Застосування методу контент-аналізу для формування інформаційних ресурсів в системах електронної контент-комерції / А. Ю. Берко, В. А. Висоцька, М. М. Сороковський // Вісник Нац. ун-ту "Львівська політехніка". – 2012. – № 743: Інформаційні системи та мережі. – С. 3–15. 54. Висоцька В. А. Утворення речень англійською та німецькою за допомогою породжувальних граматик / В.А. Висоцька, Т.В. Шестакевич, Ю. М. Щербина // Вісник Нац. ун-ту "Львівська політехніка". – 2012. – № 744: Комп'ютерні науки та інформаційні технології. – С. 142–152. 55. Висоцька В. А. Інтелектуальна система розподілу дайджестів між працівниками електронних засобів масової інформації / В. А. Висоцька, О. Ю. Окрушко // Вісник Нац. уні-ту "Львівська політехніка". – 2012. – № 744: Комп'ютерні науки та інформаційні технології. – С. 41–53. 56. Утворення українських дієприкметників за допомогою породжувальних граматик / Ю. М. Щербина, Ю. В. Нікольський, В. А. Висоцька, Т. В. Шестакевич // Вісник Нац. ун-ту "Львівська політехніка". – 2011. – № 715: Інформаційні системи та мережі. – С. 354–369.

УДК 004.912

Н. В. Борисова, З. А. Кочуєва, І. В. Оліфенко
Національний технічний університет "Харківський політехнічний інститут"

МЕТОД АВТОМАТИЗОВАНОЇ ЛЕМАТИЗАЦІЇ ДІЄСЛІВ НІМЕЦЬКОЇ МОВИ

Представлено математичне, алгоритмічне та програмне забезпечення розв'язання задачі автоматизованої лематизації німецьких дієслів з відокремлюваними префіксами.

Ключові слова: лематизація німецьких дієслів, автоматизована лематизація, розробка програм-лематизаторів.

Mathware, algorithmic support and software for problem solution of automated lemmatization of German verbs with separated prefixes are represented in the article.

Key words: lemmatization of German verbs, automated lemmatization, development of lemmatizers.

Вступ. Загальна постановка проблеми

Необхідність у створенні програмного забезпечення для автоматизованої лематизації виникає тому, що процедуру лематизації використовують для вирішення багатьох завдань автоматизованого опрацювання природної мови, а саме:

- при індексуванні веб-документів;
- у пошукових алгоритмах для підвищення релевантності пошуку;
- для визначення унікальності текстового контенту;
- у процесі схематизації веб-документів;
- для попереднього опрацювання текстів при класифікації документів;
- при створенні машинних словників;
- у системах машинного перекладу;
- при розмітці корпусів текстів;
- у системах, що навчають іноземної мови тощо.

Під лематизацією будемо розуміти одну з задач морфологічного аналізу, що полягає у приведенні різних текстових форм слова до його нормальної форми шляхом відкидання від словоформи флективних закінчень і повернення основної або словникової форми слова [2].

Особливий інтерес для дослідження представляє автоматизована лематизація німецьких дієслів з відокремлюваними префіксами, оскільки такі префікси знаходяться на відстані від дієслова та можуть частково або повністю змінювати значення дієслова [1].

Аналіз останніх досліджень та публікацій

Автоматизована лематизація здійснюється за допомогою спеціального програмного забезпечення – лематизаторів. Багато з них представлені у мережі Інтернет у відкритому доступі і можуть використовуватися безкоштовно. У межах цього дослідження розглядалися лематизатори, що працюють з німецькою мовою, а саме GermaNet [7], A Self-Learning Context-Aware Lemmatizer for German [12], The Durm German Lemmatizer [13], A freely available morphological analyzer, disambiguation and context sensitive lemmatizer for German [9], Work Package 2 – Lemmatizer [6], StaLe [10] та деякі інші. Розглянуті лематизатори є найвідомішими і використовуються найчастіше. Однак, за усієї різноманітності їх функціональних можливостей ці засоби лематизації мають такі недоліки: опрацьовують переважно іменники, часто в аналізі не враховують контексту, в якому вживається те чи інше слово, працюють під певною операційною системою, на певній апаратній платформі тощо. Усунення зазначених недоліків та розширення функціоналу засобів лематизації породжують необхідність створення досконаліших програм, які могли б ретельніше аналізувати слова.

Формулювання цілі статті

Отже, метою цього дослідження стало визначення особливостей процесу лематизації дієслів з відокремлюваними префіксами у німецькій мови та створення програми для їх автоматизованої лематизації.

Виклад основного матеріалу

Для досягнення поставленої мети необхідно розробити відповідне математичне, алгоритмічне та програмне забезпечення.

Як математичне забезпечення було обрано модель формотворення природної мови, яку запропонував О. В. Пруцков [4]. Базовими поняттями для опису цієї моделі є такі поняття:

- рядок – послідовність символів (не лише літер);
- підрядок – частина рядка або послідовність символів;
- префікс – підрядок, що знаходиться на початку рядка (ліворуч);
- постфікс – підрядок, що знаходиться у кінці рядка (праворуч).

Загальні поняття “префікс” та “постфікс” введено для узагальнення лінгвістичних понять “префікс” й “суфікс”, “закінчення” відповідно. Слово являє собою послідовність символів деякого алфавіту. Рядок може містити декілька слів.

Модель формотворення природної мови розглядає генерацію будь-якої словоформи з певним граматичним значенням як послідовність кінцевого числа перетворень основи [4]. О. В. Пруцков виділяє два основні типи перетворень рядків:

1. Додавання підрядків Р до рядка зліва (префікс) (позначається Р+) або справа (постфікс) (+Р) без зміни самого рядка.
2. Заміна у рядку першого зліва входження підрядка Н на підрядок Р (Н → Р).

Кожне перетворення має зворотне до нього, тобто таке, що робить зворотну дію, а саме:

1. Відокремлення підрядка Р від рядка зліва (Р-) або справа (-Р).
2. Зворотна заміна першого зліва підрядка Р на підрядок Н (Н ← Р).

Ця модель є відкритою, тобто до моделі можна додавати інші типи перетворень. Єдина умова додавання – перетворення повинне мати такі властивості:

- однозначність результату: перетворення завжди приводитиме до одного результату;
- оборотність дії: застосування до рядка прямого, а потім зворотного перетворень не змінює його.

Наявність цих властивостей є обов'язковою умовою для правильної роботи алгоритму.

Нехай перетворення Q – пряме, а перетворення Q' – зворотне даному. Послідовність перетворень (пряма послідовність перетворень, комбіноване перетворення) – це кінцева послідовність перетворень: $R = (Q_1, Q_2, ..., Q_n)$.

Зворотна послідовність перетворень R' є зворотною послідовністю перетворень, зворотних до даних: $R' = (Q'_n, Q'_{n-1}, ..., Q'_1)$.

Пряма послідовність перетворює основу слова S на форму слова F, а зворотна – форму F на основу S (рис. 1). Послідовність перетворень також повинна мати властивості однозначності результату та оборотності дії, як і окреме перетворення.

Рис. 1. Взаємозв'язок між основою, словоформою, прямою та зворотною послідовностями перетворень

Застосувати перетворення Q до форми F означає виконати перетворення Q над формою F для отримання форми Ф. Перетворення Q не завжди можна застосувати до форми F, наприклад, неможливо відокремити постфікс через його відсутність або неможливо зробити заміну підрядка через відсутність вихідного підрядка у формі.

Застосувати послідовність перетворень R до форми F означає застосувати послідовно зліва направо кожне перетворення Q, що складає послідовність R, до форми F. Якщо одне перетворення послідовності R не можна застосувати до форми F, то вся послідовність R вважається такою, що не може бути застосованою до форми F, і процес застосування перетворень послідовності R до форми F припиняється.

Якщо перетворення або послідовність перетворень можуть бути застосовані, то їх результатом є перетворена форма. Якщо перетворення або послідовність перетворень не можуть бути застосовані до форми, то результат є не визначеним.

Деякі перетворення можуть бути застосовані до будь-якої форми F, наприклад, додавання підрядків праворуч і ліворуч. Однак зворотні їм перетворення, що використовуються при визначенні форм слів, не завжди можуть бути застосовані.

Розглянемо роботу моделі на прикладі прямого та зворотного перетворення дієслова німецької мови з відокремлюваним префіксом ***auf\|machen*** – "*відкривати, оформляти*".

Цей глагол у теперішньому часі ***das Präsens des Indikativs*** має таку схему дієвідміни:

mache auf machen auf; machst auf macht auf; macht auf machen auf.

Для представлення дієслова ***auf\|machen*** у форми третьої особи однини послідовність R складатиметься із трьох перетворень:

Q_1: відокремити префікс "***auf***": ***auf-***;

Q_2: додати постфікс "***t***": ***+t***;

Q_3: додати пробіл і префікс "***auf***" як постфікс: ***+_auf***,

і матиме вигляд: $R = (Q_1, Q_2, Q_3) = (auf-, +t, +\_auf)$.

Нехай необхідно одержати форму третьої особи однини F = "***macht auf***" з основи невизначеної форми S = "***aufmach***". Для цього необхідно застосувати послідовність R до основи S:

1. Відокремити префікс "auf": aufmach → mach.
2. Додати постфікс "t": mach → macht.
3. Додати пробіл і префікс "auf" як постфікс: macht → macht auf.

У результаті застосування послідовності R, що складається з трьох перетворень, з основи S = "***aufmach***" отримано словоформу F = "***macht auf***".

Щоб одержати зі словоформи F = "***macht auf***" основу S = "***aufmach***", необхідно застосувати зворотну послідовність перетворень R', що складається зі зворотних перетворень послідовності R, виконаних у зворотній послідовності:

Q'_3: відокремити пробіл і постфікс "***auf***": ***-_auf***,

Q'_2: відокремити постфікс "***t***": ***-t***;

Q'_1: додати постфікс "***auf***" як префікс: ***auf+***;

і має вигляд: $R' = (Q'_3, Q'_2, Q'_1) = (-\_auf, -t, auf+)$.

Застосуємо зворотне перетворення R' до словоформи F = "***macht auf***":

1. Відокремимо пробіл і постфікс "auf": macht auf → macht.

2. Відокремимо постфікс “t”: macht → mach.

3. Додамо постфікс “auf” як префікс: mach → aufmach.

У результаті застосування зворотної послідовності R', що складається із трьох перетворень, зі словоформи F = “*macht auf*” отримано основу S = “*aufmach*”.

Для розв’язання задачі лематизації доцільно застосовувати до словоформ зворотне перетворення.

З використанням описаної вище моделі розроблено алгоритмічне забезпечення процесу автоматизованої лематизації німецьких дієслів з відокремлюваними префіксами:

1. Введення тексту. Користувач завантажує текстовий документ з потрібним йому текстом німецькою мовою або ж друкує його власноруч у спеціальному вікні. Програма отримує і записує текст до тимчасової пам’яті.

2. Поділ тексту на речення. Програма визначає речення шляхом пошуку розділових знаків, які вказують на ступінь закінченості смислових відрізків тексту.

3. Пошук відокремлюваного префікса із заданого списку у кінці речення. Після поділу тексту на речення програма починає аналізувати кожне речення, шукаючи перед вищезазначеними розділовими знаками відокремлюваний префікс.

4. Пошук дієслів у реченні. Наступним кроком є аналіз кожного речення, де було знайдено відокремлюваний префікс, та пошук дієслова у цих реченнях.

5. Приєднання префікса до дієслова. Знайшовши відокремлюваний префікс та присудок, програма приєднує префікс до дієслова. Отримане слово записується у пам’яті.

6. Нормалізація дієслова. Програма нормалізує дієслово за допомогою алгоритму лематизації.

7. Представлення користувачеві у спеціальному вікні результату – нормалізованого дієслова з відокремлюваним префіксом та його переклад.

Під час розроблення програмного забезпечення розв’язання задачі автоматизованої лематизації дієслів з відокремлюваними префіксами у німецькій мові було використано підхід об’єктивно-орієнтованого програмування, програмне забезпечення відповідає усім основним критеріям його розроблення. Для розроблення було обрано мову програмування Java [3, 5].

Для створення програмного забезпечення використовували як стандартні пакети та бібліотеки Java, такі як java.util.ArrayList, java.io.IOException, java.util.Scanner, javax.swing.JFileChooser, javax.swing.JOptionPane, java.io.BufferedReader, java.io.File, java.io.InputStreamReader, org.apache.http.HttpResponse, org.apache.http.client.methods.HttpGet, org.apache.http.client.HttpClient, org.apache.http.impl.client.HttpClientBuilder; так і спеціально розроблені для розв’язання поставленої задачі класи, функції, методи, колекції та масиви. Програмне забезпечення реалізовано у середовищі NetBeans IDE 8.1 [11].

Всі змінні та методи, необхідні для роботи програмного забезпечення, розміщено у класі Gram. Колекція ArrayList “Offers” типу String містить частини речень, знайдених в опрацьованому тексті. Змінна count слугує як лічильник вищезазначених частин речення або ж самих речень. Масив End – статичний масив, в якому зберігаються відокремлювані префікси, що розміщуються в кінці речення. Масив StopPath – це статичний масив, у якому записано список стоп-слів, які не враховано під час аналізу речення. У масив NoInf записано сильні дієслова німецької мови, що змінюють кореневу голосну в теперішньому часі. В процесі лематизації дієслова з цього масиву замінюються дієсловами з аналогічного масиву Infinitiv, які вказано в інфінітивній формі. Клас Frame успадковує клас javax.swing.JFrame і відповідає за візуалізацію програмного забезпечення. Робочі панелі jTextPane1і jTextPane2, а також кнопки ClearBox, ReadWithFile, jFileChooser1 і скроллер згенеровано за допомогою NetBeans8.1 автоматично.

Для практичної реалізації задачі також було створено такі функції:

- функція FileStream відповідає за відкриття текстового файлу через діалогове вікно. Програмне забезпечення працює тільки з файлами формату txt;
- функція readFile містить методи, необхідні для зчитування обраного файла. Програмне забезпечення використовує для опрацьованого тексту кодування UTF-8;
- функція Delete використовується для видалення різних знаків (наприклад "”", " " і т.д.) для того, щоб надалі не потрібно було аналізувати, з якого символу починається слово;

• функція ChangeTextToSentence перетворює весь текст на байтовий масив для пошуку знаків пунктуації, посимвольно йде по тексту і вирізає речення для подальшого опрацювання з урахуванням знаків пунктуації. Також ця функція містить методи SetToFrame2, який дає змогу обнулити результат другого вікна, та GetOfFrame, який відповідає за те, що опрацьовуватиметься текст саме з першого вікна;

• функція WordEnd порівнює останнє слово речення (частини речення) з закінченнями, що зберігаються в масиві End. Потім перевіряють, чи не належить інфінітив до масиву NoInf. Якщо слово міститься в масиві, то воно замінюється на відповідне слово з масиву NoInf. В іншому випадку закінчення відкидається і повертається інфінітивна форма дієслова, тобто додається закінчення -en. До закінчень дієслів відносимо -e, -t, -st. Після цього до дієслова додається знайдений префікс і виконується переклад за допомогою функції sendTranslate;

• функція sendTranslate містить конструктор, в якому вказується, що перекладають дієсліва за допомогою онлайн-сервісу Leo.org [8];

• функція ClearBoxActionPerformed очищає робочі панелі для подальшого аналізу текстів;

• функція toProccessActionPerformed запускає потоки, необхідні для виконання програми. За допомогою цієї функції очищують вікно результату та виконують метод ChangeTextToSentence у побічному потоці;

• функція ReadWithFileActionPerformed запускає потік вибору файлу для зчитування з нього інформації та подальшого аналізу;

• функції SetToFrame1 і SetToFrame2 відповідають за те, що на робочих панелях відображатиметься обраний текст або текстовий файл. SetToFrame2 додає результат з урахуванням заміни;

• функція GetOfFrame необхідна, щоб отримати текст для подальшого опрацювання;

• функція GetResult виводить результат програми, тобто знайдені нормалізовані дієслова з відокремлюваними префіксами разом з їх перекладом.

Програма має зручний інтерфейс для користування та інтуїтивно зрозуміле меню. Застосування розробленого програмного забезпечення дає змогу аналізувати будь-який текст німецькою мовою. Робоче вікно програми наведено на рис. 2.

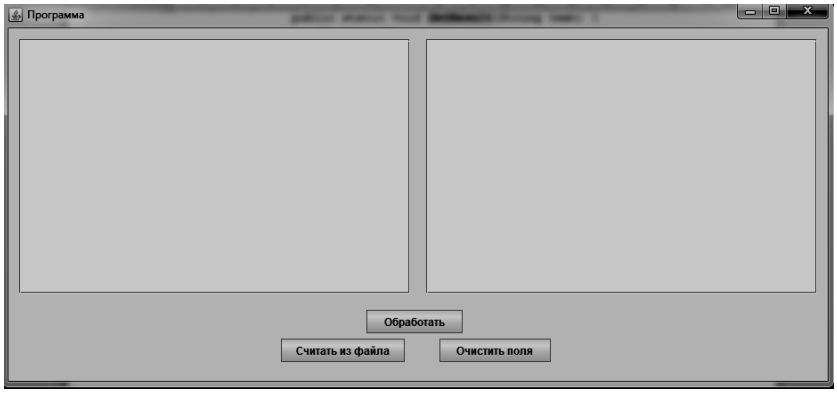

Рис. 2. Робоче вікно програми

У вікно, розташоване зліва, завантажується текст за допомогою кнопки “Зчитати з файла” або користувач вводить текст у вікно вручну. Якщо користувач намагатиметься завантажити текст не txt формату, то він отримає відповідне повідомлення.

Для подальшого аналізу тексту необхідно натиснути кнопку “Опрацювати”. Програма опрацьовує текст. У правому вікні відображається результат роботи програми – знайдені у тексті та вже нормалізовані дієслова з відокремлюваними префіксами з їх перекладом (рис. 3).

Для перекладу використовується онлайн-сервіс Leo.org. Програма надає користувачеві три найпоширеніші варіанти перекладу знайдених у тексті дієслів. Це робить програму більш якісною та досконалою, бо дає змогу враховувати контекст, у якому використовується слово. Окрім того, користувач може побачити додаткову інформацію про переклад знайдених дієслів – їх наголос та з яким прийменником вони використовуються, який відмінок іменників ставити після цих дієслів. Якщо переклад слова не було знайдено у словнику Leo, то слово виводиться у вікно без перекладу.

Якщо у тексті не було знайдено жодного дієслова з відокремлюваним префіксом, то праве вікно залишиться пустим. Також праве вікно залишиться пустим, якщо користувач захоче проаналізувати текст, написаний не німецькою мовою. Якщо користувач хоче проаналізувати ще один текст, він може або видалити текст з вікон вручну, або скористатися кнопкою “Очистити поля”. Якщо ж користувач отримав потрібну інформацію і у нього немає потреби аналізувати ще один текст, він натискає стандартну кнопку “Закрити” у правому верхньому куті вікна програми і завершує роботу з програмою.

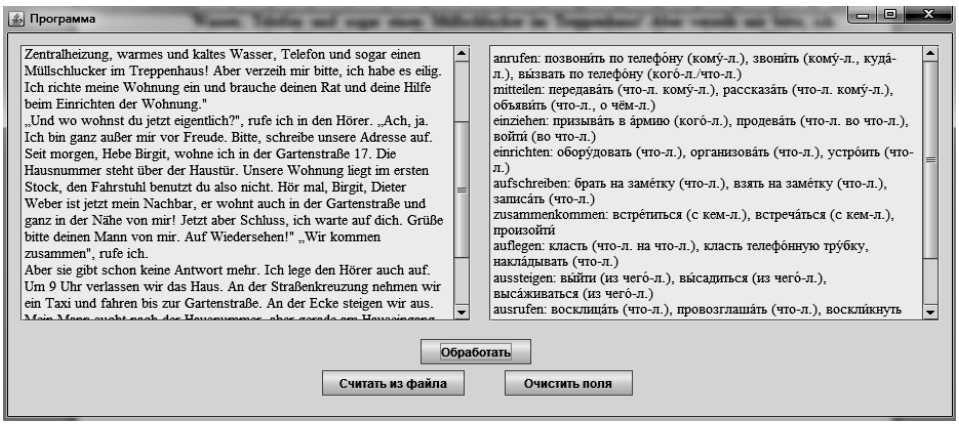

Рис. 3. Результат роботи програми

Висновки та перспективи подальших наукових розвідок

Отже, розроблене програмне забезпечення реалізує запропонований метод автоматизованої лематизації дієслів з відокремлюваними префіксами у німецькій мові та має весь необхідний для розв’язання поставленої задачі функціонал. Можливе подальше удосконалення програмного забезпечення.

1. Кияк Т. Р.Теорія і практика перекладу. Німецька мова / Т. Р. Кияк, А. М. Науменко, О. Ф. Огуй. – Вінниця, 2006. – 410 с. 2. Лингвистика и обработка текстов: [Електронний ресурс]. – Режим доступу: http://www.osp.ru/os/2013/04/13035562/ 3. Програмування на мові Java: [Електронний ресурс]. – Режим доступу: http://javaland.com.ua/programuvannya; 4. Пруцков А. В. Модели, методы и программы автоматической обработки форм слов в естественно-языковых интерфейсах: дис. … д-ра техн. наук: спец. 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей / А. В. Пруцков. – Рязань, 2015. – 253 с. 5. Що таке JAVA?: [Електронний ресурс]. – Режим доступу: http://a-yak.com/shho-take-java/ 6. Belica C. WP2 – Lemmatizer. Final Report / C. Belica]. – Режим доступу: http://www1.ids-mannheim.de/fileadmin/kl/ dokumente/glemmrep.pdf 7. GermaNet: [Електронний ресурс]. – Режим доступу: http://www.sfs.uni tuebingen.de/GermaNet/ index.shtml/ 8. Leo: [Електронний ресурс]. – Режим доступу: http://dict.leo.org/rude/index_de.html 9. Lezius W. A freely available morphological analyzer, disambiguator and context sensitive lemmatizer for German / W. Lezius, R. Rapp, M. Wettler: [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/1961142_ 10. Loponen A. A Dictionary- and Corpus-Independent Statistical Lemmatizer for Information Retrieval in Low Resource Languages / A. Loponen, K. Järvelin: [Електронний ресурс]. – Режим доступу: http://www.sis.uta.fi/ infim/julkaisut/fire/2010/StaLe_eval-preprint.pdf 11. NetBeans Release 8.1: [Електронний ресурс]. – Режим доступу: http://docs.oracle.com/netbeans/nb81/netbeans/index.html 12. Perera P. A Self-Learning Context-Aware Lemmatizer for German / P. Perera, R. Witte: [Електронний ресурс]. – Режим доступу: http://rene-witte.net/german-lemmatization 13. Perera P. The Durm German Lemmatizer / P. Perera, R. Witte: [Електронний ресурс]. – Режим доступу: http://www.semanticsoftware.info/durm-german-lemmatizer