УДК 004.934

Ю. Ю. Нестюк
А. К. Латанська

МАРКОВСЬКИЙ ПІДХІД ДО ОПИСУ МОВНОЇ ІНФОРМАЦІЇ

Анотація

Синтезовано правила граматики голосового інтерфейсу шляхом формування матриці суміжності фраз природної мови. Проведено тестування поданих алгоритмів в програмному забезпеченні, розробленому в середовищі Qt Creater 5.2.

Ключові слова: аналіз тесту, алгоритм приведення контекстно-вільної граматики.

Abstract

The rules of grammar of the voice interface are synthesized by forming a matrix of adjacency of natural language phrases. Testing of the submitted algorithms in the software developed in the Qt Creater 5.2 environment is carried out.

Keywords: Text Mining, algorithm for reducing context-free grammar.

На сьогодні в особистих ПК, локальних і глобальних мережах накопичено величезну кількість інформації і її обсяг стрімко збільшується. Пошук в гігантських масивах текстових даних і аналіз об'ємних текстів є малоефективним, тому стають затребуваними технології, які спроможні обробляти неструктуровані або слабко структуровані тексти [1].

Зазвичай, для ведення документації більшість організацій користуються природною мовою. За даними аналітиків понад 80% інформації, яка зберігається в документах представлена в текстовій формі. Text Mining – технологія з автоматичного видобутку знань з великих обсягів текстового матеріалу, що заснована на поєднанні лінгвістичних, семантичних, статистичних методик та машинного навчання. Новітня технологія Text Mining призначена для виявлення в сирих або частково оброблених даних раніше невідомих нетривіальних практично корисних і доступних до інтерпретації знань, необхідних для прийняття рішень у різних сферах людської діяльності. Text Mining часто називають текстовим Data Mining. ТМ додає до технології DM додатковий етап – переведення неструктурованих текстових масивів в структуровані. Після чого дані можуть оброблятися за допомогою стандартних методів DM. Якщо DM дозволяє видобувати нові знання (приховані закономірності, факти, невідомі взаємозв'язки тощо) з великих обсягів структурованої інформації (збереженої в сховищах даних), то ТМ призначений знаходити нові знання в неструктурованих текстових масивах [2].

Розглянемо приклад контекстно-вільної граматики з алфавітом термінальних символів {a, b} і початковим символом S:

$S \rightarrow aS \mid A \mid a$

$B \rightarrow b$

Неважко помітити, що для даної граматики допоміжні (нетермінальні) символи A і B не можуть зустрічатися в сентенціальних формах висновків термінальних ланцюжків з початкового символу S. Іншими словами, вони не беруть участі в породженні ланцюжків мови, тобто є в цьому сенсі марними [3].

Будь-яку контекстно-вільну граматику можна привести до форми, яка не містить даремних символів. Нехай G = (VT, VN, P, S) - контекстно-вільна граматика (КВ-граматика). Символ $x \in (VT \cup VN)$ називається недосяжним в граматиці G, якщо він не з'являється ні в одній сентенціальній формі цієї граматики. Символ $A \in VN$ називається безплідним в граматиці G, якщо безліч виведених з цього символу термінальних ланцюжків порожні.

КВ-граматика називається наведеною, якщо в ній немає недосяжних і непотрібних символів. Алгоритм приведення контекстно-вільної граматики до форми, що не містить непотрібних символів, складається з двох кроків. Кожен крок у свою чергу реалізується окремим алгоритмом. Ці алгоритми використовують граф граматики і будуть розглянуті нижче.

Алгоритм приведення КВ-граматики:

1. Знайти і видалити всі безплідні символи і правила, що їх містять.

2. Знайти і видалити всі недосяжні символи і правила, що їх містять [4].

Після першого кроку даного алгоритму для граматики G = (VT, VN, P, S) отримуємо еквівалентну граматику G1 = (VT, VN1, P1, S), таку, що для будь-якого A ∈ VN1 справедлива нерівність: {A ⇒ * ω | ω ∈ VT*} ≠ ∅.

На другому кроці з G1 отримаємо еквівалентну граматику G2 = (VT2, VN2, P2, S), що володіє властивостями наведеної граматики :

– для будь-якого символу x ∈ (VT2 ∪ VN2) існують α1, α2 ∈ (VT2 ∪ VN2) * такі, що S ⇒ * α1xα2;

– або для будь-якого A ∈ VN2 справедливо нерівність {A ⇒* ω | ω ∅ VT2*} ≠ ∅ або VN2 = {S} і {S ⇒* ω | ω ∈ VT2*} = ∅.

Граф контекстно-вільної граматики. Для знаходження безплідних і недосяжних символів корисний граф КВ-граматики:

– кожному символу з VT ∪ VN відповідає єдина вершина, позначена цим символом;

– якщо в P є правило з порожньою правою частиною ε, то граф має вершину, позначену ε;

– вершина, позначена символом X, з'єднується з вершиною Y дугою (стрілкою), якщо в граматиці є правило X → αY β, де α, β ∈ (VT ∪ VN) *;

– X з'єднується з вершиною ε, якщо в граматиці є правило X → ε.

Для прикладу візьмемо граматику G0:

G0 = ({a, b, c, d, e}, {S, A, B, C, D}, P0, S)

P0: S → aAB | C
D → cDc | d
C → aCD
A → aA | a | ε
B → b.

Граф для граматики G0 матиме вигляд, що зображений на рис. 1.

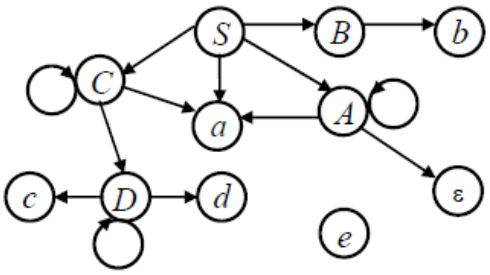

Рис. 1 – Граф для граматики G0

Алгоритм видалення непотрібних символів. Для заданої граматики G побудувати граф і виконати наступні кроки:

1. Відзначити (виділити) вершини графа, помічені термінальними символами, а також вершину ε, якщо така є.

2. Якщо в P є правило A → α, де α складається з символів уже зазначених вершин, а вершина A ще не відзначена, то відзначити цю вершину. Повторювати крок 2 поки можливо.

3. З граматики видалити символи невідмічених вершин, а також правила, що містять хоча б один символ невідміченої вершини. Для прикладу розглянемо граматику G0.

G0 = ({a, b, c, d, e}, {S, A, B, C, D}, P0, S)

P0: S → aAB | C
D → cDc | d
C → aCD
A → aA | a | ε
B → b.

Граф G0 після кроків 1, 2 алгоритму (відмічені вершини виділені подвійним кружком) матиме вигляд, як зображено на рис. 2.

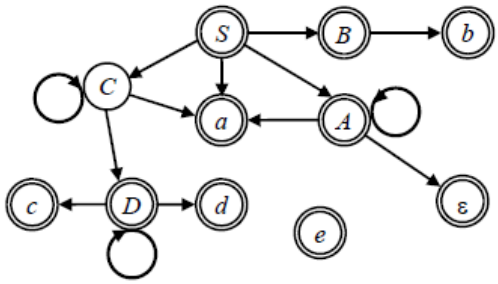

Рис. 2 – Граф для граматики G0 після виконання кроків 1,2 алгоритму видалення безплідних символів

Невідміченою на графі виявилася вершина C. Викреслюємо в граматиці G0 символ C і правила, що містять його: ({a, b, c, d, e}, {S, A, B, C, D}, P0, S)

P0: S → aAB | C
D → cDc | d
C → aCD
A → aA | a | ε
B → b.

Отримуємо еквівалентну граматику G1, що не містить безплідних символів: G1 = ({a, b, c, d}, {S, A, B, D}, P1, S)

P1: S → aAB
D → cDc | d
A → aA | a | ε
B → b.

Алгоритм видалення недосяжних символів. Побудувати граф граматики. Використовуючи граф, виконати наступні кроки:

1. Відзначити вершини графа, в які є шлях з вершини S.
2. Видалити з граматики символи невідмічених вершин і правила, що містять хоча б один такий символ.

Для прикладу розглянемо граматику G1, отриману в п. 1.3. Граф граматики матиме вигляд, як зображено на рис. 3.

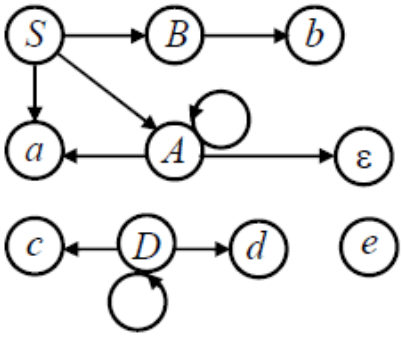

Рис. 3 – Граф граматики G1

Знаходимо недосяжні символи і викреслюємо їх з алфавіту нетерміналів і правил граматики G1.

G1 = ({a, b , c, d, e }, {S, A, B , D }, P1, S)

P1: S → aAB
D → cDc | d
A → aA | a | ε
B → b.

Граф граматики G1 (відмічені на кроці 1 вершини виділені подвійним кружком) матиме наступний вигляд, як показано на рис. 4.

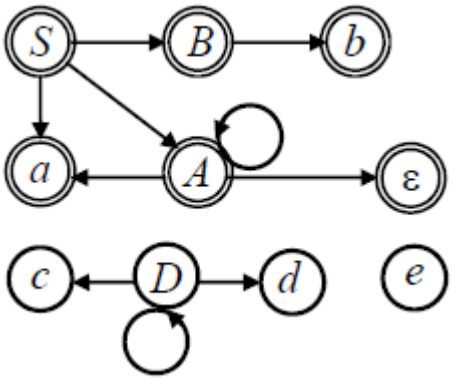

Рис. 4 – Граф граматики G1 після викреслення недосяжних символів

Невідмічені в графі символи є недосяжними і підлягають видаленню. Після видалення недосяжних символів з G1, отримаємо еквівалентну граматику G2 = ({a, b}, {S, A, B}, P2, S)

P2: S → aAB
A → aA | a | ε
B → b.

Граматика G2 – приведена, вона отримана з G0 послідовним застосуванням кроків 1 і 2 алгоритму приведення граматики

L (G0) = L (G1) = L (G2) = {anb | n ≥ 1} [3].

Порядок виконання кроків 1 і 2 важливий, його не можна змінювати.

Синтез правила граматики за матрицею суміжності відбувається в декілька кроків.

Крок 1. Формуємо матрицю суміжності на основі фраз голосової команди.

Крок 2. Сортуємо стовбці (рядки) за кількістю одиниць, як показано на рис. 5.

	OpenV	A	DocN	FileN
OpenV	0	1	1	1
A	0	0	1	1
DocN	0	0	0	0
FileN	0	0	0	0

Рис. 5 – Сортування стовпців матриці за кількістю одиниць

Крок 3. Знаходимо однакові рядки (стовбці) матриці, знаходячи паралельні вершини.

Крок 4. Об'єднуємо паралельні вершини в одну за допомогою операції АБО, як показано на рис. 6.

	OpenV	A	(DocN | FileN)
OpenV	0	1	1
A	0	0	1
(DocN FileN)	0	0	0

Рис. 6 – Об'єднання паралельних вершин за допомогою операції АБО

Крок 5. Знаходимо рядки, між якими є лише одна відмінність. Позначаємо цю вершину як опціональну, як показано на рис. 7.

	OpenV	[A]	(DocN | FileN)
OpenV	0	1	1
[A]	0	0	1
(DocN| FileN)	0	0	0

Рис. 7 – Знаходження опціональної вершини

Таким чином, в результаті обробки матриці суміжності отримано правило граматики, що співпадає з виведеним у прикладі: OpenV [A] (DocN | FileN) [5]. Задачу дослідження, яка полягає у синтезі правил граматики голосового інтерфейсу було розв'язано шляхом формування матриці суміжності фраз природної мови, що моделюється. Подано правила синтезу лінгвістичних правил, що застосовується для виведення фраз КВ-граматики мови. Даний метод забезпечує автоматизацію складання лінгвістичних правил, що полегшує роботу інженерів з розробки голосових інтерфейсів. Метод характеризується мовонезалежністю та дозволяє отримувати правила, які можна використовувати для моделювання природніх мов з одної мовної групи.

Тестуванням перевіримо коректність теоретичного апарату, дослідивши роботу створеного на його основі програмного додатку. В робоче поле програми записується набір фраз природньою мовою, які додаються до глосарію за допомогою відповідної кнопки відповідного пункту меню. Після додавання фраз до глосарію, бачимо інформаційне повідомлення про виконання даної операції, як показано на рис. 8. В результаті виконання попередніх дій формується глосарій у вигляді текстового файлу, в якому зберігаються фрази голосових команд у вигляді нетермінальних символів. Фрагмент створюваного глосарію відображений на рис. 9.

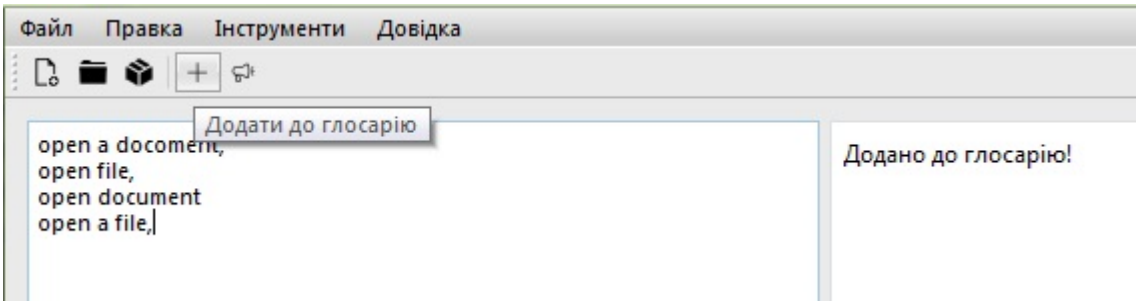

Рис. 8 – Додавання фраз голосових команд до глосарію

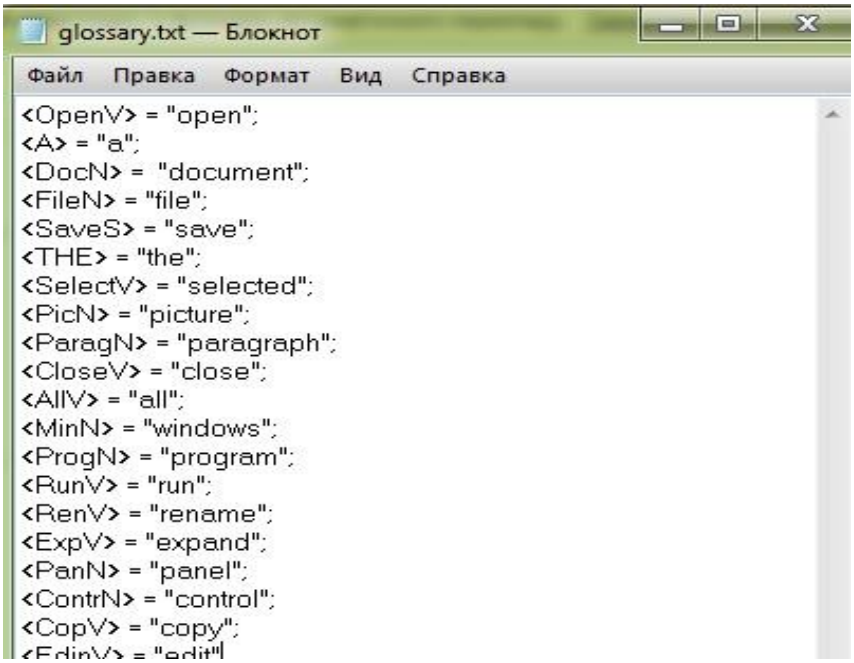

Рис. 9 – Фрагмент глосарію нетермінальних символів

Після генерування правил зберігаємо їх у вигляді граматики у текстовому файлі з кодуванням Юнікод, так як наш метод характеризується мовонезалежністю. Фрагмент файлу граматики, який створений на основі згенерованих правил, можемо побачити на рис. 10.

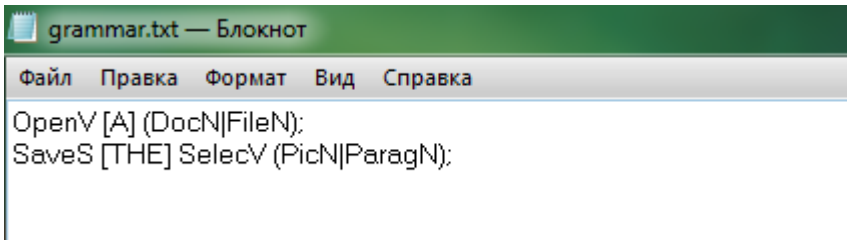

Рис. 10 – Фрагмент файлу збереженої граматики

Отже, задачу дослідження, яка полягає у синтезі правил граматики голосового інтерфейсу було розв'язано шляхом формування матриці суміжності фраз природної мови, що моделюється. Подано стандарти синтезу лінгвістичних правил, що застосовуються для виведення фраз КВ-граматики мови. Даний метод забезпечує автоматизацію складання лінгвістичних правил, що полегшує роботу інженерів з розробки голосових інтерфейсів. Метод характеризується мовонезалежністю та дозволяє отримувати правила, які можна використовувати для моделювання природніх мов з однієї мовної групи. Для реалізації програмного забезпечення розроблено схему алгоритму головної програми, відповідно до якої розроблено графічний інтерфейс для зручності здійснення взаємодії між користувачем та програмним забезпеченням. Розроблено алгоритми для підпрограм, що викликаються головною програмою. У середовищі Qt Creater 5.2 розроблене програмне забезпечення, яке реалізує відповідні алгоритми.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Беленький. А.К. Текстомайнинг. Извлечение информации из неструктурированных текстов. – [Електронний ресурс]. – Режим доступу: https://compress.ru/article.aspx?id=19605
2. Технології автоматизованого видобування знань з тексту. – [Електронний ресурс]. – Режим доступу: https://lektsii.org/7-71000.html
3. Алгоритм приведення контекстно-вільної граматики. – [Електронний ресурс]. – Режим доступу: https://docplayer.ru/30990196-Algoritmy-preobrazovaniya-kontekstno-svobodnyh-grammatik-s-pomoshchyu-grafov.html
4. Волкова И. А. Формальные грамматики и языки. Элементы теории трансляции / И. А. Волкова, Т. В. Руденко. – Издательский отдел факультета ВМиК МГУ, 1998. – 62 с.
5. Зарванська А. І. Синтез граматики голосового інтерфейсу багатомовного програмного продукту. – [Електронний ресурс]. – Режим доступу: http://ir.lib.vntu.edu.ua//handle/123456789/23308

Нестюк Юлія Юріївна – студентка групи 2АКІТ-17б, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, Вінниця, e-mail : yynestiuk@gmail.com

Латанська Анастасія Костянтинівна – студентка групи 2АКІТ-17б, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, Вінниця, e-mail : anasstasija2000@gmail.com

Nestiuk Yuliia Y. – student of 2AKIT-17b group, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail : yynestiuk@gmail.com

Latanska Anastasiia K. – student of 2AKIT-17b group, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail : anasstasija2000@gmail.com