

проблеми і дослідження: Міжнародний збірник наукових робіт. – Вип. 27. – Донецьк.: Фірма ТЕАН, 2007. – С. 132 – 139.

10. Гриб'юк О.О. Психолого-педагогічні вимоги до комп'ютерно-орієнтованих систем навчання математики в контексті підвищення якості освіти// Гуманітарний вісник ДВНЗ «Переяслав-Хмельницький державний педагогічний університет імені Григорія Сковороди» - Додаток 1 до Вип.31, Том IV (46): Тематичний випуск «Вища освіта України у контексті інтеграції до європейського освітнього простору». – Київ: Гнозис, 2013. – С. 110-123.

11. Гриб'юк О.О. Математичне моделювання при навчанні дисциплін математичного та хіміко-біологічного циклів: навчально-методичний посібник для учителів / О.О. Гриб'юк. – Рівне: РДГУ, 2010. – 207 с.

12. Гриб'юк О.О. Вплив інформаційно-комунікаційних технологій на психофізіологічний розвиток молодого покоління. “Science”, the European Association of pedagogues and psychologists. International scientific-practical conference of teachers and psychologists “Science of future”: materials of proceedings of the International Scientific and Practical Congress. Prague (Czech Republic), the 5th of March, 2014/ Publishing Center of the European Association of pedagogues and psychologists “Science”, Prague, 2014, Vol.1. 276 p. - S. 190-207.

13. Hrybiuk O. Mathematical modeling as a means and method of problem solving in teaching subjects of branches of mathematics, biology and chemistry // Proceedings of the First International conference on Eurasian scientific development. «East West» Association for Advanced Studies and Higher Education GmbH. Vienna. 2014. P. 46-53.

14. Рубинштейн С.Л. Основы общей психологии: В 2-х т., Т. II. – М.: Педагогика, 1989. – С.176.

15. Цукерман Г.А. Оценка без отметки / Г.А. Цукерман. – Москва–Рига: Педагогический центр «Эксперимент», 1999. – 137 с.

Проектно-исследовательская деятельность в процессе обучения математике учащихся общеобразовательного учебного заведения

Гриб'юк Е.А.

Аннотация: Анализируется проектно-исследовательская деятельность в процессе обучения математике с использованием отдельных компонентов компьютерно ориентированной среды обучения, обеспечивается концентрация учебных ресурсов, многогранность индивидуальных траекторий развития личности ученика и результатов формирования необходимых межпредметных и метапредметных компетентностей; доступность и равенство возможностей учащихся в обучении; полифункциональность взаимодействия субъектов учебного процесса; ориентацию содержания, форм и технологий подготовки учащихся на интеграцию образовательную, научную, исследовательскую, производственную в условиях учебно-воспитательного процесса. Рассматриваются пути построения вариативных моделей обучения с целью повышения эффективности обучения учащихся.

Ключевые слова: вариативные модели, моделирование, компьютерно ориентированная среда обучения, проектирования, предметы естественно-математического цикла, проектно-исследовательская деятельность, математика.

Project and research activity in the process of teaching mathematics students of secondary educational institutions

Hryb'yuk O.O.

Annotation. Analyzes design and research activities in learning mathematics with the use of the individual components of computer oriented learning environment, a concentration of learning resources, the diversity of individual trajectories of development of the personality of disciple and of the results of the formation of the necessary interdisciplinary competences and metapragmatic; accessibility and equality of opportunity of students in training; polyfunctionality of interaction of subjects of educational process; the orientation of the content, forms and technologies of training students to integrate educational, scientific, research, production and in the conditions of the educational process. Discusses ways to build variable models for learning to enhance learning effectiveness of students.

Key words: variable models, modeling, computer oriented learning environment, design, science, computer oriented learning environment design, design research, math.

УДК 378.147.091.3:004.4

Резіна О. В.

Кіровоградський державний педагогічний університет імені Володимира Винниченка

Технології статистичного опрацювання текстів та методика їх навчання

Анотація. У статті розглядаються особливості методики навчання технологій статистичного опрацювання текстових даних у процесі підготовки майбутніх фахівців з прикладної лінгвістики. Описані базові операції кількісного опрацювання текстів та запропоновано можливий підхід до їх реалізації з використанням мови програмування Python. Обґрунтовано доцільність використання мови Python та ресурсів платформи NLTK (Natural Language Toolkit) для виконання завдань

комп'ютерної лінгвістики. Розглядається задача опрацювання текстового файлу з метою обчислення деяких кількісних параметрів та наводяться відповідні програмні коди.

Ключові слова: статистичне опрацювання тексту, мова програмування Python, методика навчання.

З часів винайдення друкарського верстату людством розроблялися і вдосконалювалися технології швидкого розповсюдження друкованого слова. Історія розвитку систем опрацювання текстів – це історія поступової автоматизації процесів написання та редагування текстових даних, а також удосконалення відповідних технологій з метою зробити їх доступними для індивідуальних та корпоративних користувачів.

У 1964 році фірма IBM представила систему під назвою MT/ST (Magnetic Tape/Selectric Typewriter), в якій об'єднувалися функції друкарської машинки та пристроєм зберігання даних. Текст записувався із друкарської машинки на касету з магнітною стрічкою, яка була першим носієм даних багаторазового використання. Вперше з'явилася можливість редагувати текст без необхідності повного його передруковування. Дані на стрічці зберігалися, відтворювалися, редагувалися, роздруковувалися необхідну кількість разів, видалялися, і в разі видалення звільнене місце на магнітній стрічці використовувалося для інших текстів. Ця подія ознаменувала початок опрацювання текстових даних у сучасному розумінні цього поняття [1].

Розвиток інформаційно-комунікаційних технологій обумовив значне збільшення кількості, різноманіття та складності лінгвістичних даних, поданих у цифровому форматі і доступних через інтернет [2]. Сьогодні сховища оцифрованих текстових даних є невід'ємною складовою різноманітних досліджень у сфері гуманітарних наук, таких як бібліотекознавство, літературознавство, комп'ютерна та корпусна лінгвістика [3]. Одним із напрямів цих досліджень є статистичний аналіз тексту, застосування якого дає змогу реалізувати такі функції:

- здійснення контролю орфографії тексту;
- автоматичне виокремлення ключових слів та створення тезаурусів;
- створення частотних словників та визначення авторства тексту;
- обчислення лексичної щільності (насиченості) тексту та визначення авторського стилю;
- розшифрування тексту за методом частотного аналізу;
- здійснення фільтрації спаму під час користування електронною поштою;
- визначення особливостей та успішності рекламних кампаній;
- аналіз змісту відомостей у соціальних мережах з метою запобігання розповсюдження небезпечних відомостей;
- аналіз змісту відомостей, що продукуються через засоби масового інформування, та визначення ступеню їх впливу на суспільство [4, 5, 6, 7, 8, 9].

Реалізація зазначених функцій можлива за умови кількісного підходу, який відіграє допоміжну та підготовчу роль для подальших якісних досліджень. Базовими операціями кількісного опрацювання тексту є:

- обчислення загальної кількості символів, складів, слів, речень, абзаців;
- визначення середньої довжини слів та речень;
- визначення кількості частин мови (іменників, прикметників, дієслів, займенників тощо);
- визначення кількості односкладових та багатоскладових слів у тексті тощо.

Уміння здійснювати базові операції статистичного опрацювання тексту набувають більшої актуальності й у зв'язку з тим, що останнім часом усе більше людей роблять дописи на різноманітних веб-сайтах та у соціальних мережах. Оскільки текст, поданий в режимі он-лайн, потребує певних пропускових характеристик, на більшості з цих сайтів обмежується кількість символів, що можуть бути опубліковані. Наприклад, обмеження на кількість символів у Twitter складає 140, тобто неможливо опублікувати повідомлення, що містить більше, ніж 140 символів [10]. Блогери ведуть систематичний облік статистичних даних для своїх текстових файлів: кількості рядків, слів, символів тощо [11].

З огляду на вищесказане можна зазначити, що питання, пов'язані із кількісним опрацюванням тексту, є важливими та практично значущими для майбутніх фахівців з прикладної лінгвістики. Такі питання доцільно включити до навчальної програми дисципліни «Програмування» (або спорідненої спеціальності «Прикладна лінгвістика»). У статті розглядається один із методичних підходів до вивчення вказаних питань.

Засобом описування алгоритмів опрацювання текстових даних пропонується обрати мову програмування Python (версія 3.x і вище). Сьогодні мова Python визнана фахівцями одним із найкращих інструментів опрацювання текстів [12, 13]. Це потужна мова з простими синтаксисом і семантикою та об'ємною бібліотекою функцій опрацювання рядків. Доцільно звернути увагу на той факт, що провідна платформа NLTK (Natural Language Toolkit), призначення якої – робота з даними, поданими природною мовою, орієнтована на підтримку програм, описаних з використанням засобів мови Python (<http://www.nltk.org/>). В NLTK міститься інструментарій для виконання більшості

завдань комп'ютерної лінгвістики. Важливо, що цей комплект програм доступний користувачам на правах GNU (General Public License).

Розглянемо задачу написання програми, для якої

- вхідні дані: текстовий файл (у форматі .txt), в якому міститься текст для опрацювання;
- вихідні дані: текстовий файл, в якому містяться такі кількісні характеристики вхідного тексту – 1) кількість символів з пробілами, 2) кількість символів без пробілів, 3) кількість речень, 4) кількість абзаців, 5) загальна кількість слів, 6) кількість значущих слів, 7) лексична щільність.

Для розв'язування цієї задачі раціональним є створення двох модулів, що під'єднуюватимуться до основної програми. В першому модулі повинен міститися список стоп-слів. Доцільно використати список стоп-слів, запропонований на платформі NLTK [14]. Ім'я модуля – `nltk_stopwords`, його програмний код наведено нижче.

```
1 #from nltk.corpus import stopwords ('english')
2 #http://www.nltk.org/book/ch02.html
3
4 def stopwords():
5
6     stopwords=['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves',
7     'you', 'your', 'yours', 'yourself', 'yourselves', 'he', 'him',
8     'his', 'himself', 'she', 'her', 'hers', 'herself', 'it', 'its',
9     'itself', 'they', 'them', 'their', 'theirs', 'themselves',
10    'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those',
11    'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have',
12    'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an',
13    'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until',
14    'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against',
15    'between', 'into', 'through', 'during', 'before', 'after',
16    'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on',
17    'off', 'over', 'under', 'again', 'further', 'then', 'once',
18    'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any',
19    'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such',
20    'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too',
21    'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now',
22    'd', 'll', 'm', 'o', 've', 're', 'y', 'th', 'ain', 'aren', 'couldn',
23    'didn', 'doesn', 'hadn', 'hasn', 'haven', 'isn', 'ma', 'mightn',
24    'mustn', 'needn', 'shan', 'shouldn', 'wasn', 'weren', 'won', 'wouldn']
25
26    return stopwords
```

Призначення другого модуля – безпосереднє обчислення статистичних параметрів тексту. В модулі міститься функція `general_statistics()`, до якої передається параметр `text` (вхідний текст). В разі зверненні функції повертається значення параметрів `General_chars` (кількість символів з пробілами), `Spaceless_chars` (кількість символів без пробілів), `Sentences` (кількість речень), `Paragraphs` (кількість абзаців), `Words` (загальна кількість слів), `Lexical_words` (кількість значущих слів), `LD` (лексична щільність). Ім'я модуля – `gen_stat`, його програмний код наведено нижче.

```
1 import string, re, nltk_stopwords
2
3 def general_statistics(text):
4
5     General_chars = len(text)
6
7     Spaces = text.count(' ')
8
9     Spaceless_chars = General_chars - Spaces
10
11    Sentences = len(re.findall(r'[!?.]', text))
12
13    List_of_newlines = re.findall(r'\n', text, re.MULTILINE)
14    List_of_emptylines = re.findall(r'^\s*$', text, re.MULTILINE)
15    Paragraphs = len(List_of_newlines) - len(List_of_emptylines)
16
17    Words = len(re.findall(r'\S+', text))
```

```

18
19 normalizedtext = re.sub(r'\W+|\d+', ' ', text)
20 normalizedtext=normalizedtext.casefold()
21
22 fullwordlist = normalizedtext.split()
23
24 stop_words_list=nltk_stopwords.stopwords()
25 wordlist = [w for w in fullwordlist if w not in stop_words_list]
26
27 Lexical_words = len(wordlist)
28
29 LD = (Lexical_words/Words)*100
30
31 return General_chars, Spaceless_chars, Sentences, Paragraphs, Words,
    Lexical_words, LD

```

Розглянемо детальніше деякі оператори. У рядку 1 програми імпортуються модулі `string`, `re`, `nltk_stopwords`. В модулі `string` містяться константи, функції та описи методів опрацювання рядків. Підєднання модуля `re` дає можливість використовувати регулярні вирази в програмі. Регулярні вирази (regular expressions) – це формальна мова пошуку і заміни рядків у тексті. Регулярний вираз є формулою, що складається з символів і метасимволів та через який задає правило пошуку [15]. Використання регулярних виразів у програмі значно підвищує ефективність процесу опрацювання тексту. Модуль `nltk_stopwords` розглянуто вище.

За вказівкою у рядку 3 програми відбувається виклик функції `general_statistics()`. Для знаходження загальної кількості символів у тексті використовується функція `len()` модуля `string`, а для знаходження кількості пробілів – метод `count` модуля `string` (рядки 5, 7). За вказівкою у рядку 11 відбувається підрахунок кількості речень у тексті. За шаблоном регулярного виразу `'[!?.]'` відбувається пошук пунктуаційних знаків, через які позначають кінець речення. Знаки перелічені у квадратних дужках.

Для підрахунку кількості абзаців необхідно від загальної їх кількості (визначається за шаблоном `'\n'`, рядок 13) відняти кількість порожніх рядків (рядок 14). Шаблон `'^\s*$'` означає, що між початком та кінцем рядка розташовано 0 і більше пробільних символів. Використання Флага `re.MULTILINE` надає змогу здійснювати пошук у рядку, в якому містить кілька підрядків.

Для побудови шаблону підрахунку загальної кількості слів враховується той факт, що слово – це набір символів, розташований між наборами пробільних символів. Запис `'\S+'` означає один і більше пробільних символів підряд (рядок 17).

Для визначення кількості значущих слів та лексичної щільності необхідно нормалізувати опрацьований текст. Нормалізація тексту – це процес застосування певних перетворень з метою отримання такого тексту, який було б зручно опрацьовувати у подальшому (наприклад, для організації пошуку, фільтрування або сортування слів). У розглядуваній задачі для нормалізації доцільно позбавитися в тексті всіх розділових знаків, апострофів та інших символів, що не є буквами або цифрами. Замінімо їх символом пробіл. Така дія в програмі виконується оператором `re.sub(r'\W+|\d+', ' ', text)` (рядок 19). Сукупність спеціальних символів `\W+` означає набір символів, що не є цифрами або буквами, а сукупність спеціальних символів `\d+` означає набір цифр. У цілому шаблон `r'\W+|\d+'` трактується як *набір символів, що не є цифрами або буквами або набір не цифр*. Використання методу `sub` дає змогу замінити в тексті всі підрядки, що співпадають з шаблоном, на пробіли. Рядок, що повертається в разі використання методу `sub`, записується до змінної `normalizedtext`.

Наприклад, якщо змінна `text` набуде значення:

“When you want to colloquially express that you don’t care at all about something, you might say “I couldn’t care less.” This phrase first popped up in British English at the turn of the 20th century and is still popular today. In the 1960s, a controversial American variant of this phrase entered popular usage: “I could care less.””

то в результаті виконання оператора `re.sub(r'\W+|\d+', ' ', text)` до змінної `normalizedtext` буде записаний текст:

“When you want to colloquially express that you don t care at all about something you might say I couldn t care less This phrase first popped up in British English at the turn of the th century and is still popular today In the s a controversial American variant of this phrase entered popular usage I could care less”.

Наступним кроком нормалізації тексту є переведення символів до нижнього регістру, що дає можливість коректного порівняння рядків. Ця дія відбувається за допомогою використання методу

`casefold()` модуля `string` (рядок 20). В разі звернення до методу `split()` нормалізований текст розщеплюється на список, позбавлення зайвих пробілів відбувається автоматично (рядок 22).

Обчислення кількості значущих слів відбувається виключенням з тексту стоп-слів. До змінної `stop_words_list` записується список стоп-слів, перелічених у модулі `nltk_stopwords`, для цього викликається функція `stopwords()` зі вказаного модуля (рядок 24). Стоп-слова відфільтровуються з тексту за допомогою послідовного перебирання в циклі `for-in` та умови `if-not-in`. Тобто, якщо певне слово є в обох списках, воно вилучається. Результат записується до змінної `wordlist` (рядок 25).

Для рядка

```
"When you want to colloquially express that you don't care at all about something you might say I couldn't care less. This phrase first popped up in British English at the turn of the 19th century and is still popular today. In the 1950s a controversial American variant of this phrase entered popular usage. I could care less."
```

список значущих слів буде таким:

```
['want', 'colloquially', 'express', 'care', 'something', 'might', 'say', 'care', 'less', 'phrase', 'first', 'popped', 'british', 'english', 'turn', 'century', 'still', 'popular', 'today', 'controversial', 'american', 'variant', 'phase', 'entered', 'popular', 'usage', 'care', 'less'].
```

Слід звернути увагу, що наприклад, набір символів `'couldn't'` буває трансформований на вибір `'couldn't'`. Кожен з цих рядків входить до списку стоп-слів і тому не включається до списку значущих, що обраховується в рядку 27 програмного коду.

За вказівкою у рядку 29 відбувається обчислення лексичної щільності. Лексична щільність (лексична насиченість, словникова різноманітність) – це величина, за якою характеризується текст з точки зору стилістичних особливостей чи мови творів різних жанрів, авторів тощо [8]. Лексична щільність є відношенням кількості різних слів до загальної кількості слів у тексті.

Далі за оператором `return` повертається шукані параметри до програми, за якою викликається `gen_stat` як модуль (рядок 31).

Завдяки використанню модулів код основної програми є коротким та зрозумілим:

```
1 import gen_stat
2
3 def main():
4     source = open("text_data/input.txt", "r")
5     stat = open("text_data/stat.txt", "w")
6
7     myText = source.read()
8
9     myGeneral_chars, mySpaceless_chars, mySentences, myParagraphs, myWords,
10    myLexical_words, myLD = gen_stat.general_statistics(myText)
11
12    stat.write("General statistics\n" + "\n")
13    stat.write('{:40}'.format("Number of characters (including spaces)") +
14    '{:7d}'.format(myGeneral_chars) + "\n")
15    stat.write('{:40}'.format("Number of characters (without spaces)") +
16    '{:7d}'.format(mySpaceless_chars) + "\n")
17    stat.write('{:40}'.format("Number_of_sentences") +
18    '{:7d}'.format(mySentences) + "\n")
19    stat.write('{:40}'.format("Number_of_paragraphs") +
20    '{:7d}'.format(myParagraphs) + "\n")
21    stat.write('{:40}'.format("Number_of_words") +
22    '{:7d}'.format(myWords) + "\n")
23    stat.write('{:40}'.format("Number_of_lexical_words") +
24    '{:7d}'.format(myLexical_words) + "\n")
25    stat.write('{:40}'.format("Lexical_density") +
26    '{:7.2f}'.format(myLD) + "%\n")
27
28    source.close()
29    stat.close()
30
31    print("The programme is completed. Check the file.")
32
33    if __name__ == '__main__':
34        main()
```

За допомогою функції `open()` відкриваються два файли: файл з іменем `"input.txt"` містить текст для опрацювання, відкривається з параметром `"r"` (read) – для читання (рядок 4); до файлу з іменем `"stat.txt"` записуються результати, відкривається з параметром `"w"` (write) – для запису (рядок 5).

У рядку 7 до змінної `myText` записується значення тексту, зчитаного з файлу `input.txt` за допомогою методу `read()`. Фактичний параметр `myText` передається до функції `general_statistics()` модуля `gen_stat`, після виконання функції повертаються значення семи параметрів у точку її виклику (рядок 9). За допомогою методу `write()` необхідні дані у певному форматі записуються до текстового файлу (рядки 11-18 програми). Далі обидва файли закриваються за допомогою методу `close()`.

Наведемо результати роботи програми для різних текстів. Наприклад, для оповідання О'Генрі «Дари волхвів» (O. Henry, «The Gift of the Magi») результати є такими:

Number of characters (including spaces) :	11166
Number of characters (without spaces) :	9156
Number_of_sentences	: 164
Number_of_paragraphs	: 44
Number_of_words	: 2055
Number_of_lexical_words	: 1010
Lexical_density	: 49.15%

Для балади Редьярда Кіплінга про схід і захід (Rudyard Kipling «The Ballad Of East And West»):

Number of characters (including spaces) :	6631
Number of characters (without spaces) :	5401
Number_of_sentences	: 47
Number_of_paragraphs	: 102
Number_of_words	: 1332
Number_of_lexical_words	: 684
Lexical_density	: 51.35%

Для статті з комп'ютерної лінгвістики «Natural Language Processing for Social Media» (http://www.mitpressjournals.org/doi/full/10.1162/COLI_r_00270#.WPSfkPmLSUI):

Number of characters (including spaces) :	10797
Number of characters (without spaces) :	9095
Number_of_sentences	: 72
Number_of_paragraphs	: 8
Number_of_words	: 1711
Number_of_lexical_words	: 992
Lexical_density	: 57.98%

Отримані дані є основою для подальших досліджень текстів. Важливо зазначити, що ці дані зберігаються у зовнішній пам'яті комп'ютера і є придатними для багаторазового опрацювання.

Таким чином тема «Технології статистичного опрацювання текстових даних» є важливою складовою системи підготовки фахівців прикладної лінгвістиком. Під час її вивчення формуються такі професійні компетентності, як обізнаність щодо, обґрунтування вибору методів і засобів для розв'язування прикладних задач в галузі комп'ютерної лінгвістики, інтерпретування отриманих результатів та інші. Перспективою подальших досліджень може бути розробка методики навчання створення частотних словників та інших методів статистичного опрацювання текстових даних.

Список використаних джерел

1. Kunde, B. A Brief History of Word Processing [Electronic resource] / B. Kunde – Mode of access: <http://web.stanford.edu/~bkunde/fb-press/articles/wdprhist.html>
2. Linguistic Data Resources on the Internet. A topically organized list of language data resources on the Internet [Electronic resource] – Mode of access: <http://www-01.sil.org/linguistics/etext.html>
3. Lehmberg, T. Digital Text Collections, Linguistic Research Data, and Mashups: Notes on the Legal Situation [Electronic resource] / T. Lehmberg, G. Rehm, A. Witt, F. Zimmermann – Mode of access: <https://muse.jhu.edu/article/252248>
4. Бісікало О.В. Метод визначення ключових слів англomовного тексту на основі DKPro Core / О.В. Бісікало, О.В. Яхимович // Технологический аудит и резервы производства. – 2015. – № 1(2). – С. 26-30.
5. Максимів О.Г. Корпус перської мови як джерело матеріалу для навчальних словників-мінімумів / О.Г Максимів // Вісник Львівського університету. Серія: Філологічна. – Вип. 45. – Львів, 2008. – С. 164-169.
6. Jarman, J. Combining Natural Language Processing and Statistical Text Mining: A Study of Specialized Versus Common Languages. Graduate Theses and Dissertations / J. Jarman – University of South Florida, 2011. – 156 p. – Mode of access: <http://scholarcommons.usf.edu/etd/3166/>

7. Jensen, K. Linguistics and the digital humanities: (computational) corpus linguistics [Electronic resource] / K. Jensen – Mode of access: <http://ojs.statsbiblioteket.dk/index.php/mediekultur/article/viewFile/15968/17442>
8. Левус Є.О Алгоритм відображення зміни лексичної насиченості тексту / Є.О. Левус, С.Р. Бук, Є.Д. Яворський // Вісник Національного університету "Львівська політехніка". Комп'ютерні науки та інформаційні технології. – 2013. – № 771. – С. 349-353.
9. Soriano, J. Text Mining in Computational Advertising [Electronic resource] / J. Soriano, D. Banks – Mode of access: https://s3.amazonaws.com/maxpoint-main/2015/01/29/09/15/29/42/Whitepaper_TextMining.pdf
10. Why you need to count letters? [Electronic resource] – Mode of access: <http://techwelkin.com/tools/letter-count-character-count/>
11. Calculating Text File Statistics [Electronic resource] – Mode of access: <https://technet.microsoft.com/en-us/library/ee692800.aspx>
12. Mertz, D. Text Processing in Python [Electronic resource] / D. Mertz – Mode of access: <http://gnosis.cx/TPiP/>
13. Bird, S. Natural language processing with Python / S. Bird, E. Klein, E. Loper. – O'Reilly Media, 2009. – 504 с. – Mode of access: <http://www.nltk.org/book/>
14. Accessing Text Corpora and Lexical Resources [Electronic resource]. – Mode of access: <http://www.nltk.org/book/ch02.html>
15. Jurafsky, D. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd edition / D. Jurafsky, J. Martin – Prentice-Hall, 2009. – 1024 p.

Технологии статистической обработки текстов и методика их обучения

Резина О.В.

Аннотация. В статье рассматриваются особенности методики обучения технологиям статистической обработки текстовых данных в процессе подготовки будущих специалистов по прикладной лингвистике. Описаны базовые операции количественной обработки текстов и предложен возможный подход к их реализации с использованием языка программирования Python. Обоснована целесообразность использования языка Python и ресурсов платформы NLTK (Natural Language Toolkit) для выполнения заданий компьютерной лингвистики. Рассматривается задача обработки текстового файла с целью вычисления некоторых количественных параметров и приводятся соответствующие программные коды.

Ключевые слова: статистическая обработка текста, язык программирования Python, методика обучения.

Statistical text processing techniques and their teaching methods

O. V. Riezina

Resume. The article highlights the peculiarities of teaching methods of statistical text processing techniques in the process of training future specialists in Applied Linguistics. Basic operations of quantitative text processing have been outlined and a possible approach to applying them by means of Python programming language has been suggested. The appropriateness of using Python and the NLTK (Natural Language Toolkit) platform resources for solving problems in computational linguistics has been substantiated. The author considers the task of the text file processing to calculate some quantitative parameters and suggests relative program codes.

Keywords: statistical text processing, Python programming language, teaching methods.

УДК 378.014:004

Балик Н. Р.

Тернопільський національний педагогічний університет імені Володимира Гнатюка

Реалізація освітніх smart-інновацій в педагогічному університеті

Анотація. У статті аналізується понятійний апарат з проблематики дослідження: smart-навчання, smart-університет, smart-технології, smart-інновації. Обґрунтовується необхідність та можливість запровадження концепції smart-університету у навчальному процесі в педагогічному університеті. Smart-університет – це навчальний заклад, діяльність якого ґрунтується на принципах управління, основним елементом якого є розробка стратегічного бачення стану університету в майбутньому. Описуються основні ознаки, характерні для smart-університету. Зазначається, що створенню smart-університету сприяє активне запровадження освітніх smart-інновацій, таких як: цифрові носії контенту, цифрове адміністрування тощо.

Ключові слова: smart-університет, smart-навчання, технології навчання, навчання протягом життя, освітні smart-інновації.