Автоматичне Формування Словників Перекладу

Наталя Валенда
кафедра Програмної інженерії
Харківський національний університет
радіоелектроніки
Харків, Україна
natalia.valenda@nure.ua

Нікіта Павленко
магістр кафедри Програмної інженерії
Харківський національний університет
радіоелектроніки
Харків, Україна
nikitapavlenko@ukr.net

Automatic Creation of tTranslation Dictionaries

Natalia Valenda
Department of Software Engineering
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
natalia.valenda@nure.ua

Nikita Pavlenko
Department of Software Engineering
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
nikitapavlenko@ukr.net

***Анотація*—Розглядається метод створення двомовних словників перекладу на основі паралельних корпусів. Методи розробки базуються на стеку технологій Cache Intersystems. Для розробки були використані iKnow, Cache Object Script, Cache Studio. У результаті роботи здійснено програмну реалізацію системи обробки російсько-українського паралельного корпусу для створення двомовних словників з машинним інтерфейсом.**

***Abstract*—Method of creating bilingual translation dictionaries based on parallel corpora is considered. Methods of development are based on stack of Cache Intersystems technologies. iKnow, Cache Object Script, Cache Studio were used for development. As a result of the work there was created software implementation of a system that process of Russian-Ukrainian parallel corpora to create bilingual dictionaries with machine interface.**

Ключові слова—словник; паралельний корпус; Cache; iKnow

Keywords—dictionary; parallel corpora; Cache; iKnow

I. Вступ

Одним з напрямків розвитку систем обробки текстової інформації є розробка систем автоматичного перекладу. Традиційне створення системи автоматичного перекладу є досить трудомістким завданням, що вимагає залучення фахівців кількох областей. Значний обсяг роботи становить створення словників, які лежать в основі роботи системи. Для реалізації цього етапу необхідно залучення лінгвістів. Автоматизація етапу створення словників спростила б процес створення перекладача.

Якість систем автоматичного перекладу багато в чому залежить від змісту та наповнення словника. Якщо переклад здійснюється для пари близьких мов, то основними причинами помилок є - недостатнє наповнення словника і хибний підбір значень слів. Програма не враховує елементарних значень слів і не пропонує їх в якості варіанту при перекладі.

Для перекладу в спеціалізованих областях можлива побудова словника на основі паралельних текстів з даної тематики, який буде давати не самий частотний переклад слова, а орієнтований на дану предметну галузь.

Вибір правильного значення багатозначних слів є, мабуть, найбільш складною проблемою в галузі перекладу. Для вибору правильного варіанту перекладу можна використовувати контекст, який відповідає слову в даному значенні. При створенні словника на основі паралельних текстів не тільки вибирається переклад слова, але і фіксується контекст, в якому слово приймає таке значення.

Створення словника для двомовного перекладу без залучення лінгвістів, автоматичними методами на основі обробки паралельних корпусів, є актуальним завданням, вирішення якого спростить процес створення якісних систем перекладу.

Іншим напрямком застосування даної роботи може бути покращення систем багатомовного інформаційного пошуку. У цій галузі загальним підходом є переклад пошукового запиту на всі цільові мови. Це здійснюється за допомогою двомовних словників для вхідної та

цільової мови. Для ефективного двомовного пошуку існування відповідних словників має ключове значення.

II. Паралельні корпуси

Автоматична побудова словників перекладу має дві передумови:

- наявність паралельних корпусів текстів російської та української мов;
- наявність технології автоматичної обробки текстової інформації iKnow, що працює з вибраними мовами.

Застосування iKnow дозволяє робити розбір тексту, виділяючи в ньому відносини і концепти, використовуючи для цього модель мови. Для кожного концепту підраховується статистика – скільки він зустрічається в тексті. З концептів і відносин будуються ланцюжки, що відображають взаємозв'язок між поняттями.

Робота з побудови словників може здійснюватися в кілька етапів.

- обробка текстів паралельних корпусів за допомогою технології iKnow, виділення концептів і відносин;
- збереження паралельних розборів у БД Cache [1];
- аналіз розібраних паралельних корпусів за реченнями і зіставлення перекладного еквівалента слів. Підрахунок статистики. Якщо слово багатозначне, то вказуються всі варіанти перекладу і їх вірогідність;
- формування словника двонаправленого перекладу.

Отримані словники можуть бути використані як основа системи автоматичного двомовного перекладу.

Як джерело паралельних текстів можна використовувати літературні джерела, доступні українською та російською мовами, сайти перекладені на обидві мови, вже готові паралельні корпусу. На даному етапі в якості джерела даних для обробки обраний вирівняний на рівні речень українсько-російський паралельний текстовий корпус з веб-публікацій, створений в інформаційному центрі ElVisti. Обсяг корпусу - понад 2,6 млн. пар унікальних речень. Для наукових цілей відкритий вільний доступ до фрагменту в 100 тисяч пар унікальних речень [2].

Формат представлення даних наближений до XML:

<Item>

<Rus> предложение </ rus>

<Ukr> речення </ ukr>

</ Item>

III. Обробка даних Cache

Дані зчитуються з файлу, який містить паралельний корпус. Дані представлені в XML форматі тому вони потребують парсингу. Процес парсингу відбувається на боці Cache, файл з паралельним корпусом обробляється і створюються об'єкти Cache. Далі об'єкти Cache зберігаються в базу даних де перебувають до їх подальшої обробки модулем iKnow. Після збереження у базі даних об'єктів Cache ми маємо SQL таблицю, яка містить записи паралельних речень. Далі SQL записи виймаються із бази даних, речення на російській мові оброблюються модулем iKnow за допомогою російської моделі мови і зберігаються у російському домені, українські речення оброблюються відповідно української моделлю мови і потрапляють в українській домен iKnow (рис. 1).

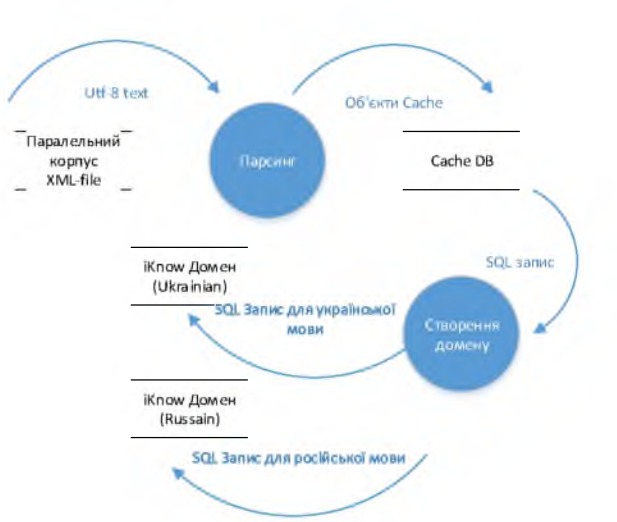

Рис. 1 – Обробка даних Cache.

Після парсингу паралельний корпус попадає до бази даних. Тому звісно потрібно розглянути схему зберігання даних. Зазвичай це робиться за допомогою моделі сутність-зв'язок. Для зберігання паралельного корпусу достатньо однієї сутності. Ця сутність - це елемент паралельного корпусу, який має ідентифікатор, речення російською мовою, а також речення українською мовою.

Мова розробки CacheObjectScript дозволяє використати об'єктно-орієнтований підхід до створення програми. Стрижнем програми є клас MainProgram, за допомогою нього можливо проініціалізувати систему, використавши паралельний корпус, а також можливо перекласти слово з української на російську та навпаки. Клас CorporaParser використовується для обробки xml-файлу і зберігання елементів корпусу в базу даних використовуючи класс CorporaElement, який у свою чергу використовує CacheLibrary. Після зберігання даних клас SQLDomainLoader використовується для створення українського та російського доменів. Він використовує iKnow API для цього. КлассTranslator використовується для перекладу слова з вхідної мови на цільову [3].

IV. Технологія iKnow

Технологія InterSystems iKnow дозволяє індексувати текстові файли і неструктуровані дані інших типів для виділення елементів представлення знань, концепцій та зв'язків між ними. На відміну від інших технологій семантичного аналізу та пошуку, технологія iKnow автоматично показує найбільш значущі елементи в даних без необхідності будь-якого втручання користувача,

навіть без завдання ключових термінів в якості пошукового критерію [3].

Система обробки паралельних корпусів базується на використанні модулю iKnow бази даних Cache. Двигун семантичного аналізу iKnow використовується для аналізу неструктурованих даних, які записуються у вигляді тексту. Надаючи можливість для швидкого доступу і аналізу даних цього типу, iKnow не вимагає додаткових попередніх знань про зміст цих даних, або навіть знань про те, якою мовою дані написані, якщо iKnow підтримує цю мову.

Неструктуровані дані складаються з декількох текстів, часто з дуже великої кількісті текстів. Текст зазвичай розділяється за знаками пунктуації на речення. Текстовим джерелом може бути файл, або результаті SQL запросу, або навіть веб-джерело таке як RSS.

iKnow забезпечує доступ до неструктурованих даних, розділяючи текст на пов'язані елементи та вираховуючи певний індекс для цих елементів. Система ділить текст на речення, а потім ділить кожне речення на послідовності понять і відносин. Система виконує цю операцію шляхом визначення мови тексту (наприклад, української), а потім застосовуючи відповідну модель мови в iKnow.

Відношення – це слово або група слів, які об'єднують два поняття, визначаючи відношення між ними. Система містить компактну мовну модель, яка здатна визначити відносини в реченні.

Поняття це слово або група слів, які пов'язані відношенням. При визначенні відношень iKnow може визначити пов'язані з ним поняття. Таким чином, двигун iKnow може визначити поняття семантично без "зрозуміння" їх змісту.

Зазвичай дієслова - це відношення, а іменники це частіше за все поняття. Проте, лінгвістична модель відносин і понять значно більш складна, ніж різниця між дієсловами і іменниками.

Таким чином, iKnow ділить речення на поняття (П) і відношення (R). Мовна модель використовує відносно невеликий і фіксований словник слів відносин і набір правил для виявлення відносин у контексті. Все, що не ідентифіковано як відношення вважається поняттям. iKnow також ідентифікує нерелевантні слова, такі як "і", і відкидає їх від подальшого аналізу.

Відносини і поняття разом називаються елементом. Проте відношення майже ніколи не має сенсу без пов'язаного з ним поняття. З цієї причини система аналізу iKnow підкреслює зв'язок понять і послідовностей, які містять поняття, з відношенням.

Оскільки iKnow аналізує текст за допомогою невеликої і стабільної моделі мови, яка зосереджена на виявленні відносин, iKnow може досить швидко індексувати тексти на будь-яку тему. Системі не потрібно використовувати словник для ідентифікації понять.

Після того, як iKnow визначив поняття і відношення у кожному реченні в тексті, або у багатьох текстах, ця інформація може бути використана для виконання наступних видів операцій: SmartIndexing, SmartMatching.

Інтелектуальне індексування (SmartIndexing) дає можливість аналізувати і перетворювати неструктуровані текстові дані в зрозумілий набір концепцій і зв'язків між ними без необхідності використання попередньо заданих словників, таксономій і онтологій. Ця функція дозволяє визначити у великих обсягах неструктурованого тексту без введення ключового слова, які тексти схожі, які мають зв'язки між собою, яка інформація є репрезентативною і є значущою.

Інтелектуальне індексування використовується для декількох різних мов. Воно також може виявляти концепції (повторювані патерни) в неструктурованих даних, які не є традиційним текстом. Зразок індексованих речень українською та російською мовами наведено на рис. 2.

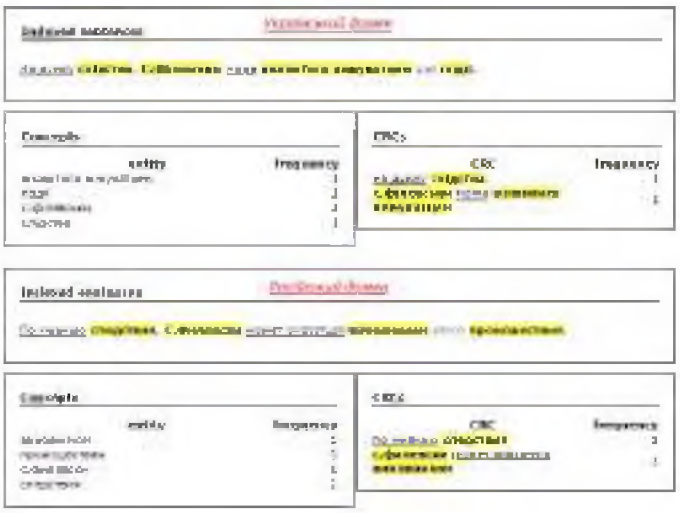

Рис. 2 – Зразок проіндексованих речень

Функція SmartMatching: надає можливість зв'язати об'єкти в вхідних текстах з зовнішніми елементами, такими як списки або словники. Ці списки можуть містити слова, фрази або речення для пошуку повного або часткового співвідношення.

При практичній роботі з системою iKnow послідовність дій поділяється на два етапи: загрузку даних у домен та запроси до цих даних. Загрузку даних можна організувати з різних джерел (файл, веб, sql-запрос). Загрузка даних виконується двома класами – це Lister та Loader. Можна також втручатись у цей процес додаючи конвертори та процесори тексту, для більш тонкого налаштування загрузки даних. Після того, як дані були завантажені в домен, доступ до них можна отримати завдяки виконанню запитів на мові CacheObjectScript або через рівень SOAP веб-сервісів.

Модель роботи iKnow представлена на рис. 3.

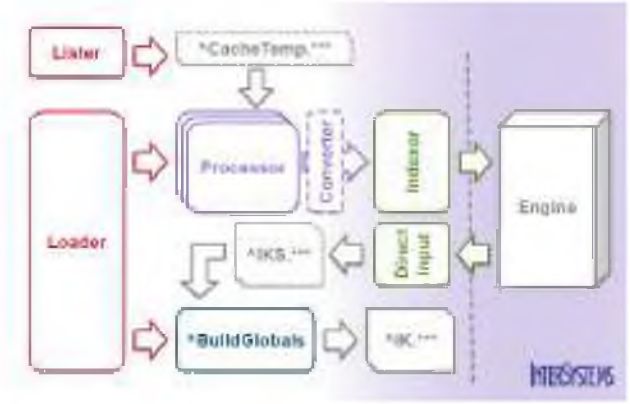

Рис. 3 – Загальна модель роботи iKnow.

V. Алгоритм обробки паралельних текстів

На основі вищенаведених технологій було розроблено програмну систему обробки паралельних корпусів, яка складається з веб-сервісу та клієнту. У систему був завантажений російсько-український паралельний корпус, який був створений для наукових цілей у інформаційному центрі ElVista.

Елементи корпусу вилучались із XML файлу, який потребував попередньої обробки. Зразок паралельного корпусу, який було завантажено, наведено на рис. 4.

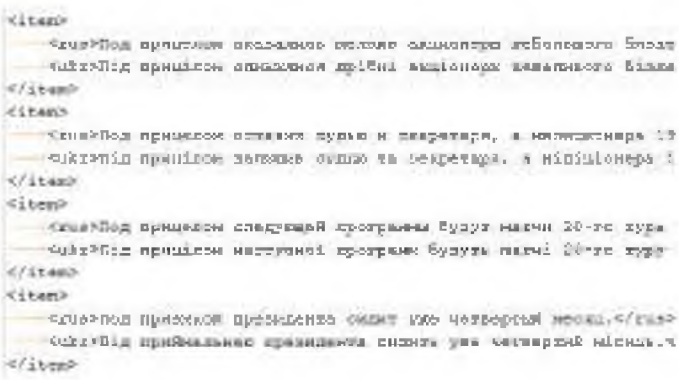

Рис. 4 – Зразок паралельного корпусу

Для реалізації перекладу був розроблений алгоритм, який використовує IKnowQueries API. Він дозволяє знайти переклад для сутності на вхідній мові на цільову мову. Сутність у термінах Cache – це самостійна частина речення, це може бути концепт, відношення або незначуща частина. Алгоритм побудований з використанням трьох класів у бібліотеці CacheIKnow, а саме EntityAPI, SourceAPI, SentenceAPI. Ці класи надають можливості для роботи з доменом.

Після аналізу результатів індексації тексту системою iKnow, було запропоновано наступний алгоритм. Беремо задану сутність у мові оригіналу та знаходимо усі речення на вхідній мові, у яких зустрічається ця сутність. Після знаходження цих речень, беремо кожне речення окремо. Для кожного окремого речення шукаємо речення на мові перекладу. Знаходимо позицію сутності у реченні на вхідній мові. Використовуємо знайдену позицію дістаючи сутність за найденою позицією у реченні на мові перекладу. Зберігаємо пару сутність-переклад до результату. Обробляємо всі речення таким чином. Після обробки аналізуємо отримані пари, відсікаємо пари з частотою нижчою ніж певний коефіцієнт. Далі збережені пари можна буде використовувати через веб-інтерфейс для отримання перекладу. Приклад веб-інтерфейсу наведено на рис. 5.

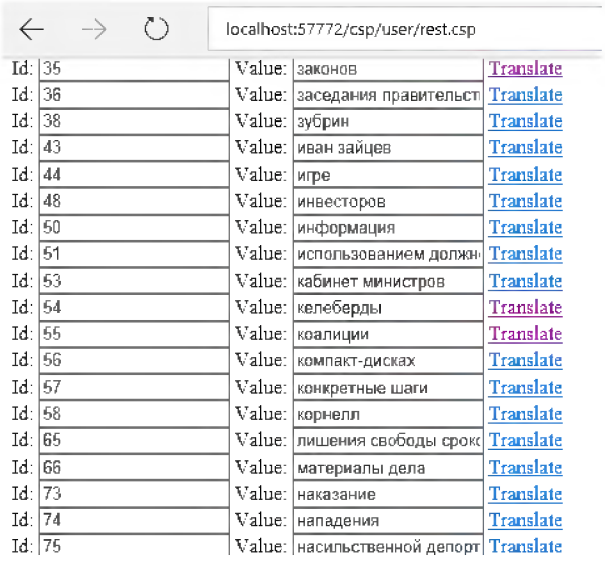

Рис. 5 - Інтерфейс словника.

У даній роботі було розглянуто техніку створення словників з машинним інтерфейсом на основі паралельних корпусів. Запропонований автоматичний метод дозволив сформувати двомовний словник перекладу. Такий підхід гарантує, що найбільш значущі переклади будуть включені у вихідний словник, якщо вибрані паралельні корпуси досить повні. Крім того, можливі кандидати для перекладу можуть бути сортовані за ймовірністю їх перекладу, тим самим гарантуючи, що швидше за все перекладні еквіваленти йдуть першими. Всі відповідні приклади речень легко доступні, що дає змогу подивитись на контекст перекладу, вибираючи відповідний переклад у разі декількох значень.

Література References

[1] В.І. Гайдаржи, І.Ю Михайлова. Об'єктно-реляційна СУБД Cache , Освіта України, 2015, 310с.

[2] [Infostream,2018]Украинско-русский параллельный текстовый корпус, 2018, Режим доступа http://ling.infostream.ua/

[3] [InterSystems, 2018] InterSystems Product Documentation, 2018, Access mode http://docs.intersystems.com