УДК 519.688

Геренко О.А. (ОНУ імені І.І. Мечникова)
к.ф.-м.н. **Шпінарева І.М.** (ОНУ імені І.І. Мечникова)
Морозова К.Ю. (ОНУ імені І.І. Мечникова)

ПИТАЛЬНО-ВІДПОВІДНА ДОВІДКОВА СИСТЕМА З ПІДТРИМКОЮ ГОЛОСОВОЇ ФУНКЦІЇ

У статті описується модель довідкової системи по туристичним місцям Одеси з мовним користувацьким інтерфейсом. Для розпізнавання мови використовується Google speech recognition API, сервіс розпізнавання мови створений корпорацією Google. Розглядаються автоматична класифікація коротких текстів, використовується наївний Байєсівський класифікатор, нечіткий пошук зразка в рядку. Для навчання Байесовского класификатора розроблено чотири класи питань: "Person", "Location, "Date", "Definition. Для формування відповіді використовується пошук по подібності ключа відповіді в рядку питання. Міру кореляції між пошуковим запитом і вихідним документом-відповіддю визначають за допомогою метрики Дамерау – Левенштейна. Розглядається перетворення звукових файлів в текстовий, а також синтез мови. Наведено відомості, необхідні для створення окремих підсистем, що відповідають за аналіз текстів на природній мові. Робочою мовою програмування в цьому випадку є C#.

Ключові слова: довідкова система, розпізнавання мови, аналіз тексту, Байєсівський класифікатор, Google speech recognition API, метрика Дамерау – Левенштейна.

Вступ. Ідея віртуальних асистентів не є новою. Ще в далекому 1997 році компанія Microsoft випустила свій черговий текстовий процесор Word 97, у якому з'явилася відома Скріпка. З тих пір створено безліч різних віртуальних помічників, особливу популярність

вони знайшли серед мобільних платформ. Найбільш видатними рішеннями є: Google Now для Android, Siri для iOS і майбутня Cortana для Windows Phone. Кожний розв'язок являє собою кишенькового порадника із вбудованою питально-відповідною системою, що підтримує введення природньою мовою [1].

Створення інтерфейсів, що підтримують і пропонують більш природні форми діалогу між користувачами і комп'ютерною технікою, рухається і прискорюється впровадженням інформаційних технологій у слід зростаючим потребам професійної та повсякденної діяльності людини. Однієї із природніх форм взаємодії для людини є мова. Голосовий інтерфейс може поліпшити існуючий призначений для користувача інтерфейс – він забезпечує більш зручний і менш обмежений спосіб взаємодії людини з комп'ютером. Сучасність теми дослідження визначається тим, що ринок інтелектуальних асистентів стрімко розвивається, охоплюючи практично всі сфери нашого життя. Світовий ринок інтелектуальних асистентів з 2012 року по 2014 рік виріс із $352 млн до $572,2 млн. ДО 2020 року очікується ріст ринку до $3,07 млрд, що складе 31% у порівнянні з ростом в 2013 році. Згідно зі звітом Transparency Market Research, найбільшою у світі виявилася частка північноамериканського ринку – 39%. З 2014 по 2022 рік, за прогнозами, найбільш швидкозростаючим стане азіатсько-тихоокеанський регіон – 33,4%.

Виклад основного матеріалу. Метою роботи є розробка програми, яка дозволяє користувачеві одержати відповідь на задане питання за допомогою голосового інтерфейсу із застосуванням технології розпізнавання мови, а також методів машинного навчання для розв'язку задачі класифікації даних. Розглянута предметна область – туристичний довідник міста Одеса.

Застосування інтелектуального асистента дозволить заощадити час людини на введення вручну запиту і пошук необхідної інформації наданої інформаційними системами.

Переваги використання голосового керування:

1) Спрощує керування програмою.
2) Ефективність, універсальність і швидкість.
3) Легко використовувати всім категоріям громадян (люди похилого віку, інваліди)

Недоліки використання голосового керування: неможливо використовувати при відсутності підключення до мережі Інтернет. Цей недолік усувається шляхом можливості використання ручного введення інформації (введення за допомогою клавіатури).

У даній роботі представлена інтерактивна довідкова система з російськомовним інтерфейсом голосового введення. Іншими словами, створений алгоритм дозволяє користувачеві вести діалог по моделі « питання-відповідь».

Основними модулями системи є:модуль розпізнавання мови, модуль класифікації тексту,модуль формування відповіді, модуль синтезу мови.

Модуль розпізнавання мови. Для розпізнавання мови використовується Google speech recognition API, сервіс розпізнавання мови створений корпорацією Google. Для розпізнавання мови використовується запит по протоколу HTTP з методом POST. За допомогою бібліотеки Naudio мова користувача записується у формат wav. Так як сервіс розпізнавання мови не працює з форматом wav, файл переводиться у формат flac за допомогою бібліотеки Cuetools. Він повинен мати частоту не більше 16 Кгц і необхідно використовувати моно канал. Записаний файл у форматі flac відправляється на розпізнавання в Google speech API. При отримання відповіді від сервісу Google speech API відбувається парсинг рядка відповіді і висновок результату на екран користувача.

Модуль класифікації тексту. Задача класифікації є закритою, тобто кожне питання відноситься як мінімум до одного тематичного класу (персона-особистість, визначення, часовий період або локація). Для формування описів класів є навчальна вибірка простих

запитань, співвіднесених з класами. Для оцінки ступеня приналежності питання тематичному класу використовується наївний Байєсівський класифікатор.

Для використання як класифікатора навчальний набір документів необхідно перетворити в зручний формат. Був використаний наступний підхід: відсіваються всі символи, які не є буквами, наприклад, розділові знаки пунктуації; віддаляються стоп-слова; проводиться стемінг.

Стоп-слова – це слова, які не несуть будь-якої самостійної смислового навантаження. До стоп-слів належать прийменники, сполучники і займенники.

Стемінг – відкидання змінюваних частин слів, головним чином, закінчень. Ця технологія більш проста, не вимагає зберігання словника слів або великого набору правил. Технологія заснована на правилах морфології мови. Недолік стемінга – це велика кількість помилок. Стемінг добре підходить для англійської мови, але гірше для російської. У роботі використаний стемінг Портеру.

Наївний Байєсівський класифікатор один з найпростіших з алгоритмів класифікації. Проте, дуже часто він працює не гірше, а то і краще більш складних алгоритмів. [1]

Байєсівський класифікатор. Наївний Байєсівський класифікатор є хорошим прикладом імовірнісного класифікатора. Цей класифікатор заснований на наступних припущеннях: усі документи мають однакову довжину; імовірності зустріти слова в документі незалежні.

На практиці обидва ці припущення не виконуються, проте, метод добре працює в деяких областях.

Навчання класифікатора. У задачі класифікації текстів метод Байєса застосовується окремо для кожної категорії і приймається рішення, належить документ категорії чи ні. Апостеріорна ймовірність приналежності запиту класу обчислюється по формулі Байєса, що зв'язує апріорну ймовірність з апостеріорної.

Нехай ймовірність зустріти слово в класі C дорівнює $P(t_i|C)$ тоді ймовірність зустріти документ D в цьому класі дорівнює

$$P(D|C) = \prod_i P(t_i|C) = \frac{P(D \cap C)}{P(C)}, \quad (1)$$

звідки

$$P = \frac{P(D \cap C)}{P(D)} = \frac{P(C)}{P(D)} P = \frac{P(C)}{P(D)} \prod_i P(t_i|C) = \frac{1}{P(d)} e^{\sum_i \ln\left(P(t_i|C)\right) + \ln(P(C))}, \quad (2)$$

реалізація припускає що ln(0)=0.

Якщо подивитися на цю формулу з іншої сторони (де i=0, якщо слова немає в документі, i=1, якщо є), то можна помітити, що коефіцієнти утворюють гіперплощину в просторі слів, по одну сторону від якої лежать слова приналежні документу, по іншу не приналежні. Таких гіперплощин існує нескінченно багато, і наївний байєсівський метод один з варіантів вибору цих коефіцієнтів. У роботі для навчання класифікатора використовуються чотири класи питань:

1)“Person” – до них ставляться власні імена, з якими можна співвіднести питальні слова – «хто» «ким»;

2) “Location” – до них ставляться слова, які визначають локацію об'єкта;

3) “Date” – тимчасові періоди походження подій;

4) “Definition” – до них ставляться невласні імена, з якими можна співвіднести питальні слова – «що» «чому».

Приклади.

«Кто такой Георгий Маразли?» Person

«Где находится отель Бристоль?» Location

«Кем был Прохаска?» Person
«Что такое фонтан?» Definition
«Когда была построена Потемкинская лестница?» Date

Модуль формування відповіді. Для формування відповіді використовується пошук по подібності ключа відповіді в рядку питання. Пошук по подібності може бути дуже корисний у системах розпізнавання тексту, де існує певна ймовірність неточного завдання терміну в пошуковому запиті. Міру кореляції між пошуковим запитом і вихідним документом звичайно визначають у вигляді відстані редагування, найбільш просте і широко-розповсюджене визначення відстані редагування задається функцією Левенштейна. Функція Левенштейна дорівнює мінімальній кількості операцій редагування, таких як вставка, заміна, видалення, необхідних для перетворення слова w у слово v.

У справжній роботі використовується метрика Дамерау – Левенштейна [3]. Відстань Дамерау − Левенштейна − це міра різниці двох рядків символів, обумовлена як мінімальна кількість операцій вставки, видалення, заміни та транспозиції (перестановки двох сусідніх символів), необхідних для перекладу одному рядка в іншу. Є модифікацією відстані Левенштейна: до операцій вставки, видалення і заміни символів, визначених у відстані Левенштейна додана операція транспозиції (перестановки) символів.

Після аналізу питання кожному слову в пропозиції зіставлена його початкова форма. Далі складається список усіх фрагментів, що підходять на роль відповіді, зіставлених ключу та відбувається пошук кандидатів для видачі відповіді. При витягу відповіді використовуються нечіткий пошук підходящого ключа в початковій формі питання. Списки відповідей із ключами складаються вручну. На основі такого алгоритму можна перефразувати питання.

Приклад: «Скажите, пожалуйста, адрес Бистоля?» та «Как проехать к гостинице Бристоль?» - Відповідь «Пушкинская,15» (рис. 1).

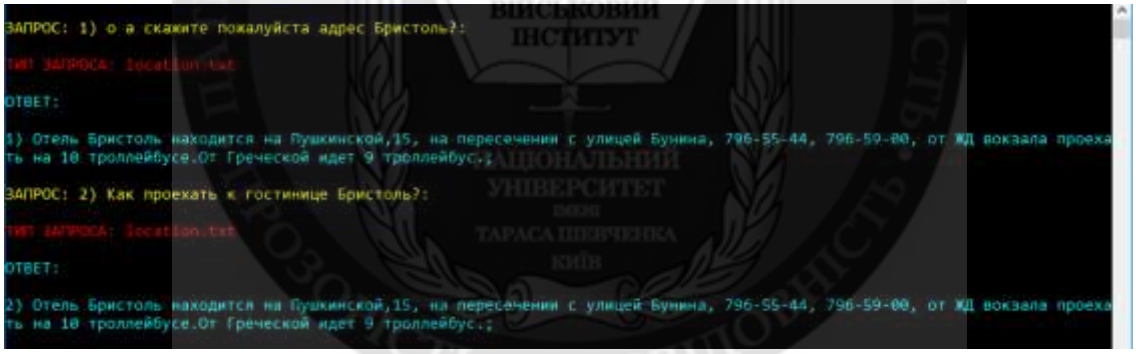

Рис. 1. Приклад можливості перефразувати питання

Оцінка якості. Точність обчислюється як відношення числа правильних позитивних прогнозів класифікатора (коли документу була присвоєна будь-яка категорія) до числа загальної кількості позитивних прогнозів. Повнота це відношення числа правильних позитивних прогнозів до числа документів, яким повинна була бути присвоєна категорія. У класифікаторів з високою точністю, зазвичай, низька повнота і навпаки. Для кожного алгоритму класифікації, як правило, є можливість управляти співвідношенням точність-повнота [4].

Нехай контрольна вибірка складається з M об'єктів. З них m класифікуються правильно. Тоді частка правильних відповідей A (від англ. Accuracy) обчислюється за формулою: A = m / M. Це одна з найпростіших оцінок, і вона не завжди відображає реальну

якість. Нехай безліч документів розбито по категоріях. Позначимо v – безліч документів, що належать класу, u – безліч документів, приписаних класу алгоритмом класифікації.

Повнота r (від англ. Recall) класифікації документів по класу обчислюється як відношення кількості документів, що належать до класу, до загальної кількості документів, що відносяться до даного класу: $r(u) = |u \cap v| \, |v|$

Точність p (від англ. Precision) класифікації документів по класу обчислюється як відношення кількості документів, правильно приписаних до класу, до загальної кількості документів, приписаних до даного класу: $p(u) = |u \cap v| \, |u|$. F-міра (F-measure) об'єднує оцінки точності і повноти в одну: $F(u) = 2 * p(u) * r(u) \, p(u) + r(u)$. Якщо p(u)=0 або r(u)=0, то тоді і $F(u) = 0$.

На рис. 2 відображені результаті тестування модулів класифікації тексту формування відповіді.

Рис. 2. Приклад тестування питально-відповідної довідкової системи

Оцінка якості класифікації в даній роботі 80%.

Деякі результати застосування системи показані на рис 3.

Модуль синтезу мови. Для завдання синтезу мови використовується бібліотека System.Speech з підключеним «голосом», який можна знайти у вільному доступі в мережі Інтернет. У процесі роботи над системою, був проведений ряд операцій по зміні тональності і тембру використовуваного голосу, для того, щоб зробити його більш підходящим для нашого робота.

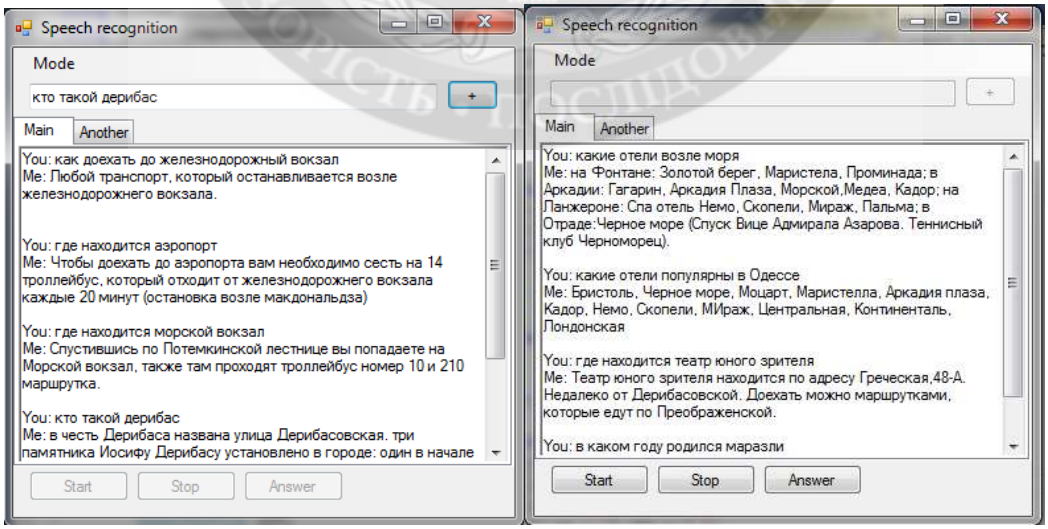

Рис. 3. Робота питально-відповідної довідкової системи

Висновки В результаті реалізації описаної в роботі моделі отримано додаток, що дозволяє задавати питання на природній мові, отримувати адекватні відповіді в рамках запропонованого підходу і предметної області. Оцінка якості класифікації в даній роботі 80%. З 100 по-різному поставлених питань отримано 84 правильні відповіді.

ЛІТЕРАТУРА:

1. Лапшин В.А. Вопросно-ответные системы: развитие и перспективы /В.А. Лапшин. – М.: 2012. – 32 с.
2. Никитин А., Райков П. Вопросно-ответные системы [Електронний ресурс] – Режим доступу: – URL: http://yury.name/internet/06ia-seminar.ppt
3. Нурбекова Ж.К., Даутова А.З., Кашкинбаева Д.Б. Технология проектирования мультимедийных обучающих систем/ К.Ж. Нурбеков, А.З. Даутова, Д.Б. Кашкинбаева – Павлодар,2003. –108 с.
4. Звенигородский А.С., Иванова С.Б., Чернышова В.Н. Модель одного ответа на вопрос в естественно-языковых системах тестирования/ А.С. Звенигородский, С.Б. Иванова, В.Н. Чернышова – "Искусственный интеллект – 2’2012".

REFERENCES:

1. Lapshin V.A. Question-answer system: development and prospects / V.A. Lapshin. - M .: 2012. – 32 s.
2. A. Nikitin, P. Raykov Question-answer system [Electron resource] - Access mode: - URL: http://yury.name/internet/06ia-seminar.ppt
3. Nurbekova JK, Dautova AZ, Kashkinbaeva DB Technology design of multimedia learning systems / KJ Nurbekov, AZ Dautova, DB Kashkinbaeva - Pavlodar, 2003-108 with.
4. Zvenigorodsky AS, SB Ivanov, VN Chernyshov Model one answer to a question in natural language testing systems / AS Zvenigorod, SB Ivanov, VN Chernyshov - "Artificial Intelligence - 2'2012"

Рецензент: д.т.н., доц. Гунченко Ю.О., професор кафедри математичного забезпечення комп’ютерних систем, Одеський національний університет імені І.І. Мечникова

Геренко О.А., к.ф.-м.н. Шпинарева И.М., Морозова К.Ю.

ВОПРОСНО-ОТВЕТНАЯ СПРАВОЧНАЯ СИСТЕМА С ПОДДЕРЖКОЙ ГОЛОСОВОЙ ФУНКЦИИ

В статье описывается модель справочной системы по туристическим местам Одессы с речевым пользовательским интерфейсом. Для распознавания речи используется Google speech recognition API, сервис распознавания речи созданный корпорацией Google. Рассматриваются автоматическая классификация коротких текстов, используется наивный Байесовский классификатор, нечеткий поиск образца в строке. Для обучения Байесовского класификатора разработаны четыре класса вопросов: “Person”, “Location, “Date”, “Definition. Для формирования ответа используется поиск по сходству ключа ответа в строке вопроса. Меру корреляции между поисковым запросом и исходным документом-ответом определяют с помощью метрики Дамерау – Левенштейна. Рассматривается преобразования звуковых файлов в текстовые, а также синтез речи. Приведены сведения, необходимые для создания отдельных подсистем, отвечающих за анализ текстов на естественном языке. Рабочим языком программирования в этом случае является C #.

Ключевые слова: справочная система, распознавание речи, анализ текста, Байесовский классификатор Google speech recognition API, метрика Дамерау– Левенштейна.

Gerenko O.A., Ph.D. Shpinareva I.M., Morozova K.Y.

QUESTION-RELEVANT BACKGROUND SYSTEMS WITH SUPPORT VOICE FEATURES

The article describes the reference model system for tourist places of Odessa with a voice user interface. For speech recognition uses Google speech recognition API, speech recognition service created by Google.The article deals with the automatic classification of short texts, using naive Bayesian classifier, fuzzy search pattern in a text string. Four classes of questions are designed to train the Bayesian hierarchical classifier: "Person", "Location"," Date "," Definition". To generate the response using similarity search response key in the line of question. The measure of correlation between the search request and the source document-answer is determined by metric Damerau - Lowenstein. Considered converting audio files into text and speech synthesis. The data needed to create individual subsystems that are responsible for the analysis of texts in natural language. The working language programming in this case is C #.

Keywords: information system, speech recognition, text analysis, Bayesian classifier Google speech recognition API, metric Damerau- Lowenstein.