СИСТЕМА АВТОМАТИЧНОГО РЕФЕРУВАННЯ

Володимир Юрійович Тарануха

Міжнародний Науково-Навчальний Центр Інформаційних Технологій та Систем
Київ, Україна
taranukha@ukr.net

Анотація

Описується система автоматичного реферування текстів складених природною мовою. Розглянуто підзадачі, описано алгоритми та виконано оцінку ефективності алгоритмів, на основі яких побудована система.

1. Вступ

Необхідність розробки такої системи зумовлена збільшенням кількості та обсягу текстів, доступних у електронному вигляді. Це не лише електронні бібліотеки, але і міграція в Інтернет різних ЗМІ, як часткова, коли газета має електронну версію на додачу до паперової, так і повна, коли інформаційна агенція має лише новини в електронному вигляді, розміщені на відповідних сайтах. Переважна більшість з новин представлена у вигляді звичайних текстів складених природною мовою.

Таким чином, в умовах, коли необхідно проаналізувати багато інформації, виникає потреба в засобах, що дозволять прочитавши менше оглянути більший об'єм інформації. Реферати вказаного користувачем об'єму та опис тематик текстів дозволяють спрости орієнтування у цьому морі доступної інформації.

2. Постановка задачі

Необхідно реалізувати систему, що орієнтована на роботу з українською та російською мовами. При цьому слід враховувати, що алгоритми, орієновані на англійську мову, для якої є досить багато систем не завжди добре справляються з аналізом текстів слов'янськими мовами, як то українською та російською.

Означення. Рефератом є текст вказаного об'єму, що менше за текст оригіналу і містить найбільш важливі для користувача думки оригіналу[1].

Таким чином з формулювання задачі вже видно першу проблему. 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. В той же час інформацію про інтереси користувача можна отримати досить швидко. Найпростіший варіант – перелік ключових слів[2]. Проте це не завжди ефективно оскільки часто користувач ще не знає що в конкретному документі або тексті його може зацікавити, принаймні до того, як почне його читати. Особливо це проявляється, коли на аналіз подаються великі обсяги новин.

Таким чином першою підзадачею виступає необхідність визначення тематичного наповнення тексту, що в свою чергу зводиться до задачі індексації.

Індексація 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. Після чого враховуючи відсоток стиску та відомості про відносну важливість різних тем представлених у тексті можна проводити вибір даних з тексту.

Після вибору даних необхідно забезпечити відповідну якість тексту результату. Якщо реферат утворений за допомогою генерації нових речень, то таке реферування називається абстрактивним[3]. Якщо текст реферату утворений вибором та спрощенням речень оригіналу, то таке реферування називається екстрактивним. В цьому випадку необхідно забезпечити зв'язність отриманого реферату. Основна причина порушення зв'язності полягає в тому, що речення змістовно наступне за вибраним реченням є в тексті оригіналу, проте відсутнє у рефераті. При цьому, в реченні лишилися елементи, що відповідають за зв'язування тексту в єдине ціле. Реферат готовий коли вибрані речення зв'язані у цілісний текст.

3. Додаткові підзадачі

В роботі система використовує такі допоміжні засоби:

- підсистему морфолексичного аналізу;
- підсистему часткового синтаксичного аналізу;
- підсистему заміни займенників.

Задача розбиття тексту на слова та визначення морфолексичних характеристик слів є критичною з ряду причин. По-перше, без визначення канонічних форм слів, зведення разом різних словоформ одного слова неможливе. Це суттєво погіршує роботу навіть простих статистичних методів аналізу текстів. По-друге, в частина алгоритмів вимагає синтаксичного аналізу і, оскільки немає гарантії, що всі слова є в словнику системи, виникає потреба в спеціальних алгоритмах евристичного морфолексичного аналізу. Робота відповідних алгоритмів описана в [4].

На основі отриманих морфолексичних даних проводиться примітивний синтаксичний аналіз. Зв'язуються прикметники(дієприкметники) з відповідними іменниками та іменники з відповідними дієсловами. Навіть такі обмежені синтаксичні дані, разом з інформацією про те, яким ролям у реченні відповідають які відмінки стають у нагоді в задачі реферування.

Для заміни займенників на повнозначні слова, на які вони посилаються, використовуються перш за все морфолексичні ознаки. Вимагається, щоб слово, що замінювало займенник зібгалося з ним у якомога більшому числі ознак, а саме: роді, числі, відмінку. Вживається евристика, що з двох кандидатів більш правдоподібним є той, що знаходиться ближче до замінюваного слова.

Лише у випадку коли інших даних недостатньо вживається простий семантичний аналіз, а саме: серед альтернатив вибачається слово, що має зміст найближчий за семантичною мірою близькості до слів контексту. Для

визначення семантичної близькості використовується локалізована до російської та української мов семантична база WordNet та алгоритм пошуку найкоротших шляхів[5].

4. Індексація

Є два способи побудови тематичного представлення - з фіксованими темами та з динамічними темами.

Індексація з фіксованими темами спирається на фіксовані тематичні словники. На сьогодні в системі представлені такі тематичні словники: біологія, хімія, комп'ютерна тематика, фінанси, геологія та географія, право, лінгвістика, математика, атомна енергетика, фізика. До цих словників входять як деталізовані поняття, так загальні поняття.

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

Означення. Лексичним ланцюжком називається послідовність слів, що розташовані в тексті поблизу, та мають якусь спільну тематику.

Для побудови таких лексичних ланцюжків вживається WordNet.

Лексичні ланцюжки можуть бути обчислені шляхом групування послідовних наборів семантично зв'язаних слів. Тотожність слів, синоніми, і гіперніми з гіпонімами – ознаки, що дозволяють групувати слова в один ланцюжок.

Означення. Гіпернім - поняття, що є узагальнюючим для даного у WordNet.

Означення. Гіпонім – поняття, що є уточненням даного у WordNet.

Необхідно зауважити, що у WordNet представлені не слова, а концепти – поняття, і кожен концепт має свій комплект слів, що його позначають – синсет.

Умови групування

1. Два входження повнозначного слова ідентичні, і використовуються в тім же самому концепті. (*Великий **корабель** на рейді. Цей **корабель** - вітрильник.*)
2. Два входження повнозначних слів використовуються в одному і тому ж самому смислі, 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 -**Вольво** .*)
4. Змісти двох входжень повнозначних слів - елементи одного рівня в гіпернім/гіпонім дереві. (*Той **аеробус** летить швидко. Проте, мій **винищувач** швидше.*)

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. Тому час від часу виникають труднощі у визначенні, до якого ланцюжка повинно бути приєднане входження слова. Наприклад, входження іменника може відповідати декільком різним змістам слова, і система повинна визначити, яке саме входження має місце. Наприклад: коса як інструмент і коса як зачіска.

Крім того, навіть якщо зміст слова може бути визначений, може трапитись, що можливо згрупувати слово у кілька різних лексичних ланцюжків, тому що це слово може бути зв'язане зі словами в різних ланцюжках.

Розглянемо алгоритм побудови лексичних ланцюжків. Для ефективного обчислення лексичних ланцюжків створюється структура, що неявно зберігає кожну інтерпретацію. А потім з цього неявного представлення обчислюється оптимальна конфігурація. Обробка документа починається зі створення великого масиву мета-ланцюжків, розмір якого – число пар зміст-слово з тексту для якого будуються ланцюжки у WordNet.

Алгоритм побудови лексичних ланцюжків

1. Для кожного повнозначного слова:{
2. Для кожного концепту слова: {
3. Обновити значення кожного мета-ланцюжка – якщо концепт входить, то збільшити відповідний лічильник.}}
4. Для кожного повнозначного слова:{
5. Для множини мета-ланцюжків що мають зміст відповідний слову:{
6. Визначити мета-ланцюжок, до якого зміст відповідний слову належить найбільше, шляхом звертання до пошуку в ширину[5].
7. Обнулити лічильник для змістів у інших ланцюжках.}}
8. Для кожного мета-ланцюжка видалити елементи, що мають нульовий лічильник.
9. Для кожного мета-ланцюжка зібрати список слів, що відповідають сенсам, що лишилися с ненульовими значеннями лічильника.

Таким чином отримані ланцюжки представляють собою динамічно сформований комплект тем документу. При цьому різні за змістом, але однакові за формою слова опиняться у відповідних ланцюжках, побудованих відповідно до тематичного наповнення тексту.

Для переставлення користувачу з 9 найбільш вагомих ланцюжків вибирається 9 слів, що мають концепти з найбільшою оцінкою. За результатами індексації користувач визначає важливість тієї чи іншої тематики.

5. Визначення важливості елементів тексту

Важливість елементів тексту у реферуванні визначаться відповідно до того, наскільки вони цікавлять користувача та наскільки вони важливі для представлення змісту тексту. При цьому дуже зручно, що вже є готові і зарані обчислені лексичні ланцюжки або принаймні фіксовані тематичні словники, які можна вжити в якості ланцюжків. Це дозволяє уникнути ряду проблем характерних для систем, що використовують частотний принцип оцінювання важливості елементу. За частотним принципом, чим частотніше слово/поняття у тексті тим воно важливіше. Проте, може трапитися, що група слів/понять разом важливі для тексту, проте, кожне з них зустрічається не досить часто, щоб частотний критерій розпізнав їх як важливі. Таким чином важливість кожного повнозначного слова в розробленій системі визначається як функція, що залежить від одної або більше складових з переліку:

відносна вага лексичного ланцюжка або тематики, куди входить зміст, що позначає вказане слово,

відносна важливість, що її визначає користувач,

відносна вага повнозначного слова серед інших слів тексту.

Відносна вага лексичного ланцюжка визначається так:

$$w_i = \frac{v_i}{\sum_j v_j} \qquad (1)$$

де v_i - сумарна кількість концептів, що входять в *i*-й ланцюжок.

Відносна вага повнозначного слова визначається двома способами. Перший, це модифікація поширеної метрики *tf*idf* [3], у вигляді специфічної *tf*isf*.

$$\text{tf*isf} = \frac{tf(t,S)}{isf(t,S)} \quad (2)$$

де S – множина речень, $tf(t,S)$ – частота слова на всій множині речень, і відповідає за те, наскільки слово важливе для тексту в цілому.

$$isf(t,S) = \log(\frac{|S|}{|s \in S, t \in s|}) \quad (3)$$

Тобто, $isf(t,S)$ – ознака того, наскільки часто в реченнях оригіналу вживається слово. Така структура функції походить від міркування, що якщо слово та зв'язане з ним поняття вживаються в оригіналі часто, то з високою імовірністю потрапить у реферат. В той же час у рефераті немає потреби весь час нагадувати про це поняття, а краще надати можливість ознайомитися і з іншими поняттями та їх зв'язками. Другий спосіб визначається у алгоритмі семантичного стиску.

6. Алгоритми стиску

6.1. Передобробка

Після визначення важливості кожного повнозначного слова окремо необхідно визначити межі тематичних областей в тексті. Розмітка тематичних областей дозволяє в подальшому точніше вибирати елементи, що будуть аналізуватися, та ігнорувати тематики, що не цікавлять користувача.

Алгоритм побудови тематичних областей

1. Для кожного речення:{
2. Для кожного повнозначного слова: {
3. Для кожного концепту слова: {
4. Встановити належність до певного ланцюжка.
5. Поставити маркер «Тема почалася» відповідно до номера ланцюжка }}}
6. Для кожного речення:{
7. Для кожного повнозначного слова: {
8. Для кожного концепту слова: {
9. Якщо немає концептів з того самого ланцюжка в сусідньому реченні:{
10. Поставити маркер «Тема скінчилася» відповідно до номера ланцюжка}}}}
11. Для кожного повнозначного слова: {
12. Для кожного концепту слова: {
13. Якщо маркери «Тема почалася» і «Тема скінчилася» для однієї теми стоять одночасно: {
14. Зняти позначки відповідних ланцюжків. }}}

6.2. Семантико-синтаксичний алгоритм стиску

В межах областей між двома маркерами «Тема почалася» «Тема скінчилася», що належать одній темі застосовувався семантико-синтаксичний алгоритм стиску. Він працює переважно з онтологією, використовуючи зв'язок „бути". В процесі узагальнення цей алгоритм пробігає по онтології від концептів нижчого рівня до концептів вищого рівня в пошуках концепту, що є водночас допустимими та досить абстрактними, для можливості узагальнення. Одною з властивостей цього алгоритму є те, що він не використовує дані про важливість окремих слів, а лише інформацію про важливість тематики. Для нього є обов'язкова синтаксична передобробка, оскільки він використовує синтаксичні дані.

Означення. Предикатною структурою будемо називати трійку „суб'єкт"-„предикат"-„об'єкт" де предикат відповідає дієслову у реченні, суб'єкт – слову що позначає те, що виконує дію, а „об'єкт" – те слово, що позначає те, над чим дія виконується.

Семантико-синтаксичний алгоритм стиску

1. Для кожного речення до кінця тематичної області:{
2. Скласти предикатну структуру відповідно до підмета і присудка.
3. Позначити її як базову.
4. Для кожного речення від даного до кінця тематичної області: {
5. Скласти предикатну структуру відповідно до підмета і присудка.
6. Порівняти предикатну структуру з базовою.
7. Якщо для підметів або присудків виконуються «Умови групування» з розділу Індексація на відстань 2 по WordNet: {
8. З двох речень будується одне, більш поширене і використовуючи більш загальні концепти. }}}

Як показали експерименти, цей алгоритм має ряд недоліків.

Чутливість до синтаксичних неоднорідностей та помилок синтаксичного аналізу. На тестових текстах відсоток повністю синтаксично правильно перебудованих речень склав 8-10% від всього числа перебудованих алгоритмом речень. Хоча відсоток зрозумілих речень складає понад 90%.

Нездатність забезпечити стабільне зв'язування, оскільки він не реагує на зв'язки між концептами 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. На тестових статтях відсоток речень де узагальнення відбулось правильно склав 7-9%. Проте, відсоток зрозумілих речень перевищує 90%, що не дозволяє стверджувати, що цей алгоритм зовсім непридатний.

Низький відсоток речень, що проходять відбір на такий стиск. Залежно від тесту – від 3 до 7%.

Іноді можливе порушення ходу викладення думки, якщо порядок подачі речень важливий для розуміння тексту. Таким чином, цей алгоритм придатний лише як допоміжний, разом з якимось іншим алгоритмом.

6.3. Семантичний алгоритм стиску

В межах областей між двома маркерами «Тема почалася» «Тема скінчилася», що належать одній темі застосовувався семантичний алгоритм стиску.

Умови зупинки для пошуку в ширину в онтології.

1. Як тільки зустрічається концепт (вершина в графі онтології), що є забороненою алгоритм припиняє обчислювати ваги для цієї вершини та всіх вершин, для яких вона є нащадком в ієрархії WordNet. До заборонених концептів належать загальні поняття, якщо вони не представлені явно в лексичному ланцюжку.
2. Якщо вершина знаходиться за межами бажаної тематики, алгоритм припиняє обчислювати ваги для цієї вершини та всіх вершин, для яких вона є нащадком в ієрархії WordNet;
3. Якщо досягнута довжина шляху, рівна 5.

Додаткове джерело даних: база синтаксичних шаблонів.

Семантичний алгоритм стиску

1. Для кожного речення до кінця тематичної області:{
2. Для кожного повнозначного слова:{
3. Для кожного концепту слова:{
4. Додати концепту у список концептів, якщо його там немає. }}}
5. Створити порожній список.
6. Виконати 5 ітерацій циклу: {
7. Для кожного концепту із списку концептів: {
8. Якщо не виконуються умови зупинки:{
9. Визначити кількість шляхів, що проходять через вказаний концептів напрямку гіперніма.
10. Занести більш загальний концепту новий список.
11. Встановити йому вагу рівну сумі всіх шляхів від нижніх концептів через нього. }}
12. Доповнити новий список значеннями зі старого.
13. Замінити список концептів на новий.}
14. Створити порожній список.
15. Для кожного речення до кінця тематичної області:{
16. Для кожного повнозначного слова:{
17. Для кожного концепту слова:{
18. Додати в список пари зв'язаних концептів відповідно до предикатної структури речення. }}}
19. Вибрати N найбільш вагомих концептів, що відповідають дієсловам.
20. Вибрати по шаблону на кожен вибраний концепт.
21. Поки шаблони не заповнені, або список зв'язків не вичерпано, робити ітерації:{
22. Вибрати найбільш зв'язаний концепт, що відповідають іменнику.
23. Спробувати вставити його в один з вільних місць в якому-небудь шаблоні, відповідно до типу зв'язку.}

Переваги алгоритму:

- Можна визначити ті концепти з WordNet, що не представленні в тексті явно, проте сильно пов'язані з його змістом. Це дозволяє, наприклад, узагальнити ряд: „стіл, стілець, ліжко" до „меблі" але не до „об'єкт".
- Відмова від використання речень з тексту і використання шаблонів дозволяє отримати синтаксично коректні речення.

Недоліки алгоритму:

- Низький відсоток речень, що проходять відбір на такий стиск. Залежно від тесту – від 10 до 23%.
- Порушення ходу викладення думки.
- Чутливість до синтаксичних неоднорідностей та помилок, хоча і менша ніж у семанитко-синтаксичного алгоритму стиску. Це пояснюється тим, що в даному алгоритмі вживаються лише часто вживані зв'язки, що в свою чергу дозволяє відсіяти менш достовірні зв'язки.

Таким чином, можна побудувати короткі набори характеристичних речень, що подають найбільш важливі поняття тексту, навіть якщо вони явно не представлені, в тексті, але маються на увазі.

6.4. Алгоритм стиску вибором

Це виявився найбільш дієвій алгоритм, оскільки він не чутливий до можливих синтаксичних неоднорідностей.

Алгоритм стиску вибором

1. Для кожного речення до кінця тематичної області:{
2. Для кожного повнозначного слова:{
3. Для кожного концепту слова:{
4. Встановити вагу відповідно до ваги ланцюжка, ваги концепту в тексті та важливості визначеної користувачем.}}}
5. Створити порожній список
6. Для кожного речення до кінця тематичної області:{
7. Занести у список оцінку речення складену як суму оцінок слів}
8. Відсортувати список.
9. Вибрати відсоток оцінок, що відповідає відсотку стиску. 10. Взяти останній елемент з вибраних в якості межі.
10. Для кожного речення до кінця тематичної області:{
11. Якщо оцінка речення менше межі – поставити маркер «Не потрібне»}
12. Відібрати всі речення без маркеру «Не потрібне».

Переваги алгоритму:

- Стабільність гарантованого стиску.
- Оскільки в реферат відбираються речення, що вважаються коректними, то в рефераті теж коректні речення.
- Зберігається порядок подачі матеріалу, що важливо, коли іде опис подій, що розгортаються в часі.

Недоліки алгоритму:

- Цей алгоритм гарантовано будує незв'язний текст який важко читається. Цю проблему розв'язує алгоритм покращення реферату, описаний в [1].
- Алгоритм тяжіє до відбору довгих речень, частина елементів у яких – іноді має низьку змістовну цінність.

Цю проблему частково розв'язує семантико-синтаксичний алгоритм стиску, бо речення відібрані алгоритмом стиску вибором частіше а ніж речення тексту-оригіналу містять елементи, що дозволяють виконати відповідний стиск.

7. Висновки

В статті описується ряд алгоритмів, що разом формують основу для створення системи автоматичного реферування.

Кожен з них окремо не забезпечує достатньої якості реферування, проте за умови використання комплекту алгоритмів можна отримати реферати необхідного об'єму та якості.

8. Література

[1] Тарануха В.Ю. Використання генетичних алгоритмів для забезпечення якості автоматично згенерованих рефератів.//Вісник Київського Національного Університету імені Тараса Шевченко. Серія: Фіз.-Мат. Науки, Вип.2, 2011 – С. 155 - 158.

[2.] Павел Браславский, Иван Колычев Автоматическое реферирование веб-документов с учетом запроса // Интернет-математика 2005. М.: Яндекс, 2005. - С. 485-501.

[3] Inderjeet Mani Natural language processing. Automatic summarization // The MITRE Corporation and Georgetown University. – Amsterdam, Philadelphia, USA. – John Benjamin Publishing Company. – 2001. – Vol 3. – 286 p.

[4] Анисимов А.В. Романик А.Н. Тарануха В.Ю. Эвристические алгоритмы для определения канонических форм и граматических характеристик слов. // Кибернетика и Системный Анализ. – Т40 – 2004. - № 2. –С. 3-15

[5] А.В. Анісімов, О.О. Марченко, А.О. Никонеко Алгоритмічна модель асоціативно-семантичного контекстного аналізу текстів природною мовою // Проблеми програмування – 2008. № 2-3. – С.379-384