

# تقرير تحليل الفجوات الشامل - مشروع كاتب

ما الذي ينقصنا؟ وما الذي يمكن تحسينه؟

التاريخ: 18-02-2026 المشروع: كاتب (Kateb) - إملاء صوتي عربي الإصدار الحالي: 1.2.0

## الجزء الأول: ما الموجود حالياً (ملخص سريع)

الميزة	الحالة	ملاحظات
تحويل صوت لنص (Whisper)	✓	نماذج streaming
تحسين بالذكاء الاصطناعي	✓	مزودين (Claude, OpenAI, Gemini, Grok, Local)
اختصار لوحة مفاتيح عالمي	✓	Double-tap + Modifier combos
كتابة تقليدية في أي تطبيق	✓	enigo
كشف الصمت والتوقف التقليدي	✓	Adaptive VAD
تسريع GPU (CUDA)	✓	NVIDIA فقط
معالجة صوتية	✓	High-pass filter, normalization, pre-emphasis
مكافحة الهدوء	✓	Pattern matching + statistics
واجهة عامة (Overlay)	✓	قابلة للسحب
System Tray	✓	تصغير الخافية
تحديث تقليدي	✓	GitHub Releases
تصدير/استيراد الإعدادات	✓	نسخ احتياطي
سجل الإملاءات	✓	مع إحصائيات
دعم ثانوي اللغة (AR/EN)	✓	i18n
بناء متعدد المنصات	✓	Linux, Windows, (macOS)

## الجزء الثاني: ما الذي ينقصنا؟ (19 نقطة)

### أولوية عالية (يجب تنفيذها)

#### 1. علامات الترقيم التقليدية (Auto-Punctuation)

المشكلة: Whisper لا يضع علامات ترقيم بشكل دقيق في العربية. المستخدم يحتاج بنطق "فاصلة" أو "نقطة" يدويًا.

الحل المقترن:

- إضافة نموذج ترقيم خفيف بعد Whisper

- أو الاعتماد على AI Refiner لإضافة الترقيم (موجود جزئياً)
  - أو استخدام نموذج مخصص مثل oliverguhr/fullstop-punctuation-multilingual-large
- الاثر: تحسين كبير في جودة النص النهائي بدون تدخل المستخدم
- 

## 2. اوامر صوتية (Voice Commands)

المشكلة: المستخدم لا يستطيع التحكم بالنص صوتيًا. لا يمكنه قول "سطر جديد" أو "احذف الكلمة الأخيرة".

الأوامر المطلوبة:

الأمر الصوتي	ال فعل
"new line" / "سطر جديد"	إدراج سطر جديد
"نقطة" / "فاصلة"	إدراج علامة ترقيم
"delete" / "احذف"	حذف آخر كلمة
"احذف الجملة"	حذف آخر جملة
"توقف" / "stop"	إيقاف الإملاء
"فقرة جديدة"	سطرين جديدين
"undo" / "تراجع"	الرجوع عن آخر تتعديل

الحل: كشف الأوامر الصوتية من النص قبل إرساله للكتابة، واستبدلها بالفعل المناسب.

---

## 3. إلغاء ضوضاء الخلفية (Noise Suppression)

المشكلة: المعالجة الصوتية الحالية (high-pass filter + normalization) أساسية جداً. لا يوجد إلغاء ضوضاء حقيقي.

الحل المقترن:

- دمج RNNNoise (مكتبة C مفتوحة المصدر، خفيفة جداً ~100KB)
- أو Whisper قبل Silero VAD + noise gate
- تزيل RNNNoise ضوضاء المروحة، لوحة المفاتيح، الخلفية

الاثر: تحسين كبير في الدقة خاصة في البيانات الصالحة. Whisper يعاني بشدة مع الضوضاء (WER يرتفع 40-60% في MASC-Noisy).

---

## 4. نظام Clipboard بدل Keystroke

المشكلة الحالية: يكتب حرف حرف = بطيء ومشاكل مع RTL وبعض التطبيقات.

الحل الأفضل:

1. نسخ النص للأ Clipboard
2. محاكاة Ctrl+V (لصق)
3. إعادة محتوى الأ Clipboard الأصلي

المميزات:

- أسرع بكثير (فوري بدل حرف حرف)
- لا مشاكل RTL
- يعمل مع كل التطبيقات
- يحافظ على التنسيق

---

## 5. معالجة انقطاع الميكروفون

المشكلة: لا يوجد معالجة لحالة فصل الميكروفون أثناء التسجيل.

الحل:

- مراقبة حالة الجهاز الصوتي باستمرار
- حفظ الصوت المسجل حتى لحظة الانقطاع
- إشعار المستخدم وتحويل ما تم تسجيله
- إعادة الاتصال تلقائياً عند توفر الميكروفون

## أولوية متوسطة (تحسينات مهمة)

### 6. دعم GPU متعدد (AMD + Apple Silicon)

المشكلة: الدعم الحالي لـ NVIDIA CUDA فقط. أجهزة AMD و Apple Silicon لا تستفيد.

الحل:

- Vulkan: يعمل على whisper.cpp NVIDIA + AMD + Intel (يدعمه)
- Metal: لأجهزة Apple whisper.cpp (يدعمه)
- OpenCL: بديل عام

الاثر: نسبة كبيرة من المستخدمين على AMD أو Mac لن يستفيدوا من التسريع حالياً.

## 7. قاموس مخصص (Custom Vocabulary)

المشكلة: Whisper لا يعرف المصطلحات المتخصصة أو أسماء الأعلام العربية.

الحل:

- استخدام initial\_prompt في Whisper مع المصطلحات المتترفة
- السماح للمستخدم بإضافة قاموس مخصص (أسماء، مصطلحات طيبة/قانونية/تقنية)
- تمرير القاموس كـ initial\_prompt لترجمة Whisper

مثال:

```
initial_prompt = "محمد، أحمد،React, TypeScript, API"
```

الاثر: تحسين كبير في دقة الأسماء والمصطلحات المتخصصة.

## 8. وضع الإملاء المستمر (Continuous Dictation)

المشكلة: حالياً يجب الضغط على الاختصار لكل جلسة إملاء. لا يوجد وضع "إملاء مستمر" يستمر حتى الإيقاف.

الحل:

- وضع Push-to-Talk (الحالى): اضغط لتداء، اضغط لتوقف
- وضع Continuous: يستمر بالإملاء مع فوائل ذكية عند الصمت
- وضع Hold-to-Talk: اضغط مع الاستمرار للتحدث، ارفع للتوقف

## 9. تصحيح تفاعلي (Interactive Correction)

المشكلة: بعد النسخ، لا يمكن المستخدم تصحيح كلمة معينة بسهولة.

: الحل

- عرض النص المنسوخ مع إمكانية النقر على أي كلمة لتصحيحها
- اقتراحات بدائل (مثل `autocorrect`)
- تعلم من تصحيحات المستخدم لتحسين الدقة مستقبلاً

## 10. مستوى الصوت والجودة (Audio Level Indicator)

المشكلة: المستخدم لا يعرف إن كان الميكروفون يلقط صوته بشكل جيد أم لا قبل البدء.

: الحل

- مؤشر مستوى الصوت المرئي (موجود جزئياً في الـ `overlay`)
- تحذير إذا كان الصوت ضعيف جداً أو عالي جداً (`clipping`)
- اختبار ميكروفون قبل البدء
- إعدادات كسب الصوت (`gain control`)

## 11. تعدد اللغات في نفس الجلسة (Language Switching)

المشكلة: يجب اختيار اللغة مسبقاً (عربي أو إنجليزي). لا يوجد كشف تلقائي.

: الحل

- كشف اللغة تلقائياً من `Whisper` (بدعمها أصلًا)
- أو وضع "متعدد اللغات" يسمح بالتبديل التلقائي
- مفيد جداً حالة `code-switching` (عربي-إنجليزي)

## 12. دعم Linux على PipeWire

المشكلة: Linux حديث يستخدم `PulseAudio`. بدل `cpal` قد لا يتعامل بشكل مثالي.

: الحل

- التأكد من توافق `cpal` مع `PipeWire`
- أو استخدام `PipeWire API` مباشرةً
- اختبار على توزيعات حديثة (+Ubuntu 24.04+, Fedora 40)

## أولوية منخفضة (ميزات إضافية للمستقبل)

### 13. تحويل الملفات الصوتية (File Transcription)

المشكلة: التطبيق يعمل فقط مع الميكروفون المباشر، لا يمكن تحويل ملف صوتي محفوظ.

: الحل

- إضافة خيار "تحويل ملف صوتي" (`drag & drop`)
- دعم: MP3, WAV, M4A, OGG, FLAC
- مفيد لتغريغ محاضرات ومقابلات

### 14. تمييز المتحدثين (Speaker Diarization)

المشكلة: لا يميز بين متحدثين مختلفين.

: الحل

- 
- `pyannote/speaker-diarization` مثل diarization
- دمج نموذج
  - يميز: "المتحدث 1: ... المتحدث 2: ..."
  - مفيد لترجمة المقابلات والاجتماعات
- 

## 15. تصدر بصيغ متعددة

المشكلة: لا يوجد تصدر للنصوص بصيغ مختلفة.

الحل:

- تصدر كـ TXT, DOCX, PDF, SRT (ترجمة من نية)
  - نسخ مع تنسيق (Markdown, HTML)
  - مشاركة مباشرة
- 

## 16. اختصارات لكل تطبيق (Per-App Profiles)

المشكلة: نفس الإعدادات لكل التطبيقات.

الحل:

- السماح باعدادات مختلفة حسب التطبيق النشط
  - مثلاً: Word يستخدم نموذج small بدون refinement، WhatsApp يستخدم large + AI refinement
- 

## 17. وضع عدم الإزعاج

المشكلة: الاختصار العالمي قد يتعارض مع تطبيقات أخرى.

الحل:

- قائمة تطبيقات مستثنية
  - وضع "عدم الإزعاج" يوقف الاختصار مؤقتاً
- 

## 18. إحصائيات متقدمة

المشكلة: الإحصائيات الحالية بسيطة (عدد الإملاءات، الكلمات، المدة).

الحل:

- رسوم بيانية للاستخدام عبر الزمن
  - دقة النسخ (مقارنة raw vs refined)
  - الكلمات الأكثر استخداماً
  - أوقات النزوة
- 

## 19. نظام Plugins

المشكلة: التطبيق مغلق - لا يمكن للمجتمع إضافة ميزات.

الحل:

- نظام plugins بسيط (مجلد /plugins)
  - يسمح بإضافة: أوامر صوتية مخصصة، مزودي AI جدد، صيغ تصدر
- 

# الجزء الثالث: تحسينات تقنية (Technical Improvements)

## أ. تحسينات Whisper

### المحسن Initial Prompt .1

:الحالي (جيد)

"...أنت نظام تحويل الصوت إلى نص"

:المقترح (أفضل)

initial\_prompt : مخصص حسب السياق

"...إذا المستخدم طبيب: "مريض، تشخيص، علاج، أشعة، تحليل -

- إذا المستخدم مبرمج: "function, API, React, component, database..."

- إذا المستخدم عام: "بسم الله، إن شاء الله، القاهرة، محمد -"

### محسن Chunk Overlap .2

overlap مع 0.75 ثانية chunks 2 ثانية

دینامیکیہ حسّب VAD chunks

أكبر (4-3 ثوانٍ) = دقة أعلى chunk: إذا كلام متصل -

أصغر (1-2 ثانية) = سرعة أعلى chunk: إذا جمل قصيرة -

### Temperature Fallback .3

(ثابت) temperature = 0.0 :الحالي

:المقترح temperature fallback

- أول محاولة: 0.0

منخفض: أعد مع 0.2 إذا confidence

- إذا لا يزال منخفض: أعد مع 0.4

هذا يقلل الهلوسات مع الحفاظ على الدقة

### Whisper Prompt Chaining .4

التالي chunk للـ prompt من النص السابق كـ token 224

يحسن استمرارية الحمل بين الـ chunks

- يقلل تكرار الكلمات

- Whisper initial\_prompt يدعم هذا أصلاً عبر

## ب. تحسينات الأداء

### 1. تحميل النموذج المسبق (Model Preloading)

الحالي: النموذج يحمل عند أول إملاء (2-5 ثوانٍ تأخير)

المقترح: تحميل النموذج عند بدء التطبيق في الخلفية

- بدء التطبيق → تحميل النموذج فوراً

- أول إملاء → جاهز فوراً بدون انتظار

### 2. تقليل استهلاك الذاكرة

المشكلة: يستهلك large-v3-turbo ~5-3 GB RAM

المقترح:

- INT8 quantization (50%) يقلل الذاكرة عبر whisper.cpp
- (يقلل 75% مع خسارة دقة طفيفة) INT4 quantization أو
- تفريغ النموذج بعد فترة عدم استخدام (5 دقائق)

### 3. Processing Batch للملفات

المقترح: عند تحويل ملفات صوتية (ميزة مستقبلية)

- تقسيم الملف لأجزاء
- معالجة متوازية على عدة cores
- عرض تقدم بنسبة مئوية

## ج. تحسينات الأمان

### 1. خصوصية الصوت

السؤال	الحالة الحالية	المقترح
هل يحفظ الصوت؟	لا (جيد)	إضافة خيار حفظ اختياري
هل يرسل الصوت لسيرفر؟	لا (جيد)	واضح في السياسة
AI النص المرسل له	نعم (التحسين)	تشفير + إشعار واضح
API Keys	System Keychain	أقوى fallback جيد، لكن يحتاج تشفير

### 2. سياسة خصوصية واضحة

المشكلة: لا توجد صفحة خصوصية في التطبيق توضح:

- ما البيانات التي تجمع؟
- هل يرسل الصوت أو النص لأي جهة خارجية؟
- أين تخزن البيانات محلياً؟

الحل: إضافة صفحة "الخصوصية" في التطبيق مع شرح واضح.

## الجزء الرابع: مقارنة مع المنافسين

الميزة	كاتب (نحن)	Windows Dictation	Google Voice Typing	Whisper.cpp CLI	Munxit
(offline) يعمل محلياً	✓	✓	✗	✓	✗
مجاني	✓	✓	✓	✓	✗
دقة العربية	جيد	ضعيف	جيد جداً	جيد	ممتاز
اللهجات	ضعيف	لا	جيد	ضعيف	ممتاز
أوامر صوتية	✗	✓	✓	✗	✗

ترقيم تلقائي	(AI) جزئي	✓	✓	✗	✓
تصحيح AI	✓	✗	✗	✗	✗
تخصيص قاموس	✗	✗	✗	✓	✗
إملاء مستمر	✗	✓	✓	✗	✓
تحويل ملفات	✗	✗	✗	✓	✓
تمييز متحدثين	✗	✗	✗	✗	✓
cross-platform	✓	Windows فقط	Chrome فقط	CLI	Web
إلغاء ضوضاء	بسيط	✓	✓	✗	✓
كتابة في أي تطبيق	✓	✓	Chrome في فقط	✗	✗
GPU acceleration	✓ CUDA	✓	Cloud	✓	Cloud

#### نقاط القوة الفريدة لكاتب:

- الوحيد الذي يجمع: محلي + مجاني + AI refinement + يكتب في أي تطبيق
- الوحيد المفتوح المصدر مع واجهة رسومية كاملة
- الوحيد الذي يدعم 5 مزودي AI للتصحيح

#### نقاط الضعف مقارنة بالمنافسين:

- لا أوامر صوتية (المنافسون يدعمونها)
- لا إملاء مستمر
- لا إلغاء ضوضاء حقيقي
- لا تحويل ملفات صوتية

## الجزء الخامس: خارطة الطريق المقترحة

### المرحلة 1: التحسينات الجوهرية (الأولوية القصوى)

#	الميزة	الجهد	الاثر
1	إلغاء ضوضاء (RNNoise)	متوسط	🔥🔥🔥
2	Clipboard paste بدل keystroke	سهل	🔥🔥🔥
3	أوامر صوتية أساسية	متوسط	🔥🔥🔥
4	مخصص + قاموس initial_prompt	سهل	🔥🔥
5	معالجة انقطاع الميكروفون	سهل	🔥🔥

### المرحلة 2: تجربة مستخدم متقدمة

#	الميزة	الجهد	الاثر
---	--------	-------	-------

6	إملاء مستمر (Continuous mode)	متوسط	
7	كشف لغة تلقائي	سهل	
8	تصحيح تفاعلي	صعب	
9	تحميل النموذج المسبق	سهل	
10	مؤشر جودة الصوت	سهل	

### المرحلة 3: ميزات متقدمة

#	الميزة	الجهد	الاثر
11	تحويل ملفات صوتية	متوسط	
12	دعم Vulkan/Metal GPU	صعب	
13	تصدير بصيغة متعددة	سهل	
14	ملفات شخصية لكل تطبيق	متوسط	
15	INT8 quantization	متوسط	

### المرحلة 4: ميزات مستقبلية

#	الميزة	الجهد	الاثر
16	تمييز المتحدثين	صعب	
17	احصائيات متقدمة	سهل	
18	نظام Plugins	صعب	
19	صفحة خصوصية	سهل	

## الجزء السادس: ملخص تنفيذي

### ما الذي يميّزنا؟

كاتب هو التطبيق الوحيد في العالم الذي يجمع بين:

- محي ومجاني ومفتوح المصدر
- تصحيح بالذكاء الاصطناعي
- يكتب في أي تطبيق
- واجهة رسومية كاملة

### ما الذي ينقصنا بشدة؟

1. أوامر صوتية - المناقشون كلهم يدعمنها
2. الغاء ضوضاء - يؤثر بشكل كبير على الدقة
3. keystroke clipboard paste - أسرع وأدق من

## ما الذي سيحدث فرقاً كبيراً بأقل جهد؟

1. تحسين فوري في سرعة الكتابة **Clipboard paste** (ساعات قليلة) → تحسين فوري في سرعة الكتابة
2. تحسين دقة الأسماء والمصطلحات **initial\_prompt** (ساعة واحدة) → تحسين دقة الأسماء والمصطلحات
3. تحميل النموذج المسبق (ساعة واحدة) → بدون تأخير عند أول إملاء تحميل النموذج المسبق

---

### المصادر

- [Open Universal Arabic ASR Leaderboard](#)
- [RNNoise - Noise Suppression](#)
- [faster-whisper](#)
- [Silero VAD](#)
- [whisper.cpp](#)
- [Munsit Arabic ASR](#)
- [Deepgram Nova-3 Arabic](#)
- [Speechmatics Arabic](#)
- [fullstop-punctuation model](#)