ПРОГРАМНА СИСТЕМА АВТОМАТИЧНОГО ГЕНЕРУВАННЯ ОПИСІВ ГОТЕЛІВ ІЗ ВИКОРИСТАННЯМ ШАБЛОНІВ НА ОСНОВІ ЛАНЦЮГІВ МАРКОВА

С. Я. Крепич

Тернопільський національний економічний університет
вул. Львівська, 11, м. Тернопіль, 46000, Україна. E-mail: s.krepych@tneu.edu.ua

Робота присвячена вирішенню проблеми інформаційної присутності готелів у мережі Інтернет. У вагомій частині готелів немає текстового опису для представлення в профілі готелю. Але є набори структурованих даних, які характеризують готель і по ним можна сформулювати текстовий опис. Звідси і постає актуальність задачі, а саме розробка комбінованого підходу до автоматизованої генерації тексту на основі шаблонів та з використання ланцюгів Маркова для поширення різноманітності текстів. Підхід продемонстровано на прикладі генерування описів готелів. Науковою новизною отриманих результатів є удосконалений підхід до автоматичного генерування опису об'єкта за набором даних його характеристик та відповідний алгоритм. Розроблено підсистему, яка здатна автоматично підготувати вхідні дані по об'єкту від всіх доступних провайдерів і генерувати текстовий опис цього об'єкта за доступними характеристиками із використанням шаблонів.

Ключові слова: генерація опису, шаблони, ланцюг Маркова, алгоритм, програма.

ПРОГРАММНАЯ СИСТЕМА АВТОМАТИЧЕСКОГО ГЕНЕРИРОВАНИЯ ОПИСАНИЯ ОТЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ ШАБЛОНОВ НА ОСНОВЕ ЦЕПЕЙ МАРКОВА

С. Я. Крепич

Тернопольской национальной экономической университет
ул. Львовская, 11, г. Тернополь, 46000, Украина. E-mail: s.krepych@tneu.edu.ua

Работа посвящена решению проблемы информационного присутствия отелей в сети Интернет. У большинства гостиниц нет текстового описания для представления в профиле отеля. Но есть наборы структурированных данных, которые характеризуют отель, и по ним можно сформировать текстовое описание. Отсюда и возникает актуальность задачи, а именно разработка комбинированного подхода к автоматизированной генерации текста на основе шаблонов и с использованием цепей Маркова для расширения разнообразия текстов. Подход продемонстрирован на примере генерации описаний отелей. Научной новизной полученных результатов является усовершенствованный подход к автоматическому генерированию описания объекта по набору данных его характеристик и соответствующий алгоритм. Разработана подсистема, которая способна автоматически подготовить исходные данные по объекту от всех доступных провайдеров и генерировать текстовое описание этого объекта по доступным ему характеристикам с использованием шаблонов.

Ключевые слова: генерация описания, шаблоны, цепь Маркова, алгоритм, программа.

АКТУАЛЬНІСТЬ РОБОТИ. Робота присвячена проблемі інформаційної присутності готелів по яким працює пошук, зокрема і на проєкті «Hotelsbroker». У вагомій частині готелів немає текстового опису для представлення в профілі готелю. Але є набори структурованих даних, які характеризують готель і по ним можна сформулювати текстовий опис.

Актуальною задачею є саме створення автоматизованої системи, яка могла б вирішувати великий об'єм роботи по генеруванні описів готелів, який у порівнянні з виконанням цієї самої роботи людиною буде не менш якісний, однак значно дешевший.

Поставлену задачу можна розв'язувати двома шляхами. Перший шлях – написати усі описи, яких не вистачає, за допомогою редактора. А другий шлях – це автоматизувати поставлену задачу за допомогою комп'ютерної програми.

Такі задачі на сьогоднішній день вирішуються із використанням багатьох методів, зокрема методів штучного інтелекту, методу ланцюгів Маркова тощо. Однак основним недоліком усіх таких методів є те, що вони не можуть реалізувати задачі з потрібним рівнем якості генерованого опису. Для підвищення рівня якості генерованого тексту запропоновано використовувати модифікований алгоритм генерації опису об'єкту із використанням шаблонів на основі методу ланцюгів Маркова [1, 2].

MeanoTek є компанією, яка займається аналізом і генерацією текстів. Останньою їх розробкою є нейронні мережі, які генерують текст. Результати своєї розробки вони представили на конференції Діалог-2015 [3].

На рис. 1 представлена схема шарів рекурентної нейронної мережі [1].

Рисунок 1 – Схема шарів рекурентної нейронної мережі

Шар проекції перетворює індекс слова в вектор. Рекурентний шар описується формулою (1).

$$h(t) = f(Wx(t) + Vh(t-1) + b), \quad (1)$$

де f – нелінійна функція (у нашому випадку гіперболічна дотична функція), W і V – матриці ваги між проекцією і рекурентним шаром.

Список аспектів кодується вектором, де кожний змінний відповідає один аспект. Змінна може приймати значення 1 чи 0 в залежності від того, кодує змінна присутність цього аспекту у вказаному реченні чи ні. Задачею нейронної мережі є передбачення наступного слова, знаючи нинішнє слово і вектор аспектів [4].

Така нейронна мережа може сама генерувати речення, хоча вони і не ідеальні. Такий текст місцями втрачає зміст і має порушення правил граматики.

Приклад такого тексту:

Акумулятор, швидкість гарна. Дизайн, звук, функціонал, маса різних днів вистачає. Красивий, тому трохи назад, процесор, чуйний сенсор. Гарний екран, передача кольору. Дизайн, батарея, не гальмують, практичний.

Одним з найпопулярніших методів генерації тексту є метод генерації тексту за допомогою ланцюгів Маркова [2]. Ланцюги Маркова широко використовуються для моделювання поєднання слів в тексті. Отже, розглянемо текст, який складається з w слів. Представимо процес, в якому станами є слова, так що коли він знаходиться в стані S_i система переходить в стан S_j згідно матриці перехідних ймовірностей. По-перше, потрібно «навчити» систему: подати на вхід достатньо великий текст для оцінки перехідних ймовірностей. Потім можна будувати траєкторії марківських ланцюгів. Збільшення змістового навантаження тексту, побудованого за допомогою алгоритму ланцюгів Маркова можливо лише при збільшенні порядку, де станом є не одне слово, а множина з більшою потужністю – пари (u, v) , трійки (u, v, w) тощо. При чому, що в ланцюгах першого, що п'ятого порядку, змісту буде ще не дуже багато. Зміст почне проявлятися при збільшенні розмірності порядку як мінімум до середньої кількості слів в типовій фразі вихідного тексту. Але таким шляхом просуватись не можна, адже ріст змістового навантаження в ланцюгах Маркова високого порядку відбувається значно повільніше, ніж спад унікальності тексту. А текст, побудований на марківських ланцюгах, для прикладу тридцятого порядку, все ще буде не настільки змістовним, щоб зацікавити користувача, але вже достатньо схожим з оригінальним текстом.

Ця технологія зараз дуже широко використовується в Інтернеті для створення контенту вебсторінок. Люди, які бажають збільшити трафік на свій сайт та підвищити його рейтинг в пошукових системах, намагаються розмістити на своїх сторінках якомога більше ключових слів для пошуку. Також ланцюги Маркова широко використовують для роботи з текстом, їх застосовують для визначення авторства, аналізу унікальності тексту тощо.

Чим більший корпус, тобто база на основі якої в подальшому буде генеруватися текст, тим більша

буде різноманітність вихідних текстів при прогонах. В реальності цей метод з корпусом в 500000 слів, згенерує текст в якому прослідковуватиметься зміст, однак для задачі генерації описів готелів метод має обмежене застосування.

І наостанок, проблему генерування текстів висвітлював в 2015-му році на конференції Moscow Django MeetUp №28 доповідач від компанії Rambler у доповіді «Генерація літературних текстів» [5]. Автором була розглянута подібна ситуація. На вибір було подано два варіанти вирішення проблеми – навчити генерувати тексти нейронну мережу або генерувати тексти по шаблону. Обидва варіанти цікавіші за написання описів редакторами. Їх лозунг – «Це той випадок, коли робота програміста коштує менше за роботу редактора». Але перший варіант не підходить для генерації описів готелів через те, що найскладніше в цьому процесі є навчання самої нейронної мережі. Бізнес, в якому ця проблема актуальна, вимагає чітких термінів і гарантій. А гарантії успіху з цим підходом компанія не змогла забезпечити. І додатковим тягарем було те, що вимоги до опису, з високою ймовірністю, можуть змінитися і проблему доведеться вирішувати з нуля. Тому команда Rambler Travel схилилася в сторону шаблонних текстів [5].

Отож, за результатами проведеного аналізу літературних джерел, показано доцільність обрання генерації описів по шаблонах і ланцюгів Маркова, які дозволяють розширити варіативність описів. Для збільшення вибірки характеристик готелю пропонується використання text mining, який надасть можливість добути нові дані для шаблонів із відгуків туристів. Також, на противагу рішенням компанії Rambler, пропонується перевести управління ймовірностями з емпіричного на теоретичний рівень, щоб ймовірності вибору шаблонів проставлялись редактором не випадковим чином, а приймалися на основі існуючих описів, наявних вхідних даних та інших коефіцієнтів, які можуть вплинути на кінцевий результат.

МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ.

Опис готелю завжди починається з даних про готель. Це його стартовий блок і для нього дані будуть у будь-якому випадку. Далі всі доступні дані розподіляються по спеціальних блоках. Це може бути блок комплектації номера, послуг, які надаються на території готелю чи інформації по харчуванню. Зручно такі дані представляти у вигляді графа, де стартовою вершиною є готель, а інші блоки спускаються від цієї вершини донизу (див. рис. 2).

Рисунок 2– Граф розбиття опису на блоки і шаблони

На рис. 2 вершина A являє собою перший блок опису готелю, так би мовити «початкова» вершина, опис по якій буде завжди, адже це безпосередньо назва готелю. Всі вершини B – блоки, які можуть далі включатись до опису. Змінною n позначено їх кількість. Вершини C – це окремі шаблони, якими може бути представлений опис блоку. Змінною m позначена кількість всіх шаблонів блоку. У кожній із вершин B та C є свої умови, які накладаються на вхідні дані. Блок може бути доданий до опису лише у випадку, коли наявні всі необхідні для його шаблонів вхідні дані.

Граф, представлений на рис. 2, є ланцюгом Маркова, адже складається із послідовності випадкових подій (вузлів), в якій ймовірність кожної події залежить лише від стану, в якому процес знаходиться в поточний момент і не залежить від більш ранніх станів [2]. Ланцюг визначається наступними характеристиками:

1. Множиною станів S ;
2. Вектором початкових ймовірностей $\vec{p}_0$;
3. Матрицею перехідних ймовірностей P .

Усі описані категорії вершин графа, представленого на рис. 2, і є множиною станів ланцюга. Потужність множини станів для кожного окремого процесу генерування опису готелю буде різною і залежатиме від наявності необхідної кількості даних для представлення певного блоку чи шаблону. Ймовірності переходів між початковою вершиною та блоками B визначається за такими показниками, а саме:

- DV_i – кількість вхідних даних для i -го блоку

при описі готелю;

- k_i – коефіцієнт важливості додавання i -го блоку в опис готелю;

- DS_i – статистичний показник кількості вживання i -го блоку при попередніх описах готелів (на самому першому кроці, коли статистика відсутня, цей показник по кожному блоці дорівнює 1).

Показник ймовірності переходу від початкового стану до стану B_i визначається за формулою:

$$P_{0i} = \left(\frac{DV_i}{\sum_{i=1}^n DV_i} + \frac{k_i}{\sum_{i=1}^n k_i} + \frac{DS_i}{\sum_{i=1}^n DS_i} \right) / 3. \quad (2)$$

Кожен із блоків B може бути представлений шаблоном C . Так як і блоки, шаблони мають свої умови для відбору. Серед всіх шаблонів, які за умовами підходять до представлення блоку, вибирається один випадковим чином. Ймовірність використання того чи іншого шаблону визначається наступним чином: користувач пропонує цілі числа в якості ймовірності шаблону, а ймовірність використання одного шаблону з набору є відносною до ймовірностей використання інших. Сумарна ймовірність використання всіх шаблонів із набору дорівнює 1.

На вихід подається масив із діапазонів ймовірності відносно суми всіх ймовірностей і сама сума ймовірностей. Розглянемо процес визначення ймовірності

використання певного шаблону на прикладі. Наприклад, є три шаблони. Ймовірність, проставлена користувачем, кожного із них рівна 2, 40 і 60 відповідно. Сумарна «ймовірність» рівна 102. Число 102 є ніщо інше як сумарна ймовірність (=1) використання кожного шаблону із множини, а ймовірність вказаних шаблонів і є частками цієї сумарної ймовірності. Частки мають свої діапазони в межах суми. Схематично це зображено на рис. 3(а).

Щоб визначити, який із шаблонів буде використаний треба взяти випадкове число в діапазоні $[1...102]$ і передати його в функцію `_choice_from_group_probability` разом із інтервалами кожного із шаблонів. Припустимо, що наше випадкове число із того інтервалу дорівнює 71. Інтервали передаються списком і функція циклом проходить кожен із інтервалів і порівнює число 71 з межами інтервалу. В результаті всіх цих операцій вибрано третій шаблон так як 71 належить до діапазону $[61...101]$. Для представлення ймовірності вибору вказаного шаблону перетворимо всі цілі величини у частки відносно 1. Аналогічна схема але з інтервалами, об'єднання яких рівне 1 проілюстрована на рис. 3(б). Значення 71 відносно 102 має частку 0,7. І як і в попередньому випадку визначає що буде вибрано третій шаблон.

Рисунок 3 – Схема розподілу ймовірності використання шаблону

Побудуємо, для прикладу, ланцюг Маркова, який представляє процес генерування опису готелю на основі вхідних даних по 5 блоках, кожний з яких має по 2 шаблони і припустимо, що ці шаблони є рівноцінними і ймовірність вибору одного з них рівна 0,5.

У табл.1 представлені необхідні дані для визначення ймовірностей переходів від початкового стану ланцюга до станів ланцюга, які є блоками опису готелю, за формулою (2).

Таблиця 1 – Вхідні дані для визначення ймовірності переходу від початкового стану до стану,

що є блоком B_i

Назва блоку	DV_i	k_i	DS_i	P_{0i}
Номери	5	1	100	0,3
Харчування	7	0,5	90	0,26
Проїзд	3	0,5	60	0,17
Паркінг	2	0,2	80	0,13
Розваги	5	0,2	40	0,14
$\sum_{i=1}^n$	22	2,4	370	1

Ланцюг Маркова, побудований для цього прикладу, зображений на рис. 4.

Матриця перехідних ймовірностей для ланцюга набуде вигляду:

$$P = \begin{pmatrix} 0 & 0,3 & 0,26 & 0,17 & 0,13 & 0,14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,5 & 0,5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,5 & 0,5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,5 & 0,5 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Рисунок 4 – Ланцюг Маркова для прикладу

Однак досить часто виникає ситуація, коли одні блоки вибираються рідше від інших в міру того, що для них часто не вистачає вхідних даних. Щоб уникнути такої ситуації необхідно обов'язково вести статистику використання блоків. Коли буде чітке бачення ситуації, можна буде деяким блокам розширити кількість доступних шаблонів, а деякі можливо замінити.

Текст генерується по шаблону. В шаблоні спеціальними вставками вказується, що і як представити.

На даний момент є три типи вставки:

1. Підстановка змінної. Вважається, що значення змінної перебуває в готовому стані.
2. Підстановка з вибором одного із значень, задається список значень серед яких треба вибрати одне. Змінна з пункту 1 може фігурувати в списку.
3. Виконання функції. В окремих випадках треба на місці перетворити змінну.

Наприклад форматування дати, числа, визначення форми слова відносно числа з яким пов'язана змінна, встановлення першої літери великою тощо.

Розширює цей список перелік функцій перетворення змінної пов'язаної з морфологією. Функції склеювання масиву через сепаратор.

Порядок виконання підстановок має значення. Так як результат одної функції може бути вхідними даними для іншої або для вставки виду – підстановка з вибором одного із значень.

Розглянемо послідовність операцій, яка виконується при відборі блоків, шаблонів блоків і безпосередньо процесу генерування опису блоку по шаблону. Послідовність цих дій представлена на рис. 5 [6].

Рисунок 5 – Блок-схема роботи алгоритму генерації опису

Код функцій, задача яких полягає у виборі одного шаблону із множини можливих, представлений нижче.

```
def choice_tpl_by_probability(list_tpl, vars_tpl):
    group = _group_tpl_by_id(list_tpl, vars_tpl)
    return _choice_from_group_probability(group,
    random.randint(1, group[1]))
```

```
def _group_tpl_by_id(list_tpl, vars_tpl):
    probability_list = []
    max_num_probability = 0
    for item in list_tpl:
        if not item.get('conditions', False) or
        validate_conditions(
            item.get('conditions', vars_tpl)
        ):
            probability = item.get('probability', 1)
            probability_list.append((max_num_probability,
            max_num_probability + probability, item))
            max_num_probability += probability
    return probability_list, max_num_probability
```

```
def _choice_from_group_probability(group_probability,
    num_probability):
    probability_list, max_num_probability =
    group_probability
    if max_num_probability:
        for start_prob, end_prob, item in probability_list:
            if start_prob < num_probability <= end_prob:
                return item
```

Викликається функція `choice_tpl_by_probability`. Вона є публічною і саме вона використовується в генераторі. Сама ж функція для своєї роботи використовує дві приватні, які займаються

опрацюванням масиву шаблонів, які передаються на вибір. Функція `_group_tpl_by_id` із масиву всіх шаблонів відбирає тільки ті, які проходять перевірку на умови шаблону і паралельно запам'ятовує ймовірність їх використання. Ця ймовірність прописана в шаблоні. Та якщо користувач забув вказати її при створенні шаблону, то її значення по замовчуванню буде рівне 1.

Схема взаємодії модулів програми, зокрема рух даних між ними та базою даних, представлена на рис. 6. Зокрема модуль TGen – головний модуль програми. Він приймає дані від модулів Selectors і займається саме генерацією тексту по шаблонах [7].

Рисунок 6 – Схема організації модулів програми

Це основний модуль всієї програми. Він працює з морфологією і шаблонами. Для морфології до програми підключено сторонню бібліотеку `pymorphy` [8]. `Pymorphy` – це морфологічний аналізатор для слів. Він може:

- приводити слово до нормальної форми;
- приводити слово до потрібної форми (наприклад поставити слово у множину чи потрібний відмінок);
- повертати інформацію про слово (рід, форма, відмінок).

При роботі використовує словник `OpenCorpora`, а для незнайомих слів будуються гіпотези [9]. Ліцензований під MIT, що дозволяє вільно використовувати його в роботі. Розглянемо функції модуля з використанням цього розширення. Функція `_take_word_form` відповідає за отримання потрібної форми слова.

```
# -*- coding: utf-8 -*-
import pymorphy2
MORPHY = pymorphy2.MorphAnalyzer()
def _take_word_form(word, form):
    for item in MORPHY.parse(word):
        word_form = item.inflect({form})
        if word_form:
            return word_form.word
    return word
```

Розглянемо приклади генерування описів. Всі тести будуть проведені по готелю `Beach Albatros Resort`, який розташований в курорті Хургада, Єгипет. Профіль цього готелю представлений на рис. 7.

Рисунок 7 – Профіль готелю Beach Albatros Resort

В конструктор генератора переданий такий контекст.

```
placesList = ['u'Большой аквариум Хургады',
'u'Дайвинг-центр Colona Divers', 'u'Лунный пляж',
'u'Музей песчаных скульптур Sand City']
places = u" и ".join([u", ".join(placesList[0:-1]),
placesList[-1]])
servicesList = ['u'Обслуживание номеров',
'u'Парковка', 'u'Парковка автомобиля служащим
отеля', 'u'Лифт', 'u'Круглосуточная рецепция'];
services = u" и ".join([u", ".join(servicesList[0:-1]),
servicesList[-1]])
print gen.generate_text(
    ['Heading', 'Services', 'Places'],
    {
        'city': 'u'Хургада',
        'distance': '27800',
        'first_sea_line': '1',
        'name': 'u'Beach Albatros Resort',
        'type': 'u'Гостиничный комплекс',
        'places': places,
        'places_count': '4',
        'services': services,
        'services_count': '5'
    })
```

Ось такий фрагмент коду склеює список однорідних елементів через кому, а останній елемент через “і”.

```
u" и ".join([u", ".join(placesList[0:-1]), placesList[-1]])
```

В генератор такий фрагмент передається вже готовим тому, що методи рендеру в текст шаблону не можуть вкраплювати такий тип, як list. Вони працюють з примітивними типами як рядок чи число.

Для цього прикладу були підготовані шаблони тексту, умови і ймовірності, яких показано в табл. 2.

Даних, які передані для генерації опису блоку вистачає для показу всіх шаблонів крім першого. Перший орієнтований на те, що готель розташований не далеко відносно центру міста. Цей готель піддану умову не підлягає. Проте для нього можна застосувати шаблон заголовку із зазначенням, що го-

тель розташований в районі першої берегової лінії. Отже в результуючому тексті опису будуть представлені всі три блоки. Після першого запуску отримали опис представлений нижче.

Таблиця 2 – Таблиця з шаблонами блоків

Блок	Умова	Ймовірність	Шаблон
Заголовок	distance > 10 and distance < 20	40	<name~inflect_case:nomn~capfirst> {разместился расположен находится} в [distance] от центра <city~inflect_case:gent~capfirst>.
Заголовок	first_sea_line == 1	40	<name~inflect_case:nomn~capfirst> {разместился расположен находится} в <city~inflect_case:gent~capfirst> на первой береговой линии.
Заголовок	—	20	Отель <name~inflect_case:nomn~capfirst> {разместился расположен находится} в <city~inflect_case:loct~capfirst>.
Сервіси	services_count > 2	40	К удобству гостей неподалеку можно посетить {множество много} местных достопримечательностей таких, как <services~lower>.
Сервіси	services_count > 2	70	Этот <type~lower> предлагает {множество много} первоклассных удобств, таких как <services~lower>.
Місця	places_count > 2	40	К удобству гостей неподалеку можно посетить множество местных достопримечательностей таких, как <places>.
Місця	places_count > 1	20	Рядом с <name~capfirst> находятся такие достопримечательности, как <places>.

Отель Beach Albatros Resort находится в Хургаде. Этот гостиничный комплекс предлагает множество первоклассных удобств, таких как обслуживание номеров, парковка, парковка автомобиля служащим отеля, лифт и круглосуточная рецепция. К удобству гостей неподалеку можно посетить множество местных достопримечательностей таких, как Большой аквариум Хургады, Дайвинг-центр Colona Divers, Лунный пляж и Музей песчаных скульптур Sand City.

Логічно припустити, що після наступного запуску генератора опису по тих же вхідних даних отримаємо, дещо інший текст опису. Текст опису після другого запуску представлений нижче.

Beach Albatros Resort расположен в Хургада на первой береговой линии. Рядом с Beach Albatros Resort находятся такие достопримечательности, как Большой аквариум Хургады, Дайвинг-центр Colona Divers, Лунный пляж и Музей песчаных скульптур Sand City. Этот гостиничный комплекс предлагает много первоклассных удобств, таких как обслуживание номеров, парковка, парковка автомобиля служащим отеля, лифт и круглосуточная рецепция.

длагает много первоклассных удобств, таких как обслуживание номеров, парковка, парковка автомобиля служащим отеля, лифт и круглосуточная рецепция.

На рис. 8 представлено як вписується цей опис в профіль готелю [10].

Для того, щоб порівнювати описи генеровані генератором з описами, які були доступні по деяким готелям, в адміністративному розділі проекту hotelsbroker реалізовану сторінку зі співставленням двох описів. Теоретично дані у них мають співпадати, особливі ті, що стосуються сервісів та комплектації. На рис. 9 представлено приклад такого співставлення.

У такій перевірці є ще одне призначення. Не всі згенеровані описи ідеальні. Приблизно 1% усіх описів містить деякі неточності або моменти дублювання і в цьому розділі можна відредагувати його вручну, щоб таким чином виправити проблему або згенерувати опис повторно.

Такси — от 236 ₴

Время в пути: 20 мин. в пути
Место встречи: Аэропорт

Встреча в аэропорту, помощь с багажом, гарантированная подача

Описание отеля

Beach Albatros Resort расположен в Хургада на первой береговой линии.

Рядом с Beach Albatros Resort находятся такие достопримечательности, как Большой аквариум Хургады, Дайвинг-центр Colona Divers, Лунный пляж и Музей песчаных скульптур Sand City.

Этот гостиничный комплекс предлагает много первоклассных удобств, таких как обслуживание номеров, парковка, парковка автомобиля служащим отеля, лифт и круглосуточная рецепция.

Подарки и бонусы отеля

Комплимент от отеля по приезду

Рисунок 8 – Опис готелю в профілі на проєкті HotelsBroker

Описание отелей

+ Добавить описание

Выберите отель...

Применить **Сбросить**

Показаны записи 1-20 из 173 617.

Отель	Включено	Автоописание	Описание LS
Grecian Sands Hotel (73)	✓ Да	<p>Уютный и стильный, Grecian Sands Hotel размещается в месте под названием Айя-Напа. Рецепция работает круглосуточно, благодаря этому вы сможете оперативно решить все возникшие вопросы.</p> <p>Отель предоставляет услуги, которые смогут удовлетворить самые необычные требования гостя. В апартаментах предусмотрен чайник / кофеварка, вы сможете насладиться горячим напитком не покидая своего номера.</p> <p>Испытайте удовольствие от проживания в этом отеле.</p> <p>Постояльцы могут прогуляться в саду. Отель открывает туристам легкий доступ к целому ряду популярных достопримечательностей: Мост влюбленных, Парк скульптур в Айя-Напе, Пляж Пантакху, Церковь Святого Епифания.</p>	<p>"Grecian Sands Hotel" - это современный, комфортабельный и гостеприимный отель, принадлежащий гостиничной сети Grecian Hotel Group. Пятиэтажный отель уютно расположился на прекрасном белоснежном берегу Айя-Напы, на территории огромного ухоженного частного парка с тропической растительностью. Grecian Sands Hotel, как и все отели Grecian Hotel Group, предлагает современные условия размещения, прекрасный сервис и множество предложений для проведения досуга. ...</p>

Рисунок 9 – Співставлення генерованих описів із раніше написаними редактором

ВИСНОВКИ. В роботі розглянуто основні методи, підходи та практичні реалізації задачі генерації описів. Зокрема розглянуто рішення компанії MeanoTek, які пропонують генерувати текст за правилами формальної граматики і нейронних мереж. Описано також метод ланцюгів Маркова, як інструменту поширення варіативності текстів.

Розроблено підхід, що поєднує механізм генерації описів по шаблонах, та ланцюг Маркова для вибору шаблонів для розширення варіативності описів.

Наведено опис етапу підготовки вхідних змінних для генерації опису готелю. Представлено перелік всіх даних, які можуть бути використані в процесі генерації.

На основі пропонованого підходу розроблено алгоритм генерації опису готелю, який використовує як шаблони так і ланцюг Маркова. Опис готелю подано у вигляді графа, основною вершиною якого є назва готелю. Наведено умови переходу до гілок та приклад розрахунку ймовірностей переходу до шаблону. Показано блок-схему роботи алгоритму генерації опису готелю.

Описана програмна реалізація підсистеми генерування описів готелів на прикладі системи пошуку готелів "Hotelsbroker". Наведені приклади результатів роботи генератора описів готелів.

ЛІТЕРАТУРА

1. Марков А. А. Распространение закона больших чисел на величины, зависящие друг от друга, *Известия физико-математического общества при Казанском университете*. 2-я серия. Том 15. 1996. С. 135–156.
2. Olle Haggstrom, *Finite Markov Chains and Algorithmic Applications*, Cambridge University Press, 2002, 126 p.
3. Tarasov D. S. Natural language generation, paraphrasing and summarization of user reviews with recurrent neural networks, *Computational linguistics and Intellectual Technologies, Paper from the Annual International Conference "Dialogue"*(2015), Issue 14, Vol. 1. 2015, P. 595–603.
4. Тарасов Д. С. Нейросетевая модель для решения задачи ответа на вопросы пользователя в произвольной предметной области, XVIII «Нейроинформатика-2016». Сб.н.т. М.: НИЯУ МИФИ. 2016. Т.3. С. 205–212.
5. Шаламов М. Генерация литературных текстов. Выступление на on-line конференции MoscowPython Meetup 28, 2015. [Електронний ресурс] - Режим доступа до ресурсу: <http://www.moscowpython.ru/meetup/28/literate-text-gen>.
6. Крепич С. Я., Литвинчук М. В. Подсистема автоматического генерирования описов готелей для системы поиска готелей «Hotelsbroker». Сучасні комп'ютерні інформаційні технології: Матеріали Всеукраїнської конференції з міжнародною участю АСІТ'2017. Тернопіль: ТНЕУ, 2017. С. 151–152.
7. К. Дж. Дейт, Введение в системы баз данных. 7-е изд. М.: Вильямс, 2001. 1072 с.
8. Даг Хеллман, Стандартная библиотека Python 3. Справочник с примерами. 2-е изд., 2018. 1376 с.
9. Словник OpenCorpora [Електронний ресурс] – Режим доступа до ресурсу: <http://opencorpora.org/>
10. Krepych S.Ya, Spivak I.Ya, Algorithm of Automatic Generation of Hotel Descriptions Using Templates Based on Markov Chains, 2018 International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PICST - 2018) Kharkiv, 2018. P. 187–200.

SOFTWARE SYSTEM OF AUTOMATIC GENERATION OF THE HOTELS DESCRIPTION USING THE TEMPLATES ON THE BASIS OF MARKOV CHAINS

S. Krepych

Ternopil National Economic University

vul. Lvivska, 11, Ternopil, 46000, Ukraine, E-mail: s.krepych@tneu.edu.ua

Purpose. The work is devoted to the problem of information presence of hotels with which the tour agency co-operates. In a significant part of the hotels, there is no text description for presentation in the profile of the hotel. But there are sets of structured data that characterize the hotel and using which you can form a text description of the hotel. The actual task is to create an automated system for describing hotels through a large volume of work that is unprofitable either economically or in time in case it is performed by a person or even a group of people. The problem can be solved in two ways. The first one is the person (editor) writes all the descriptions of the hotels that are missing. The second one is to automate the task with a computer program. **Methodology.** Nowadays such tasks are solved by means of many methods, in particular, methods of artificial intelligence, Markov chains methods, etc. However, the main disadvantage of all such methods is that they cannot perform tasks with the desired level of quality of the generated description. To improve the quality of the generated text proposed to use a modified algorithm for generating a description of the object based on templates using the Markov chains. **Findings.** The stage of preparation of input variables of the hotel description generator is presented. A list of all data that can be used during the generation process is presented. An algorithm for generating hotel descriptions based on templates using the Markov chains was developed. The hotel description is presented in the form of a graph whose main peak is the name of the hotel. The conditions for switching to the branches and the probability of transition to the template are given. The block diagram of the algorithm of hotel description generation is shown. **Originality.** Within the work, for the first time, an algorithm for the automatic generation of a description of an object for a set of data of its characteristics is developed on the basis of templates using the method of the Markov chains. **Practical value.** A subsystem is developed which can automatically prepare input data for the object from all available providers and generate a text description of this object according to the characteristics available to it using templates. As a result, the uniqueness of the source text is increased

due to the splitting of text templates into high order links in the Markov chain.. **Conclusions.** The article describes the structure of the description of hotels for a better understanding of the specifics of their work. It is shown the expediency of breaking the description into blocks and briefly given examples of their description. The approaches and methods for automation of generation of descriptions are given. In particular, the solutions of the companies MeanoTek and Rambler Travel, as well as the method of the Markov chains. Proved the expediency of using templates in the generation of descriptions and suggested to use Markov chains to increase the variability of descriptions. An algorithm for generating hotel descriptions based on templates using the Markov chains was developed. Using the block diagram the algorithm of the generator is shown. It's described the program implementation of the subsystem of generating hotel descriptions on the example of the hotel search system "Hotelsbroker".

Key words: template, Markov chain, description generation, algorithm, block diagram, program.

REFERENCES

1. Markov, A. A. (1996), "Propagation of the law of large numbers by values dependent on each other", *Yzvestiya fizyko-matematicheskoho obshchestva pry Kazanskoy unyversytete*, no. 15, pp. 135–156.
2. Haggstrom, O. (2002), *Finite Markov Chains and Algorithmic Applications*, Great Britain: Cambridge University Press, 126 p.
3. Tarasov, D. S. (2015), Natural language generation, paraphrasing and summarization of user reviews with recurrent neural networks, *Computational linguistics and Intellectual Technologies: paper from the Annual International Conference "Dialogue"*, Vol. 1, no. 14, p. 595-603.
4. Tarasov, D. S. (2016), "A non-grid model for solving the problem of answering user questions in an arbitrary domain", *XVIII «Neyroinformatika-2016»*, no. 3, pp. 205-212.
5. Shalamov, M. (2015), Generation of literary texts, *On-line conference MoscowPython Meet*, URL: <http://www.moscowpython.ru/meetup/28/literate-text-gen>. (date of reference: December, 1, 2018)
6. Krepych, S. Ya., Litvinchuk, M. V. (2017), "Subsystem of automatic generation of descriptions of hotels for the search system of hotels "Hotelsbroker"", *Suchasni kompiuterni informatsiini tekhnologii: Materialy Vseukrainskoi konferentsii z mizhnarodnoi uchastiu* [Modern Computer Information Technologies: Materials of the All-Ukrainian Conference with International Participation of ASIT'2017], Ternopil, May 19, 2017. pp. 151-152
7. Dait, K. J. (2001), *Vvedeniye v sistemy baz dannykh. 7-e izd.* [Introduction to Database Systems. 7th ed], Williams, Moscow, 1072 p.
8. Hellman, D. (2018), *Standartnaya biblioteka Python 3. Spravochnik s primerami. 2-e izd.* [Standard Python Library 3. Reference book with examples. 2nd ed.], Williams, Moscow, 1376 p.
9. Dictionary OpenCorpora URL: <http://opencorpora.org/> (date of reference: December, 1, 2018)
10. Krepych, S. Ya, Spivak, I. Ya. (2018), Algorithm of Automatic Generation of Hotel Descriptions Using Templates Based on Markov Chains, *International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PICST - 2018)*, (Kharkiv, 10 October 2018), p. 187-200.

Стаття надійшла 12.11.2018.