Завдання.

Parse product description from ROZETKA and build classifier for products using only word vectors.  
Use kNN as baseline. Make at least 2 iterations of enhancement:  
- use best vectors (<https://lang.org.ua/en/models/#anchor4>)  
- preprocessing  
- any classification algorithms

1. **Парсинг опису товарів із сайту «Розетки»:**

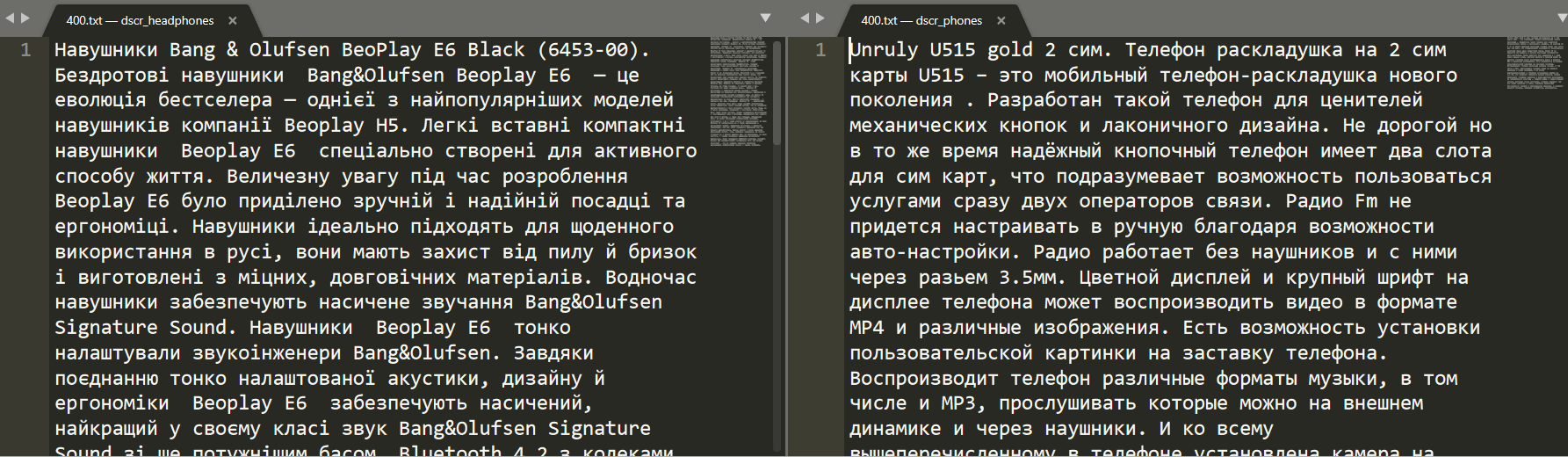
Код програми:



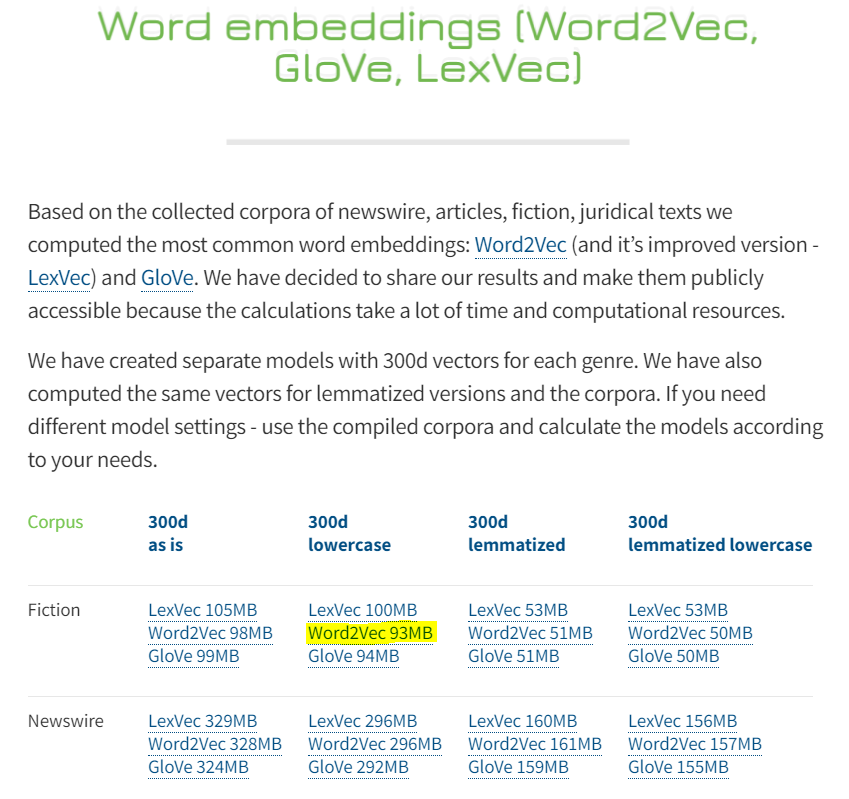


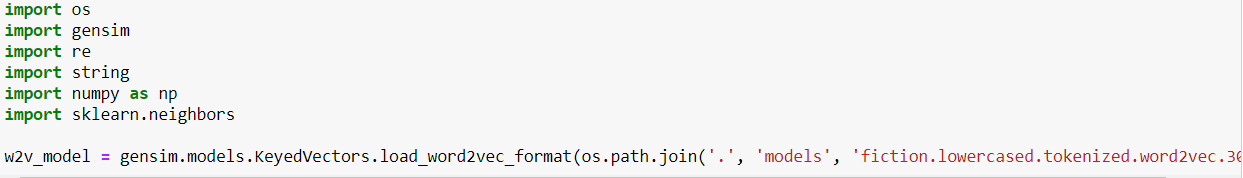
Результати роботи програми:

Було зібрано по 400 описів мобільних телефонів та навушників (усього 800 екземплярів).



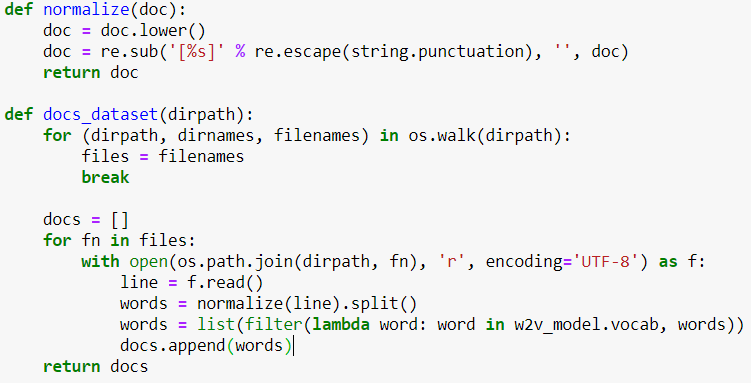
1. **Підготовка вибірки даних та навчання класифікатора:**
   1. Завантаження та ініціалізація моделі «word embedding»:

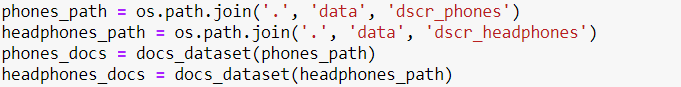
****

****

* 1. Попередня обробка та розбиття описів продуктів на слова:

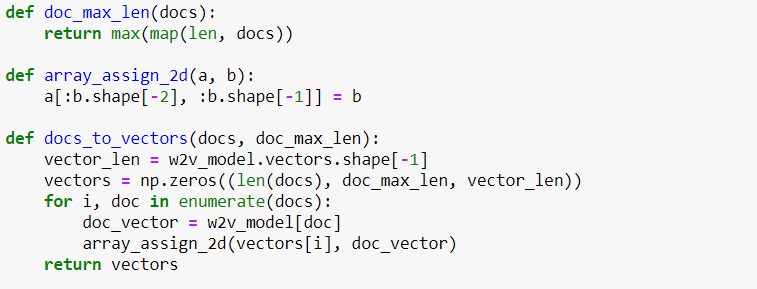
Кожен документ (представлений текстовим рядком) переводиться у нижній регістр (lower); із нього вилучаються розділові знаки (string.punctuation); документ розбивається на окремі слова (split); після фільтрації залишаються лише ті слова, які наявні в словнику обраної моделі (w2v\_model.vocab).

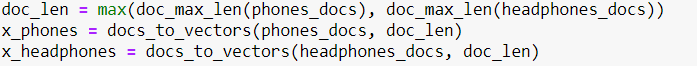
****

****

* 1. Відбувається перетворення масивів слів у вектори чисел:

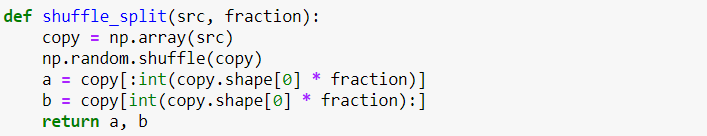
Для цього визначається максимальна кількість слів у документі, а всі інші документи мають бути доповнені до цієї довжини нулями (техніка padding, тут це значення 780 слів); для кожного документу знаходиться відповідна матриця (масив розмірністю 780х300, де 300 – розмірність вихідного вектору word2vec). На виході отримується багатомірний масив розмірністю 400х780х300 для кожного з двох наборів даних.

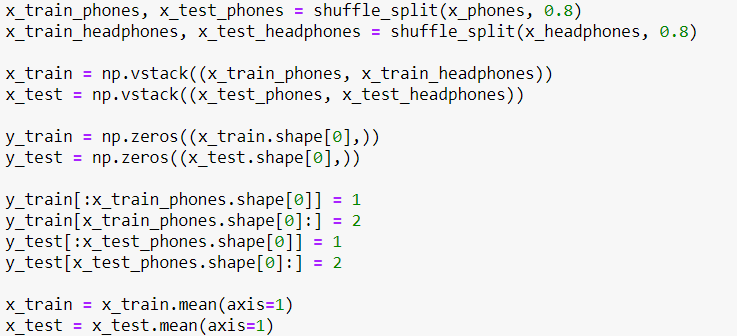


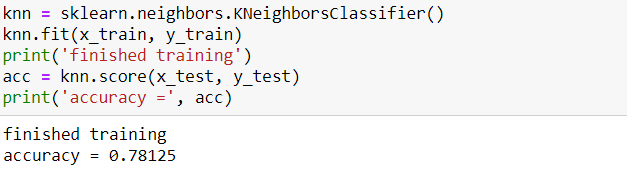


* 1. Остаточна обробка даних та навчання класифікатора:

Відбувається розбиття кожного з наборів на навчальну і тестову частини (shuffle\_split); їхнє подальше об’єднання (np.vstack). На останньому етапі із масиву розмірністю Nх780х300 було отримано масив Nх300 шляхом обчислення середніх значень по другій осі. Це було виконано з метою отримання певної загальної характеристики документа, незалежної від перестановки слів та інших особливостей. Наприклад, коли два документи містять однакове речення, але воно зміщене на певну кількість слів, що передують йому. В результаті документи матимуть однакові вектори, але розташовані на різних осях. Але, як показано далі, використання цього усереднення (averaging) тільки погіршило точність.







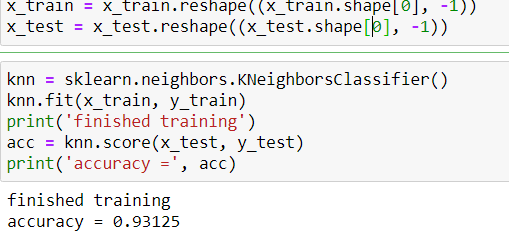
1. **Cпроби підвищити точність моделі:**

По-перше, було використано моделі word2vec з інших словників. Вони спричинили наступні зміни в точності класифікатора:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва словнику моделі | Простір на диску | Кількість слів словника моделі | Максимальний розмір документу (залежить від кількості розпізнаних слів) | **Досягнута точність** |
| Fiction | 301 Мб | 110363 | 780 | **78-82%** |
| Newswire | 899 Мб | 328958 | 1135 | **88-92%** |
| Ubercorpus | 1,49 Гб | 538431 | 1165 | **92-96%** |

Найкращий показник точності виявився при застосуванні моделі, що була натренована на словнику «Ubercorpus». Її ініціалізація тривала в межах декількох хвилин, але у швидкості роботи вона не поступається меншим за розміром альтернативам.

По-друге, була здійснена перевірка ефективності техніки averaging, результати якої були досить неоднозначні. Для моделі зі словником «Fiction» (найменшої за розміром), використання averaging лише погіршувало точність класифікатора. Точність 78-82% із її застосуванням проти 93%, досягнутих без неї (в такому випадку кожен документ характеризується 780\*300 = 234,000 значеннями, замість 300, що значно сповільнює виконання, але підвищує точність):



Натомість, при застосуванні моделі зі словником «Ubercorpus», використання averaging виявилось виправданим: якщо з нею точність класифікатора складає 92-96%, то без неї класифікатор має точність 88-93%, а швидкість його роботи значно погіршується.

По-третє, було до попередньої обробки слів було включено фільтр за розміром, який відкидав всі слова, довжина яких менше 3-х символів. Він ґрунтується на припущення, що слід відкинути вплив таких слів різноманітних сполучників («та», «і», «або»), прийменників («на», «в») та інших частин мови. Це покращило точність на декілька відсотків, яка тепер становить 97-94%.

Нарешті, застосування класифікатора Support Vector Machines з параметром gamma = ‘scale’ додатково підвищило точність до 99-95%.