, системні граматики Холідея, дерева підпорядкування та системи складових Гладкого, розширенні мережі переходів; для морфологічного аналізу розроблено багато різних моделей, орієнтованих на конкретні групи мов.

Найбільш складні проблеми обробки природномовних текстів зумовлені явищами полісемії, омонімії тощо, які привносять у мову неоднозначність і значно ускладнюють задачу встановлення коректного відображення семантично-синтаксичної структури тексту в його формальне логічне представлення. Всі ці проблеми вирішуються на рівні семантичного аналізу.

З іншого боку, застосування ресурсномістких функцій логічно-семантичного аналізу робить програми обробки тексту занадто складними та

повільними. Людина в процесі розуміння тексту не так часто застосовує логіку – лише по мірі виникнення логічних задач, а в решті випадків відбувається застосування інших механізмів, у першу чергу – пошук за асоціацією по за формою чи контекстом.

Пошук за асоціацією – це оцінювання поняття, що відповідає даному слову та є контекстно близьким до свого оточення. При цьому асоціативний пошук є швидким та економічним засобом розв’язання неоднозначності інтерпретації тексту. Тому частина методів пов’язана з визначенням асоціацій за контекстом. Для роботи цих методів необхідна онтологічно-словникова база (онтологія), яка містить інформацію про концепти (поняття) мови, зв’язки концептів зі словами мови та зв’язки між концептами (синонімія, антонімія, гіперо-, гіпонімія та інші). Разом з онтологією використовується ряд алгоритмів, а саме: алгоритм визначення концептів за словами в тексті, алгоритми відновлення значень мовних вказівників на основі онтології, алгоритм визначення тематичної належності слів та понять тексту та тексту в цілому, алгоритм визначення змістовної близькості слів та понять, алгоритм узагальнення.

Пошук за формою мовної конструкції – оцінювання фрази/речення через його форму (синтаксичну структуру, регулярний вираз, наявність визначених елементів), з створенням відповідної реакції на віднайдений шаблон. При цьому вчені намагаються збудувати такі шаблони або маркери, які б дозволяли охопити якомога більше мовних явищ.

Для аналізу складних ситуацій використовуються фрейми. Вони що забезпечують найбільш зручний механізм для представлення жорстко структурованих знань про предметну область чи задачу.

Використання всіх цих методів у поєднанні з адаптованими методами статистики суттєво спрощує та прискорює створення систем інтелектуальної обробки текстів та їх використання.

Виклад зосереджено на задачах автоматичного та автоматизованого реферування, автоматичного та автоматизованого перекладу, діалогових

систем. Для кожної задачі наводяться також спеціалізовані методи, що мають основне використання лише у вказаній задачі.

Відповідно, Главу 1 присвячено реферуванню та пов'язаним з ним задачам, а саме: індексації (визначенню тематики), вилученню дублів, тощо. Главу 2 присвячено перекладу, з розглядом двох напрямків – автоматичного та автоматизованого перекладу, та відповідних механізмів що оптимізовані під конкретний напрямок. Глава 3 присвячена діалоговим системам та питально-відповідальна системам.

## 1. АВТОМАТИЧНЕ РЕФЕРУВАННЯ

Необхідність дослідження та розробки систем автоматичного реферування зумовлено збільшенням кількості та обсягу електронних документів, які потребують обробки, оскільки більшість таких документів має вигляді неструктурованих текстів, складених природною мовою, а більшість програмного забезпечення орієнтовано на роботу зі структурованими даними. Крім того, весь час зростає кількість новинних інтернет-сайтів, і для однієї і тієї ж події різні сайти надають різні інтерпретації. Разом з поширенням мобільного Інтернету та пристроїв класу смартфон, це створює попит на системи, які, зібравши дані з різних джерел, можуть дати користувачу короткий, проте достатній за охопленням звіт-реферат про поточні новини. Ще один спосіб використання автоматичного реферування пов'язаний з системами підтримки прийняття рішень. Експертам для виконання швидкого огляду необхідно аналізувати велику кількість документів, і вдалі системи реферування скорочують час, необхідний для читання. Замінити систему реферування пошуковою системою не вдається, тому що пошукова система буде шукати те, про що експерт вже знає або здогадується.

Системи автоматичного реферування здебільшого належать до двох видів. Це або системи, вбудовані в якийсь великий продукт, від новинних агрегаторів до Microsoft Office Word, з різним рівнем залучення користувача до процесу, або он-лайн системи. Останні в основному безкоштовні та пропонують генерацію з простими алгоритмами і не дуже високої якості.

### 1.1. ПОСТАНОВКА ЗАДАЧІ

*Реферат* – це текст заздалегідь визначеного об'єму, який менший за текст оригіналу і містить найбільш важливі для користувача думки оригіналу. *Реферування* – це процес побудови реферату [5].

В основному реферати будуються за двома напрямками – екстракція та абстрагування.

*Екстракція* – здобуття з тексту оригіналу елементів, які описують його зміст.

*Абстракція* – побудова висновків на основі тексту, які максимально стисло передають зміст тексту. Цей підхід передбачає застосування додаткових джерел даних про навколишній світ. Дві ці категорії не виключають одна одну і допускають застосування гібридних підходів.

За типами реферати бувають: інформативні, критичні та оповідні.

*Інформативний реферат* замінює собою текст первинного документа і містить основну або нову фактичну інформацію у скороченій формі.

*Критичний реферат* повідомляє не тільки інформацію, а й пропонує певну думку про неї. Критичні реферати мають додаткову цінність у порівнянні з оригіналом, оскільки пропонують висновки, яких немає у самому реферованому тексті.

*Оповідний реферат* формується за принципом здобуття інформації і повинен надати достатній обсяг інформації, щоб створити у користувача уявлення про джерело, з тим щоб можна було вирішити - звертатися до оригіналу чи відкинути текст як нерелевантний.

За орієнтацією на споживача реферати бувають загальні або, орієнтовані на задоволення спеціальних потреб.

*Загальний реферат* орієнтуються на широке коло читачів; до нього не висуваються спеціальні вимоги, оскільки реферат не призначений для якоїсь однієї групи читачів.

*Реферат, орієнтований на потреби*, адресований конкретному користувачеві або групі користувачів з їхніми специфічними потребами.

За кількістю документів: звичайні та мультиреферати.

*Мультиреферат* – реферат з багатьох документів одразу.

У матеріалі посібника розглядається побудова рефератів, орієнтованих на потреби читачів, при потребі – з кількох джерел.

Які саме думки найбільш важливі для користувача? Це залежить від знань та інтересів користувача. Доступу до знань користувача система не має

і не може мати, оскільки для цього треба, щоб користувач витратив час і зусилля для навчання системи всьому тому, що він знає сам. У той же час інтереси користувача можна визначити досить швидко. Найпростіший варіант – перелік ключових слів. Проте це не завжди ефективно, оскільки часто користувач ще не знає, що в документі його може зацікавити. Таким чином, першою задачею виступає задача визначення тематичного наповнення тексту.

*Індексація* – визначення тематик, що представлені в тексті, та елементів, що є індикаторами. Вона спирається на перелік ключових слів та понять, які цими словами позначаються. Після проведення індексації результати індексації надаються користувачеві для того, щоб він міг вказати необхідні йому дані.

Враховуючи, що мультиреферування має на меті побудову рефератів з кількох документів, необхідно якимось чином встановити можливі зв'язки між текстами та їх частинами. Це в свою чергу знову вимагає розв'язання задачі індексації та задачі визначення близькості між текстами.

*Близькість між текстами* – близькість за тематикою, близькість за метою подачі матеріалу (висновками, до яких підштовхує текст), також можливе запозичення тексту, часткове або повне. Запозичення найбільш характерне для текстів новин, коли одна інформагенція передруковує новину, посилаючись на іншу.

Окремо постає задача забезпечення читацьких якостей реферату. Якщо текст реферату утворений вибором та редагуванням речень оригіналу, то необхідно забезпечити зв'язність тексту.

*Зв'язність тексту* – наявність елементів, які дозволяють коректно визначати зв'язки між змістовними елементами тексту.

Основна причина порушення зв'язності полягає в тому, що речення змістовно наступне за вибраним реченням є в тексті оригіналу, проте відсутнє у рефераті. Реферат готовий, коли вибрані речення зв'язано у цілісний текст. У випадку мультиреферування, крім зв'язності, необхідно

забезпечити мінімальний рівень повторів, оскільки часті та невиправдані повтори роблять текст реферату психологічно неприємним та стилістично невиправданим.

*Рівень стиску* – частка тексту, що лишилася від оригіналу. Звичайно, рівень стиску складає від 30% до 5% від вхідного тексту чи текстів.

Отже, в цілому задача реферування ставиться так: за множиною текстів (можливо з одного з елементів) побудувати текст з мінімальним рівнем самоповторів, який відображає головні елементи вхідних текстів (основні події), та легко читається.

Тут не розглядаються задачі реферування із стиском одного тексту до одного абзацу, хоча вони є досить актуальними, особливо для агрегаторів новин.

## 1.2. ДОПОМІЖНІ ЗАСОБИ

Допоміжні засоби перераховуються, проте не деталізуються, оскільки можуть бути реалізовані багатьма способами, і від їх заміни на рівноцінні робота системи в цілому не змінюється.

Допоміжні засоби, що використовуються у роботі системи реферування:

- підсистема морфологічного аналізу (наприклад [6], або словники, вбудовані в систему вищого рівня аналізу [7]), результатом її роботи є граматичні ознаки та нормальні форми (леми);
- підсистема часткового синтаксичного аналізу (припускається, що вона реалізує принаймні зв'язки між іменниками та прикметниками, можна використати і більш потужну [7]);
- підсистема заміни займенників (описано в [9,12]);
- підсистема семантичного аналізу (опціонально [2,13]);

Таким чином, вважається, що задачу розбиття тексту на слова та визначення морфологічних характеристик слів вже розв'язано. На основі отриманих морфологічних даних проводиться частковий синтаксичний

аналіз. Зв'язуються прикметники (дієприкметники) з відповідними іменниками, що є мінімально необхідним для роботи системи. Для заміни займенників на повнозначні слова, на які вони посилаються - антецеденти, використовуються перш за все морфологічні ознаки. І лише у випадку, коли їх недостатньо, використовується простий семантичний аналіз, а саме: серед альтернатив вибирається слово, яке має зміст, найближчий за семантичною мірою близькості до слів контексту займенника. Для визначення семантичної близькості рекомендується використовувати семантичну базу WordNet (якщо є домовленість, то і її версія локалізовану до російської та української мов) та алгоритм пошуку найкоротших відстаней.

### 1.3. ІНДЕКСАЦІЯ

Є два способи побудови тематичного представлення - з фіксованими темами та з динамічними темами. Індексція з фіксованими темами спирається на фіксовані тематичні словники.

Найпростіше скласти теми з чітко вираженою термінологією: біологія, хімія, комп'ютерна тематика, фінанси, геологія та географія, право, лінгвістика, математика, атомна енергетика, фізика тощо. У таких темах до цих ключових слів входять як деталізовані, так і більш загальні поняття.

Створення тематичних словників для загального політичного дискурсу ускладнене, так само як і для історичного. Рекомендується вживати спеціалізовані словники реалій відповідного місця(регіону) та часу.

Динамічна індексція використовує динамічно створювані комплекти повнозначних слів, що належать приблизно до однієї тематики.

Для задачі мультиреферування, яке в першу чергу орієнтується на тексти новин, використання фіксованих списків ключових слів не є раціональним. Окремо використовується список слів, які при тематичному аналізі тексту будуть ігноруватися.

### 1.3.1 ЗМІСТОВНА БЛИЗЬКІСТЬ

Змістовна близькість є одним з найбільш важливих елементів зв'язності тексту. Розглянемо такий фрагмент тексту:

«Розвиток **обчислювальної техніки** відбувався одночасно з розвитком технологій. На зміну електронним лампам прийшли напівпровідники. На їх основі було створено **інтегральні схеми**. Щодалі, то більше і більше **логічних елементів** стали розміщувати на одному кристалі.»

У даному тексті виділено два типи змістовної близькості:

- пов'язаної за значеннями термінів;
- пов'язаної за структурою тексту.

Виділені в тексті **жирним** шрифтом словосполучення позначають поняття, пов'язані значеннями відповідних слів або термінів. Зв'язок першого типу будується на основі знань про те, що два елементи (поняття) пов'язані між собою в базі знань.

Виділені підкресленням слова позначають поняття, пов'язані з іншими поняттями за рахунок того, що зустрічаються поблизу. Цей тип зв'язку задається автором тексту і є відображенням того, що саме хотів повідомити автор цим текстом.

Зв'язки можуть бути реалізовані як термінами безпосередньо, так і мовними вказівниками (у лінгвістиці це називається анафора). Задача встановлення елементів, на які вказують мовні вказівники, називається *розв'язанням анафори*[12].

### 1.3.2 ЛЕКСИЧНІ ЛАНЦЮЖКИ

Лексичні ланцюжки представляють змістовні єдності серед довільного числа зв'язаних слів.

По-перше, це лексичний ланцюжок тотожних об'єктів. До нього заносяться ті елементи тексту, що вказують на один і той самий об'єкт або одне і те саме поняття. Задачу знаходження всіх згадувань одного і того самого об'єкту називають задачею *розв'язання корелювання*[12].

По-друге, це лексичний ланцюжок семантично зв'язаних об'єктів. Такий ланцюжок не обмежує типи елементів та види зв'язків між ними, поки вони пов'язані між собою з точки зору автора. Надалі вживається саме таке значення терміну, оскільки це дозволяє визначати не лише тотожність/відмінність між об'єктами, а і визначати тематичну належність. Для побудови таких лексичних ланцюжків в якості джерела знань вживається WordNet.

### 1. 3.3 СПОСІБ ОБ'ЄДНАННЯ ЛЕКСИЧНИХ ЛАНЦЮЖКІВ

Лексичні ланцюжки обчислюються шляхом групування послідовних наборів семантично зв'язаних слів. Тотожні слова, синоніми, гіперніми і гіпоніми, мероніми, голоніми – ознаки, що дозволяють групувати слова в один ланцюжок.

Гіпернім – поняття, що є узагальнюючим для даного у онтології (WordNet[8]).

Гіпонім – поняття, що є уточненням даного у онтології (WordNet).

Меронім – поняття, що позначає «ціле» у відношенні «ціле-частина».

Голонім – поняття, що позначає «частина» у відношенні «ціле-частина».

Необхідно зауважити, що у WordNet представлені не слова, а сенси(концепти) – поняття, і кожен сенс має свій комплект слів, які його позначають. Такий комплект слів називається „синсет”

#### **Умови групування**

1. Два входження повнозначного слова ідентичні, і використовуються в тому ж самому сенсі. (*Великий **корабель** на рейді. Цей **корабель** – вітрильник.*)

2. Два входження повнозначних слів використовуються в одному і тому ж самому сенсі, але текстуально різні тобто, є синонімами. (*Той **аероплан** летить швидко. Проте, мій **літак** швидше.*)

3.Змісти двох входжень повнозначних слів мають гіпернім/гіпонім відношення між ними. (*Я маю **автомобіль**. Це –вантажівка.*)

4. Змісти двох входжень повнозначних слів – елементи одного рівня в гіпернім/гіпонім дереві і мають спільного предка. (*Той **аеробус** летить швидко. Проте, мій **винищувач** швидше.*)

5. Два входження повнозначних слів означають відповідно частину та ціле – є зв'язок меронімії. (*Дмитро відчинив **двері**. **Замок** голосно клацнув.*)

В обчисленні лексичних ланцюжків, входження повнозначних слів повинні бути згруповані згідно з вищезгаданими правилами, але кожне входження повнозначного слова повинно належати точно одному лексичному ланцюжку.

Розглянемо загальний метод побудови колекції лексичних ланцюжків для заданого тексту, що був запропонований Регіною Барзілай.

1. Вибрати *слово* або *словосполучення*, який взято з тексту (надалі **об'єкт**) і який має представлення у WordNet.
2. Для кожного об'єкту знайти відповідний ланцюжок і вставити об'єкт туди.
3. Якщо ланцюжок не існує, то створити новий на основі заданого об'єкту.

Як видно з опису методу, немає однозначного способу для відповіді на певні питання.

1. Як визначити відповідний ланцюжок? Наприклад, входження іменника може відповідати декільком різним сенсам слова, і система повинна визначити, яке саме входження має місце. Наприклад, «коса» як інструмент і «коса» як зачіска.
2. Як забезпечити однозначність? Навіть якщо сенс слова може бути визначений, може трапитись, що слово можна занести до декількох різних лексичних ланцюжків, тому що це слово може бути зв'язане зі словами в різних ланцюжках.

Для коректності вводяться параметри об'єднання об'єкту з ланцюжком. До цих параметрів входять відстані та напрямки для зазначених у

попередньому розділі умов групування за зв'язками. Необхідно враховувати, що комбінуючи умови групування, можна будувати як завгодно складні зв'язки.

1. Для умови групування 3. накладається обмеження на довжину такого зв'язку при обчисленні його від більш деталізованого поняття до більш загального, але не навпаки.

2. Для умови групування 4. накладається обмеження на відстань від узагальнюючого поняття. В нашому випадку – 2.

3. Для умови групування 5. накладається обмеження на довжину такого зв'язку, якщо в шляху є різнотипні переходи. У нашому випадку – 2.

Необхідно зауважити, що два слова будуть зв'язані разом з більшою вірогідністю, якщо в тексті, який аналізується, вони стоять поблизу. З цього випливає ще один параметр об'єднання: об'єкти зв'язані, якщо вони знаходяться у сусідніх реченнях.

Для різних умов групування варто задати різні допустимі відстані для визначення сусідства.

1. Для умов 1-3 відстань може сягати 7 речень.

2. Для умов 4-5 відстань не повинна перевищувати 4 речення.

#### 1.3.4 АЛГОРИТМ ПОБУДОВИ ЛЕКСИЧНИХ ЛАНЦЮЖКІВ

Для ефективного обчислення лексичних ланцюжків створюється структура, яка неявно зберігає кожну інтерпретацію кожного слова. А потім з цього неявного представлення обчислюється оптимальна конфігурація. Обробка документа починається зі створення великого масиву мета-ланцюжків, розмір якого дорівнює числу сенсів слів тексту, знайдених у WordNet, плюс число слів у документі, оскільки можливо, що слова не будуть знайдені у WordNet. Довжина кожного такого мета-ланцюжка дорівнює кількості повнозначних слів у тексті.

Коли алгоритм знаходить повнозначне слово, збільшується лічильник у відповідному мета-ланцюжку, який містить сенс цього слова, та у кожному ланцюжку, куди це слово входить за однією з вище визначених ознак.

Коли переший прохід закінчено, текст проглядається ще раз, і для кожного повнозначного слова визначається ланцюжок, до якого воно вносить якнайбільшу вагу. З решти ланцюжків слово вилучається.

Позначення:  $w_i$  – повнозначне слово з тексту,  $c(w_i)_j$  – сенс слова  $w_i$  визначений за WordNet,  $\{c(w_i)_j\}_k$  – мета-ланцюжок.

#### *Алгоритм побудови лексичних ланцюжків.*

Для кожного  $w_i \in T$

Для кожного  $c(w_i)_j$

Для кожного  $\{c(w_i)_j\}_k$  оновити значення

Для кожного  $w_i \in T$

Для кожного  $\{c(w_l)_m\}_k$ , що має  $c(w_i)_j$

Визначити  $k$  мета-ланцюжа, до якого  $c(w_i)_j$  належить найбільше

Обновити таблицю, видаливши зайві елементи.

Оцінка складності роботи алгоритму:  $O(N * s_{max} * M^2)$ , де  $N$  – довжина тексту в словах,  $s_{max}$  – найбільша кількість сенсів слова,  $M$  – довжина мета-ланцюжка. Таким чином, отримані ланцюжки представляють собою динамічно сформований комплект тем документу.

За відсутності даних про конкретне слово чи поняття у локалізації WordNet можна застосувати евристичне об'єднання елементів тексту у тематичні ланцюжки. Таке об'єднання спирається на припущення, що автор тексту писав його як осмислений текст, і користувався однією термінологією у межах викладення однієї думки або ідеї.

Алгоритм TextTilling розділяє текст на сукупність фрагментів, які мають внутрішні зв'язки. Тоді два об'єкти будуть зв'язані з більшою достовірністю, якщо вони знаходяться в одному фрагменті.

Відстань між вікнами -  $\cos \Theta = (A, B) / (|A| |B|)$ ,

де  $A, B$  – частотні вектори відповідних вікон.

Межа розриву – параметр  $Th$

#### *Алгоритм TextTilling*

Задати розмір  $n$  вікна в словах -  $w_1 \dots w_n$

Пересуваючи вікно по тексту, створити з  $c(w_i)_j$  вектори, що відповідають двом сусіднім вікнам;

Порахувати відстані між кожними двома сусідніми вікнами  $w_i \dots w_{n+i}; w_{n+i+1} \dots w_{i+2n}$ ;

Якщо  $\cos \Theta > Th$  - розрив.

Встановити маркер «Розрив»

Оцінка складності роботи алгоритму:  $O(N \cdot w^2)$ , де  $N$  – довжина тексту в словах,  $w$  – довжина вікна у словах.

### 1.3.5 ВИКОРИСТАННЯ РЕЗУЛЬТАТІВ ІНДЕКСАЦІЇ

Результати індексації, а саме побудовані лексичні ланцюжки, необхідні для двох задач:

- побудови оцінки важливості тих чи інших елементів у рефераті;
- побудови оцінки близькості фрагментів тексту з метою вилучення дублів.

Ані частотний підхід, ані зважування окремих елементів семантичного представлення (концептів) не працює коректно, якщо в тексті(текстах), що реферуються, спостерігається велика кількість повторів або запозичень.

Також треба враховувати, що задача оцінювання імовірних запозичень має високу обчислювальну складність, якщо її виконувати на всьому наборі вхідних текстів. Можливість розбиття тексту на фрагменти та вилучення очевидних дублів значно прискорює роботу системи реферування.

### 1.4. ВИЗНАЧЕННЯ МОЖЛИВИХ ЗАПОЗИЧЕНЬ У ТЕКСТАХ

При визначенні запозичень потрібно враховувати, за яких обставин відбулось запозичення. Якщо людина вдається до плагіату, швидше за все, вона внесе до тексту зміни, для того щоб приховати факт неправомірних

запозичень. При цьому потрібно добитися мінімізації випадків помилкового спрацьовування. Якщо газета передруковує новину, то нормальним є зазначення посилання на джерело, при, можливо, новій інтерпретації та переписуванні тексту.

У досліджуваному тексті для підозрілого фрагмента можливі такі спотворення:

- Межі підозрілого фрагмента не обов'язково збігаються з межами абзацу або пропозиції;
- У підозрілий фрагмент могли бути внесені зміни, як шляхом вставки, так і шляхом викидання елементів;
- У підозрілому фрагменті можуть бути переставлені елементи;
- У підозрілому фрагменті можливі синтаксичні зміни, виражені у зміні граматичних форм слів;
- У підозрілому фрагменті можливі стилістичні зміни, виражені в заміні слів на синоніми;
- Межі пропозицій в підозрілому фрагменті можуть не збігатися з вихідними.

У загальному випадку текст є особливою формою представлення знань про зовнішню тексту дійсність. Текст повинен мати властивості: членимості, смислової цілісності та зв'язності. При цьому треба пам'ятати, що різні спостерігачі (читачі), в одному і тому ж наборі символів побачать різний текст.

Членимість тексту має на увазі, що текст ділиться на абзаци, службові угруповання однорідних одиниць викладу за сюжетом або темою. Послідовність абзацив формує порядок виведення однієї одиниці з іншої або інших. Це в свою чергу формує ті висновки, які хотів донести до читача автор. Абзац у свою чергу ділиться на пропозиції - сукупності декількох пов'язаних слів, що виражають певний сенс. Пропозиції складаються зі слів і словосполучень, які відображають смисли і відносини між ними.

Смислова цілісність виникає, якщо читач може узагальнити текст і дати йому назву.

Зв'язність тексту визначається наявністю у ньому зв'язок, забезпечених смисловими або синтаксичними засобами. До смислових відноситься згадка раніше описаного, не обов'язково у вигляді повторення слова. Це може бути згадка аспекту або якості. Таке згадка породжує відношення «нове-старе» між елементами пропозиції. Синтаксичні засоби складаються у застосуванні слів або навіть синтаксичних конструкцій, які не мають власного сенсу, але вказують на раніше згадану сутність. Будемо називати їх мовними вказівниками.

#### 1.4.1 МОДЕЛЬ СТРУКТУРИ ТЕКСТУ

Текст можна вважати послідовністю тематичних або сюжетних елементів, при цьому кожний наступний набирає відношення слідування до всіх попередніх. Не обмежуючи загальності, вважаємо, що слова в тексті приведено до нормальних форм, - це значно спрощує роботу.

$$T_k = T_k(A(T), R(T)),$$

де  $T_k$  – окремий текст,

$A(T)$  - множина абзаців,

$R(T)$  – відношення слідування, визначене на абзацах, і залежить від того, як і яку думку хотів донести до читача автор.

Абзац розбивається на сукупність речень, кожне з яких вводить, уточнює або зв'язує певні сенси.

$$a = a_i(S(a_i), R(a_i))$$

де  $a_i$  –  $i$ -й абзац,

$S(a_i)$  – множина речень в абзаці,

$R(a_i)$  – відношення слідування визначене на реченнях, і залежить від того, як і яку думку хотів донести до читача автор. При цьому між реченнями одного абзацу діють змістовні зв'язки та мовні вказівники.

Додатково вводиться структура для речення, щоб можна було оперувати словами, а не окремими реченнями.

$$s_{i,j} = s_j(W(s_j), R(s_j))$$

де  $s_j$  –  $j$ -е речення,  $i$ -го абзацу,

$W(s_j)$  – множина слів в абзаці,

$R(s_j)$  – множина синтаксичних відношень між словами.

Можлива ситуація, коли між елементами тексту (наприклад, абзацами) відсутня змістовна зв'язність. Це подається у вигляді

$$T = U T_k$$

Тоді текст, представлений для аналізу, виступає як сукупність окремих текстів, кожен з яких має свій зміст.

Зауважимо, що в задачі мультиреферування ситуація, коли текст насправді являє собою мультитекст – звичайне явище.

#### 1.4.2 ЗАПОЗИЧЕННЯ В ПОДРОБИЦЯХ

Запозичення можуть бути двох типів: легальні, оформлені у вигляді цитат, можливо, без лапок, і приховувані, можливо з додатковими спотвореннями.

Позначимо  $G$  - вихідний текст, з якого щось запозичалося,  $D$  - текст-мета, текст до якого щось було запозичене.

Виявлення запозичень першого типу не являє собою особливої проблеми для читача, тому що автор тексту сам робить все необхідне, щоб читач впізнав запозичення. Можна вважати, що в межах такого запозичення більшість відносин визначених у  $G$  буде існувати і в  $D$ , на всіх рівнях ієрархії керуючого простору тексту [1], незалежно від того, як ми їх визначимо. Проте це лишається досить складною задачею для автоматичного аналізу.

Успішність виявлення запозичень другого типу залежить від:

- Успіху розбиття тексту на блоки по смислової цілісності,
- Структури відношення  $R(T)$ ,
- Структури відношення  $R(a(T))$ ,
- Структури відношення  $R(s(a(T)))$ .

Для розбиття тексту  $D$  на блоки за смисловою цілісністю у загальному випадку потрібен апарат, порівнянний з штучним інтелектом. Оскільки поки

що його немає, то для розбиття пропонується використовувати метод, що спирається на ознаку більш низького рівня: на наявність зв'язок, забезпечуваних смисловими або анафоричними засобами. Розрив такої структури будемо вважати кінцем блоку.

У системі, яку ми розглядаємо, кінець блоку визначається за результатами, отриманими у процесі індексації. А саме, розриви за TextTiling та розриви лексичних ланцюжків вказують на межі блоків.

Відношення  $R(T)$  можна апроксимувати відношенням порядку. Таким чином, щоб абзац в  $D$  приводив до того ж висновку, потрібно, щоб він перебував після тих самих вихідних положень, що і в  $G$ .

Позначимо  $N(a, T)$  - функцію, яка повертає номер абзацу в тексті. Нехай виконуються умови:  $a_1, a_2 \in D, a_1, a_2 \in G, N(a_1, D) > N(a_2, D) \ \& \ N(a_1, G) > N(a_2, G) \Rightarrow a_1 R(D) a_2 \ \& \ a_1 R(G) a_2$

Відношення  $R(a_i)$  задається подібно до відношення  $R(T)$ . Позначимо  $a(G)$  – абзац з тексту  $G$ ,  $a(D)$  – абзац з тексту  $D$ . Позначимо  $N(s, a)$  – функцію, що повертає номер речення в абзаці.

Тоді слабке відношення задається так:

$s_1, s_2 \in a(D), s_1, s_2 \in a(G), N(s_1, a(D)) > N(s_2, a(D)) \ \& \ N(s_1, a(G)) > N(s_2, a(G)) \Rightarrow s_1 R(a(D)) s_2 \ \& \ s_1 R(a(G)) s_2$

Сильне відношення задається з урахуванням збереження змістовних зв'язків і мовних вказівників. Позначимо  $P(s, a)$  – функцію, котра повертає речення, на які є вказівники. Позначимо  $Q(s, a)$  – функцію, котра повертає речення, з якими є змістовні зв'язки.

Тоді сильне відношення задається так:

$s_1, s_2 \in a(D), s_1, s_2 \in a(G), P(s_1, a(D)) \cup Q(s_1, a(D)) = P(s_1, a(G)) \cup Q(s_1, a(G)) \ \& \ P(s_2, a(D)) \cup Q(s_2, a(D)) = P(s_2, a(G)) \cup Q(s_2, a(G)) \Rightarrow s_1 R(a(D)) s_2 \ \& \ s_1 R(a(G)) s_2$

Відношення  $R(s_j)$  задається так. Позначимо  $s(a)$  – речення абзацу. Позначимо  $V(w, s)$  – функцію, яка повертає слова, зв'язані с даним словом у реченні.

$w_1, w_2 \in s(a(D)), w_1, w_2 \in s(a(G)), V(w_1, s(a(D))) = V(w_1, s(a(G))) \& V(w_2, s(a(D))) = V(w_2, s(a(G))) \Rightarrow w_1 R(s(a(D))) w_2 \& w_1 R(s(a(G))) w_2$

Багато задавати зв'язок виключно за допомогою синтаксичних зв'язків слів речення, наскільки це дозволяє якість синтаксичного аналізатора. Такий підхід дозволяє гарантовано опрацювати все можливі коректні перестановки слів у реченні. За відсутності хорошого аналізатора  $V(w, s)$  задається як індикаторна функція для слів по реченню.

Побудовані вищеописаним чином відношення  $R(a), R(s), R(w)$  гарантують, що запозичені елементи будуть відзначені, але власний текст буде проігноровано. Тоді можна визначити оцінку запозичення тексту.

$F_1 = |\{ (a_1, a_2) : a_1, a_2 \in D, a_1 R(D) a_2 \& a_1 R(G) a_2 \}|$ , якщо  $|D(a)| > 1$ , інакше 1.

$F_2 = |\{ (s_1, s_2) : s_1 R(a(D)) s_2 \& s_1 R(a(G)) s_2 \}|$ , якщо  $|D(s)| > 1$ , інакше 1.

$F_3 = |\{ (w_1, w_2) : w_1 R(s(a(D))) w_2 \& w_1 R(s(a(G))) w_2 \}|$ ,

$F(D, G) = F_1 * F_2 * F_3 / (|D(a)|^{^2} * |D(s)|^{^2} * |D(w)|^{^2})$ , (1)

де  $|D(a)|, |D(s)|, |D(w)|$  – кількість абзаців, речень і слів у тексті.

Властивості оцінки:

1)  $F(D, G) \geq 0$ , для довільних  $D, G$ .

2)  $F(G_1, G_2) \neq F(G_2, G_1)$

З практичних міркувань у мультиреферуванні більш корисною є така сама оцінка але для блоків, що визначаються на основі індексації текстів.

Альтернативним способом побудови оцінки запозичень є застосування моделі на  $n$ -грамах [7]. Послідовність слів мови  $w_1 \dots w_n$  називається  $n$ -грамою порядку  $n$ , якщо вони розташовані в тексті одне за одним.

Позначається  $w_1^n$

Зафіксуємо  $Th$  – як рівень допуску для збіжності.

Алгоритм побудови оцінки запозичень на  $n$ -грамах

Для кожного тексту  $T \in \{T\}_t$

Створити порожній словник  $D$   $n$ -грам, де ключем буде  $w_1^n$ , значенням – її частота

Для всіх  $w_i \in T$ :

сформувані  $w_i^{n+i-1}$ , до яких вони входять,

внести  $w_i^{n+i-1}$  в  $D$ , відповідно збільшуючи лічильники частот

Створити квадратну матрицю збіжностей текстів  $A=\{a_{lm}\}$ ,

Для всіх  $D_l$ :

Для всіх  $w_i^{n+i-1}$  у  $D_k$ ,  $l < k$

За кожну знайдену  $w_i^{n+i-1}$  в  $D_k$  збільшувати  $a_{lk}$

Для всіх  $a_{lm} > Th$ ,

Для  $T_l$  і  $T_k$ :

Вказати збіжні елементи.

Такий алгоритм працює швидше, проте не стійкий до складних запозичень.

## 1.5. КЛАСТЕРИЗАЦІЯ

Важливим етапом мультиреферування є кластеризація фрагментів текстів, для того, щоб прибрати дублі та зібрати подібні тексти в групи. Якщо замість набору текстів є огляд, то зникає потреба в першому етапі кластеризації, а саме в кластеризації за результатами індексації.

### 1.5.1 МІРИ ЯКОСТІ КЛАСТЕРІВ ТА КЛАСТЕРНОГО РОЗБИТТЯ

Важливою проблемою кластерного аналізу є вибір методології визначення якості одержаних кластерів та визначення цільової функції кластерного аналізу. Найбільш ефективні результати досліджень одержані при використанні так званих зовнішніх мір, тобто при порівнянні результатів автоматичного аналізу з ручним.

Критерії якості кластеризації, як правило, ґрунтуються на наступних вимогах:

- усередині груп об'єкти повинні бути тісно пов'язані між собою;
- значення відстані між об'єктами різних груп повинне бути достатньо великим;
- при інших рівних умовах розподіл об'єктів по групам повинен бути рівномірним.

Міри якості кластерного розбиття поділяються на два класи. До першого класу відносяться міри, які базуються на оцінках експертів, так звані зовнішні міри якості, наприклад  $F$ -міра. До другого класу, відповідно, відносяться міри, які не використовують додаткову інформацію – внутрішні міри якості, наприклад – *загальна внутрішня подібність*.

У задачі мультиреферування порушуються всі такі вимоги. У той же час, існує хороша оцінка загальної внутрішньої подібності, за рахунок можливості обчислити ступінь запозичення одного тексту відносно іншого. Це, у свою чергу, підштовхує до застосування кластеризації на графах, про яку буде йтися нижче.

Міра близькості, заснована на визначенні запозичень, добре спрацьовує, коли тексти документів є запозиченнями одне з одного. Це, зазвичай є нормою в текстах новин, опублікованих в Інтернеті, коли новину передрукують, лише додавши посилання на джерело. Проте для повноцінного аналізу такої міри близькості не досить.

Альтернативою виступає метод оцінки за словниками текстів. Два тексти будуть тематично близькими, якщо набори сутностей, згаданих у них, збігаються. Що більша збіжність, то більша подібність тематичного наповнення.

Для оцінювання такої близькості використовується косинусна міра.

$$\cos \Theta = (A, B) / (|A||B|)$$

де  $A, B$  – частотні словники текстів, представлені як вектори.

Подібність, обчислена таким чином, досить добре описує тематичне наповнення текстів, за умови, що коректно виділені основні термінологічні одиниці.

Оцінка складності швидкість порівняння текстів за такою мірою  $O(\text{оцінки подібності}) = O(N)$

### 1.5.2. КЛАСТЕРНИЙ АНАЛІЗ НА ЗВАЖЕНИХ ГРАФАХ

Методи кластеризації, які базуються на використанні зважених графів, об'єктам вибірки ставлять у відповідність деякий набір вершин  $V$ . Дві вершини  $v_a$  та  $v_b$ , що відповідають векторам ознак  $x_a$  та  $x_b$  об'єктів  $A$  та  $B$ , можуть бути з'єднані ненаправленим ребром  $E_{ab}$  з додатковою вагою  $S(x_a; x_b)$ , що являє собою міру близькості між векторами. Кількість ребер графа  $|E|$  дорівнює кількості ненульових значень близькості між усіма парами точок. Набір ребер, видалення яких розбиває граф  $G = (V, E)$  на  $k$  підграфів, що не перетинаються, називається роздільником ребер. Отже, метод кластерного аналізу з використанням зважених графів полягає в знаходженні роздільника ребер з мінімальною сумою належних йому ребер. При цьому часто накладається додаткова умова – отримання приблизно рівної кількості об'єктів (вузлів) у кожному кластері (підграфі).

Відповідно до розглянутого методу загальна складність алгоритму кластеризації складе  $O(N^2) * O(\text{оцінки запозичень})$ .

Треба врахувати, що сама по собі формула (1) вказує на складність принаймні

$$O(\text{оцінки запозичень}) = O(N^4),$$

де  $N$  - кількість слів у тексті.

Таким чином, оптимальною виглядає попередня кластеризація за результатами індексації або якимось іншим способом, з подальшою додатковою кластеризацією за мірою запозичення.

### 1.6. ВИДІЛЕННЯ ТЕРМІНОЛОГІЧНИХ ОДИНИЦЬ

Задача виділення термінологічних одиниць виникає, коли в тексті зустрічаються терміни, які не відомі базі знань (WordNet). Особливо це важливо, коли терміни складаються з декількох слів.

Звичайно використовують такі методи: підрахунок кількості пар,  $t$ -критерій Стьюдента, критерій узгодженості Пірсона. Методи простого

частотного підрахунку та  $t$ -критерій також є досить ефективними і можуть бути використані для складання списку термінів-кандидатів у системах напівавтоматичного формування термінів. Основний тип помилок обох методів – виділення стійких загальноживаних словосполучень, які задовольняють шаблонам-обмеженням.

Найбільша проблема всіх частотних методів наступна: при збільшенні довжини терміна падає його частота, навіть у спеціалізованому корпусі термін може зустрічатися один-два рази.

Звичайними методами, які дозволяють обійти такі обмеження, є метод максимальної довжини та метод контролю у вікні.

Метод максимальної довжини. Виділення максимальних ланцюжків, які містять терміни. Ці ланцюжки визначаються через негативний відбір: складається список слів і знаків, які не можуть входити в термін. У нашій реалізації в якості таких роздільників ми розглядаємо розділові знаки, стоп-слова, дієслова, дієприслівники; послідовності слів між цими роздільниками розглядаються як кандидати в терміни.

Метод контролю у вікні. До частоти сумісного входження включається також частот входжень слів в одне вікно певного розміру (8 слів). Вважається, що якщо пари елементів зустрічаються як безпосередні сусіди більш ніж у половині випадків їх появи в тому самому текстовому вікні, те ця пара являє собою термін або фрагмент терміна. Відбувається склейка пари у єдиний елемент, таблиці перераховуються так, ніби цей елемент був відомий із самого початку, до початку обробки тексту, що дає можливість і далі нарощувати термін.

За результатами перехресної перевірки частина слів об'єднуються в багатослівні терміни.

## 1.7. РЕФЕРУВАННЯ

Базується на результатах індексації, тематичного аналізу, аналізу запозичень та результати кластеризації.

### 1.7.1 ВИЗНАЧЕННЯ ВАЖЛИВОСТІ ЕЛЕМЕНТІВ ТЕКСТУ

Важливість елементів тексту визначається відповідно до того, наскільки вони цікавлять користувача та наскільки вони важливі для представлення змісту тексту. Є декілька підходів до визначення важливості елементів.

*Частотні критерії*: частота вживання терміну, та TF\*IDF.

Частота – кількість випадків вживання терміну у тексті  $F(w_i)$ ,  $w_i \in T$ .

TF\*IDF – спирається на роздільну силу терміну. Що термін частотніший, то більш він важливий. Проте, чим до більшої кількості речень чи текстів він входить, то слабкіше їх розрізняє.

В TF\*IDF, TF – частота терміну, IDF – інверсна частота документу.

$$TF = \frac{n_i}{\sum_k n_k}$$

де,  $n_i$  є число входжень терміну в документ, а в знаменнику — сума частот термінів у даному документі.

$$IDF = \log \frac{|D|}{|(d_i \supset t_i)|},$$

де  $|D|$  — кількість документів в корпусі;  $|(d_i \supset t_i)|$  - кількість документів, у яких зустрічається  $t_i$  (коли  $n_i$  більше нуля).

З указаних елементів збирається статистична оцінка (StatTerm) для термінів, і за потреби - як сума ваг для блоку з  $T_b$ .

*Конекціоністські критерії*: зв'язність у графі, утвореному з  $w_i \in T$ , входження у відповідні лексичні ланцюжки та TextRank (Connect).

Зв'язність у графі – кількість вершин, які пов'язані з вершиною, що відповідає  $w_i \in T$ .

Якщо використовуються лексичні ланцюжки, то кожен з них може мати власну вагу, і тим самим модифікує вагу  $w_i$ .

TextRank – аналог міри PageRank, що вживається для визначення.

*Критерії за структурними ознаками тексту*: розташування  $w_i$ , наявність поблизу спеціальних слів або фраз маркерів.

Ознака розташування (Location) залежить від того, де у всьому тексті або в окремо взятому параграфі з'являється фрагмент, що містить  $w_i$  - на початку, всередині або в кінці, а також чи використовується  $w_i$  в ключових розділах, наприклад, вступній частині або у висновках. Важливим критерієм є і те, чи з'являється  $w_i$  також у заголовку, у колонтитулі, першому параграфі.

Ключова фраза (CuePhrase) представляє собою лексичні або фразові конструкції, такі як: „на закінчення”, „у даній статті”, „згідно з результатами аналізу” і так далі. Ваговий коефіцієнт ключової фрази може залежати також і від прийнятого в даній предметній області оцінного терміна, типу „відмінний” (найвищий коефіцієнт) або „малозначне” (значно менший коефіцієнт).

*Машинно вивчені критерії:* для їх побудови вживається алгоритм машинного навчання того чи іншого виду над множиною текстів та множиною відповідних рефератів. У межах цього посібника вони не розглядаються.

*Задані користувачем:* наявність  $w_i$  профілі визначеному користувачем системи (AddTerm). Виділення пріоритетних термінів, що найбільш точно відображають інтереси користувача, - це один із шляхів налаштувати реферат або анотацію на конкретну людину або групу.

*Спеціалізовані критерії:* у випадку мультиреферування замість важливості, яку визначає користувач, використовується індекс цитування новини як зовнішня експертна оцінка її важливості. (Special)

Використання лише якогось одного типу критеріїв не є оптимальним. Наприклад, за частотним принципом: чим частотніше слово/поняття у тексті, тим воно важливіше. Проте, може мати місце випадок, коли група слів/понять разом важлива для тексту, хоча кожне з них зустрічається не досить часто, щоб частотний критерій розпізнав їх як важливі.

При цьому застосовується модель лінійних вагових коефіцієнтів. Основу аналітичного етапу в цій моделі складає процедура призначення

вагових коефіцієнтів для кожного блоку тексту відповідно з вказаними вище характеристиками.

Сума індивідуальних ваг визначається після додаткової модифікації у відповідності з параметрами налаштування і дає загальну вагу всього блоку тексту  $T_b$ :

$$\text{Weight}(T_b) := \text{Location}(T_b) + \text{CuePhrase}(T_b) + \text{StatTerm}(T_b) + \text{Connect}(T_b) + \text{AddTerm}(T_b) + \text{Special}(T_b)$$

### 1.7.2 ПЕРЕДОБРОБКА

Перш за все, після визначення важливості кожного повнозначного слова, необхідно визначити межі тематичних областей в тексті. Це робиться простим проходом по тексту з розстановкою маркерів. Після чого проводиться розстановка обмежувачів, які чітко визначають, де певна тема скінчилася або почалася. Позначимо  $s \in T$  – речення тексту. Повнозначне слово, як завжди, –  $w_i, c(w_i)_j$  – сенс слова  $w_i$ ,  $\{c(w_i)_j\}_k$  – ланцюжок, що містить відповідний сенс слова.

#### *Алгоритм розмітки тематичних областей*

Для кожного  $s \in T$

    Для кожного  $w_i \in s$

        Для кожного  $c(w_i)_j$

            Встановити належність до певного  $\{c(w_i)_j\}_k$

            Поставити маркер «Тема почалася» відповідно до номера ланцюжка

Для кожного  $s_l \in T$

    Для кожного  $w_i \in s$

        Для кожного  $c(w_i)_j$

            Чи є  $c(w_i)_j$  з того  $\{c(w_i)_j\}_k$  в  $s_{l+1}$  (сусідньому реченні).

            Немає: Поставити маркер «Тема скінчилася» відповідно до номера ланцюжка

Для кожного  $w_i \in T$

    Для кожного  $c(w_i)_j$

        Якщо маркери «Тема почалася» і «Тема скінчилася» стоять одночасно

        Зняти позначки відповідних  $\{c(w_i)_j\}$ .

Оцінка складності роботи алгоритму:  $O(N_s * l_{max} * k_{max} * M)$ , де  $N_s$  – довжина тексту в реченнях,  $k_{max}$  – найбільша кількість сенсів слова,  $M$  – довжина мета-ланцюжка,  $l_{max}$  – найбільша довжина речення в тексті.

Також застосовується вже описаний алгоритм TextTilling. Отримані за його допомогою зони розриву накладаються на тематичні області. Якщо відбувається зміна теми без порушення зв'язності за TextTilling, то таке речення отримує додатковий маркер «Критична Область».

#### *Алгоритм визначення критичних областей*

Для кожного  $s_i \in T$

Якщо  $s_{i+1}$  (наступне речення) належить іншій темі

Якщо  $s_i$  не має маркеру «Розрив»

Поставити маркер «Критична Область»

Оцінка складності роботи алгоритму:  $O(N_s * l_{max} * M)$ , де  $N_s$  – довжина тексту в реченнях,  $M$  – довжина мета-ланцюжка,  $l_{max}$  – найбільша довжина речення в тексті.

### 1.7.3. АЛГОРИТМ ПЛАНУВАННЯ

Як уже зазначалося в розділі ПЕРЕДОБРОБКА, на основі результатів роботи алгоритму TextTilling та алгоритму тематичної розмітки було визначено критичні області. Справді, якщо відбувається зміна теми без порушення зв'язності за TextTilling, то таке речення часто є переходом від однієї теми до іншої. Отже, такі речення містять у собі сформовану автором структуру зв'язків між темами тексту.

За результатами визначених критичних областей та відповідно до структури відношень  $R(T)$  можна зафіксувати найбільш важливі переходи логічної структури.

#### *Алгоритм вибору зв'язків між темами*

Для кожного  $s \in T$  з маркером «Критична Область»

Для кожного  $w_i \in s$

Для кожного  $c(w_i)$

Встановити вагу відповідно до ваги ланцюжка, ваги поняття в ланцюжку, важливості переходу та важливості визначеної користувачем (групою експертів)

Створити порожній список  $L$

Для кожного  $s \in T$  з маркером «Критична Область»

Занести у  $L$  оцінку речення складену як суму оцінок  $w_i$

Відсортувати  $L$

Вибрати відсоток оцінок, що відповідає відсотку стиску

Взяти останній елемент з вибраних у якості межі

Для кожного  $s \in T$  з маркером «Критична Область»

Якщо оцінка речення менше межі – поставити маркер «Не потрібне»

Оцінка складності роботи алгоритму на одному проході:  $O(N_K^2 * l_{max} * k_{max} + N_K * \log(N_K))$ , де  $N_K$  – кількість речень з маркером „Критична Область”,  $k_{max}$  – найбільша кількість сенсів слова,  $l_{max}$  – найбільша довжина речення в тексті.

Таким чином, побудований перелік опорних речень дозволяє наближено відобразити той хід думки, який хотів передати автор тексту. У випадку мультиреферування це лише певне наближення, що відображає не хід думки одного автора, а зміну способу подачі матеріалу та акцентів з часом у різних джерелах.

#### 1.7.4 СЕМАНТИКО-СИНТАКСИЧНИЙ АЛГОРИТМ СТИСКУ

Цей алгоритм належить до групи алгоритмів, які виконують абстрагування, з опорою на зовнішні джерела інформації. У наведеному вигляді він не здатний прореферувати весь текст, або стиснути його до малого об'єму. Проте він є корисним, оскільки дозволяє отримати для ряду випадків стиск там, де простий вибір буде змушений втратити інформацію.

Передбачається, що є ряд додаткових механізмів, а саме: синтаксичний та семантичні аналізатори, синтаксичний синтезатор для речень.

У межах областей між двома маркерами „Тема почалася” і „Тема скінчилася”, які належать одній темі, застосовується перший алгоритм узагальнення. Він працює переважно з онтологією, використовуючи зв’язок „бути” (is\_a). У процесі узагальнення цей алгоритм пробігає по онтології від понять нижчого рівня до понять вищого рівня у пошуках поняття, яке є водночас допустимими та досить абстрактними, для можливості здійснення узагальнення. Для нього є обов’язковою синтаксична передобробка, оскільки він інтенсивно використовує синтаксичні дані.

Позначимо тематичну область як  $To$ .

#### *Алгоритм стиску 1*

Для кожного  $s_l \in To$

Скласти предикатну структуру відповідно до підмета і присудка

Назвати її базовою

Для кожного  $s_{l+j} \in To$  (від даного і до кінця області)

Скласти предикатну структуру відповідно до підмета і присудка

Порівняти предикатну структуру з базовою.

Якщо для підметів присудків або і підметів і присудків виконуються «Умови групування» з розділу ІНДЕКСАЦІЯ на відстань 2

З  $s_l$  та  $s_{l+j}$  будується одне  $s_l$ , більш поширене і, використовуючи більш загальні поняття,  $s_{l+j}$  вилучається.

Оцінка складності роботи алгоритму:  $O(N_T^2 * l_{max} * k_{max})$ , де  $N_T$  – довжина блоку тексту в реченнях,  $k_{max}$  – найбільша кількість сенсів слова,  $l_{max}$  – найбільша довжина речення в тексті.

Недоліком цього алгоритму є його чутливість до синтаксичних неоднорідностей та «короткозорість», оскільки він не реагує на зв’язки між поняттями у WordNet, що мають довжину більшу за 2. Проте, якщо збільшити відстань до 3х або 4х, часто відбувається надлишкове узагальнення, що негативно впливає на якість реферату.

Попри очевидні перевагу такого алгоритму, а саме здатність до складання висновків, хоч би і обмежену, необхідно зауважити його малу частоту очікуваного використання.

### 1.7.5. СЕМАНТИЧНИЙ АЛГОРИТМ СТИСКУ

Аналогічно до попереднього, цей алгоритм належить до групи алгоритмів, які виконують абстрагування, з опорою на зовнішні джерела інформації. У наведеному вигляді він не здатний прореферувати весь текст, або стиснути його до малого об'єму. Проте він є корисним, оскільки дозволяє отримати для ряду випадків стиск там, де простий вибір буде змушений втратити інформацію.

Передбачається, що є ряд додаткових механізмів, а саме: синтаксичний та семантичні аналізатори, синтаксичний синтезатор для речень.

У межах областей між двома маркерами „Тема почалася” „Тема скінчилася”, які належать одній темі, також застосовується другий алгоритм узагальнення. Його основою є пошук у ширину в орієнтованому графі онтології. Умови зупинки:

1. Як тільки зустрічається вершина (концепт), що є забороненою (зайвою), алгоритм припиняє обчислювати ваги для цієї вершини та всіх вершин, для яких вона є нащадком в ієрархії WordNet. До заборонених сенсів належать загальні поняття, якщо вони не представлені явно в лексичному ланцюжку.
2. Якщо вершина поза бажаною тематикою, алгоритм припиняє обчислювати ваги для цієї вершини та всіх вершин, для яких вона є нащадком в ієрархії WordNet;
3. Якщо досягнуто довжину шляху, рівну 5.

Позначимо тематичну область як  $To$ . Повнозначне слово –  $w_i$ ,  $c(w_i)_j$  – сенс слова  $w_i$

**Алгоритм узагальнення**

Створити список  $L$

Для кожного  $s_i \in To$

Для кожного  $w_i \in s_i$

Для кожного  $c(w_i)_j$

Додати  $c(w_i)_j$  у  $L$

///

Для кожної ітерації

Створити список  $L_n$

Якщо не виконуються умови зупинки

Для  $c(w) \in L$

Визначити кількість шляхів, що проходять через  $c(w)$  в напрямку більш загального  $c(w)_n$  у WordNet

Занести  $c(w)_n$  у  $L_n$

Встановити  $c(w)_n$  вагу, рівну сумі всіх шляхів від нижніх  $c(w)$  синсетів через нього  $c(w)_n$

Замінити  $L = L_n$

///

Для кожного  $c(w) \in L$

Для кожного  $s_l \in T_o$

Для кожного  $w_i \in s_l$

Для кожного  $c(w_i)_j$

Виконати розмітку за WordNet.

///

Для кожного  $s_l \in T_o$

Створити список  $L$  (речень кандидатів)

Для кожного  $s_{l+j} \in T_o$

Порівняти маркери  $s_l$  та  $s_{l+j}$

Якщо маркери співпадають,

Додати  $s_{l+j}$  у  $L$

Інакше:

Для кожного  $s_k \in L$  (речення зі списку)

Порівняти предикатну структуру  $s_l$  зі  $s_k$ .

Якщо немає відповідностей -вилучити  $s_k$

Опрацювати  $L$ , створивши більш загальне речення  $s_l$

Вилучити використані  $s_k$  з тексту

Зробити список порожнім

Оцінка складності роботи алгоритму на одному проході:  $O(N_T^2 * l_{max} * k_{max} + SS^3 + SS * N_T * l_{max} * k_{max} + N_T^2 * l_{max} * k_{max})$ , де  $N_T$  – довжина блоку тексту в реченнях,  $k_{max}$  – найбільша кількість сенсів слова,  $l_{max}$  – найбільша довжина речення в тексті,  $SS$  – кількість сенсів у списку сенсів. Очевидно, що чим більше різних змістовно наповнених елементів є в межах теми, то повільніше працює алгоритм.

Таким чином можна визначити ті концепти з WordNet, що не представлені в тексті явно, проте сильно пов'язані з його змістом.

Це дозволяє, наприклад, узагальнити „стіл, стілець, ліжко” до „меблі” але не до „об'єкт”. Проблема полягає у тому, що як і попередній алгоритм, цей також чутливий до синтаксичних структур. Так само, як і у попередньому випадку, необхідно зауважити малу ефективність (частоту очікуваного вживання) даного алгоритму.

#### 1.7.6. АЛГОРИТМ СТИСКУ ВИБОРОМ

Практично це найбільш дієвий алгоритм, оскільки він не чутливий до можливих синтаксичних неоднорідностей.

Позначимо тематичну область як  $To$ . Повнозначне слово –  $w_i$ ,  $c(w_i)_j$  – сенс слова  $w_i$

##### *Алгоритм стиску вибором*

Для кожного  $s_l \in To$

    Для кожного  $w_i \in s_l$

        Для кожного  $c(w_i)_j$

            Встановити вагу відповідно до ваги ланцюжка, ваги поняття в ланцюжку та важливості визначеної користувачем

Створити порожній список  $L$

Для кожного  $s_l \in To$

    Занести у список оцінку  $s_l$  як складену як суму оцінок  $w_i$

Відсортувати список  $L$

Вибрати відсоток оцінок, що відповідає відсотку стиску

Взяти останній елемент з вибраних в якості межі

Для кожного речення  $s_l \in To$

Якщо оцінка речення менше межі – поставити маркер «Не потрібне»

Оцінка складності роботи алгоритму на одному проході:  $O(N_T^2 * l_{max} * k_{max} + N_T * \log(N_T))$ , де  $N_T$  – довжина блоку тексту в реченнях,  $k_{max}$  – найбільша кількість сенсів слова,  $l_{max}$  – найбільша довжина речення в тексті. Даний алгоритм дозволяє стискати текст до необхідного розміру. Проте, він гарантовано буде незв'язний текст, який важко читається.

### 1.7.7 ЗАГАЛЬНИЙ АЛГОРИТМ РЕФЕРУВАННЯ

Алгоритм реферування базується на послідовному застосування алгоритмів, описаних вище. Вони застосовуються в такому порядку:

1. Texttilling
2. Алгоритм побудови лексичних ланцюжків
3. Алгоритм побудови зв'язків між лексичними ланцюгами
4. Індксація
5. Алгоритм розмітки тематичних областей
6. Алгоритм визначення критичних областей
7. Алгоритм планування.
8. Семантико-синтаксичний алгоритм стиску
9. Семантичний алгоритм стиску
10. Алгоритм стиску вибором

Відповідно до часових оцінок роботи алгоритмів, застосування алгоритму семантичного стиску допускається, тільки якщо час побудови реферату не має значення, оскільки його часова оцінка  $O(S^3)$ , де  $S$  – кількість понять у тексті. Якщо його не застосовувати, то оцінка складності реферування в цілому буде  $O(N^2)$ , де  $N$  – довжина тексту у словах.

Побудований таким чином реферат має бажаний відсоток стиску. Застосування алгоритму стиску вибором після інших алгоритмів стиску гарантує коректність роботи алгоритмів стиску, що використовують семантику.

Проте у побудованого таким чином реферату є ряд суттєвих недоліків. Попри намагання передати логічну структуру тексту результат може погано читатися.

#### 1.7.8 ПОКРАЩЕННЯ РЕЗУЛЬТАТІВ РЕФЕРУВАННЯ

Є два підходи до покращення реферату, особливо отриманого екстракцією речень: перевпорядкування речень та екстракція одразу груп речень.

Перевпорядкування спирається на припущення, що елементи, віднесені до однієї тематики та вибрані користувачем, було рознесено в оригіналі. Тоді для покращення їх необхідно зібрати разом. Даний метод не розглядається, оскільки він вступає у протиріччя з ідеєю, що послужила основою для методу виділення критичних областей.

Екстракція груп речень. Граф зв'язності тексту представляє собою щось схоже на ланцюжок, де речення найчастіше пов'язане зі своїми сусідами і не пов'язане з віддаленими реченнями. Залежно від контексту, в якому знаходиться речення у рефераті, його оцінка може бути збільшена або ж зменшена. Якщо речення, що розглядається, відповідає принципу нерозривності для речення, яке передує йому або йде за ним у рефераті, то його оцінка збільшується. Те, яким чином визначається, які речення передують йому та слідує за ним, залежить значною мірою від того, які алгоритми використовуються. Якщо це перше або останнє речення (тобто перед ним або після нього немає речень), то для цього речення не проводиться оцінювання.

Якщо порушується принцип нерозривності, то оцінка зменшується. Після експериментів з різними значеннями було вирішено збільшувати оцінку речення за допомогою максимальної оцінки сенсу у тексті штрафувати також, за допомогою максимальної оцінки сенсу у тексті документа.

Задача в загальному випадку є повноперебірною, тому застосовується алгоритм, який здатний досить ефективно виконати перебір і знайти хоча б локально оптимальну оцінку. Тому застосовується генетичний алгоритм.

Хромосомою будемо називати список номерів речень, які входять до реферату. Хромосома може мутувати, змінюючи значення елементу списку. Дві хромосоми можуть поелементно обмінюватися даними, при цьому утворюється дві нові хромосоми – нащадки, не тотожні батькам.

*Генетичний алгоритм*

Генерується хромосома за рефератом

Хромосома заноситься у список.

На хромосому накладаються випадкові мутації  $N-1$  разів. Мутанти заносяться у список.

Для всіх кроків

Створюється новий список

Для всіх хромосом зі списку

Для всіх хромосом зі списку

Пара хромосом породжує нащадків

Нашадки заносяться у новий список

Хромосоми в новому списку сортуються за оцінкою

Відбираються  $N$  кращих.

Результат стає новим списком.

Оцінка роботи алгоритму:  $O(R*(2N^2* + N*l_{max}^2 *V + N*log(N)))$ , де  $N$  – довжина тексту в реченнях,  $l_{max}$  – найбільша довжина речення в тексті,  $V$  – складність побудови оцінки хромосоми,  $R$  – кількість ітерацій.

На особливу увагу заслуговують такі моменти:

- створення мутацій;
- створення нащадків;
- оцінка хромосоми.

При створенні мутацій, як і при породженні нащадків, забороняється вилучати з реферату елементи, відмічені як «Критичні області». При обчисленні оцінки головним є врахування максимальних ваг лексичних

ланцюжків, обчислених на рефераті. Це принаймні дозволить враховувати слова, що мають сенси поєднані зв'язком гіпернім/гіпонім.

Швидкодія генетичного алгоритму: в практичних експериментах за 100 кроків вдавалося досягти пристойного результату.

## 1.8. ОЦІНЮВАННЯ ЯКОСТІ РЕФЕРАТІВ

Метою методів оцінки рефератів є визначення адекватності (та достовірності) або корисності реферату по відношенню до оригінального тексту. Використовуються дві методики оцінки.

Оцінка „зсередини” (або нормативна оцінка). Користувачі приймають рішення про якість реферату, аналізуючи сам реферат. Користувачі оцінюють гладкість тексту, роблять висновок про те, наскільки добре реферат відображає основні ідеї оригіналу, або порівнюють його з „ідеальним” рефератом, написаним автором вихідного тексту або іншим фахівцем. Жодна з цих оцінок не може вважатися повністю задовільною. Зокрема, отримати достатню кількість „ідеальних” рефератів важко.

Подібно до того, як існує безліч способів описати якусь подію, користувачі можуть визнати прийнятними кілька рефератів, отриманих системами різного призначення. Як показує практика, люди взагалі рідко приходять до згоди щодо того, які положення чи висловлювання слід включати в реферат [\*8\*].

Оцінка „ззовні”. Користувачі оцінюють якість реферату по тому, як він впливає на завершення тієї чи іншої роботи, наприклад, допомагає знайти джерела інформації з даного питання, або наскільки добре він дозволяє відповісти на певні питання, пов'язані зі змістом всього тексту. Одним із способів є оцінювання часу на пошук та аналіз інформації з використанням та без використання реферату.

Машинні метрики спираються на використання множини еталонних рефератів. Основні з них: Оцінка, орієнтована на повноту основної суті (ROUGE), Найдовша Спільні Підпоследовність (LCS)

ROUGE спирається на n-грами, що повинні бути спільними у рефераті-еталоні та рефераті, зробленому машиною. Є варіації, де n-грами генеруються з розривами (слова одне за одним, але зібрані з пропуском деяких слів), з штрафами за розрив структури.

LCS спирається на ідею, що чим більша спільна підпоследовність, тим краще реферат відповідає еталону.

## 1.9 ВИСНОВКИ

Описано ряд алгоритмів, що разом формують основу для створення системи автоматичного реферування або мультиреферування. Кожний з них окремо не забезпечує достатньої якості реферування. Проте за умови використання їх у комплексі більшість проблем вдається розв'язати. Це дозволяє досить ефективно будувати реферати, що добре передають зміст оригіналу та водночас добре читаються.

### *Контрольні запитання до розділу*

1. Задача автоматичного реферування.
2. Визначення тематичної структури документів.
3. Визначення основних змістовних елементів. Частотна модель.
4. Визначення основних змістовних елементів. Модель на основі опорних елементів та тематик.
5. Використання онтологій для визначення основних змістовних елементів.
6. Генерація реферату: реферування вибором, забезпечення зв'язності.
7. Реферування багатьох документів

## 2. МАШИННИЙ ПЕРЕКЛАД

Машинний переклад передбачає виконання комп'ютером перекладу тексту з однієї природної мови на іншу без участі людини та результат такої роботи. *Задача машинного(автоматичного) перекладу* потребує морфологічного аналізу, аналізу і перекладу лексики, синтаксичного аналізу і синтезу семантичних трансформацій, які б забезпечували смислову рівність введеної і виведеної текстової інформації. Звідси випливає, що *задача машинного(автоматичного) перекладу* – це задача штучного інтелекту, який би зміст не вкладався в поняття штучного інтелекту [4,9].

На даний час серед найбільш досліджуваних є системи з використанням мови-посередника (проміжної мови), змішаного підходу, на основі навчання машин, на основі корпусів, на основі прикладів та гідридних методів на основі статистик та прикладів. Працюють як одномовні, так і багатомовні системи, не лише для письмових текстів, але й для усного мовлення.

Окремо стоїть *задача автоматизованого перекладу*, яка передбачає тісну взаємодію перекладача та системи на всіх етапах перекладу. Завдяки успіхам в розробці систем автоматизованого перекладу професійні перекладачі масово користуються такими системами, особливо для перекладів з повторюваною тематикою або перекладів, де треба узгоджувати роботу багатьох перекладачів.

### 2.1. АВТОМАТИЧНИЙ ПЕРЕКЛАД

Оскільки першопрохідцями були математики і програмісти, для першого етапу розвитку машинного перекладу було характерне так зване „кодування-декодування”.

Цей підхід називається прямим методом, у ньому переклад розглядається як звичайний аналог тексту оригіналу. Відповідно до методу прямого перекладу, вихідний і цільовий тексти повинні бути схожі і за своєю формою, і за концептуальним змістом. Ця ідея виявилась обмеженою

вузьким колом текстів спеціалізованої тематики – прогноз погоди тощо.

Схему подано на Рис. 2.1.



Рис. 2.1. Прямий метод

Протягом 1970х-80х рр. відбувався розвиток так званих систем „другого покоління”, побудований на правилах, спрямованих на лінгвістичну обробку, як правило, у три етапи: синтактико-семантичний аналіз вихідного тексту, застосування правил перетворень з більш-менш абстрактним рівнем представництва та генерування цільового тексту з синтаксичного представлення вхідного тексту. У той же час точилися дебати про те, як можна використовувати системи, побудовані за цієї архітектурою, для забезпечення прийняттого рівня перекладу для реальних користувачів. Найпопулярнішими були ідеї обмеження входу (підмова і контрольовані мови), або переклад за участю користувача в перед- і пост-редагуванні[3].

Схему подано на Рис. 2.2.



Рис. 2.2 Оптимізований прямий метод

Щоб поліпшити якість прямого перекладу, застосовуються два наступні методи, а саме: синтаксичні фільтри і статистичне ранжування перекладних еквівалентів, які б дозволили вибрати найбільш ймовірні з них для конкретного документа, що перекладається.

Синтаксичні фільтри мають форму логічних фреймів, де слоти заповнені синтаксичними структурами з зазначенням функції. Зазвичай в системах машинного перекладу на основі прямого методу досить багато фільтрів для „згладжування” сирого перекладу.

Другий основний метод машинного перекладу - це спосіб переносу інформації, заснований на правилах перетворень[12]. (Першим вважається прямий метод).

У системі на основі перетворень процес перекладу включає наступні стадії обробки: морфологічний та синтаксичний аналіз, власне перенос інформації у проміжному представленні, синтез синтаксичних структур, морфологічний синтез (побудова тексту перекладу). Досить часто системи на основі переносу містять семантичну складову. Мережа семантичних описів і відносин накладається на синтаксичні структури вихідного тексту і тексту мети (тобто власне перекладу). Метою семантичної компоненти є підвищення точності перекладу. Схему подано на Рис. 2.3.



Рис 2.3. Метод, заснований на правилах перетворень

Третім основним методом є використання проміжної мови[13]. У певному сенсі це схоже на попередній метод, однак існує декілька важливих відмінностей. На відміну від процедур переносу, які застосовуються здебільшого на синтаксичному рівні з деякими коригуванням семантики, представлення інформації через проміжну мову включає всю доступну лінгвістичну інформацію.

Крім того, системи на основі проміжної мови претендують на універсальність, тобто поширюються на будь-які мови.

Проміжна мова є формальним описом морфологічних, синтаксичних і семантичних характеристик мовної одиниці у вигляді співвідношення один-до-одного. Кожна одиниця мови пов'язана з конкретним незмінним атомом у структурі проміжної мови і навпаки - кожен атом структури проміжної мови незмінно пов'язаний з одиницями різних мов.

В ідеалі, модель із застосуванням проміжної мови у машинному перекладі має включати наступні етапи обробки: морфологічний, синтаксичний та семантичний аналіз вихідного тексту, використовуючи інформацію зі словника мови оригіналу і парадигм; формування представлення мови вихідного тексту модулем проміжної мови; перетворення початкового представлення модулем проміжної мови у текст перекладу, використовуючи відповідні семантичні, синтаксичні, лексичні та морфологічні дані зі словника мови перекладу і парадигм.

Зазвичай формалізм проміжної мови має вигляд графічного мережі або її аналітичного еквівалента. Це дуже складна система морфологічних, синтаксичних та семантичних одиниць і відносини. Схему подано на Рис.2.4.



Рис 2.4. Модель на основі проміжної мови:

На особливу увагу заслуговують системи на основі методу штучного інтелекту (ШІ) – artificial intelligence (AI), які спираються на енциклопедичні дослідження.

Основним компонентом моделі перекладу на основі ШІ є його так звана „база знань”. Відповідно до моделі перекладу, заснованій на ШІ, - основі результати лінгвістичного аналізу на всіх рівнях мови перевіряються за допомогою позамовної інформації, що міститься в базі знань.

У всіх трьох вищезгаданих способах моделювання перекладу усунення неоднозначності здійснюється тільки за допомогою контексту. Жодна з цих моделей, однак, не використовує двох інших інструментів усунення неоднозначності, тобто ситуації та довідкової інформації.

У моделях перекладу, заснованих на ШІ, процедури усунення неоднозначності радикально відрізняються і ґрунтуються перш за все на аналізі ситуації та довідкової інформації (бази знань), в той час як лінгвістичні методи аналізу контексту служать тільки в якості вторинних резервних засобів. Бази знань містять особливим чином впорядковані ієрархії фактів про реальний світ, а вербальна інформація відіграє підлеглу роль, лексично позначаючи факти і ситуації. Ще одним важливим компонентом моделювання перекладу на основі ШІ є модуль прийняття рішень, який включає структурну ієрархію логічних побудов з оцінкою імовірності.

Нинішній рівень складності моделювання перекладу на основі ШІ досить неоднозначний - з одного боку, результати розвитку моделей ШІ, призначених для перекладу як такого вельми обмежені, а з іншого, однак, розробка моделей ШІ, призначених для інтерфейсу природною мовою, особливо для експертних систем, дуже ефективна.

Машино-орієнтовані статистичні методи складають наступну групу підходів[11]. У статистичних методах моделювання перекладу передбачається, що з певною ймовірністю кожне слово тексту перекладу може бути перекладом кожного слова вихідного тексту, але різні статистичні моделі відрізняються щодо подальших імовірнісних оцінок. Модель може оцінювати:

- імовірності узгодження порядку слів у тексті оригіналу і результуючому тексті перекладу
- імовірності словосполучень у тексті оригіналу і результуючому тексті перекладу тощо.

Схему подано на Рис 2.5.



Рис.2.5. Статистична модель

У кінці 1980-х - на початку 1990-х рр. розвивається модель з використанням пар приклад-переклад для довгих конструкцій. Це стало можливим завдяки таким факторам:

1. Розширенню можливості ЕОМ зберігати великі бази прикладів.
2. Наявності великих двомовних корпусів текстів в електронних форматах.

Схему подано на Рис 2.6.



Рис.2.6. Діаграма 6. Модель перекладу, побудована на прикладах

Слід зазначити, однак, що жоден з методів машинного перекладу не використовується в реальних системах у чистому вигляді.

У даний час розробляються також змішані або гібридні системи, що використовують як імовірнісні, так і лінгвістичні методи для отримання найкращого результату.

### 2.1.1 МІКРОКОСМОС

Проект Мікрокосмос розроблявся під керівництвом С. Ніренбурга в 1991-99 рр[13]. В ньому доведено до реалізації метод перекладу з проміжною

мовою. Розробники Мікрокосмосу займалися в основному семантичним аналізом, морфологічні і синтаксичні аналізатори були ними запозичені. Проект покладається на ідею максимально продуктивно синтезувати множину існуючих на сьогодні теоретичних розробок (т.з. мікротеорій) в єдину систему. До числа найбільш цікавих мікротеорій, що адаптувалися і були покращені в Мікрокосмосі відносяться:

- 1) теорія організації онтології, принципів виводу інформації за нею;
- 2) засоби застосування онтології до реальних текстів, засоби розв'язання омонімії;
- 3) інтеграція конкретних семантичних мікротеорій.

Концепти реалізуються в тексті в словах. Слова можуть бути омонімічні або полісемічні, їм може бути приписано декілька концептів, з яких потрібне вибрати один. Проблема вибору потрібного значення слова - одна з найбільш складних, повний перебір варіантів значення на великих текстах не є можливим через велику складність. Тому використовуються евристики: лінгвістичні та логічні. Лінгвістичні евристики є застосовними в конкретних мовах, вони звичайно звужують область пошуку конкретними правилами, зменшують відрізки тексту, на яких потрібно застосовувати повний перебір. Логічні евристики - зв'язані з припущенням, що семантична структура речення найчастіше буває деревом.

*Обмеження задані текстом:* значення слова зумовлює вибір значення іншого слова. Граф, в вузлах якого містяться слова, а на дугах - знаходяться обмеження, будемо називати *графом обмежень*. Стрілки графу обмежень отримують з синтаксичного аналізу, а самі обмеження - з лексикону та онтології, вони називаються *селективними обмеженнями*. Частина обмежень може бути записані в словникові або онтології. (Подробиці будуть далі.) Наприклад, може бути, що прямий об'єкт даного присудка знаходиться в такому-то семантичному зв'язку підметом іншого речення. Зрозуміло, що між цими об'єктами прямого синтаксичного зв'язку немає.

Знаходження всіх розв'язків приписування значень складає окрему теорію обмежень (Constraint satisfaction theory). Основним положенням теорії

обмежень, служить той факт, що граф обмежень в природних мовах найчастіше буває деревом або майже деревом в тому сенсі, що можна розбити граф на такі підграфи, коли число внутрішніх залежних вузлів(залежних від зовнішніх для даного дерева), істотно менше загального числа вузлів, що входять в цей підграф. Це розбиття виконується рекурсивно, поки не доходять до найменших підграфів, для яких ця задача розв'язна безпосередньо.

Основні результати в проекті Мікрокосмос:

1) Застосування більш ніж попарного бінарного способу злиття підграфів, що дозволяє оптимально розглядати варіанти і здійснювати повернення;

2) Використання кількісних обмежень, що не просто можуть виконуватися або не виконуватися, а мають деяку оцінку від 0 до 1.

Для впровадження останнього вдосконалення потрібне використати засіб знаходження мінімального шляху в зваженому графі для отримання кращої комбінації рішень на підграфах.

Відбулося успішне впровадження теорії обмежень в реально працюючу систему і ця теорія дозволила скоротити кількість переборів з мільйонів до сотень варіантів.

Таким чином онтологія в Мікрокосмосі являється ієрархією фреймів, де для кожного слота, що має бути заповнений, є: обмеження, значення за замовчуванням та, можливо, перелік виключень/заборон.

Мікротеорія прикметників. Перше лінгвістичне спостереження полягає в тому, що в мовах, де немає спеціальних форм для прикметників, роль прикметників грають або іменники, або дієслова. Це розмежування можна спроектувати на мови з прикметниками, виділивши два класи прикметників: іменникові та дієслівні, хоча, це не покриває всієї множини прикметників. Хоч би тому, що прикметники означають тільки одну властивість, а іменники – набори властивостей.

Автори запропонували свою інтерпретацію семантики прикметників. Основним елементом їхньої теорії є поняття шкали, що бувають двох видів:

чисельні і символні. Наприклад, шкала розмірів “маленький, середній, великий і т. д.” - чисельна, а шкала кольорів “червоний, синій, зелений” - символна. Всі прикметники діляться на дві категорії: шкальні прикметники (scalar adjectives) - ті, що прив'язані до якої-то шкали, і нешкальні прикметники (nonscalar adjectives), у яких немає шкал.

Таким чином, значень у прикметників не багато (для прикметника *ефективний* - одне), але зате при кожному об'єкті прописане, що в цьому об'єкті може змінюватися і по яким шкалам або які значення може набувати.

Довгі переліки значень для прикметників, схожі на ті, що є в WordNet, неприпустимі в Мікрокосмосі, тут діють два додаткові правила.

1) Спробувати поставити двох кандидатів на різні значення в одне речення. Якщо потрібен додатковий контекст, щоб реалізувалося одне з значень, те це значення не є самостійним, і повинно бути включене в інше.

2) Якщо кандидат на окреме значення застосований тільки до обмеженого класу семантично подібних іменників, значить це значення потрібно включити або підпорядкувати(успадкувати) від вже існуючого значення.

Іменники утворюють піддерево онтології, запозичуючи значення та обмеження у предків. Дієслова є головними елементами, що формують фрейми речення, задають часову послідовність, тощо.

Загальний результат роботи показує, що створення моделі значення тексту можливе, але обсяг предметної області, для якого можна реалізувати проміжну мову серйозно обмежений можливостями до залучення висококваліфікованих експертів-лінгвістів. Відповідно, це обмежує дієвість таких систем в цілому.

### 2.1.2. СТАТИСТИЧНИЙ ПЕРЕКЛАД

Переклад[9] спирається на розподіл  $P(T=t, A=a, S=s)$ , в якому  $T$  відповідає за рядок цільовою мовою,  $S$  за рядок початковою мовою,  $A$  за співставлення. Найбільш цікавий тут розподіл  $P(T=t/S=s)$ .

$$P(t | e) = \sum_a P(t, a | s)$$

Ключовим елементом є співставлення (alignment): відповідність між словами оригіналу та перекладу. Приклад подано у Таблиці 2.1.

Таблиця 2.1

	1	2	3	4	5	6	7
S	I	do	not	want	to	waltz	anymore
	S1-A1	S2-A2	S3-A3	S4-A2	S5-A4	S6-A4	S7-A5
A	A1	A2	A3		A4		A5
	A1-T1	A5-T2	A3-T3	A2-T4	A4-T5	A4-T6	
T	Я	більше	не	хочу	танцювати	вальс	

Як видно з таблиці, кількість елементів що є ключовими структурними елементами не збігається з кількістю слів у реченнях. Тому, виникає потреба у способі фіксації зв'язків виду  $m:n$  у системі. Для цього вживаються коефіцієнти співставлення. В свою чергу, такі коефіцієнти, разом з відповідними імовірностями краще всього здобувати з паралельних двомовних корпусів.

Для рядочків  $t_1 t_2 \dots t_m$ ,  $s_1 s_2 \dots s_l$  та співставлення  $a_1 a_2 \dots a_m$  (де елементи  $a_i$  набувають значень від 1 до  $l$ ) можна записати загальне рівняння:

$$P(t, a | e) = P(m | s) \prod_{j=1}^m P(a_j | a_1^{j-1}, t_1^{j-1}, m, s) P(t_j | a_i^j, t_1^{j-1}, m, s)$$

За цією схемою утворюються такі моделі (IBM Models 1-5), або *генеративні моделі*.<sup>[\*\*]</sup>.

Модель 1.  $P(m | s)$  не залежить від  $m, s$ . Тому  $P(a_j | a_1^{j-1}, t_1^{j-1}, m, s)$  залежить тільки від  $l$ ,  $P(t_j | a_i^j, t_1^{j-1}, m, s)$  залежить тільки від  $t_j$  та  $s$ . Модель 1 приймає всі співставлення однаковою ймовірністю (таким чином, порядок слів в  $t$  і  $s$  не впливає на результат). Тоді рівняння набуває вигляду:

$$P(t, a | s) = \frac{\mathcal{E}}{(l+1)^m} \prod_{j=1}^m P(t_j | s_{a(j)})$$

Для кожного слова-цілі  $t_i$ , що відповідає початковому слову  $s_{a(j)}$  визначеного функцією відповідності  $a(j)$  є своя імовірність.  $P(m | s) = \mathcal{E}$  - в цій моделі виконує роль нормуючого множника.

Е.М. Алгоритм складається з двох етапів

1. Оцінка параметрів:

- Застосувати модель до даних. З використанням моделі, призначити ймовірності можливих значень.

2. Максимізація: Оцінка моделі з даних

- Прийняти значення як задані

- Зібрати (псевдо)частоти (зважені за ймовірностями)

- Переоцінити модель за частотами

Умова зупинки максимізації: поки не перестануть змінюватися частоти.

Дану модель варто використовувати в якості допоміжного інструмента для наступних моделей, а не самостійно.

Модель 2. Є узагальненням Моделі 1. При обчисленнях враховуються для  $P(a_j | a_1^{j-1}, t_1^{j-1}, m, s)$  також  $j$ ,  $m$ ,  $a_j$ . В цілому обчислюється за тим самими алгоритмом.

Модель 3. Допускається переклад одного слова з  $s$  у кілька слів  $t$ . Це задається ймовірностями розмноження. Допускається перевпорядкування елементів. Складність віднаходження коректного перевопрякування обумовлює проблеми застосування моделі.

Модель 4. Розроблялася як спроба врахувати, що насправді елементи рухаються блоками, що зумовлені властивостями конкретної мови.

Моделі 3 та 4 мають спільний недолік, через те, що для приведення їх до вигляду, коли вони можуть бути обчислені виконуються спрощення. Тому що ймовірності перевпорядкування для призначення позицій слів в кінці рядка не залежать від позицій, присвоєних словам, що стоять на початку, Моделі 3 та 4 називаються дефіцитними, бо витрачають частину ймовірнісної маси зайве. Деякі з ймовірностей стосуються, „узагальнених рядків”, тобто рядків, які мають деякі з відповідностей між словами виду  $1:n$ . Проте в цілому частка таких співставлень є не такою великою, а отже видатки певною мірою є невиправданими.

Модель 5. Розроблялася, як обчислювальний механізм, щоби побороти дефіцитність. Має надмірну складність.

Через зазначені недоліки, найбільш популярним механізмом зараз є *переклад побудований на фразах*. Для цього, речення розбиваються на фрази(які зовсім не обов'язково є граматичними) і набір фраз перекладається та перевпорядковується. Ключовими елементами є співставлення, що обчислюється над фразами, та перевпорядкування.

Для коректного обчислення співставлення корпуси підготовлених текстів є обов'язковим елементом, оскільки без них, неможливо коректно розбити на фрази. Евристикою, що дозволяє спростити цю задачу, є обчислення співставлень в обидві сторони  $s \rightarrow t$  так і  $t \rightarrow s$ . Цей процес називається *симетризацією* співставлення.

Окремий вид співставлення, що може бути вжитий альтернативно до генеративного підходу – а саме розрізняючий (*discriminative*), обчислюється за формулою:

$$\hat{a} = \arg \max_a \sum_i \lambda_i h_i(s, a, t)$$

де  $\lambda_i$  - вагові коефіцієнти, а  $h_i$  - різні ознаки. Недоліком цього співставлення є необхідність в зарані розмічених коректних корпусах.

Щодо перевпорядкування, то значна частина систем використовує лексикалізовані моделі зміни порядку, в яких перевпорядкування визначаються безпосередньо на фразами (або блоками). Ці моделі навчаються синхронно з фразовою моделлю перекладу. Кожна пара фраза в лексикалізованій моделі п перевпорядкування отримує присвоюється один з трьох напрямків: монотонний ( $m$ ), перестановка( $s$ ), або перенос( $d$ ). Орієнтація задається на основі положення фрази по відношенню до інших слів для пари речень  $t, s$ .

Коли пара фраз аналізується для моделі перекладу, орієнтації також записуються. Розподіл ймовірностей  $p_o$  для моделі зміни порядку оцінюється на основі підрахунку як часто конкретні пари фраз мають кожен з трьох типів орієнтації.

$$p_o(\text{орієнтація}|t, s) = \frac{\text{частота}(\text{орієнтація}|t, s)}{\sum_o \text{частота}(o, t, s)}$$

де *орієнтація*  $\in \{m, s, d\}$  прогнозується для кожної фрази пари джерело-ціль по всім можливим орієнтаціям *o*.

Найбільш дієвим зразком статистичного перекладу є перекладач Гугль(Google Translate). Завдяки доступу до величезних обсягів даних система може оперувати n-грамами великої довжини.

Оскільки Google Translate використовує статистичні відповідності у перекладі, а не правила та словник, перекладений текст часто включає явні помилки, часто вживаючи загальні терміни для схожих, але нееквівалентних загальних термінів на іншій мові, іноді інвертуючи сенс.

Через застосування мови-посередника часто втрачаються ознаки відмінків, навіть якщо вони були в початковій та кінцевій мові. Також Гугль був звинувачений в сексизмі через статистичний спосіб присвоєння статі при перекладі. Наприклад, в дієсловах, коли певні дії прив'язуються лише до одного роду.

Наступні мови не мають прямого перекладу на англійську і опрацьовуються через вказану проміжну мову (яка у всіх випадках тісно пов'язана з потрібною мовою, але більш широко поширена):

Білоруська мова (be ↔ ru ↔ en ↔ інші)

Каталонська мова (ca ↔ es ↔ en ↔ інші)

Галісійська мова (gl ↔ pt ↔ en ↔ інші)

Гаїтянська креольська мова (ht ↔ fr ↔ en ↔ інші)

Македонська мова (mk ↔ bg ↔ en ↔ інші)

Словацька мова (sk ↔ cs ↔ en ↔ інші)

Українська мова (uk ↔ ru ↔ en ↔ інші)

Урду (ur ↔ hi ↔ en ↔ інші)

### 2.1.3 ГІБРИДНИЙ ПЕРЕКЛАД

Для гібридного підходу характерне застосування імовірнісного підходу в комбінації зі знаннями про мову[11]. Наприклад, при розборі речення використовуються імовірнісні контекстно-вільні граматики, для побудови розбору і після того для генерації перекладу вже за правилами, по готовому розбору.

Для отримання ймовірностей проводиться підрахунок числа раз ( $N$ ), коли використовується деякий варіант розгортання вузла ( $\alpha \rightarrow \beta$ ) з наступною нормалізацією :

$$P(\alpha \rightarrow \beta | \alpha) = \frac{N(\alpha \rightarrow \beta)}{\sum N(\alpha \rightarrow \beta)} = \frac{N(\alpha \rightarrow \beta)}{N(\alpha)}$$

Значення ймовірності використовуються в процесі граматичного розбору. Кожному дереву  $T$  присвоюють ймовірність ( $P$ ) кожному дереву для речення  $S$ . Ця інформація є ключовою для розв'язання неоднозначності синтаксичних структур.

Ймовірність кожного можливого дерева розбору  $T$  визначається як добуток ймовірностей всіх правил  $r$ , використовуваних для розгортання кожного вузла  $n$  в дереві розбору:

$$P(T, S) = \prod_{n \in T} p(r(n))$$

Ймовірність повного розбору речення обчислюється з урахуванням категоріальної інформації для кожної головної вершини кожного вузла. Нехай  $n$  - синтаксична категорія деякого вузла  $n$ ,  $h(n)$  - головний вершина вузла  $n$ ,  $m(n)$  – батьківський вузол для вузла  $n$ , таким чином, обчислюється ймовірність  $P(r(n) | n, h(n))$ , для цього вираз перетворюється таким чином, що кожне правило стає обумовленим своєю головною вершиною.

$$P(T, S) = \prod_{n \in T} p(r(n) | n, h(n)) \times p(h(n) | n, h(m(n)))$$

Це дозволяє коректно застосувати правила подальшого перетворення розборів речень у текст.

## 2.2. АВТОМАТИЗОВАНИЙ ПЕРЕКЛАД

Паралельно з машинним перекладом, який за визначенням є повністю автоматичною системою, розвивається автоматизований переклад (АП) – computer-assisted translation (CAT), що є інструментом, який допомагає суттєво прискорити людський письмовий переклад науково-технічної літератури, у той час як в усному перекладі АП обмежується доступом до словників оффлайн/онлайн та інтернет-ресурсів онлайн[14].

Основою автоматизованого перекладу є системи типу „машинна пам'ять перекладача” (МПП) - translation memory (TM). Вони з'явилися після масового поширення персональних комп'ютерів. Системи МПП є найбільш широко використовуваними прикладними програмами в локалізації цифрової інформації, тобто перекладі і культурній адаптації електронного контенту для місцевих ринків.

Ідея їх ключового елементу – „пам'ять” або архів перекладів, яка зберігає оригінали та їх переклади людиною в комп'ютерній системі з розбивкою на певні одиниці. З часом, величезні колекції речень та їх відповідні переклади накопичуються в системах МПП. Це дозволяє перекладачам використовувати такі перекладені сегменти, вибираючи з автоматично запропонованих відповідний переклад з пам'яті як повний (точний) збіг (perfect match) або як частковий (нечіткий) збіг (fuzzy match). Нечіткий збіг виникає, коли речення схоже, але не співпадає дослівно. Крім всього іншого, це допомагає гарантувати, що термінологія і вирази вживаються послідовно, без переходів до інших тематик. Схему роботи МПП представлено на Рис. 2.7.

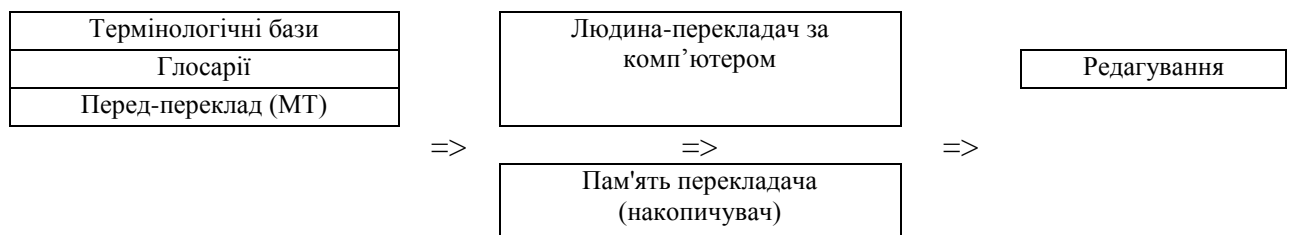


Рис. 2.7. Автоматизований переклад.

Оскільки МПП полегшує і прискорює переклад спеціалізованих текстів, кількість яких дедалі швидше зростає, більшість професійних перекладачів використовує технологію МПП на регулярній основі. Ніяка інша технологія не змінила загальні умови перекладу так радикально, як професійне програмне забезпечення за останні 20 років. Це може бути пов'язано з тим, що професійні перекладачі виконують величезну кількість повторюваної, рутинної роботи над типовими документами, без істотного залучення до ситуацій, які вимагають творчого підходу.

На сьогодні в Інтернет-ресурсах нараховується понад 50 таких - систем АП, які невпинно вдосконалюються і перетворюються на *середовище перекладу*(СП) - translation environment (TeNT). СП є вже третім поколінням інструментів автоматизованого перекладу. У СП основною метою ставиться створення не лише баз перекладів та допоміжного інструментарію, але і надання найбільш зручного інтерфейсу та забезпечення всіма необхідними допоміжними функціями. Приклад інтерфейсу наведено на Рис. 2.8. Найбільш відомі серед СП - SDL Trados, Transit, Deja Vu, Wordfast, AIT, MemoQ.

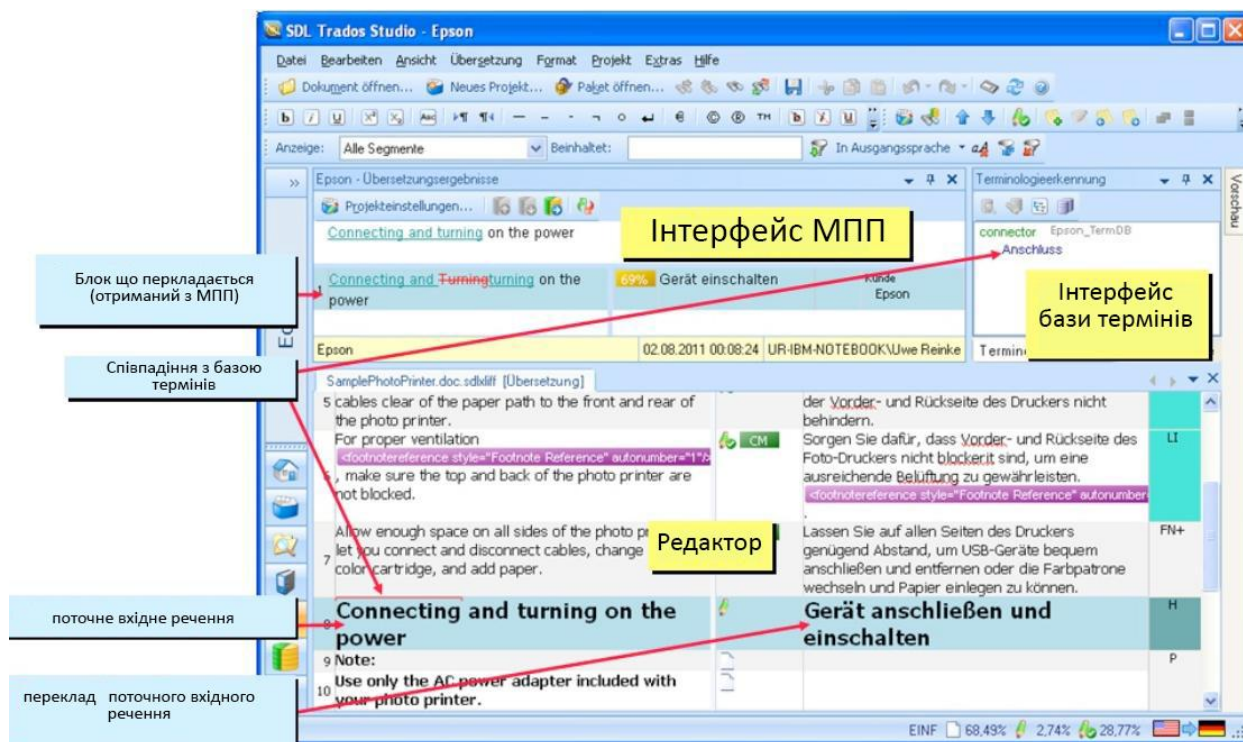


Рисунок 2.8 : Інтерфейс користувача SDL Trados Studio Цитується за: Uwe Reinke *State of the Art in Translation Memory Technology*[14 ]

Мережеві технології також заявили про себе, пропонуючи доповнювати АП та СП машинним перекладом на основі статистичного методу у таких потужних інструментах як сервіс Google Translate, що використовує власне програмне забезпечення Google. Цей сервіс дозволяє автоматично перекладати слова, фрази тексти та web-сторінки з однієї мови на іншу, оперуючи понад 80-ма мовами. Переклад на 8 мов, включаючи українську, білоруську, каталонську, відбувається через споріднену, але більш поширену проміжну мову. Працюють із сервісом Google Translate також Globefish, gTranslate та UnofficialGoogleTranslate.

Гібридизація моделей продовжується і іншими розробниками, і не лише із залученням англійської мови, як скажімо, португальсько-китайська PCTAssist. Почали поєднувати МП і автоматизований переклад з хмарними технологіями, як от Memsource Cloud. Це повноцінне перекладацьке середовище, запущене спочатку у закритій бета-версії у 2011р., що включає пам'ять перекладів, інтегрований модуль машинного перекладу, управління термінологією і перекладацький редактор у вигляді веб-додатків та автономної програми і використовує хмарний сервіс. <http://www.memsource.com>

### 2.2.1 КОМПОНЕНТИ МПП

Типова система МПП складається з масиву інструментів і функцій для допомоги перекладачу. До неї входять:

- „Пам'ять” чи архів перекладів.
- Інструмент *створення баз даних* з раніше перекладених документів. Використовує інші інструменти для співставлення елементів перекладів та оригіналів.
- Інструмент *автоматичного розпізнавання термінів* для виділення з тексту та автоматичного пошуку у базі всіх термінів, що містяться у вихідному текстовому сегменті, над яким працює перекладач у даний час.
- База термінів.

- Інструмент *автоматичного співставлення блоків текстів*. Генерує за вхідним текстом структуру вихідного, або маючи вхідний та вихідний тексти, встановлює взаємозв'язки між їх елементами.

- Програма *керування термінологією* для підтримки бази термінів, отримання та оновлення специфічної термінології про предмет, клієнта та проект. Може включати інструмент вилучення термінології у якості додаткової або комплексної функції для надання допомоги в заповненні термінологічних баз та створення термінології для локалізації електронного контенту проекту, витягуючи одно- або двомовні списки потенційних термінів з конкретних електронних текстів оригіналу та/або цільових текстів.

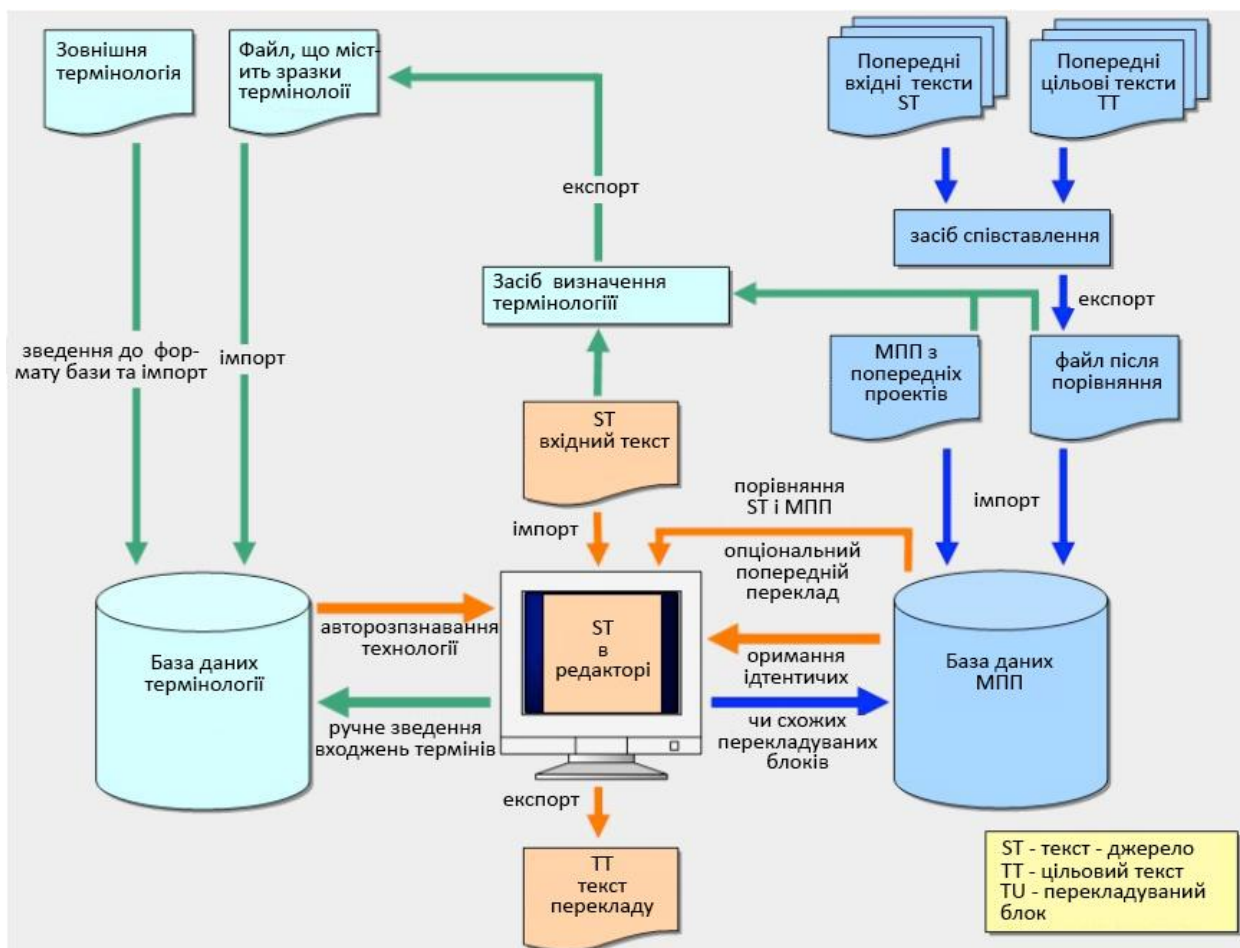


Рисунок 2.9 : Компоненти й процеси в МПП. Цитується за: Uwe Reinke *State of the Art in Translation Memory Technology*

- Багатомовний редактор для читання вихідних текстів і введення перекладу у всіх відповідних форматах файлів різних програм обробки

текстів, систем верстки і т.д., захищаючи теги макетів цих форматів від випадкового видалення або перезапису

- Інструмент *конкорданс* (узгодження), який дозволяє користувачам побачити всі випадки використання конкретної одиниці (окремі слова, групи слів, фрази і т. д.) у їхньому безпосередньому контексті у документах з архіву перекладів.

- Бібліотека з функцією статистики, що дозволяє приблизно побачити кількість сегментів тексту, які можна повторно використати для перекладу нового вихідного документа.

Додатково можуть бути реалізовані:

- Підтримка *машинного перекладу*: може бути реалізований інтерфейс до якої-небудь системи МП, або навіть система МП може бути інтегрована в систему МПП.

- Підтримка *управління проектом*, зокрема: обробка файлів і керування ними (специфікація всіх файлів мовою оригіналу, термінологічні бази даних проекту, допомога у визначенні структури папок), управління даними про клієнта і перекладача (адреса, контактні особи, сфера діяльності перекладача, обладнання, доступність і т. д.), управління робочим процесом (терміни, хід виконання проекту і т.д.).

На Рисунку 2.9 представлено огляд того, як взаємодіють основні компоненти стандартного середовища МПП.

### 2.2.2. ВИДІЛЕННЯ ТЕРМІНІВ

Виділення *за максимальною довжиною та граматичними ознаками*. Перший етап роботи алгоритму - виділення максимальних ланцюжків, що містять терміни. Ці ланцюжки визначаються через заборону: складається список слів і знаків, які не можуть входити до термінів. Це знаки пунктуації, стоп-слова, за потребою додаються інші частини мови. Рядки між цими роздільниками розглядаються як кандидати в терміни. Використовує інформацію про допустимість зв'язування певних елементів. Відповідно,

ланцюжки слів, які за зовнішніми ознаками повинні мати синтаксичні зв'язки, збираються разом.

*C-value*. Метод виділення багатослівних термінів, запропонований К.Франці та колегами, орієнтований на словосполучення, що не входять до складу інших, більш довгих. Частоти довгих термінів у тексті нижчі, ніж коротких, і тому було запропоновано метод *C-value* для компенсації цього ефекту. Значення термінологічності розраховується так:

$$C - value(a) = \begin{cases} \log_2 |a| * freq(a) \\ \log_2 |a| * freq(a) - \frac{1}{P(T_a)} * \sum_{b \in T_a} freq(b), \text{вкладений} \end{cases}$$

де  $a$  - кандидат у терміни,  $|a|$  - довжина словосполучення, вимірювана в кількості слів,  $freq(a)$ - частотність  $a$ ,  $T_a$  - множина словосполучень, які містять  $a$ ,  $P(T_a)$ - кількість словосполучень, що містять  $a$ .

Як видно, чим більша частота терміна-кандидата і його довжина, тим більша його вага. Але якщо цей кандидат входить до великої кількості інших словосполучень, то його вага зменшується.

*Віконний метод*. Розроблений Добровим Б.В. та колегами. Ідея методу - нарощувати словосполучення, якщо більш короткі часто зустрічаються у складі більш довгих. Однак, на відміну від інших методів, враховується не тільки частота контактних випадків (слова безпосередньо слідує одне за одним), а й спільна наявність у вікні (послідовності слів з тексту, вибраних підряд). На кожній ітерації для кожного елемента списку запам'ятовуються його безпосередні сусіди і сусіди в текстовому вікні. Створюються відповідні таблиці, обчислюється частотність утворення пар у вікні. Передбачається, що якщо пара елементів (на першому етапі - окремих слів) зустрічається як безпосередні сусіди більш ніж у половині випадків їх появи в одному і тому ж текстовому вікні, то ця пара являє собою термін або фрагмент терміна. Відбувається склейка пари в єдиний елемент, таблиці перераховуються так, наче цей елемент був відомий з самого початку (до початку обробки тексту), це дає можливість і далі нарощувати термін. Якщо не накладати обмежень на

частоту народження склеюваних елементів, то метод об'єднає унікальні (з частотою 1) ланцюжки допустимих слів (тобто повторить результат MaxLen).

Гібридні методи використовують комбінування вищезазначених підходів.

### 2.2.3. МЕТОДИ СПІВСТАВЛЕННЯ ТЕРМІНІВ ТА БЛОКІВ

Два основних способи співставлення термінів – точне та нечітке співставлення. Відповідно, співставлені фрагменти мають точний або нечіткий збіг. Найпростішим є точне співставлення, оскільки для нього треба, щоб вихідний текст та аналог в базі перекладів збігалися дослівно в на певній послідовності слів.

Для прискорення пошуку будь-яких кандидатів у збіги можна використовувати метод, який спирається на пошук дублів у текстах новин, описаний у розділі «Автоматичне реферування». Через малу довжину кожного блоку, що перекладається застосування загальних алгоритмів пошуку підрядка у рядку є недоцільним. Тобто алгоритми Бойера — Мура, Рабина — Карпа, Ахо — Корасика, та інші – не потрібні.

Основні спотворення, що мають місце при нечіткому збігу: вставка, видалення, заміна слова. Тому задачу нечіткого збігу можна трактувати як задачу пошуку редакторської відстані. Псевдокод наведено нижче.

```
int EditDistance(char s[1..m], char t[1..n])
  let d array of int [0..m, 0..n]

  for i in [0..m]
    d[i, 0] ← i // Відстань будь-якого першого рядка до порожнього другого
    рядка
  for j in [0..n]
    d[0, j] ← j // Відстань будь-якого другого рядка до порожнього першого
    рядка
  for j in [1..n]
    for i in [1..m]
      if s[i] = t[j] then
        d[i, j] ← d[i-1, j-1]
      else
        d[i, j] ← minimum of
          (
            d[i-1, j] + 1, // видалення
            d[i, j-1] + 1, // вставка
            d[i-1, j-1] + 1 // заміна
          )
```

`return d[m, n]`

Цей алгоритм називають алгоритмом Вагнера-Фішера.

Оскільки повний перебір не є раціональним, то як вже згадувалося, використовуються методи виділення запозичень, принаймні для відбору кандидатів та розбиття на субблоки, у яких уже використовується алгоритм Вагнера-Фішера.

#### 2.2.4. ПРОБЛЕМИ, ЩО ВИНΙΚАЮТЬ ПРИ ВИКОРИСТАННІ МПП

Іноді використання систем МПП може мати негативний вплив на якість перекладу. Одним з основних недоліків систем МПП є те, що вони зазвичай працюють на рівні речень. Таким чином, існує серйозна небезпека того, що перекладач приділяють занадто багато уваги ізольованим реченням, можливо, без урахування контексту, де вони зустрічаються.

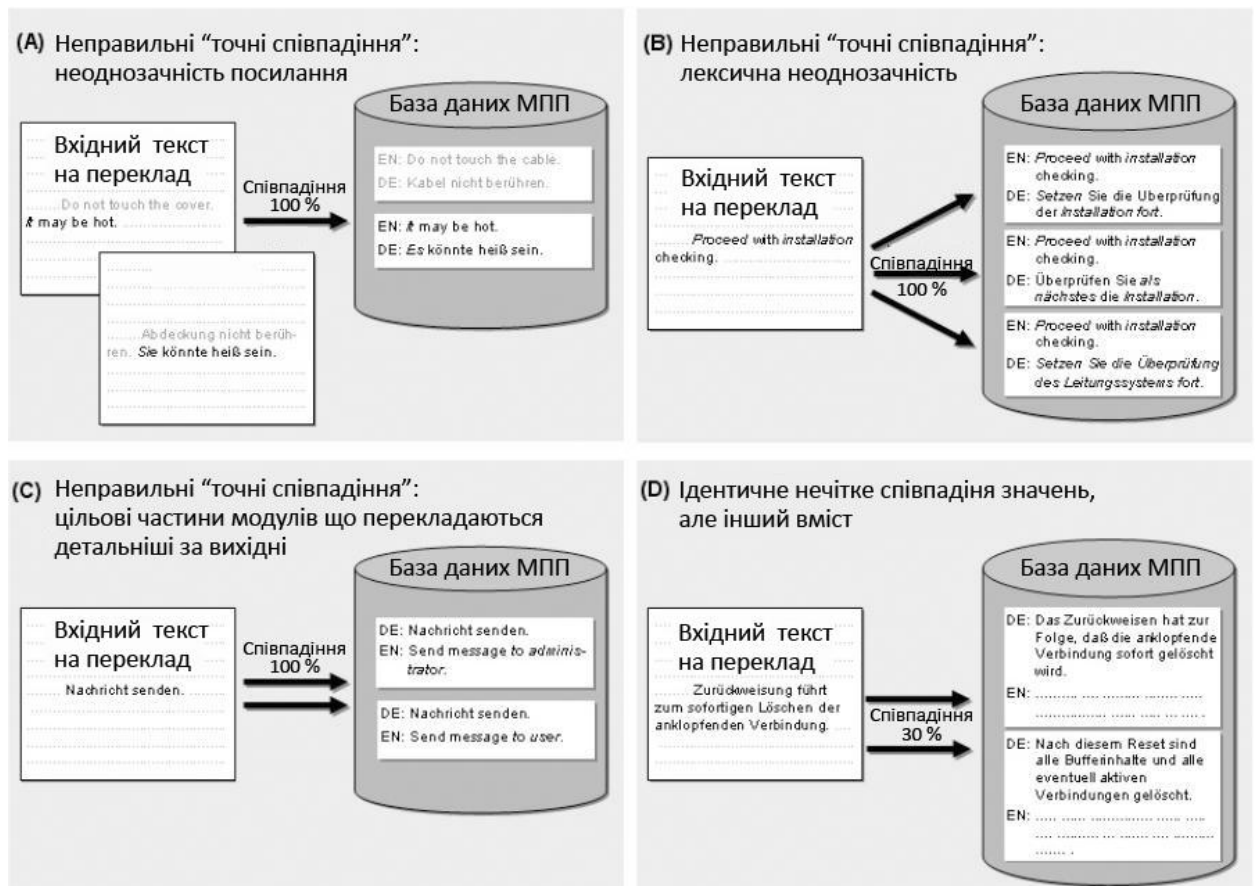


Рисунок 2.10 : Нечіткі відповідності та розпізнавання термінології. Цитується за: U. Reinke *State of the Art in Translation Memory Technology*

Приклади (А) і (В) на Рис.2.10 демонструють проблему відносно мовних вказівників і лексичної неоднозначності. У прикладі (А) займенник задає посилання на словосполучення попередньому реченні. У прикладі (В) такі терміни мають по декілька сенсів, таким чином задаючи мовну неоднозначність. Звідси випливає, що точний збіг у МПП може надати неправильний переклад. Відповідні алгоритми систем МПП засновані на дуже простих формальних критеріях подібності символічних рядків. Таким чином, уявлення людини-перекладача про ступінь подібності між сегментом для перекладу і сегментом з бази даних може істотно відрізнятись від ступеня подібності, розрахованого системою МПП. У прикладі (С) точний збіг насправді дає хибний сенс через те, що втрачено контекст. У перекладах з „нечітким збігом” шанси на помилку ще вищі. Приклад (D).

#### 2.2.5. МЕТОДИ ПОКРАЩЕННЯ СПІВСТАВЛЕННЯ ТЕРМІНІВ ТА БЛОКІВ

Хоча комерційні системи МПП були доступні протягом більше двох десятиліть, продуктивність їх пошуку не значно покращилася з точки зору точності та повноти.

Звичайно, алгоритми співставлення з часом змінювалися, але вони, як і раніше, покладаються на простий збіг символів або на маркери узгоджувальних процедур без урахування лінгвістичних аспектів, як то: морфологічного, синтаксичного або семантичного. Проте ці особливості можуть вказувати на схожість перекладу текстових блоків (ТБ).

Розміщені елементи: теги, графіки і динамічні поля, як правило, не містять перекладних елементів. Вони часто можуть бути скопійовані (розміщені) у цільовий текст без необхідності подальшої модифікації. Мітки є елементи розмітки в HTML і XML файлах; вкладені графічні та динамічні поля, як правило, трапляються в файлах Microsoft Word.

Локалізовані елементи: номери, дати, URL або адреси електронної пошти, у свою чергу, складаються з простого тексту у певному порядку, вони можуть бути ідентифіковані без „мовних знань”. Локалізація (або переклад)

цих елементів передбачає врахування правил і часто не впливає на інші частини в ТБ.

Аналіз розміщуваних та локалізованих елементів впливає на ефективність співставлення комерційних систем МПП. Розміщувані елементи іноді призводять до порівняно низького значення нечіткого співставлення, тому що деякі системи ставляться до них як до стандартного тексту при порівнянні довжини відрізків. Замість цього, було доцільніше використовувати фіксований штраф, коли цільовий текст та вихідний текст розрізняються тільки місцем переміщуваних елементів, в той час як інший текст є ідентичний. Проте, якщо є різниця в розміщенні, то це не тотожні ТБ, це треба враховувати.

Локалізовані елементи. Замість того, щоб розглядати їх як звичайний текст, їх слід розглядати в якості спеціальних елементів, які слідує певним шаблонам. Ці моделі можуть бути визначені за допомогою регулярних виразів. Для розрахунку співставлень можуть бути застосовані ті ж принципи, що вже запропоновані для переміщуваної елементів.

Підходи, які застосовують „лінгвістичні знання”. По-перше, це використання морфологічного та часткового синтаксичного аналізу. Недоліком є обмеження по кількості пар мов, для яких можна проводити такий аналіз. Також, не виключається використання семантики, звісно, за наявності доступу до семантичних баз знань. Можливо таким джерелом стане EuroWordNet. Інше джерело – Вікіпедія, завдяки наявності багатьох зв’язків, представлених посиланнями та наявністю різних мов.

Пізніші дослідження щодо підвищення точності та повноти в системах МПП в основному зосереджувалися на поліпшення повторного використання нечітких збігів, застосовуючи методи з статистичного МП. Залежності буде машинний перекладач, і вони можуть бути нетривіальними, як у Табл. 2.1. Математична основа перекладу виражається тоді такою формулою:

$$\Pr(t, a|s) = 1/Z_s \exp \sum_{m=1}^M \lambda_m \phi_m(s, t, a)$$

де  $s$  є джерелом для перекладу,  $t$  - кандидат у цільове речення,  $a$  - „вирівнювання” між ними,  $\phi_m$  - дійсне значення ознаки (частота такого перекладу пар, синтаксична якість, тощо),  $\lambda_m$  дійсний ваговий коефіцієнт ознаки,  $Z_s$  – нормуючий множник. Задача полягає у віднаходженні такої пари  $(a, t)$ , що максимізує  $P(t, a /s)$ .

### 2.3. ВИСНОВКИ

Описано системи автоматичного та автоматизованого перекладу, основні підходи та задачі. Деталізовано ряд алгоритмів та методів. Останнім часом ці два підходи гібридизують один одного, і надають все більш зручні механізми для автоматизації перекладу.

#### *Контрольні запитання*

1. Задача машинного перекладу. Основні підходи та моделі.
2. Імовірнісні моделі перекладу.
3. Переклад з проміжною мовою. Перетворення структур однієї мови у структури іншої мови.
4. Змішаний підхід на основі статистик та правил перетворення.
5. Системи типу „машинна пам’ять перекладача”.
6. Задача автоматичного співставлення фрагментів.

### 3. СИСТЕМИ ПРИРОДНОМОВНОГО ДІАЛОГУ

Існує багато різних архітектур діалогових систем (ДС). Набори компонентів і їхні функції різні. Головний компонент будь-якої ДС - це менеджер діалогу, який керує станом та стратегією діалогу. Обмежимося лише системами текстового діалогу, хоча діалогова система в цілому може використовувати текст, мову, графіку, тактильні, жести і інші режими для спілкування як на вході, так і на виході[10].

Діалогові системи (ДС) можна розділити на два типи:

- ДС із запрограмованим діалогом;
- ДС, що настроюються на різні класи задач на основі відповідних описів предметної області.

На відміну від ДС з запрограмованим діалогом, обмежених, як правило, певною предметною областю, ДС загального типу дає користувачу більш широкі можливості, оскільки зміна предметної області діалогу не вимагає регенерації (або перетрансляції) системи. Модифікація системи забезпечується шляхом введення нового сценарію, що визначає схему діалогового взаємодії (структуру та зміст діалогу), стан діалогу та запускаються в діалозі процедури (функції). При сценарному підході досягається максимально можлива незалежність діалогу від програмних засобів ДС, що дозволяє спростити і прискорити розробку ДС. Як правило, ДС створюють, користуючись типовою архітектурою систем автоматичної обробки мови. Саме такі ДС, орієновані на допомогу в обробці даних шляхом діалогу, утворюють групу питально-відповідальних систем.

Системи природно мовного діалогу орієнтовані на режим питання-відповідь (англ. QA - Question-answering system)– це інформаційні системи, здатні сприймати питання і відповідати на них природною мовою, іншими словами, це системи спілкування з природно-мовним інтерфейсом. Надалі будемо називати їх питально-відповідальними системами (ПВС).

Сучасні ПВС зазвичай включають особливий модуль - класифікатор питань, який визначає тип питання і, відповідно, тип очікуваної відповіді.

Після цього аналізу система поступово застосовує за наданими документами все більш складні і тонкі методи обробки природної мови, відкидаючи непотрібну інформацію. Найбільш грубий метод - пошук в документах - передбачає використання системи пошуку інформації для відбору частин тексту, що потенційно містять відповідь. Потім фільтр виділяє фрази, схожі на очікувану відповідь (наприклад, на питання «Хто...» фільтр поверне шматочки тексту, що містять імена людей). І, нарешті, модуль виділення відповідей знайде серед цих фраз правильну відповідь.

Функціонування більшості ПВС зводиться до здійснення наступних основних дій через інтерфейс:

- введення повідомлень (питань) користувача;
- виведення повідомлень (питань) системи;
- запуск модулів (функцій), які обирають відповідно з повідомленнями користувача та умовами вибору;
- вибір подальших шляхів продовження (або завершення) діалогу, який визначається сповіщенням користувача і задається умовою.

Діаграма діяльності такої системи зображена на Рис 3.1.

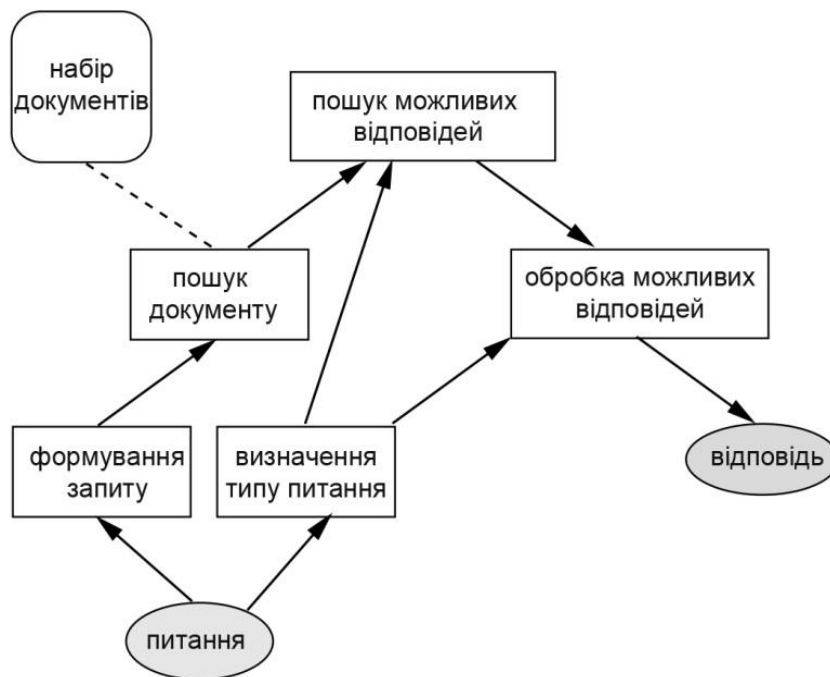


Рис. 3.1. Діаграма діяльності ПВС

Складність розробки та взаємодії продемонстровано у таблиці 3.1

Таблиця 3.1. Характеристики користувачів, питань і відповідей. Цитується за: Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A). [10]

Рівень 1	Рівень 2	Рівень 3	Рівень 4
<i>Пересічний користувач</i>	Типовий репортер	Допитливий молодий репортер	Аналітик-професіонал
<b>СКЛАДНІСТЬ ПИТАНЬ І ДІАПАЗОН ВІДПОВІДЕЙ</b>			
<b><u>ВІД:</u></b>		<b><u>ДО:</u></b>	
<p><b>Питання:</b>  <i>Прості факти</i></p>		<p><b>Питання:</b>  <i>Складні, використовує оціночні судження; Необхідне знання контексту; Широкий діапазон;</i></p>	
<p><b>Відповіді:</b>  <i>Прості, в межах одного документа</i></p>		<p><b>Відповіді:</b>  <i>Потребують пошуку у багатьох джерелах(мультимедійних системах/багатьма мовами Потребують поєднання інформації; узгодження суперечливих даних; велика кількість альтернативних відповідей і додаткового розтлумачення; Необхідні висновки</i></p>	

Характеристику питання і відповіді ПВС наведено на Рис. 3.2. При роботі з ПВС фахівець-аналітик діє у спосіб, показний на Рис. 3.3.

ПВС почали розробляти у 60-роки ХХ ст. як оболонки для експертних систем у конкретних галузях. Сучасні системи в більшості призначені для пошуку відповідей на питання в базах документів з використанням технологій обробки природної мови. ПВС можна умовно розділити на *загальні (open-domain) вузькоспеціалізовані (closed-domain)*. Загальні ПВС працюють з інформацією з усіх галузей знань, даючи можливість вести пошук і в суміжних областях. Вони характеризуються універсальністю. Найбільш відома система - Start (1993р.). Вузькоспеціалізовані ПВС працюють у конкретних областях – телефонних системах, настільних та інших комп'ютерах та ін. У них висуваються вимоги до точності відповідей і вони потребують складання онтологій для предметних областей.

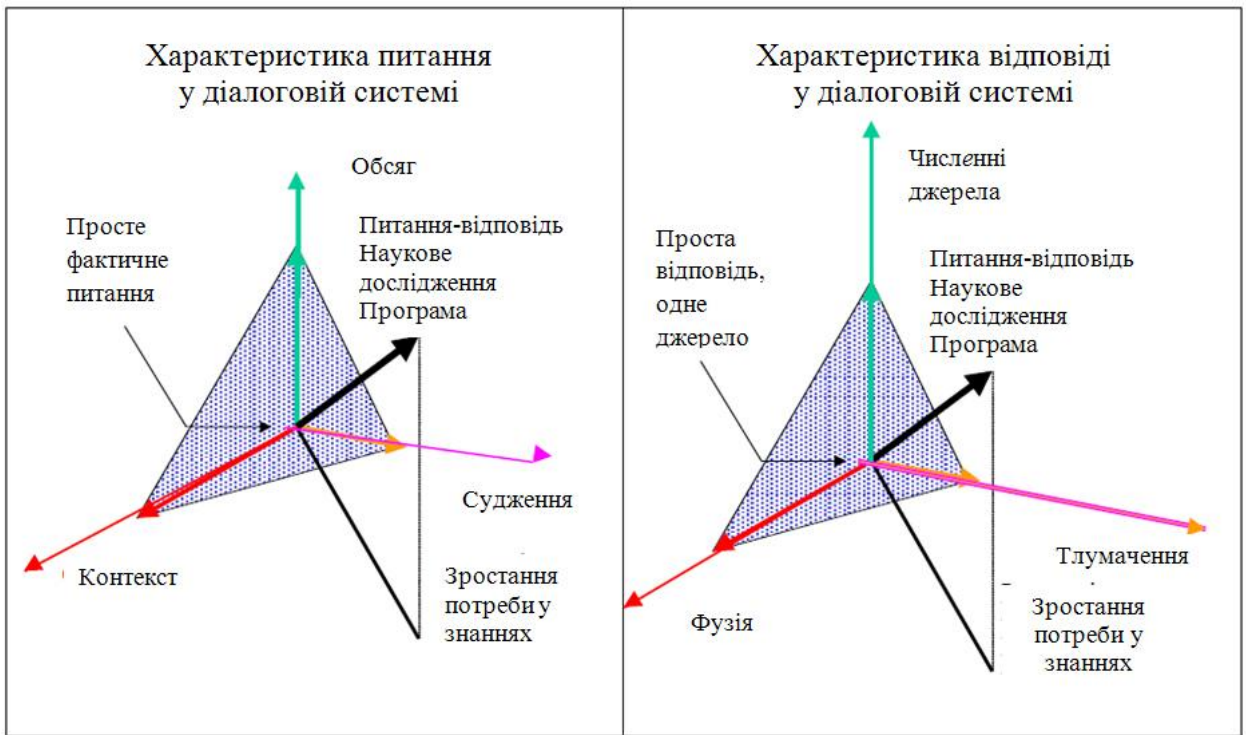


Рис. 3.2. Характеристика питання і відповіді ПВС. Цитується за: Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A)



Рисунок 3.3. Цитується за: Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A)

Роботу ПВС можна формалізувати як задачу перекладу, в якій цільова природна мова збігається з природною мовою джерела, але множини понять та виразних засобів – відрізняються. Це дозволяє без обмежень застосовувати методи описані в Розділі Машинний переклад.

### 3.1. МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Сучасні технології проектування ПВС ґрунтуються на використанні методології моделювання предметної області. Моделлю предметної області називається деяка система, яка імітує структуру або функціонування досліджуваної предметної області і відповідає основній вимозі - бути адекватною цій галузі. Без проведення моделювання предметної області велика ймовірність допущення великої кількості помилок у відповідях на питання. Внаслідок цього до моделей предметних областей висувуються такі вимоги:

- Формалізація, що забезпечує однозначний опис структури предметної області.
- Зрозумілість для замовників і розробників на основі застосування графічних засобів відображення моделі;
- Реалізованість, тобто наявність засобів фізичної реалізації моделі предметної області в ІС.
- Забезпечення оцінки ефективності реалізації моделі предметної області на основі обраного формалізму.

Для реалізації перерахованих вимог, як правило, будується система моделей, яка:

- складається з певних методів і обчислюваних показників;
- відображає структурний і оціночний аспекти функціонування предметної області.

Структурний аспект передбачає побудову:

- об'єктної структури, яка відбиває склад взаємодіючих в процесах матеріальних та інформаційних об'єктів предметної області;

- функціональної структури, яка відбиває взаємозв'язок функцій (дій) щодо перетворення об'єктів в процесах;
- структури управління, яка відбиває події та бізнес-правила, які впливають на виконання процесів;
- організаційної структури, яка відбиває взаємодію організаційних одиниць підприємства і персоналу в процесах;
- технічної структури, яка описує топологію розташування і способи комунікації комплексу технічних засобів.

Для відображення структурного аспекту моделей предметних областей в основному використовуються графічні методи, які повинні гарантувати представлення інформації про компоненти системи. Головна вимога до графічних методів документування - простота. Графічні методи повинні забезпечувати можливість структурної декомпозиції специфікацій системи з максимальним ступенем деталізації та погоджень описів на суміжних рівнях декомпозиції.

Першим стандартом програмної архітектури є стандарт IEEE 1471: ANSI/IEEE 1471-2000: Рекомендації по опису переважно програмних систем. Його було прийнято в 2007 році, під назвою ISO/IEC 42010:2007. Спосіб дії повинен враховувати доступ до інформаційних ресурсів. Продуктивність ПВС залежить від якості та обсягів текстової бази – локальної чи глобальної. Для локальної необхідні певні потужності для зберігання інформації, а великі сховища (такі як Інтернет) містять багато надлишкової інформації. Проте великі сховища мають і позитивні характеристики. Так як інформацію у великому сховищі представлено в різних формах, ПВС швидше знайде необхідну відповідь, проминувши стадію поглибленого аналізу тексту. У великих масивах правильна інформація частіше повторюється, тому помилки в документах краще відсіваються.

## 3.2. ФРЕЙМИ

Розглянемо конкретні структури моделей пошуку знань фреймами. Теорія фреймів - це теорія наукових концепцій, і хоча моделі подання знань фреймами ґрунтуються на цій теорії, вона повністю не охоплює їх. Останнім часом замість назви „фреймова” використовуються назви „об'єктно-орієнтована”, „структурованих об'єктів” і т.п. Ці назви, як далі буде показано, характерні для мови типу Smalltalk, що, власне, і відноситься до так званих об'єктно-орієнтованих мов, у яких є багато спільного із структурою програм і механізмами управління виконанням. Для порівняння розглянемо Таблицю 3.2:

Таблиця 3.2.

Фреймова термінологія	Об'єктно-орієнтована термінологія
Фрейд	Клас об'єктів
Слот	Властивості і атрибути об'єктів
Тригер	Методи Accessor і Mutator
Метод (певні мови, напр., Loom, KEE)	Метод

Іншими словами, змінні (структури даних) і процедури, які стосуються їх обробки, утворюють об'єкт, а схожі об'єкти мають ієрархічну структуру типу онтології. Управління виконанням проводиться за допомогою передачі повідомлень між об'єктами, що майже аналогічно механізму управління виводу за допомогою передачі повідомлень між фреймами. Однак на відміну від об'єктно-орієнтованої мови, яка має парадигми для універсального програмування, моделі подання знань фреймового типу мають парадигми для управління представленням знань (або пам'яттю) і висновками, тобто за основними ідеями і конкретною реалізацією вони мають багато відмінностей.

*Фреймова система* - це ієрархічна структура, вузлами якої є фрейми. Кожен фрейм складається з елементів, значення кожного елемента, розглянуто нижче.

Структуру вузла фреймової системи подано на Рис. 3.4.

Ім'я фрейма(1)	Показчик наслідування(3)	Показчик атрибутів слота (4)	Значення слота (5)	Демон (6)
----------------	--------------------------	------------------------------	--------------------	-----------

	(текст, чисельне значення, приєднана процедура, показчик тощо	Назва, значення, процедура(7), показчик тощо
Слот 1 (2)		
Слот 2		
Слот N		

Рис 3.4. Структура вузла фреймової системи

(1). Ім'я фрейма. Це ідентифікатор, який присвоюється фрейму, фрейм повинен мати ім'я, єдине в даній фреймовій системі (унікальне ім'я). Кожен фрейм, як показано на Рис. 3.2, складається з довільного числа слотів, причому декілька з них зазвичай визначаються самою системою для виконання специфічних функцій, а інші визначаються користувачем. До їх числа входять: слот „IS-A”, що показує на фрейм-батька даного фрейма, слот показчиків дочірніх фреймів, який є списком показчиків цих фреймів, слот для введення імені користувача, дати визначення, дати зміни, тексту коментаря та інші слоти. Кожен слот, у свою чергу, також представлений певною структурою даних.

(2). Ім'я слота. Це ідентифікатор, який присвоюється слоту; слот повинен мати унікальне ім'я у фреймі, до якого він належить. Зазвичай ім'я слота не несе ніякого смислового навантаження і є лише ідентифікатором даного слота, але в деяких випадках воно може мати специфічний сенс. До таких імен крім IS-A (відношення IS-A), DDESENDANTS (показчик прямого дочірнього фрейму), DEFINEDBY (користувач, який визначає фрейм), DEFINEDON (дата визначення фрейму), MODIFIEDON (дата модифікації фрейму), COMMENT (коментар) і т.п. відносяться імена, використовувані для представлення структурованих об'єктів, наприклад HASPART, RELATIONS та інші. Ці слоти називаються системними і використовуються при редагуванні бази знань і управлінні висновком.

(3) Показчики наслідування. Ці показчики стосуються тільки фреймових систем ієрархічного типу, заснованих на відносинах „абстрактне –

конкретне”, вони показують, яку інформацію про атрибути слотів у фреймі верхнього рівня успадковують слоти з такими ж іменами у фреймі нижнього рівня. Типові покажчики наслідування Unique (U: унікальний), Same (S: такий же), Range (R: встановлення границь), Override (O: перезаписати ) і т.п., U показує, що кожен фрейм може мати слоти з різними значеннями: S - що всі слоти повинні мати однакові значення, R - значення слотів фрейма нижнього рівня повинні знаходитися в межах, зазначених значеннями слотів фрейма верхнього рівня, O - при відсутності вказівки значення слота фрейма верхнього рівня стає значенням слота фрейма нижнього рівня, але у разі визначення нового значення слотів фреймів нижніх рівнів вказуються як значення цих слотів. Незважаючи на те що в більшості систем допускається кілька варіантів вказівки наслідування, існує чимало й таких, де допускається тільки один варіант.

(4). Вказівка типу даних. Вказується, що слот має чисельне значення, або служить покажчиком іншого фрейму (тобто показує ім'я фрейму). До типів даних відносяться FRAME (покажчик), INTEGER (цілий), REAL (дійсний), BOOL (булеві), LISP (приєднана процедура), TEXT (текст), LIST (список), TABLE (таблиця), EXPRESSION (вираз) та інші.

(5). Значення слота має збігатися з указаним типом даних цього слоту, крім того, повинна виконуватися умова наслідування.

(6). Демон. Демоном називається процедура, що автоматично запускається при виконанні деякої умови. Демони запускаються при зверненні до відповідного слоту. Наприклад, демон IF-REQUIRED запускається, якщо в момент звернення до слоту його значення не було встановлено, IF-ADDED запускається при підстановці в слот значення, IF-REMOVED запускається при стиранні значення слота. Крім того, демон є різновидом приєднаної процедури.

(7). Приєднана процедура. Як значення слота можна використовувати програму процедурного типу. Коли ми говоримо, що в моделях представлення знань фреймами об'єднуються процедурні та декларативні

знання, то вважаємо демони і приєднані процедури процедурними знаннями. Крім того, у мові представлення знань фреймами відсутній спеціальний механізм управління виводу, тому користувач повинен реалізувати цей механізм за допомогою приєднаної процедури. Однак дана мова має дуже високу універсальність, що дозволяє крім ієрархічного і мережевого представлення знань за допомогою фреймової системи ефективно писати будь-яку програму управління виводу за допомогою приєднаної процедури. Водночас це додаткове навантаження для користувача. Отже, мову представлення знань фреймами можна назвати мовою, орієнтованою на фахівців з штучного інтелекту, а також мовою, орієнтованою на складні прикладні проблеми. Відомі також приклади систем, що допускають застосування правил продукції як типів даних. Це обумовлено, з одного боку, тим, що більшість систем, орієнтованих на вирішення складних проблем, містить як складову продукційну систему, а з іншого боку - зниженням навантаження на користувача. Крім того, відомі приклади систем типу ZERO, що допускають застосування функцій Прологу як приєднаної процедури. Можливий розвиток діалогу з уточненням питання. Такий сценарій описується фреймом - прототипом, а конкретна реалізація діалогу фіксується у вигляді фрейму - екземпляра.

Можливість включення в структуру фрейму процедурної інформації дозволяє, у разі незадовільного уточнення, організувати цикл щодо заповнення вмісту слотів фрейма. Недоліком використання апарату фреймів при організації діалогу є відсутність хронологічної інформації у фреймах і неможливість безпосереднього урахування історії діалогу

### 3.3. ГЕНЕРАЦІЯ ВІДПОВІДІ

Два основних способи: на базі статистичного підходу та на базі фреймів.

Основою генерації відповіді на базі статистичного механізму є обчислення співставлення, що вказує на відповідні блоки, що слугуватимуть

ключами для пошуку елементів з бази відповідей, що треба включити в результат. Після того, генерується перестановка, також на базі статистичних закономірностей, а на базі перестановки – речення-відповідь. При потребі система також змінює свій стан по плану ведення діалогу.

Генерації відповіді на базі фреймів передбачає застосування приєднаних процедур, що виконують рекурсивний пошук в базі знань, при потребі викликаючи інші процедури. При цьому головна складність полягає у визначенні фрейму, з якого треба починати обхід. Для економії обчислювальних ресурсів застосовується класифікатора питань, це дозволяє зменшити кількість потенційних стартових фреймів. Також, оптимізація можлива за допомогою побудови таблиці відповідностей, аналогічно до механізму для статичної побудови відповідей.

Основний допоміжний інструмент - тезаурус або онтологічна база (наприклад, типу WordNet). З її допомогою можна успішно вирішувати питання синонімії, та заміни термінів на більш загальні або більш деталізовані, хоча і без гарантії якості. Це можна виконувати подібно до реалізацій з Розділу „Реферування”, або Розділу „Машинний переклад”.

### 3.4. ПЕРСПЕКТИВИ РОЗВИТКУ ПВС

ПВС розвиваються, і Спеціальний комітет, що складається із провідних дослідників з організацій, наукових установ та вишів, визначив структуру програм, здатних вирішувати проблеми обробки питань, у 2002 р. розробив підзадачі і об'єднав їх у більш складні способи, щоб уможливити задоволення потреб найвимогливіших запитувачів. Комітет визначив наступні вимоги, які повинні виконуватися для ПВС:

Відповіді повинні надаватися вчасно в реальному режимі часу, бути вчасними навіть за умови одночасної постановки питань тисячами запитувачів. Нова інформація повинна ставати доступною негайно після її отримання, навіть коли йдеться про найновіші факти чи останні події.

Достовірність. Достовірність ПВС є надзвичайно важливою, адже

неправильні відповіді гірші, ніж відсутність відповідей. Дослідження ПВС мають бути зосереджені на шляхах оцінки правильності наданих відповідей та розробляти методи для виявлення випадків, коли наявні дані не містять відповіді. Необхідно виявляти суперечності в джерелах даних і на постійній основі опрацьовувати суперечливу інформацію. Для забезпечення достовірності система повинна включати в себе знання всього світу і механізми, які імітують доходження висновків особистістю у здоровому глузді.

Зручність. Часто знання в ПВС необхідно пристосовувати до конкретних потреб користувача. Необхідно розробляти спеціальні онтології і предметно-орієнтовані процедурні знання. Дуже важливим є швидке прототипування з предметно-орієнтованих знань та їх включення до відкритих онтологій вузькоспеціалізованих. Часто використовуються гетерогенні джерела даних - інформація може бути доступною в текстах, у базах даних, у відеокліпах або інших засобах масової інформації. ПВС повинна бути в змозі здобувати відповіді незалежно від формату джерела даних, і повинна надати відповідь у будь-якому форматі, який є бажаним для користувача. Навіть більше того, ПВС повинна дозволяти користувачеві описувати контекст запитання, і повинна давати всілякі пояснення та забезпечити його способами візуалізації і навігації.

Повнота. Бажано давати повні відповіді на питання користувача. Іноді в інформаційних джерелах відповідь роззосереджена, або міститься у декількох документах. Потрібно зливати інформацію в єдину відповідь. Генерація повної відповіді повинна покладатися на імплікатури, оскільки люди висловлюються у економний спосіб, і також враховувати розрідженість даних. Крім того, загальні та предметно-орієнтовані знання повинні поєднуватися і досить складно вмотивовуватись. ПВС повинна включати в себе можливості, які дозволять все обміркувати і зуміти скористатися високопродуктивними базами знань. Іноді необхідно проводити аналогії з іншими питаннями, і вони мають бути або в заданому користувачем

контексті, або в контексті профілю користувача. Автоматичне отримання профілю користувача являє собою метод, що дозволяє співробітництво з ПВС та отримання інформації зворотного зв'язку про ДС.

Актуальність. Відповідь на питання користувача повинні бути актуальними в певному контексті. Часто може виникати необхідність у інтерактивній ПВС, де послідовність питань допомагає прояснити потрібну інформацію. Складність питання і відповідну таксономію неможливо вивчити, не маючи точок дотику з користувачем і ПВС та без урахування наступного дослідження „по холодних слідах”. Оцінка ПВС повинна проводитись з огляду на задоволення потреб користувача: саме люди повинні приймати остаточне рішення про доцільність і достовірність інформації, отриманої від ДС та легкість її використання.

### 3.5. ВИСНОВКИ

Описано задачі та структуру питально-відповідальних систем. Розглянуто взаємозв'язки з іншими задачами обробки природної мови. Описано вимоги до перспективних ПВС.

#### *Контрольні запитання*

1. Довідкові системи, питально-віповідальні системи.
2. Фреймові моделі предметних областей, бази фактів.
3. Використання фреймів для аналізу недосить специфічних питань.
4. Використання онтологій для роботи з полісемією у запитаннях.
5. Побудова оптимальної відповіді на основі запитання та бази відповідей (застосування механізмів перекладу).

## СПИСОК ЛІТЕРАТУРИ

1. Анисимов А.В. Компьютерная лингвистика для всех: Мифы. Алгоритмы. Язык Киев: Наук. думка, 1988.- 223 с.
2. Белоногов Г.Г. Компьютерная лингвистика и перспективные информационные технологии М.: Русский мир. 2004, - 248 с. Ел. ресурс. Режим доступа: <http://www.twirpx.com/file/134393/>
3. Волошин В.Г. Комп'ютерна лінгвістика: Навчальний посібник. – Суми: Університетська книга, 2004. –382 с.
4. Марчук Ю.Н. Компьютерная лингвистика М.: Изд-во Восток-Запад, 2007 г., - 317 с Ел. ресурс. Режим доступа: <http://www.twirpx.com/file/398578/>
5. Партико З.В. Прикладна і комп'ютерна лінгвістика, Львів, «Афіша», 2008, - 221 с.
6. Сайт «Автоматическая обработка текстов» Ел. ресурс. Режим доступа: <http://aot.ru>
7. Сайт проекту Link Grammar Ел. ресурс. Режим доступа <http://www.link.cs.cmu.edu/link/>
8. Сайт проекту WordNet Ел. ресурс. Режим доступа: <http://wordnet.princeton.edu/>
9. Clark A. The Handbook of Computational Linguistics and Natural Language Processing /Clark A., Fox C., Lappin S.// Blackwell Publishing, 2010, - 775р. Ел. ресурс. Режим доступа: [stp.lingfil.uu.se/~santinim/sais/ClarkEtAl2010\\_HandbookNLP.pdf](http://stp.lingfil.uu.se/~santinim/sais/ClarkEtAl2010_HandbookNLP.pdf)
10. Burger J Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A) /John Burger et al.// Ел. ресурс. Режим доступа^ [http://www.inf.ed.ac.uk/teaching/courses/tts/papers/qa\\_roadmap.pdf](http://www.inf.ed.ac.uk/teaching/courses/tts/papers/qa_roadmap.pdf)
11. Jurafsky D. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition, / Jurafsky D., Martin J.// 2009 Ел. ресурс. Режим доступа: <http://www.cse.iitk.ac.in/users/mohit/Speech-and-Language-Processing.pdf>
12. Mitkov R. The Oxford handbook of computational linguistics /Oxford University Press, 2003 –786 p. Ел. ресурс. Режим доступа: <http://www.google.com.ua/books?hl=uk&lr=&id=y16AnaKtVAkC&oi=fnd&pg=PP2&dq=5.%09The+Oxford+handbook+of+computational+linguistics>
13. Nirenburg S. Ontological semantics / Nirenburg S., Raskin V. //MIT Press, 2004, – 420 p. Ел. ресурс. Режим доступа: [http://books.google.com.ua/books/about/Ontological\\_semantics.html?id=OPek3LpMIgC&redir\\_esc=y](http://books.google.com.ua/books/about/Ontological_semantics.html?id=OPek3LpMIgC&redir_esc=y)
14. Uwe Reinke State of the Art in Translation Memory Technology Ел. ресурс. Режим доступа <http://www.t-c3.org/index.php/t-c3/article/view/25>

## ЗМІСТ

	Передмова	3
1	Автоматичне реферування	6
1.1	Постановка задачі	6
1.2	Допоміжні засоби	9
1.3	Індексація	10
1.3.1	Змістовна близькість	11
1.3.2	Лексичні ланцюжки	11
1.3.3	Спосіб об'єднання лексичних ланцюжків	12
1.3.4	Алгоритм побудови лексичних ланцюжків	14
1.3.5	Використання результатів індексації	16
1.4	Визначення можливих запозичень у текстах	16
1.4.1	Модель структури тексту	18
1.4.2	Запозичення в подробицях	19
1.5	Кластеризація	22
1.5.1	Міри якості кластерів та кластерного розбиття	22
1.5.2	Кластерний аналіз на зважених графах	24
1.6.	Виділення термінологічних одиниць	24
1.7	Реферування	25
1.7.1	Визначення важливості елементів тексту	26
1.7.2	Передобробка	28
1.7.3	Алгоритм планування	29
1.7.4.	Семантико-синтаксичний алгоритм стиску	30
1.7.5	Семантичний алгоритм стиску	32
1.7.6	Алгоритм стиску вибором	34
1.7.7	Загальний алгоритм реферування	35
1.7.8	Покращення результатів реферування	36
1.7.9	Оцінювання якості рефератів	38
1.8	Висновки	39
2.	Машинний переклад	40
2.1	Автоматичний переклад	40
2.1.1	Мікрокосмос	45
2.1.2	Статистичний перекладач	48
2.1.3	Гібридний переклад	53
2.2.	Автоматизований переклад	54
2.2.1	Компоненти МПП	56
2.2.2	Виділення термінів	58
2.2.3	Методи співставлення термінів та блоків	60
2.2.4	Проблеми, що виникають при використанні МПП	61
2.2.5	Методи покращення співставлення термінів та блоків	62
2.3	Висновки	62
3.	Системи природномовного діалогу	65
3.1	Моделювання предметної області	69
3.2	Фрейми	71

3.3	Генерація відповіді	74
3.4	Перспективи розвитку ПВС	75
3.5	Висновки	77
	Список літератури	78
	Зміст	79