نظام إكمال النصوص التلقائي الذكي

إعداد : غيث كسار

المقدمة:

يقوم هذا المشروع بتطوير نظام إكمال تلقائي للنصوص يهدف إلى تحسين سرعة ودقة الكتابة للمستخدمين، خاصةً عند التعامل مع اللغتين العربية والإنجليزية. يعتمد النظام على تحليل أنماط الكلمات المتكررة في النصوص لتقديم اقتراحات ذكية، مما يقلل من الجهد المطلوب للكتابة ويساعد في تجنب الأخطاء الإملائية والنحوية.

بنية المشروع:

يتكون المشروع من ثلاثة ملفات رئيسية:

- main.py:نقطة الدخول الرئيسية للتطبيق.
- app.py:يحتوي على منطق واجهة المستخدم الرسومية (GUI) باستخدام مكتبة PyQt6 .
- model.py:يتضمن منطق بناء وتحميل نماذج اللغة (Bigram و Trigram) ووظائف التنبؤ بالكلمات والجمل.

التقنيات المستخدمة:

Python 3 :لغة البرمجة الأساسية.

PyQt6:لإنشاء واجهة المستخدم الرسومية.

NLTK (Natural Language Toolkit):لمعالجة اللغة الطبيعية (تقسيم الكلمات، N-grams).

arabic-reshaper: لإعادة تشكيل الحروف العربية بشكل صحيح.

python-bidi :لمعالجة اتجاه النص ثنائي الاتجاه (من اليمين إلى اليسار ومن اليسار إلى اليمين).

pickle: لتخزين وتحميل نماذج اللغة.

re (Regular Expressions): لتنقية وتحليل النصوص.

collections.Counter: لحساب تكرارات N-grams .

سير عمل المشروع:

- يبدأ المستخدم في كتابة النص في مربع الإدخال.
- عند كل تغيير في النص، يتم استدعاء وظيفة on_text_changed.
- تقوم هذه الوظيفة بتحديد آخر كلمة أو كلمتين مكتوبتين.
- بناءً على وضع التشغيل (كلمة أو جملة)، يتم استدعاء predict_next_wordsأو generate_sentences من model.py للحصول على الاقتراحات.
- يتم عرض الاقتراحات في قائمة أسفل مربع النص.
- يمكن للمستخدم تحديد اقتراح بالنقر عليه أو باستخدام مفاتيح الأسهم و Enter لإدراجه في النص.
- يتم تحديث قائمة الاقتراحات باستمرار بناءً على النص الحالي

المكونات والوظائف الرئيسية:

model.py معالجة البيانات ونماذج اللغة:

- **تحميل النصوص العربية** :تقوم الدالة load_arabic_text() بتحميل النص العربي من الملف arabic_corpus.txt. يتم تنقية النص من الأحرف غير العربية وتوحيد المسافات.
- **تقسيم النصوص (Tokenization) :**تستخدم الدالة arabic_tokenizer() التعبيرات النمطية لتقسيم النص العربي إلى كلمات (tokens).
- **تحميل النماذج(N-gram Models) :**
  - تقوم الدالة load_models() بتحميل نماذج Bigram و Trigram المدربة مسبقاً من ملفات bigram_model.pkl و trigram_model.pkl إذا كانت موجودة.
  - إذا لم تكن النماذج موجودة، يتم تدريبها من البداية باستخدام نص إنجليزي من nltk.corpus.gutenbergعلى وجه التحديد (austen-emma.txt)ونص عربي من arabic_corpus.txt.
  - يتم دمج الكلمات (tokens) من كلا اللغتين لتدريب النماذج.
  - يتم استخدام collections.Counter لحساب تكرارات Bigrams وTrigrams .
  - يتم حفظ النماذج المدربة باستخدام pickle لإعادة الاستخدام لاحقاً.

- يتم تحديث النماذج أيضًا ببيانات المستخدم من الملف user_data.txt، حيث يتم إضافة تكرارات Trigrams الموجودة في هذا الملف.

- **التنبؤ بالكلمات التالية(predict_next_words)** :
  - تأخذ هذه الدالة كلمتين سابقتين (word1, word2) وتبحث في نموذج Trigram عن الكلمة التالية الأكثر احتمالاً.
  - إذا لم يتم العثور على اقتراحات كافية باستخدامTrigram ، فإنها تعود إلى نموذج Bigramباستخدام الكلمة الثانية فقط (word2) .
  - تقوم بإرجاع قائمة بأكثر الكلمات المرشحة شيوعاً(أعلى N).
- **توليد الجمل (generate_sentences)** :
  - تأخذ هذه الدالة الكلمات الأولية (start_words) وتحاول توليد جمل كاملة باستخدام نماذج اللغة.
  - تستخدم التنبؤ بالكلمات التالية بشكل متكرر لإنشاء سلسلة من الكلمات حتى الوصول إلى علامة ترقيم نهاية الجملة أو تجاوز الحد الأقصى للكلمات.
  - تهدف إلى توليد عدد محدد من الجمل الفريدة.

app.py واجهة المستخدم الرسومية(GUI) :

- **الفئة AutoCompleteApp**:
  - تقوم بتهيئة واجهة المستخدم الرسومية باستخدام PyQt6 .
  - تقوم بتحميل نماذج اللغة (trigram_model و bigram_model)عند بدء التشغيل.
  - تحدد وضع التشغيل (كلمة أو جملة) ويتم التبديل بينهما.
- **واجهة المستخدم(init_ui)** :
  - تتكون الواجهة من مربع نص (QTextEdit) لإدخال النص، وقائمة (QListWidget) لعرض الاقتراحات، وأزرار للتبديل بين الأنماط، ومسح النص، والخروج.
  - يتم تعيين اتجاه النص في مربع النص وقائمة الاقتراحات من اليمين إلى اليسار لدعم اللغة العربية.

- **معالجة النص المدخل (on_text_changed)** :
  - يتم استدعاء هذه الدالة في كل مرة يتغير فيها النص في مربع الإدخال.
  - تقوم بتحليل النص المدخل واستخراج آخر كلمتين كـ "سياق" للتنبؤ.
  - إذا كان الوضع هو "جملة"(sentence_mode) ، فإنه يتم استخدام generate_sentences لتوليد اقتراحات الجمل.
  - إذا كان الوضع هو "كلمة"، يتم استخدام predict_next_wordsللتنبؤ بالكلمة التالية.
  - يتم عرض الاقتراحات الفريدة (مع استبعاد الاقتراحات المستخدمة بالفعل) في قائمة الاقتراحات.
- **تنسيق النص العربي(reshape_arabic, is_arabic) :**
  - يتم استخدام مكتبتي arabic_reshaperو bidi.algorithm لمعالجة وعرض النص العربي بشكل صحيح في الواجهة الرسومية، لضمان صحة الاتصال (joining) واتجاه النص(bidi) .
- **إدراج الاقتراحات(on_suggestion_clicked, insert_suggestion)** :
  - عند النقر على اقتراح في القائمة، يتم إدراجه في مربع النص تلقائيًا.
  - يتم إضافة الاقتراح المستخدم إلى مجموعة used_suggestions لمنع تكرار الاقتراحات لنفس السياق.
- **مسح النص (clear_text)** :
  - يقوم زر "مسح" بمسح مربع النص وإخفاء الاقتراحات وإعادة تعيين الاقتراحات المستخدمة.

main.py تشغيل التطبيق:

- يقوم بإنشاء مثيل لتطبيق QApplication وبدء تشغيل الواجهة الرسومية AutoCompleteApp.
- يضمن إغلاق التطبيق بشكل صحيح عند الخروج.