

تقرير تحليل الفجوات الشامل - مشروع كاتِب

ما الذي ينقصنا؟ وما الذي يمكن تحسينه؟

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

الجزء الأول: ما الموجود حالياً (ملخص سريع)

الميزة	الحالة	ملاحظات
Whisper) تحويل صوت لنص	✓	streaming، نماذج 5
تحسين بالذكاء الاصطناعي	✓	5 مزودين (Claude, OpenAI, Gemini, Grok, Local)
اختصار لوحة مفاتيح عالمي	✓	Double-tap + Modifier combos
كتابة تلقائية في أي تطبيق	✓	enigo
كشف الصمت والتوقف التلقائي	✓	Adaptive VAD
GPU (CUDA) تسريع	✓	فقط NVIDIA
معالجة صوتية	✓	High-pass filter, normalization, pre-emphasis
مكافحة الهلوسات	✓	Pattern matching + statistics
(Overlay) واجهة عائمة	✓	قابلة للمسح
System Tray	✓	تصغير للخلفية
تحديث تلقائي	✓	GitHub Releases
تصدير/استيراد الإعدادات	✓	نسخ احتياطي
سجل الإملاءات	✓	مع إحصائيات
(AR/EN) دعم ثنائي اللغة	✓	i18n
بناء متعدد المنصات	✓	Linux, Windows, (macOS)

الجزء الثاني: ما الذي ينقصنا؟ (19 نقطة)

● أولوية عالية (يجب تنفيذها)

1. علامات الترقيم التلقائية (Auto-Punctuation)

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

الحل المقترح:

- 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`

الأثر: تحسين كبير في جودة النص النهائي بدون تدخل المستخدم

2. أوامر صوتية (Voice Commands)

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

الأوامر المطلوبة:

الأمْر الصوتي	الفعل
"new line" / "سطر جديد"	إدراج سطر جديد
"نقطة" / "فاصلة"	إدراج علامة ترقيم
"delete" / "احذف"	حذف آخر كلمة
"احذف الجملة"	حذف آخر جملة
"stop" / "توقف"	إيقاف الإملاء
"فقرة جديدة"	سطرين جديدين
"undo" / "راجع"	الترجع عن آخر تعديل

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

3. إلغاء ضوضاء الخلفية (Noise Suppression)

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

الحل المقترح:

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

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

4. نظام Clipboard بدل Keystroke

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

الحل الأفضل:

1. نسخ النص للـ Clipboard
2. محاكاة Ctrl+V (لصق)
3. إعادة محتوى الـ 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 NVIDIA + AMD + Intel (whisper.cpp يدعمه)
- **Metal**: لأجهزة Apple (whisper.cpp يدعمه)
- **OpenCL**: بديل عام

الأثر: 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 المتوقعة
- السماح للمستخدم بإضافة قاموس مخصص (أسماء، مصطلحات طبية/قانونية/تقنية)
- تمرير القاموس كـ `initial_prompt` لتوجيه Whisper

مثال:

```
initial_prompt = "الأهر، القاهرة، أحمد، محمد، React، TypeScript، API"
```

الأثر: 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 (الحالي): اضغط لتبدأ، اضغط لتوقف
- وضع Continuous: يستمر بالإملاء مع فواصل ذكية عند الصمت
- وضع Hold-to-Talk: اضغط مع الاستمرار للتحديث، ارفع للتوقف

9. تصحيح تفاعلي (Interactive Correction)

المشكلة: بعد النسخ، لا يمكن للمستخدم 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 يلتقط 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 (يدعمها أصلاً)
- أو وضع "متعدد اللغات" يسمح بالتبديل التلقائي
- مفيد جداً لحالة code-switching (عربي-إنجليزي)

12. دعم PipeWire على Linux

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

الحل:

- التأكد من توافق cpal مع PipeWire
- أو استخدام PipeWire API مباشرة
- 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 للتقريغ محاضرات ومقابلات

14. تمييز المتحدثين (Speaker Diarization)

المشكلة: لا يميز بين متحدثين مختلفين.

الحل:

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

15. تصدير بصيغ متعددة

المشكلة: لا يوجد تصدير للنصوص بصيغ مختلفة.

الحل:

- تصدير كـ: TXT, DOCX, PDF, SRT (ترجمة مرئية)
- نسخ مع تنسيق (Markdown, HTML)
- مشاركة مباشرة

16. اختصارات لكل تطبيق (Per-App Profiles)

المشكلة: نفس الإعدادات لكل التطبيقات.

الحل:

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

17. وضع عدم الإزعاج

المشكلة: الاختصار العالمي قد يتعارض مع تطبيقات أخرى.

الحل:

- قائمة تطبيقات مستثناة
- 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 الذروة

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 تصدير

الجزء الثالث: تحسينات تقنية (Technical Improvements)

أ. تحسينات Whisper

1. Initial Prompt المحسن

الحالي (جيد):

"...أنت نظام تحويل الصوت إلى نص"

المقترح (أفضل):

initial_prompt مخصص حسب السياق:

- "...إذا المستخدم طبيب: "مريض، تشخيص، علاج، أشعة، تحليل"
- "function, API, React, component, database...": إذا المستخدم مبرمج
- "...إذا المستخدم عام: "بسم الله، إن شاء الله، القاهرة، محمد"

2. Chunk Overlap محسن

overlap مع 0.75 ثانية chunks الحالي: 2 ثانية

VAD ديناميكية حسب chunks: المقترح

- أكبر (3-4 ثواني) = دقة أعلى chunk: إذا كلام متصل
- أصغر (1-2 ثانية) = سرعة أعلى chunk: إذا جمل قصيرة

3. Temperature Fallback

الحالي: temperature = 0.0 (ثابت)

المقترح: temperature fallback

- أول محاولة: 0.0
 - منخفض: أعد مع 0.2 confidence إذا
 - إذا لا يزال منخفض: أعد مع 0.4
- هذا يقلل الهلوسات مع الحفاظ على الدقة

4. Whisper Prompt Chaining

التالي chunk لل prompt من النص السابق كـ token المقترح: تمرير آخر 224

- chunks يحسن استمرارية الجمل بين الـ
- يقلل تكرار الكلمات
- initial_prompt يدعم هذا أصلاً عبر Whisper

ب. تحسينات الأداء

1. تحميل النموذج المسبق (Model Preloading)

الحالي: النموذج يُحمّل عند أول إملاء (2-5 ثواني 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: large-v3-turbo 5-3 GB RAM يستهلك

المقترح:

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

3. Batch Processing للملفات

المقترح: عند تحويل ملفات صوتية (ميزة مستقبلية)

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

ج. تحسينات الأمان 🔧

1. خصوصية الصوت

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

2. سياسة خصوصية واضحة

المشكلة: لا توجد صفحة خصوصية في التطبيق توضح:

- ما البيانات التي تُجمع؟
- هل يُرسل الصوت أو النص لأي جهة خارجية؟
- أين تُخزن 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	Munsit
(offline) يعمل محلياً	✓	✓	×	✓	×
مجاني	✓	✓	✓	✓	×
دقة العربية	جيد	ضعيف	جيد جداً	جيد	ممتاز
اللهجات	ضعيف	لا	جيد	ضعيف	ممتاز
أوامر صوتية	×	✓	✓	×	×

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

نقاط القوة الفريدة لكاتب:

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

نقاط الضعف مقارنة بالمنافسين:

- 1. لا أوامر صوتية (المنافسون يدعمونها)
- 2. لا إملاء مستمر
- 3. لا إلغاء ضوضاء حقيقي
- 4. لا تحويل ملفات صوتية

الجزء الخامس: خارطة الطريق المقترحة

المرحلة 1: التحسينات الجوهرية (الأولوية القصوى)

#	الميزة	الجهد	الأثر
1	(RNNoise) إلغاء ضوضاء	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	الأثر
---	--------	-------	-------

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

المرحلة 3: ميزات متقدمة

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

المرحلة 4: ميزات مستقبلية

#	الميزة	الجهد	الأثر
16	تمييز المتحدثين	صعب	🔥
17	إحصائيات متقدمة	سهل	★
18	Plugins نظام	صعب	★
19	صفحة خصوصية	سهل	★

الجزء السادس: ملخص تنفيذي

ما الذي يميزنا؟

كاتب هو التطبيق الوحيد في العالم الذي يجمع بين:

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

ما الذي ينقصنا بشدة؟

1. أوامر صوتية - المنافسون كلهم يدعمونها
2. إلغاء ضوضاء - يؤثر بشكل كبير على الدقة
3. Clipboard paste - أسرع وأدق من keystroke

ما الذي سيحدث فرقاً كبيراً بأقل جهد؟

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

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](#)