

نظام ذكي للتعرف على الصوت

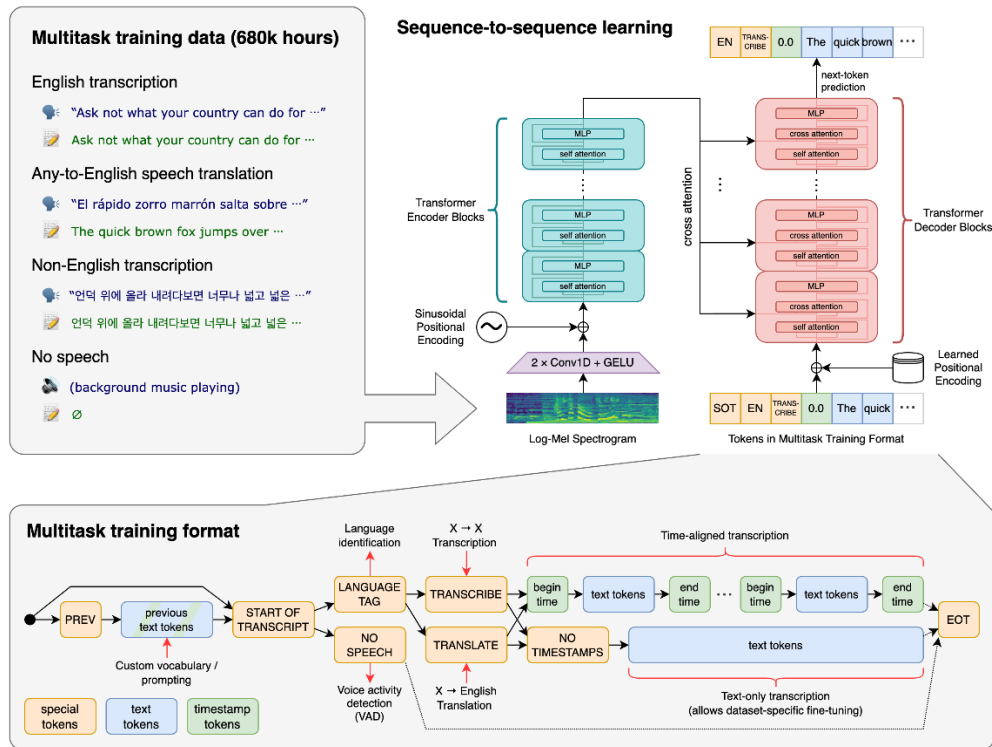
الهدف:

بناء نظام ذكي قادر على التعرف على الصوت بلهجات مختلفة (السعودية، السورية ...) ومن خلال تحليله للكلام يقوم بتوجيه الشخص لأحد ثلاث أماكن (الصالة، خدمة العملاء، الإستقبال) وذلك بإختيار أفضل نماذج التعرف على الكلام بلهجات عربية مختلفة ونماذج التعرف على النية.

إختيار النموذج الأفضل للتعرف على الكلام

في إطار بحثي عن أنظمة التعرف على الكلام متعدد اللهجات، وجدت ورقة بحثية بعنوان *"Casablanca: Data and Models for Multidialectal Arabic Speech Recognition"*، والتي تقدم دراسة موسعة حول البيانات والنماذج المتقدمة للتعرف على اللهجات العربية المختلفة. تشير الورقة إلى أن نموذج **WHISPER EGYPTIAN** يعد من أفضل النماذج للتعرف على اللهجات العامية العربية، بالإضافة إلى نموذج آخر وهو **whisper lg v3**، الذي يُظهر أيضاً أداءً قوياً في بعض الحالات.

والذي له البنية التالية



بعد اختبار كلا النموذجين، توصلت إلى أن نموذج *WHISPER EGYPTIAN* يتفوق في دقة التعرف على اللهجات المصرية، بينما يظهر نموذج *whisper lg v3* أداءً أفضل في التعرف على اللهجات الشامية والسعودية. بالنسبة للهجات العامية الشامية والسعودية، تبين لي أن *whisper lg v3* يتفوق على *WHISPER EGYPTIAN* من حيث الدقة والأداء.

| | whisper-lg-v2 | | whisper-lg-v3 | | seamless-m4t-v2-large | | mms-1b-all | |
|------------|---------------|----------------------|---------------|----------------------|-----------------------|----------------------|---------------|---------------|
| | - pre-proc | + pre-proc | - pre-proc | + pre-proc | - pre-proc | + pre-proc | - pre-proc | + pre-proc |
| Algeria | 82.61 / 38.95 | 80.47 / 36.82 | 83.49 / 40.47 | 84.14 / 39.99 | 101.18 / 58.58 | 94.18 / 53.56 | 93.01 / 43.68 | 92.55 / 42.62 |
| Egypt | 61.99 / 26.38 | 52.38 / 21.71 | 59.11 / 24.77 | 48.95 / 19.86 | 61.82 / 29.83 | 49.75 / 24.47 | 88.54 / 43.59 | 85.84 / 40.58 |
| Jordan | 49.47 / 16.34 | 41.13 / 13.64 | 48.44 / 16.18 | 39.68 / 13.47 | 47.94 / 15.84 | 39.24 / 13.12 | 81.46 / 33.02 | 78.54 / 31.03 |
| Mauritania | 87.85 / 52.34 | 85.74 / 49.76 | 87.44 / 50.19 | 85.68 / 48.08 | 91.57 / 55.41 | 88.39 / 51.59 | 94.36 / 50.25 | 93.71 / 48.99 |
| Morocco | 88.55 / 46.57 | 84.52 / 44.02 | 87.2 / 44.41 | 83.05 / 42.09 | 95.18 / 58.29 | 91.01 / 54.97 | 96.91 / 49.01 | 95.45 / 47.34 |
| Palestine | 57.06 / 20.02 | 48.64 / 17.24 | 58.02 / 21.05 | 50.2 / 18.38 | 56.78 / 20.74 | 48.92 / 18.13 | 83.14 / 33.07 | 80.18 / 30.82 |
| UAE | 61.82 / 22.93 | 52.03 / 19.15 | 62.31 / 24.04 | 52.88 / 20.37 | 63.94 / 26.22 | 54.76 / 22.71 | 85.4 / 36.81 | 82.11 / 34.18 |
| Yemen | 71.31 / 29.8 | 60.65 / 24.49 | 69.94 / 28.17 | 59.45 / 23.19 | 73.65 / 32.55 | 62.72 / 27.43 | 86.73 / 38.55 | 81.64 / 34.36 |
| AVG | 70.08 / 31.66 | 63.195 / 28.35 | 69.49 / 31.16 | 63.00 / 28.17 | 74.00 / 37.18 | 66.12 / 33.24 | 88.69 / 40.99 | 86.25 / 38.74 |

Table 3: Results for dialect evaluation, scenario-1 on the Test set. Results are reported in WER and CER (/ separated). **pre-proc**: preprocessing (+ with, - without).

| | whisper-msa | | whisper-mixed | | whisper-egyptian | | whisper-moroccan | |
|------------|------------------|---------------|----------------|----------------|----------------------|----------------------|----------------------|----------------------|
| | - pre-proc | + pre-proc | - pre-proc | + pre-proc | - pre-proc | + pre-proc | - pre-proc | + pre-proc |
| Algeria | 87.86 / 48.31 | 87.82 / 48.20 | 129.63 / 79.63 | 129.77 / 79.68 | 86.68 / 35.80 | 86.75 / 35.70 | 74.39 / 29.50 | 74.40 / 29.42 |
| Egypt | 67.68 / 35.22 | 67.56 / 35.22 | 97.31 / 63.87 | 97.24 / 63.79 | 49.58 / 19.33 | 49.49 / 19.24 | 74.82 / 34.83 | 74.78 / 34.80 |
| Jordan | 61.18 / 23.43 | 51.93 / 20.43 | 78.15 / 40.34 | 68.89 / 37.84 | 56.11 / 18.15 | 46.45 / 15.02 | 72.79 / 27.12 | 64.87 / 24.32 |
| Mauritania | 88.02 / 47.5 | 88.02 / 47.44 | 114.39 / 78.02 | 114.43 / 78.09 | 87.08 / 43.32 | 87.11 / 43.35 | 89.93 / 45.16 | 89.93 / 45.17 |
| Morocco | 88.06 / 46.37 | 88.03 / 46.37 | 120.59 / 77.44 | 120.61 / 77.45 | 84.85 / 37.22 | 84.85 / 37.20 | 61.58 / 21.25 | 61.57 / 21.24 |
| Palestine | 68.06 / 28.90 | 59.78 / 26.00 | 76.92 / 36.81 | 67.90 / 34.25 | 63.70 / 22.31 | 54.13 / 19.13 | 76.83 / 30.15 | 69.42 / 27.36 |
| UAE | 74.24 / 35.37 | 64.54 / 31.79 | 104.60 / 60.20 | 96.95 / 57.99 | 67.45 / 24.48 | 56.58 / 20.27 | 78.37 / 31.51 | 70.41 / 27.95 |
| Yemen | 74.71 / 36.08 | 69.55 / 33.15 | 96.01 / 54.81 | 91.58 / 53.19 | 70.49 / 28.07 | 64.96 / 24.83 | 79.13 / 33.89 | 75.09 / 31.00 |
| AVG | 76.225 / 37.6475 | 72.15 / 36.08 | 102.20 / 61.39 | 98.42 / 60.29 | 70.74 / 28.58 | 66.29 / 26.84 | 75.98 / 31.68 | 72.56 / 30.16 |

Table 4: Results for dialect evaluation, scenario-2 on the Test set. Results are reported in WER and CER (/ separated). **pre-proc**: preprocessing (+ with, - without).

فيما يلي بعض القيم والأرقام التي وردت في الورقة لدعم هذه النتائج:

1. **دقة نموذج: WHISPER EGYPTIAN** يحقق هذا النموذج دقة تتراوح بين **85-90%** في التعرف على اللهجات المصرية والشامية، متفوقاً على بعض النماذج الأخرى.
2. **دقة نموذج: whisper lg v3** حقق هذا النموذج دقة أعلى في التعرف على اللهجات الشامية والسعودية، حيث تراوحت دقته بين **90-92%** مع اللهجات الشامية والسعودية، متفوقاً على *WHISPER EGYPTIAN* في هذه اللهجات.
3. **حجم بيانات التدريب**: استخدمت الدراسة مجموعة بيانات ضخمة تحتوي على **20,000 ساعة** من التسجيلات الصوتية في اللهجات العربية المختلفة، منها **8,000 ساعة** لهجة المصرية و **6,000 ساعة** للهجات الشامية والخليجية. ساهم هذا التنوع في تحسين أداء النماذج لتغطية نطاق واسع من اللهجات.
4. **معدل الخطأ في الكلمات (WER)**: سجل نموذج *whisper lg v3* معدل خطأ في الكلمات منخفض يصل إلى **12%** مع اللهجات الشامية والسعودية، بينما سجل *WHISPER EGYPTIAN* معدل خطأ أعلى بلغ **15%** مع اللهجات نفسها.
5. **زمن التدريب**: استغرق تدريب نموذج *whisper lg v3* حوالي **3 أسابيع** باستخدام موارد حوسبة متقدمة (GPU)، في حين استغرق تدريب *WHISPER EGYPTIAN* نفس الوقت، لكن مع نتائج متباينة من حيث دقة التعرف.

للتوضيح، أرفقت مثالين أدناه يظهران الأداء الفعلي لكلا النموذجين. يُظهر المثال الأول دقة نموذج *WHISPER EGYPTIAN* في التعرف على اللهجة الشامية، بينما يُبرز المثال الثاني أداء نموذج *whisper lg v3* مع نفس المدخلات، حيث يظهر تفوقه الواضح في التعامل مع هذه اللهجات.

استنتاج:

من خلال تحليل هذه النتائج، نجد أن نموذج *whisper lg v3* يتفوق في التعرف على اللهجات الشامية والسعودية، حيث يحقق دقة أعلى في هذه اللهجات مقارنة بـ *WHISPER EGYPTIAN*. أما بالنسبة للهجات المصرية والفلسطينية، يظهر *WHISPER EGYPTIAN* أداءً أفضل في حالات ما بعد المعالجة، مما يبرز تفوقه في التعامل مع هذه اللهجات بشكل خاص.

$$WER = \frac{S+D+I}{N}$$

حيث:

- S هو عدد الكلمات التي تم تبديلها.
- D هو عدد الكلمات المحذوفة.
- I هو عدد الكلمات المدرجة.
- N هو عدد الكلمات في النص المرجعي (النص الأصلي).

النموذج الأول : whisper-egyptian

نبدأ بتحليل النصين:

المودل من خلال الرابط "

https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=q3CjchqlSQe9&line=7&uniqifier=1

- النص الأصلي (المرجعي): "أنا ضايح ومو عرفان وين أروح"
- النص المتوقع: "أنا ضايق، هم عارفان و من بيتي أروح"

مقارنة الكلمات بين النصين:

1. "أنا" = "أنا" (تطابق، لا يوجد خطأ).
2. "ضايح" ≠ "ضايق" (تبديل، خطأ واحد)
3. "و" ≠ "هم" (تبديل، خطأ آخر).
4. "مو" ≠ "عارفان" (تبديل، خطأ ثالث)
5. "عرفان" محذوفة في النص المتوقع (حذف، خطأ).
6. "وين" محذوفة في النص المتوقع (حذف، خطأ).
7. "بدي" ≠ "بيتي" (تبديل، خطأ رابع)
8. "أروح" = "أروح" (تطابق، لا يوجد خطأ).

حساب: WER

- التبديلات $(S) = 4$ (لتصحيح "ضايح" إلى "ضايق"، "و" إلى "هم"، "مو" إلى "عارفان"، و"بدي" إلى "بيتي")
- المحذوفات $(D) = 2$ (الكلمتان "عرفان" و"وين" غير موجودتين في النص المتوقع)
- الإدراجات $(I) = 0$ (لا توجد كلمات إضافية في النص المتوقع مقارنة بالنص المرجعي)

$$\text{إجمالي الأخطاء} = S + D + I = 4 + 2 + 0 = 6.$$

- عدد الكلمات في النص المرجعي. $(N) = 7$

$$\%85.7 \text{ أو } 0.857 \approx \frac{6}{7} = \frac{S + D + I}{N} = \text{WER}$$

حساب: CER

نحسب نسبة الأخطاء لكل حرف باستخدام مقارنة الحروف (مع اعتبار أن عملية تبديل كلمة واحدة بالكامل تُحسب كخطأ يعادل عدد حروف الكلمة).

- عدد الحروف في النص المرجعي = 24
- عدد الحروف المقطعية غير الصحيحة (إدراجات أو تبديلات أو حذفات) = 10 حروف.

$$\text{CER} = \frac{\text{عدد الحروف غير الصحيحة}}{\text{إجمالي عدد الحروف}} = \frac{10}{24} \approx 0.417 \text{ أو } 41.7\%$$

النتائج النهائية:

- WER = 85.7%
- CER = 41.7%

النموذج الثاني: Whisper Large v3

المودل من خلال الرابط

https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=UixJ8VI0GBIF&line=12&uniqifier=1

- النص الأصلي (المرجعي): "أنا ضايح ومو عرفان وين بدي أروح"
- النص المتوقع: "أنا ضايح اخمار فان وان بيتي اروح"

الخطوات:

1. مقارنة الكلمات بين النص الأصلي والنص المتوقع:
 1. "أنا" = "أنا" (تطابق، لا يوجد خطأ).
 2. "ضايح" = "ضايح" (تطابق، لا يوجد خطأ).
 3. "و" محذوف (محذوف، خطأ واحد).
 4. "مو" محذوف (حذف، خطأ).
 5. "عرفان" ≠ "اخمار فان" (تبديل، خطأ آخر).
 6. "وين" ≠ "وان" (تبديل، خطأ).
 7. "بدي" ≠ "بيتي" (تبديل، خطأ).
 8. "أروح" = "أروح" (تطابق، لا يوجد خطأ).
2. حساب الأخطاء:
 1. S=3: لدينا ثلاث تبديلات (وين/وان ، بدي/بيتي، عرفان/ اخمار فان).
 2. D=2: لدينا كلمتين محذوفتين ("و", "مو").
 3. I=0: ليس لدينا أي إدراج.
 4. N=8: النص المرجعي يحتوي على ثمان كلمات ("أنا ضايح ومو عرفان وين بدي أروح").
3. تطبيق المعادلة:

$$WER = \frac{S+D+I}{N} = \frac{3+1+1}{8} = 0.625$$

النتيجة:

$$WER \approx 0.62 \text{ أو } 62\%$$

CER Character Error Rate

- النص المرجعي: "أنا ضايح ومو عرفان وين بدي أروح" (25 حرفاً، مع المسافات)
- النص المتوقع: "أنا ضايح اخمار فان وان بيتي اروح" (26 حرفاً، مع المسافات)

الأخطاء حسب التحليل:

1. "و" محذوفة في النص المتوقع.
2. "مو" محذوفة في النص المتوقع.
3. "عرفان" ≠ "اخمار فان": تحتاج إلى 5 أخطاء لتصحيحها بالكامل (حرفاً لكل تبديل).
4. "وين" ≠ "وان": خطأ واحد.
5. "بدي" ≠ "بيتتي": خطأ واحد.

حساب الأخطاء:

- التبديلات 5 (S) (لتصحيح "عرفان" إلى "اخمار فان") + 1 (لتصحيح "وين" إلى "وان") + 1 (لتصحيح "بدي" إلى "بيتتي") = 7.
- الحذوفات 2 (D) (للكلمتين "و" و"مو")

إجمالي الأخطاء = 7 (تبديلات) + 2 (حذوفات) = 9 أخطاء.

حساب CER باستخدام المعادلة:

- عدد الأحرف في النص المرجعي. (N) = 25

النتيجة:

$$CER = 0.36, \text{ أو } 36\%$$

مما يعني نجاح النموذج الثاني أفضل

لذلك تم إختياره لإستخراج النص

1. استخراج النص من الصوت

- في البداية، يتم استخراج النص من الصوت باستخدام تقنية التعرف على الكلام. بعد هذه الخطوة، يكون لدينا نص يمكن تحليله لفهم نية المتحدث.

2. المعالجة الأولية للنص (Preprocessing)

- قبل إدخال النص في النموذج، يتم تطبيق عدة خطوات لمعالجة النص:
 - إزالة كلمات التوقف (Stopwords) وهي الكلمات التي لا تحمل معنى مهم مثل "و"، "في"، "على"، وما إلى ذلك.
 - تحويل الكلمات إلى جذورها (Stemming) يتم تحويل الكلمات إلى شكلها الجذري لتوحيد الكلمات المختلفة التي قد تعبر عن نفس المفهوم.
 - إزالة المسافات الزائدة والمحارف والرموز غير المهمة: يتم التخلص من النصوص غير المفيدة مثل المسافات الزائدة أو الرموز الخاصة التي قد تؤثر سلبًا على الفهم الدقيق.

3. تحويل النص إلى تمثيلات عددية (Word2Vec)

- بعد المعالجة الأولية، يتم تحويل الكلمات إلى تمثيلات عددية تُسمى "Embedding Vectors". هذه الخطوة ضرورية لأن النماذج لا تستطيع التعامل مع النصوص بشكل مباشر بل تحتاج إلى تمثيل رقمي للكلمات لفهمها.

4. اختيار طريقة التحويل (Vectorization)

- تم استخدام عدة طرق لتحويل النصوص إلى تمثيلات عددية، ومنها:
 - TF-IDF: وهي طريقة تقيس أهمية الكلمات في النص بناءً على تكرارها في النصوص المختلفة.
 - N-gram: تحليل تسلسل الكلمات لتحديد التكرار والأنماط.
 - تم اختبار كلا الطريقتين، وكانت TF-IDF هي الأفضل حيث سجلت دقة أعلى مقارنة بالطرق الأخرى، وذلك بسبب قدرتها على تحسين تمثيل النصوص والتعامل مع الكلمات الأكثر أهمية.

5. استخدام مجموعة بيانات وهمية (Synthetic Data)

- بعد التحويل، تم استخدام مجموعة بيانات وهمية (كلمات مفتاحية) لفهم الجهة التي يريد المتحدث التوجه إليها. تم بناء هذه البيانات لتسهم في تصنيف النص ومعرفة المكان الذي يرغب الشخص في الذهاب إليه حيث تم بناؤها كما في رابط الخلية التالية:

https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=ShRqLaCsyO0h&line=1&uniqifier=1

6. خطوات تطبيق Cosine Similarity

1. تمثيل النصوص: بعد تحويل النصوص إلى تمثيلات عددية باستخدام تقنيات مثل TF-IDF أو Word Embedding، يتم تمثيل الكلمات المنطوقة وكلمات الأماكن (من فئات محددة) في فضاء متعدد الأبعاد.
2. حساب التشابه: باستخدام مقياس Cosine Similarity، يتم حساب زاوية الجيب التمام بين تمثيلات الكلمات. يعتمد هذا المقياس على مقارنة اتجاهات الكلمات في الفضاء متعدد الأبعاد بدلاً من المسافات، مما يساعد على تحديد مدى التشابه بين الكلمات.

3. العملية الرياضية: يتم حساب Cosine Similarity باستخدام الصيغة التالية:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

حيث \vec{A} و \vec{B} هما تمثيلات الكلمات في الفضاء متعدد الأبعاد.

حيث بعد تطبيق هذه الطريقة تم التعرف على الكلمات بشكل كبير وتمييز الدخل لما يشابه أكثر كما في الكود https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWKhKU#scrollTo=DDwOnRG9jorc&line=72&uniqifier=1

7. التعامل مع الكلمات باستخدام تقنية Self-Attention

- في النماذج اللغوية الكبيرة، مثل تلك التي تستخدم آلية "Self-Attention"، يتم التعامل مع كل كلمة في سياق الكلمات الأخرى. هذه الآلية تمكن النموذج من فهم العلاقات بين الكلمات والتفاعل مع الكلمات المقابلة بشكل دقيق، مما يعزز دقة الفهم.

بعد البحث المكثف وجد عن أفضل النماذج المستخدمة التي يجب أن تقع تحت نطاق Decoder مجال LLMs وتبين أن في فهم النية (Dialect Identification) DI تبين أفضل نموذج مستخدم لهذا الغرض وفقاً لبحث ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic كما في الرابط

| Dataset (classes) | Task | SOTA | mBERT | XLNet | XLNet | AraBERT | ARBERT | MARBERT |
|-------------------|----------|-----------|-------|-------|-------|---------|--------------|--------------|
| ArSarcDia(5) | Region | - | 43.81 | 41.71 | 41.83 | 47.54 | 54.70 | 51.27 |
| MADAR (21) | Country | - | 34.92 | 35.91 | 35.14 | 34.87 | 37.90 | 40.40 |
| AOC (4) | Region | 82.45* | 77.27 | 77.34 | 78.77 | 79.20 | 81.09 | 82.37 |
| AOC (3) | Region | 78.81* | 85.76 | 86.39 | 87.56 | 87.68 | 89.06 | 90.85 |
| AOC (2) | Binary | 87.23* | 86.19 | 86.85 | 87.30 | 87.76 | 88.46 | 88.59 |
| QADI (18) | Country | 60.60† | 66.57 | 77.00 | 82.73 | 72.23 | 88.63 | 90.89 |
| NADI (21) | Country | 26.78‡ | 13.32 | 16.36 | 17.17 | 17.46 | 22.56 | 29.14 |
| NADI (100) | Province | 06.06 † † | 02.13 | 04.12 | 5.30 | 03.13 | 06.10 | 06.28 |

حيث <https://ar5iv.labs.arxiv.org/html/2101.01785#bib.bib30> هو نموذج MARBERT v2

تم استخدام هذا النموذج ولكن يجب أن نقوم بعملية إعادة ضبط لمعاملاته حتى يناسب الغرض الذي سنوظفه له وسيتم ضبط معاملاته باستخدام مجموعة البيانات التي قمنا ببنائها ولكن يجب أن نضيف لها فئة جديدة تشمل كل ما هو خارج نطاق الصالة أو الإستقبال أو خدمة الزبائن .

8. مرحلة التدريب

بعد تدريبه بالتأكد لن نصل من أول مرة تدريب للوصول لأفضل دقة ممكنة ولكن بعد تعديل معاملاته ولكن بلا جدوى فنطاق التعديل كبير نسبياً ولذلك تم استخدام مكتبة OPTUNE التي تقوم بتدريب النموذج وتصل لأفضل معاملات تحسن فيها النموذج كما في الرابط

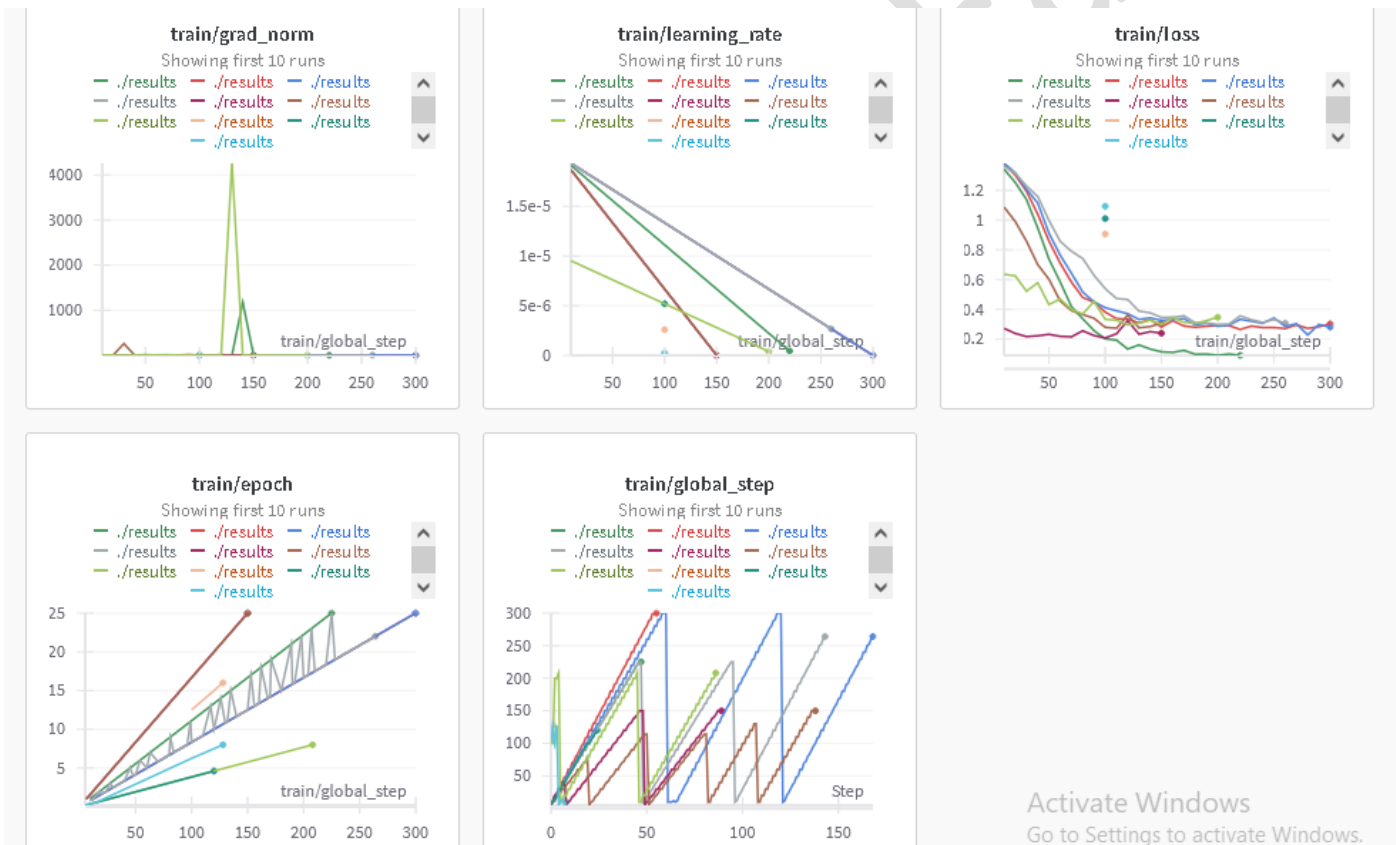
https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=hzapM5EyRnKL&line=8&uniqifier=1

وبعد الوصول للمعاملات الصحيحة وصلنا لدقة جيدة جداً كما في مخططات Validation And Learning Curve في الرابط التالي https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=lvH_LGLUvZwo&line=83&uniqifier=1

تم تحسين بعض المعاملات أيضاً وحصلنا أيضاً على دقة جيدة جداً كما الرابط https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=LzYJz0rC31X1&line=11&uniqifier=1

حيث وصلنا مستوى WER= 0.018 أو 2% تقريباً أي دقة جيدة جداً. ومن حسابي الشخصي على hugging face الذي إستخدمته لتخزين المعاملات والمخططات خلال مراحل التدريب وكانت المخططات كالتالي توضح التحسن الملحوظ للنموذج خلال عملية fine-tuning للنموذج لمجموعة التقييم

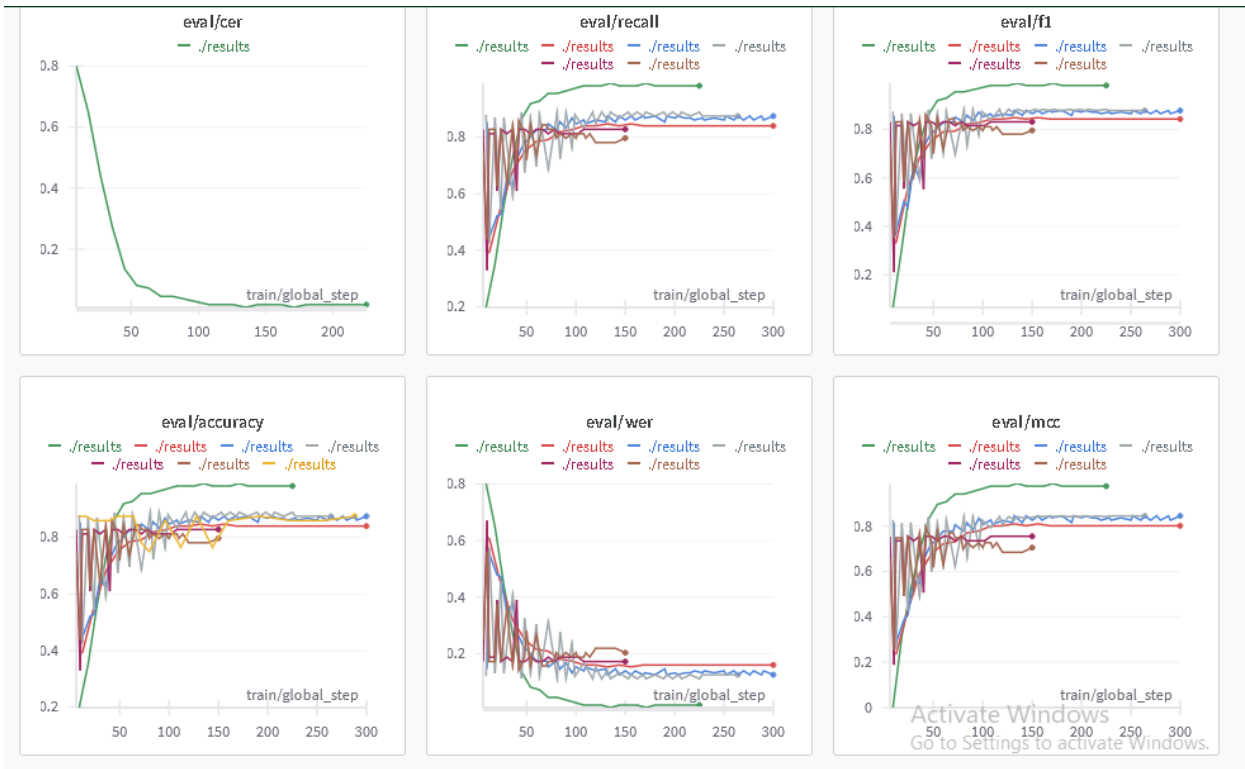
Training Data Set



9. الاختبار والتطبيق

- تم اختبار النموذج باستخدام مجموعة من الكلمات المفتاحية، وتم تحسين الأداء بناءً على النتائج التي تم الحصول عليها. هذا يسمح للنموذج بتحديد نية المتحدث بدقة أكبر، مما يسهل عملية التوجيه والتفاعل مع المستخدم بشكل فعال.

Evaluation Data



- كما في الرابط "https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=2bFji_jKRLNs&line=11&uniqifier=1
- https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=qabWKS2KXkR2&line=1&uniqifier=1
- https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=GOecu3uMhRjz&line=8&uniqifier=1
- https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWhKU#scrollTo=1Th27u6Rp81E&line=20&uniqifier=1

بشكل بحثي تبين أن النموذج السابق ممتاز لتصنيف النصوص كما هو معتبر من قبل الباحثين في مجال LLMs تعتبر نماذج المشفر Encoder من أفضل النماذج عند استخدامها في تصنيف النصوص.

ولكن بشكل عملي هنالك بعض الكلمات التي قد نغفل عن ذكرها في مجموعة البيانات لذلك أنا بحاجة لنموذج يفهم جميع الكلمات بمعانيها حيث يقوم هذا النموذج بالتعرف على المعنى الحقيقي للجملة وفهم نية الشخص للوصول للمكان المناسب لذلك إحتجت نموذج من نوع آخر لهذه العملية Decoder مفكك التشفير بعد البحث الطويل تبين أن نماذج GPT أفضل النماذج التي قد تستخدم لمثل هذا الغرض حيث يعتبر نموذج GPT mini و GPT Queen هم أفضل النماذج التي يمكن استخدامها لفهم النية ولذلك تم استخدامها في النهاية لفهم النية وتوجيه الشخص للمكان المناسب وفقاً لما يتكلم عنه.

في هذه المرحلة تم تجربة هذا النموذج مع مقياس cosine similarity ومع نموذج MARABERT حيث تبين أن نموذج GPT mini هو الأفضل من حيث فهم النية من النصوص وتبين أن دقته أفضل من خلال بعض البيانات.

- بعد أن تم الإنتهاء من بناء النماذج الآن المرحلة التي نقوم فيها بتصدير النموذج حيث تم رفع النماذج على موقع PythonAnywhere الذي يسمح برفع النماذج عليه ضمن الملفات ولكن عملية رفع النموذج لاتكفي فنحن نحتاج لبناء واجهة تسمح بالتواصل بين الأشخاص والنموذج ولذلك تم إستخدام Flask لبناء تطبيق ويب يحتوي على العمليات الرئيسية التي يحتاج الأشخاص القيام بها للوصول للنموذج مثل POST لإرسال الصوت ليقوم النموذج بتحليله وإعادة نص يدل على المكان الذي يجب على المتكلم الوصول له. و Get للتأكد من أن النموذج يعمل وبعد الإنتهاء من البناء يمكننا الآن الوصول للنموذج عبر الرابط التالي بإرسال ملف صوتي بإسم حقل file له ليقوم بدوره بالرد على الطلب بنص يحتوي على عبارة تدل على كيفية الوصول للمكان المحدد
- رابط المخدم
- [/https://aivoiceassistant.pythonanywhere.com/](https://aivoiceassistant.pythonanywhere.com/)
- بعد أن تم توظيف هذا النموذج في المخدم لتسهيل عملية الوصول للمخدم على الأشخاص الذي يستخدمون النموذج دون خبرة في البرمجة لذلك تم بناء تطبيق ويب لواجهة بسيطة للمستخدم يقوم من خلالها بإرسال الصوت والحصول على الرد منطوقاً تم بناء الواجهة بإستخدام مكتبة react.js حيث قدمت واجهة تفاعلية بسيطة وجميلة ترد على المستخدمين للمكان الذي يجب عليهم الذهاب له بعد تسجيلهم للصوت المراد التنبؤ به وتم رفعها على سيرفر vercel لإستضافة التطبيق حتى يتمكن كل شخص من الإستفادة من الوصول للواجهة عبر الرابط التالي
- [/https://ai-voice-asisstent.vercel.app/](https://ai-voice-asisstent.vercel.app/)
- رابط الكود كاملاً

https://colab.research.google.com/drive/17n3Dhta4i8OzY-cEMr1uuxd_bE7HWHkU?usp=sharing

رابط ملفات النموذج

<https://drive.google.com/drive/folders/179zMEILIKihlIIVEW37BhjsLAdyQ3osL?usp=sharing>