مقدمةٌ في معالجةِ اللغةِ العربيةِ

Serry Sibaee سري السباعي
Intern researcher at PSU in NLP
باحث متدرب في معالجة اللغة العربية في جامعة الأمير سلطان

github: https://github.com/serrysibaee
linkedin: https://www.linkedin.com/in/serry-sibaee/

مقصدنا السؤال والنقاشُ ... لا كثرة الكلام والرِّقَاشُ

فاسأل وناقش دونما تحرُّجِ ... فالعلمُ نصفُه السؤالُ تدرجِ

لَكِنْ بشرطِ الذَّوْقِ والتَّأَدُّبِ ... فسيِّئُ الأخلاقِ يُحَرَمْ يُسْلَبِ

ومن يجدْ من خطأٍ أو سهوِ ... يُهدِيْ جَميلا دونما تَرَوِّ

فلتحزموا العقول والأذهانا ... فالرحلة خليطٌ استبانا

الرقاش هو زخرفة اللوح وهو الحية لذلك في جلدها

[بعض الكلمات مكتوبةٌ عروضيا فزيدَ فيها الألف مثل "الأذهانا"]

فهرست المحاضرة:

١. تعريف بالمجال وأصوله وتعريفها

٢. أركان العلم الأساسية

٣. أهم أقسام العلم بالنسبة للعربية

٤. أحدث التقنيات

٥. تطبيق عملي على ثلاثة تطبيقات

٦.مفاتيح للبدء في المجال

كيْفَ نَسْتَوْعِبُ مَا نَقرَأُ ؟

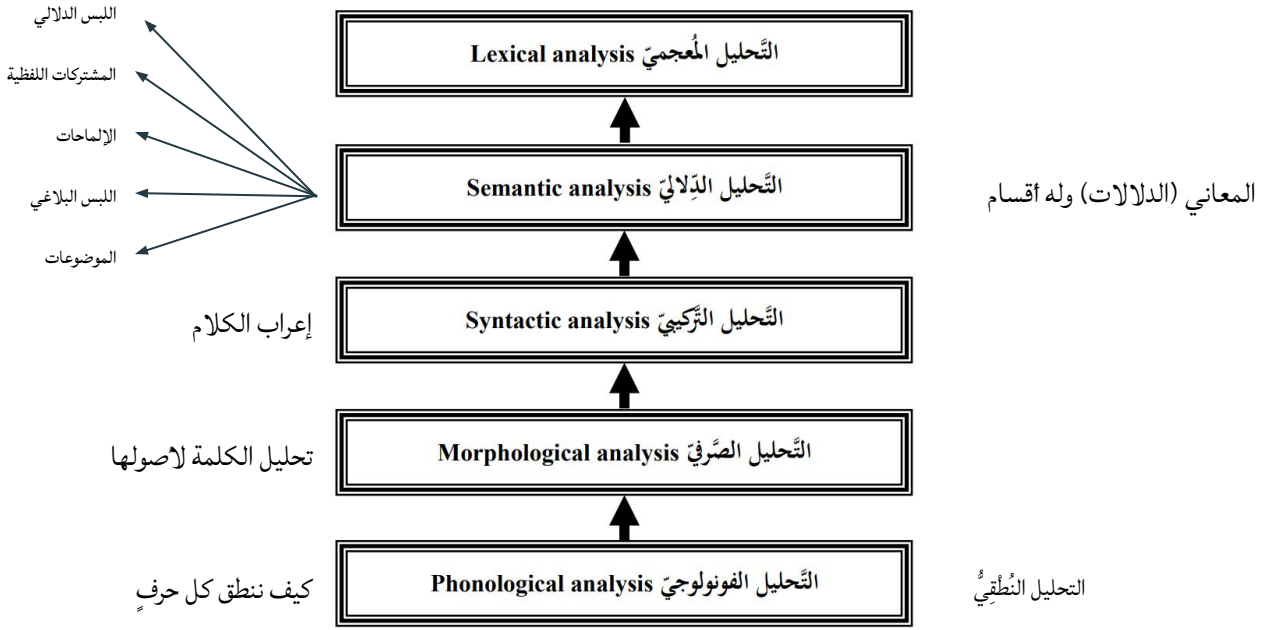
الحمدُ لله المُعِيدِ المُبْدِي ... حمدًا كثيرًا وهو أهلُ الحمدِ

وما مِثْلُهُ في النَّاسِ إلَّا مُمَلَّكًا ... أبو أُمِّهِ حيٌّ أبوه يُقَاربُه

أي ليس في الناس حي يقاربه إلا ممّلكًا أبو أمه أبوه أي هشام

انظر منظومة التاريخ لعلي بن الجهم
وكتاب تلخيص المفتاح

الحمدُ لله المُعِيدِ المُبْدِي ... حمدًا كثيرًا وهو أهلُ الحمدِ



كتاب مقدمة في حوسبة اللغة العربية

تعريف المجال

هو مجال بَيْنِيٌّ (أي أنه خليطٌ عدة مجالات أخرى) هدفه معالجة النصوص صوتيةً أو مكتوبة لإنتاج خدماتٍ مفيدة

فماذا تتوقعون هو مكوَّنٌ من ؟

1. علوم الحاسب
2. علوم اللغويات
3. الرياضيات (علمُ التَعَاليم)
4. علم تعلّم الآلة (وهو فرع من السوابق ولكن أُفْرِدَ لبَسْطَته)

المقصد من الذكر: المقصد الاساسي من ذكر هذه العلوم هو الإلماح لعلاقة كلٍّ منها بهذا المجال لا ذكرَ تفاصيلها فهذا لا يكون إلا في محاضرات كثيرة

علوم الحاسب

خلاصتها في دقائق:

- الأصول سبعة:
  - سير البرنامج (التصور البرمجي)
  - المتغيرات
  - العمليات وأنواعها
  - الإدخال والإخراج
  - الشروط المنطقية
  - الحلقات (المَكْرَر)
  - تتبع الأخطاء ومعالجتها

برزخ نوع البرمجة والمشهور الشيئية وركناها

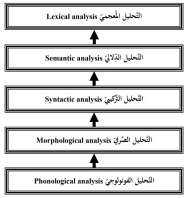
- القولبة
- الوراثة
- الفرع التطبيقي (هياكل البيانات والخوارزميات)
  - هياكل البيانات وأصولها
    - المصفوفات وفروعها كالارتال و الأكداس
    - العقديَّات (الرؤوسية) منها المتصلات والبيانات
    - المُرمزات مثل القاموس في مُبيثن
  - الخوارزميات وأصولها ثلاثة
    - حساب التعقيد (الكركبة)
    - الترتيب
    - البحث

Computer Science I Dr. Chris Bourke

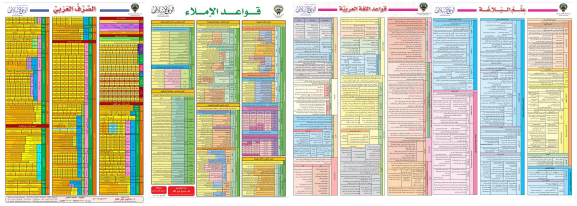
علوم الحاسب مع المجال

فن الحوسبة وعلاقته بالمجال أمران أساسيان:

1. التفكير المنطقي (البرمجي): كيف نستطيع أن نمثل أفكارنا على ما يفهمه **الحاسوب** وخاصة في مسألة المصفوفات والأرقام
2. الخوارزميات وهياكل البيانات: كيف يمكن أن نخزن ونسترجع البيانات ونتعامل معها بشكل سريع فعال فما زالت المعلومات تزيد أُسًّيًا
    a. قواعد البيانات (المتجهية) وغيرها
    b. وحدة معالجة المرصوصات (الموترات) Tensor Processing Unit
    c. أي لغة نختار وكيف نبرمج
    d. تصميم خوارزميات جديدة أو تحسين الموجود



اللغويات (كلها في ورقة .)



- الدلالة
- أساليب العربية من التراث إلى المعاصرة
- فنون اللغة العربية وعلاقتها بالمعالجة (انظر الملخصات)

صَرْفٌ بَيَانٌ مَعَانِي النَّحْوُ قَافِيَةٌ شِعْرٌ عَرُوضُ اشْتِقَاقُ الخَطُّ إِنْشَاءُ

مُحَاضَرَاتٌ وَثَانِي عَشْرِهَا لُغَةٌ تِلْكَ العُلُومُ لَهَا الآدَابُ أَسْمَاءُ

- الفصحى والعامية: الاصل أن نستعمل الفصحى ولكن هذا المجال غايته التجارة والربح ولذلك استُعملت العامية بأخرة وهذا ضار على المدى البعيد
- العلم هذا بشكل مختصر قائم على النظرية الوصفية (ومنها جاءت التعبيرات الإحصائية) وهناك رأي آخر لتشومسكي في نظريته التحويلية وقد نفعت في اللغات الاصطلاحية (مثل لغات البرمجة والمنطق) ولكنها لم تُجْدِ كثيرا في النماذج وتمثيلها

يرجع لسلسة اللغة في ورقة وفيها النحو والصرف والبلاغة والإملاء في ورقة ولمن أحب فهناك ألفية لسان العرب في علوم الأدب للآثاري جمع فيها عشرة علوم من علوم العرب (رابط)

خلاصة الرياضيات

- المنطق الرياضي (الجبر المُجرد)
- التحليل العددي (المقاربة العددية)
- الجبر الخطي
- طوبولوجيا (الفضاءات العامة)
- الهندسة التفاضلية

لمزيد من التفاصيل عن دراسة الرياضيات البحتة انظر المقطع

الرياضيات [علم التعاليم] مع المجال

خلاصة العلم وتطبيقه في هذا الفن هو التحليل الرياضي والإحتمالات ولكن هذين مضمَّنان التالي (انظر الملخصات):

- الجبر الخطي (الجخط [نحت]): وهو معالجة البيانات مصفوفيًّا تسريعا وتوفيرا للوقت [كل شيء يجب أن يعرض كمصفوفة]
- الإحصاء والإحتمالات وخلاصته التوزيع الإحصائي على أي توزيع يكُن.
- التفاضل: حيْثُ يُدرسُ فيه التغير وتأثيره على الدوال وتغييرها

ولمن أحب تذوق شيء من ذلك لمن بلغ مبلغًا فلينظر ورقة ستانفورد أو بيركلي

لمزيد من تفاصيل دراسة الرياضيات لتعلم الآلة انظر الكتاب رابط

أصول تعلم الآلة (التعالئ) إلى التعلم العميق

- أنواعه الأساسية ثلاثة
  - تعلم إشرافي (س$_{ن}$,ص$_{ن}$)
    - توقع
    - تصنيف
  - تعلم غير إشرافي (س$_{1}$,س$_{2}$,س$_{3}$, ... , س$_{ن}$)
    - تجميع
  - تعلم تعزيز: بيئة مع جزاءٍ للفعل حسنا وسوءا

ولعلي ألخص لكم المجال وجميع تطبيقاته في توضيح بسيط وثلاث أسس: تمثيل المدخلات و حُسنَ العرض للسرعة (المصفوفات والمرصوصات) واستعمال النموذج المناسب

لمن أحب مقدمة قصيرةً عن تعلم الآلة كتاب تعلم الآلة في 100 صفحة لطيف جدا ""machine learning in 100 pages

تعلم الآلة (التعالئ) مع المجال

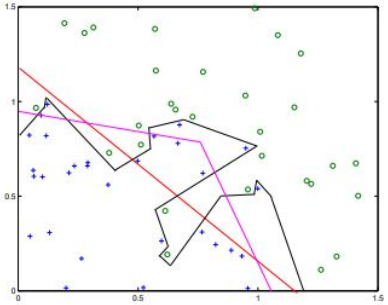
- ضحلٌ: وهو قواعد فيها نوع (بعض) الذكاء وأهمها الانحدار الخطي واللوجستي وهو أساس وقاعدة التعلم العميق واستعملوا كثيرا في التصنيف خاصة
- وعميقٌ باستعمال الشبكات العصبية وهي أشكال وألوان ولكن الذي منها للمعالجة كالتالي
  - فهم الكلمات: word2vec , BERT
  - فهم السياق
  - تصنيف الكلام
  - الترجمة (نموذج من سياق (تتابع) لآخر)
  - توليد النصوص والكلام
  - تحويل الصور لنصوص

ولعلي ألخص لكم المجال وجميع تطبيقاته في توضيح بسيط وثلاث أسس: تمثيل المدخلات و حُسنَ العرض للسرعة (المصفوفات) واستعمال النموذج المناسب

نظرية التعلم الإحصائي (اختياري) ؟

نظرية التعلم الإحصائي هي الوحش ومن أبوابه:

- التنبؤ inference
- الصوغ formalizaion
- التعميم generalizaion
- تقارب منظم Uniform convergence
- الحساب التجريبي Empirical error
- الحجم البعدي
- التنظّم
- لا شيء بلا مقابل (نظرية لا غداء مجانيا)



هنا تسقط كل التوقعات لأننا نثبت أن كل نموذج وخوارزمية لا بُدَّ تفشل

ولمن أحب ورقةً فلينظر الرابط

تاريخ العلم ١

لخصت تاريخ العلم من عدة مصادر ولعلي أذكر هنا خلاصته أو أكتب رؤوس أقلام مع رابط للورقة إن تيسر.

ما زال الناس يريدون الحواتل (الروبوتات) لتساعدهم وبرز منهم (يان تشي) و (بديع الزمان الجزري) و(هنري لوس وأبوه) ثم نهضت الرياضيات وملكتنا أدوات أنجعت علوم الطبيعة وظهر فيها ميكنة الحل بعد التداول والمعالجة.

ثم جاءت مرحلة اتقان الرياضيات مع غوتليب وراسل ثم كسر الأمر ونقضه غودل وظهرت نظرية الصعوبة (التعقيد) complexity وهي لُب لباب علوم الحاسب.

ثم في بداية الأربعينيات وبزغ نجم جون نيومان وهذا في الحرب العالمية الثانية وما بعدها. واقترب حُلم الحاتل (شبيه البشر) ومن أضغاثه ظهرت صعوبة وتعقيد اللغة.

ورجا الناس محاكاة العقل البشري وظهر تورينج وتشيرش فالأول شرّط الذكاء ونمذجه والثاني يسره لتبين وتُدرس فعاليته بعرض الدالة من منظور حوسبي

وعام ١٩٥١ صمم أحد الباحثين ذكاء ليلعب الضامة وظهر أيضا البحث الشجري وحوسبة المنطق وهذا بين [١٩٥٧-١٩٧٤] وهو العصر الذهبي وفيه ظهرت معالجة اللغات متعمقةً دلاليا مع الشبكات الدلالية وجاء مفهوم الدراسة المثالية ثم بعد الاستنتاج توضع في بيئة حقيقية (واقعية)

ومع هذا النجاح تصاعدت آمال الناس وظنوا أنهم ختموها ونسوا أنها دنيا [كمالها في نقصها] وتناقص التمويل تتابعا حتى انقطع عام ١٩٨٠ (وهنا ملاحة أن في عام ١٩٥٨ صمم روزنبلانت الشبكة العصبونية وهو مفتاح التعلم الحاسوبي)

وفي الثمانينات طوّر هينتون وروميل هارت خوارزمية تعلم الشبكات متعددة الطبقات ووقتها قامت بريطانيا واليابان بتكلّف مشاريع باهظة لمعالجة اللغة

ملخص من مقال محمد عطية أحمد [حوالي ٧٠ صفحة]
انظر كتاب العربية والذكاء الصناعي

تاريخ العلم ٢ أشهر الخوارزميات تاريخيا [المصدر]

تحليل المركبة الأساسية PCA	١٩٠١	الانتشار الخلفي وخوارزمية 3iD	١٩٨٦
المنطق الثلاثي (بدايات الضباب)	١٩٢٠	خوارزمية 4.5iD	١٩٩٣
وضوح فكرة عدم التأكد والضبابية	١٩٣٧	المتجهات المعتمدة وخدع الآنوية وخوارزمية التجريب bs	١٩٩٥
المنطق العتبي	١٩٤٣	الغابات العشوائية وأيضا الشبكة الإلتفافية cnn	١٩٩٥
اختبار تورنج	١٩٥٠	الذاكرة العصبونية (التداخلية) RNN LSTM	١٩٩٧
الخلية العصبية المحوسبة (روزنبلات)	١٩٥٨	الغابات العشوائية ذات القرار	٢٠٠١
المنطق الضبابي (لطفي زاده)	١٩٦٥	التعلم السريع	٢٠٠٦
تطور الخلية المحوسبة	١٩٦٩	تعلم الطبقات الآناني (قطعة قطعة)	٢٠٠٧
أول تطبيق للمنطق الضبابي	١٩٧٢	تدريب image net	٢٠١٢
الخرائط ذاتية الترتيب	١٩٨٢	وانتشر المجال وبزغ نجمُه وتنوعت فروعه بعد ذلك	٢٠١٢-إلى الآن

بدايات الحوسبة اللغوية العربية

بدايتها مع ابراهيم انس وعلي حلمي دراسات احصائية على الصحاح واللسان والتاج وكذا عدد من الإحصائية القرآنية	١٩٧١-١٩٧٣
مؤتمر الرباط وظهر منه كتاب اللسانية العربية	١٩٨٣
أدلة الكتابة والنظام الصرفي	١٩٨٥
محمد مراياتي نظام الاشتقاق الحوسبي	١٩٨٧
بناء معجم تركيبي وكذا نظام للتوليد والتحليل الصرفي والنحوي	١٩٨٩
الحوسبة وخصائص العربية مع بناء مُشَكِّلات ومترجمات	١٩٩٢

ملخص محاضرات في اللسانيات التطبيقية رابط وهناك كتاب من أربع أجزاء لم أطلع عليه إلا سريعا

ملخص مجال معالجة اللغة

- النصوص جمعها واستيرادها وتنظيفها
- العرض (التمثيل) كيف نعرض البيانات (النصوص)
  - الكلمة والجملة
    - العرض العَدِّي: في مصفوفة صفرية أو عد نسبي أي إحصائي (خجم المصفوفة كبيرة عادةً هنا)
    - مُتعلم (صغير الحجم ولكنه معصور الفائدة وأشهر نموذج الكلمة المُتجِهة word2vec
  - تقسيم النصوص
    - حرفا حرفا
    - كلمة كلمة
    - أجزاء
- النموذج (كيف نجعل الحاسب يتعلم العلاقات)
  - إحصائي: مثل الرمزي المنطقي (لوجستك) و حدْسِ بايز و المتجهات المعتمدة وسلاسل ماركوف والتقسيم النوني (الكلنوني) احصائيا وهو يستخدم مع النماذج السابقة
  - نموذج عميق: الشبكة العصبونية بأنواعها تلافيفية أو ارتدادية أو بدائية أو تنبهية وغير ذلك
- التطبيقات: وأصلها أمران توليدي أو فهمي [تَوَقُّعي و استيعابي] حيث الأول ينتج (يكتب) نصوصا والثاني يستوعب النصوص وعلاقاتها
- ربطها باللغويات: (وهو آخر فصل من كتاب جرفاسكي) وفيه محاولة تفسير النماذج العميقة مع مقارنتها بالهياكل المعنوية [انظر شريحة الدلالة] والتراكيب النحوية وغيرها

لمزيد من التفاصيل انظر الرابط

معالجة النصوص العربية

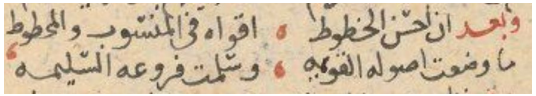
الأصول التي ذكرها الأستاذ حبش في كتابه "مقدمة في معالجة اللغة العربية": أربعة (استخلصتها) و سَأُرَكِّزُ على ما كان منها عن معالجة المكتوب وهي

1. الخط وما فيه من إعجام ورسم وتنوع الحروف فيما تستخدمه اللغات الأخرى من الحروف العربية
    a. ويدخل تحت هذا الفصل ما تعارف عليه الباحثون من تنظيف (تهيئة) النصوص العربية قبل استعمالها في النماذج
2. الأصوات والتهجي: ولم أُطِلْ فيه ولكن يدخل فيه كتابة الأسماء الأعجمية من لغتها للغتنا وتصحيح الأخطاء
3. الصرف وهو أطولها وفيه
    a. التحليل الصرفي
    b. رفع الإبهام
    c. التقطيع
    d. التشكيل
4. السياق ويدخل فيه معاني الدلالات

ثم بعد ذلك يكون بناء النماذج

لمزيد من التفاصيل انظر كتاب الدكتور نزار حبش وترجمته للدكتورة هند خليفة "مقدمة الى معالجة اللغة العربية"

الخَطُّ وتوابعه



وبعد إن أحسن الخطوط ... أقواه في المنسوب والمخطوط
ما وُضعت أصوله القويمة ... وسلمت فروعه السليمة

تنظيف (تهيئة النصوص) وهي عملية تعارف عليها الباحثون لإزالة مالا نفعَ فيه ومن ذلك:

- إزالة التطويلات [مرحبا , مـــرحـــبـــا]
- والحركات وهذه فيها إشكالات [تراجع الورقة البحثية]
- والهمزات
- والمدات
- إزالة الكلمات المعتادة المتكررة بكثرة (حروف الجر والضمائر وغيرها)

الأبياتُ فوقُ من منظومة بضاعة المجود في الخط وأصوله للسنجاري

وبَعْـــدُ: إنَّ أحْـــسَنَ الخُطوط ... أقْـــواهُ فى المنسُـــوب والمخطوط
مـــا وُضِـــعَتْ أصـــولُهُ القَـــوِيمَهْ ... وسَلِمَتْ فـــرُوعُـــهُ السليـــمـــهْ

مثال برمجيٌ

```python
from camel_tools.utils.normalize  import normalize_alef_maksura_ar
from camel_tools.utils.normalize  import normalize_alef_ar
from camel_tools.utils.normalize  import normalize_teh_marbuta_ar

sentence = "هل ذهبت إلى المكتبة"
print(sentence)

# Normalize alef variants to 'ا'
sent_norm = normalize_alef_ar(sentence)
print(sent_norm)

# Normalize alef maksura 'ى' to yeh 'ي'
sent_norm = normalize_alef_maksura_ar(sent_norm)
print(sent_norm)

# Normalize teh marbuta 'ة' to heh 'ه'
sent_norm = normalize_teh_marbuta_ar(sent_norm)
print(sent_norm)
```

```python
from camel_tools.utils.dediac  import dediac_ar

sentence = "هَلْ ذَهَبْتَ إِلَى المَكْتَبَةِ"
print(sentence)

sent_dediac = dediac_ar(sentence)
print(sent_dediac)
```

```
هل ذهبت إلى المكتبة
هل ذهبت الى المكتبة
هل ذهبت الى المكتبة
هل ذهبت الي المكتبة

هَلْ ذَهَبْتَ إِلَى المَكْتَبَةِ
هل ذهبت إلى المكتبة
```

الأصوات والتهجي

أغلب تطبيقات هذا القسم متعلقة بالصوتيات وليس هذا موضع درسنا ولكن دخل فيه أيضا قسمان يُمكن ويحسن الإشارة لهما لتعلُّقِهما بموضوعنا وهما كتابة الأسماء الأعجمية وكذا تصحيح الأخطاء الكتابة (تبعا للكتاب)

ويجمل ذكر بعض الطرق المتبعة في تصحيح الكتابة ومن ذلك خوارزميات الكشف (heuristic) وأيضا نظام التقاسيم مثل الثنائي (Bigram)

وهذا على المستوى التركيبي

مثالٌ برمجي

```
# Correct single word

from ar_corrector.correctorimport Corrector
corr = Corrector()

all_corrections = corr.spell_correct('بختب') # return 5 corrections with top frequencies
# [('61 ,'بكتب'), ('22 ,'برتب'), ('21 ,'بختم'), ('9 ,'بختي'), ('7 ,'بخت')]
print(all_corrections)



corr.spell_correct('من') # return true


# Correct with context

from ar_corrector.correctorimport Corrector
corr = Corrector()

sent = 'أكدت قواعص التمذد في تشاد أنها تواضضل طريقها للعاحمة'
print(corr.contextual_correct(sent))
#أكدت قوات التمرد في تشاد أنها تواصل طريقها للعاصمة

sent = 'اتنتهى حدث آبل المنتظو بالإعلاخ عن مموعة من المنتجات'
print(corr.contextual_correct(sent))

#انتهى حدث آبل المنتظر بالإعلان عن مجموعة من المنتجات
```

الصرف

باب التصريف وفروعه من أكثر الأبواب أهمية حيث تظهر فيه معاني اللغة في تقليباتها واشتقاقاتها ولعلي أخص الأقسام الأربعة التي ذكرناها في أول هذا القسم وهي التحليل الصرفي ورفع الإبهام والتقطيع والتشكيل

أما أولها وهو التحليل الصرفي حوسبيا بتقعيد وتأسيس ما عليه يُحلل الحاسبُ النصَّ ولنرى مثالا مكتوبا "أنسألكموها" ولنحلله من جذره حتى مركباته

مثال برمجي

```
from camel_tools.morphology.database import MorphologyDB
from camel_tools.morphology.analyzer import Analyzer

# First, we need to load a morphological database.
# Here, we load the default database which is used for analyzing
# Modern Standard Arabic.
db = MorphologyDB.builtin_db()

analyzer = Analyzer(db)

analyses = analyzer.analyze('موظف')

for analysis in analyses:
    print(analysis, '\n')
```

```
diac: وَسَيَكْتُبُونَها
lex: كَتَبَ-u_1
bw: وَ/CONJ+سَ/FUT_PART+يَ/IV3MP+كْتُبْ/IV+ونَ/IVSUFF_SUBJ:MP_MOOD:I+ها/IVSUFF_DO:3FS
gloss: and_+_will_+_they_(people)+write+it;them;her
pos: verb
root: ك.ت.ب
catib6: PRT+PRT+VRB+NOM
ud: CONJ+AUX+VERB+PRON
d1seg: وَ+_سَيَكْتُبُونَها
d1tok: وَ+_سَيَكْتُبُونَها
atbseg: وَ+_سَ+_يَكْتُبُونَ+_ها
d3seg: وَ+_سَ+_يَكْتُبُونَ+_ها
d2seg: وَ+_سَ+_يَكْتُبُونَها
d2tok: وَ+_سَ+_يَكْتُبُونَها
atbtok: وَ+_سَ+_يَكْتُبُونَ+_ها
d3tok: وَ+_سَ+_يَكْتُبُونَ+_ها
bwtok: وَ+_سَ+_يَ+_كْتُبْ+_ونَ+_ها
pos_lex_logprob: -3.648503
caphi: w_a_s_a_y_a_k_t_u_b_uu_n_a_h_aa
```

رفع الإبهام

بأن تعرف مكان وقسم (نوع) كل كلمة في النص هل هو فعل أو اسم و إن كان فهل هو صفة أم اسم فاعل الخ ولنرى مثالا

يا صاحب الهم إن الهم منفرج ... أبشر بخير فإن الكاشفَ الله

الم الم الم بانه ... ان ان ان اوانه

عِشِ اِبقَ اِسمُ سُد قُد جُد مُرِ اِنهَ رِفِ اِسرِ نَل ...غِظِ اِرِمِ صِبِ اِحمِ اِغزُ اِسبِ رُع زَع دِلِ اِثنِ نُل

وَهَذا دُعاءٌ لَو سَكَتُّ كُفِيتَهُ ...لِأَنّي سَأَلتُ اللَهَ فيكَ وَقَد فَعَل

مثال برمجي

```
from camel_tools.tokenizers.word import simple_word_tokenize
from camel_tools.disambig.mle import MLEDisambiguator

mle = MLEDisambiguator.pretrained()

# The disambiguator expects pre-tokenized text
sentence = simple_word_tokenize('نجح بايدن في الانتخابات')

disambig = mle.disambiguate(sentence)

# For each disambiguated word d in disambig, d.analyses is a list of analyses
# sorted from most likely to least likely. Therefore, d.analyses[0] would
# be the most likely analysis for a given word. Below we extract different
# features from the top analysis of each disambiguated word into seperate
lists.
diacritized = [d.analyses[0].analysis['diac'] for d in disambig]
pos_tags = [d.analyses[0].analysis['pos'] for d in disambig]
lemmas = [d.analyses[0].analysis['lex'] for d in disambig]

# Print the combined feature values extracted above
for triplet in zip(diacritized, pos_tags, lemmas):
    print(triplet)
```

→

```
('نَجَحَ', 'verb', 'نَجَح')
('بايدن', 'noun_prop', 'بايدن')
('فِي', 'prep', 'فِي')
('الاِنْتِخاباتِ', 'noun', 'اِنْتِخاب')
```

```
from camel_tools.tokenizers.word import simple_word_tokenize
from camel_tools.disambig.mle import MLEDisambiguator
from camel_tools.tagger.default import DefaultTagger

mle = MLEDisambiguator.pretrained()
tagger = DefaultTagger(mle, 'pos')

# The tagger expects pre-tokenized text
sentence = simple_word_tokenize('نجح بايدن في الانتخابات')

pos_tags = tagger.tag(sentence)

print(pos_tags)
```

→ ['verb', 'noun_prop', 'prep', 'noun']

تقسيم (تقطيع) الكلام

وهذا من أهم أقسام هذا المجال وينبني عليه أغلب التطبيقات الحديثة وله أنواع كثيرة سنشرح منها أهم ثلاثة:

1. تقسيم مبنيٌّ على الحروف
2. وآخر على الكلمات نفسها
3. وأشهرها على التجزيء فتصير الكلمات أجزاء مقطعةً

وسنشرح أشهر خوارزميةٍ لكل واحدةٍ ان شاء الله

التقسيم على الحروف

وهو مبني على أن نعتبر الحرف هو المركب الأساسي للكلمة فمثالا:

كلمة "ملعب" إذا حللناها تصير [م,ل,ع,ب] وهكذا

وهذه الطريقة لها فائدة كبيرة إذ لا يعيقها شيءٌ عن تقسيم أي معطى كيفما كان ولكن يعيبها طول مدخلاتها وقلة بياناتها

مثال برمجي

```python
# we build this by our selfs

chars = "ابتشجحخدذرزسشصطظعغفقكلمنهويًٌإ'÷×؛<>ٍ[&،/:ْ~{}آ, .ﷲةأىّ'"

stoi = { ch:i for i,ch in enumerate(chars) }
itos = { i:ch for i,ch in enumerate(chars) }
encode = lambda s: [stoi[c] for c in s] # encoder: take a string,
output a list of integers
decode = lambda l: ''.join([itos[i] for i in l]) # decoder: take a
list of integers, output a string

print(encode("أنْ يَطَّلِعَ على صُورَةٍ حَقيقيَّةٍ"))
```

[58, 30, 25, 50, 29, 28, 30, 16, 1, 30, 23, 41, 18, 30, 29, 18, 23, 59, 29, 15, 32, 27, 11, 30, 57, 42, 29, 7, 30, 21, 28, 21, 28, 1, 30, 57, 42]

التقسيم على الكلمات

وهو أن نعتبر كل كلمة قسما مميزا كما هي بدون أي تغيير على أجزائها

كلمة "ملعب" تبقى "ملعب" و"ملعبا" تبقى نفسها

وهذه الطريقة تنفع حيث أن لكل كلمة متجها خاصا بها فيقل عدد المدخلات وتزيد دلالاتها ولكن يعيبها عيب قاتل وهو كبر حجم قاعدة الكلمات المستعملة وهذا سيؤدي إلى صعوبات في الحسابات وأيضا المقسم لن يستطيع فهم الكلمات الجديدة إن لم تكن موجودةً قبلُ.

مثال برمجي

```
# we can build this by our self

text = """
هذا النص هو مثال لنص يمكن أن يستبدل في نفس المساحة، لقد تم توليد هذا النص من مولد النص العربى، حيث يمكنك أن تولد مثل هذا النص أو العديد من النصوص الأخرى إضافة إلى زيادة عدد
الحروف التى يولدها التطبيق.
إذا كنت تحتاج إلى عدد أكبر من الفقرات يتيح لك مولد النص العربى زيادة عدد الفقرات كما تريد، النص لن يبدو مقسما ولا يحوي أخطاء لغوية، مولد النص العربى مفيد لمصممي المواقع على وجه
الخصوص، حيث يحتاج العميل فى كثير من الأحيان أن يطلع على صورة حقيقية لتصميم الموقع.
ومن هنا وجب على المصمم أن يضع نصوصا مؤقتة على التصميم ليظهر للعميل الشكل كاملاًدور مولد النص العربى أن يوفر على المصمم عناء البحث عن نص بديل لا علاقة له بالموضوع الذى يتحدث
عنه التصميم فيظهر بشكل لا يليق.
هذا النص يمكن أن يتم تركيبه على أي تصميم دون مشكلة فلن يبدو وكأنه نص منسوخ، غير منظم، غير منسق، أو حتى غير مفهوم لأنه مازال نصًا بديلاً ومؤقتًا.
        """

un_set = set(text.split()+[" ","<مجهول>"])

word_indx = {word:index for index,word in enumerate(un_set)}
indx_word = {index:word for index,word in enumerate(un_set)}


# put the unkown part
def encoder(text):
  collected = []
  for word in text.split():
    if word in word_indx.keys():
      collected.append(word_indx[word])
    else:
      collected.append(word_indx['<مجهول>'])
  return collected

decode = lambda x: " ".join([indx_word[indx] for indx in x])

encoded = encoder("من أن يطلع على صورة حقيقية لو يستطيع الطالب أن يفهم مفهوم اللعبة")
print(encoded)
print(decode(encoded))
```

```
[17, 70, 55, 33, 30, 45, 51, 51, 51, 70, 51, 8, 51]
من أن يطلع على صورة حقيقية <مجهول> <مجهول> <مجهول> أن <مجهول> مفهوم <مجهول>
```

التقسيم المُجَزِّءُ (الجزئي)

وهو أشهر أنواع المقسمات وهو الذي يُستعمل في أغلب إلا تكن كلَّ التطبيقات الحديثة من المحادثات (تشات جبت chat gpt) وغيرها من النماذج الكبيرة وهو مبني على مبدأ لطيفٍ جدا وهو أن تعرض الكلمات بأجزاء كلٍّ منها وكمثالٍ عليه فكلمة "ملعب" مثلا قد تصير "مل" و "عب" فتخيل كم عدد الكلمات التي أجزاءٌ منها السوابقُ

وأشهر هذه المقسمات BPE "مقسم بايَتِيٌّ" وهدفه كان ضغط البيانات وهذا في الثمانينيات ولكن استخدم في معالجة اللغة بأخرة وظهر وبزغ وما زال. ولعلي أشرح الخوارزمية بسرعة

وعليه بُنيَ عدد من المقسمات الجديدة التي فيها تحسينات إحصائية مثل "جزء الجملة" Sentence Piece

الخوارزمية المستعملة

Algorithm 1 Learn BPE operations

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

r · → r·
l o → lo
lo w → low
e r· → er·

function BYTE-PAIR ENCODING(strings C, number of merges k) **returns** vocab V

$V \leftarrow$ all unique characters in C # initial set of tokens is characters
for $i = 1$ **to** k **do** # merge tokens til k times
  $t_L, t_R \leftarrow$ Most frequent pair of adjacent tokens in C
  $t_{NEW} \leftarrow t_L + t_R$ # make new token by concatenating
  $V \leftarrow V + t_{NEW}$ # update the vocabulary
  Replace each occurrence of t_L, t_R in C with t_{NEW} # and update the corpus
return V

Figure 2.13 The token learner part of the BPE algorithm for taking a corpus broken up into individual characters or bytes, and learning a vocabulary by iteratively merging tokens. Figure adapted from Bostrom and Durrett (2020).

نبدأ بأصغر مكونات النصوص: ا ب ت ث ... ه و ي ,١ ,٢ ,٣ ,٤ ,٥ ,٦ ,٧ , ... ٩

بعد اللفة الاولى: اب , اج , ... , هو , هي

بعد اللفة الثانية: ... ...

بعد لفَّاتٍ معدودة ... ...

يتكون عندنا مجموعة أجزاء نستعملها و الأجزاء المذكورة أعلاه مثال من مقسم أرابيرت العربي

مثال برمجي من مكتبة أرابيرت ٢

```
ids = tokenizer("من أن يطلع على صورة حقيقية لو يستطيع الطالب أن يفهم مفهوم
اللعبة")['input_ids']
# tokenizer.tokenize("من أن يطلع على صورة حقيقية لو يستطيع الطالب أن يفهم مفهوم اللعبة")
tokenizer.convert_ids_to_tokens(ids)
cod = tokenizer.encode("مرحبا بك في لعبتنا")
tokenizer.decode(cod)
print(cod)
```

```
ar_toks = tokenizer.tokenize( "السلام عليكم و رحمة الله و بركاته")
```

['السلام', 'علي', '##كم', 'و', 'ر', '##حم', '##ة', 'الله', 'و', 'بر', '##كات', '##ه']

التنبيه على اختلاف البرمجة

Word Piece

لمزيد من التفاصيل يُرجع للمقالة التالية (رابط)

وهذا فيه إضافة حسابات إحصائية

$$\mathcal{O}_{\mathrm{ML}}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log P_{\theta}(Y^{*(i)} \mid X^{(i)}) \ .$$

Sentence Piece

وهذه طريقة أخرى تستهدف اللغات التي لا تفصل كلماتها بالمسافات المعروفة

Bigram

```
["b", "g", "h", "n", "p", "s", "u", "ug", "un", "hug"],
```

"hugs" could be tokenized both as `["hug", "s"]`, `["h", "ug", "s"]` or `["h", "u", "g", "s"]`. So which one to choose? Unigram saves the probability of each token in the training corpus on top of saving the vocabulary so that the probability of each possible tokenization can be computed after training. The algorithm simply picks the most likely tokenization in practice, but also offers the possibility to sample a possible tokenization according to their probabilities.

Those probabilities are defined by the loss the tokenizer is trained on. Assuming that the training data consists of the words $x_1, \ldots, x_N$ and that the set of all possible tokenizations for a word x_i is defined as $S(x_i)$, then the overall loss is defined as

$$\mathcal{L} = -\sum_{i=1}^{N} \log \left(\sum_{x \in S(x_i)} p(x) \right)$$

الدلالة

الدلالة: علاقة بين تعبير ومعنى وهي في العربية أصل وقرائن خمسةٌ ملحق وصوتي وتركيبي وصرفي وسياقي

وأصنافها: لفظ واشارة وعقد (حساب) وخط وحال

وعلم الدلالة هو علم المعنى ومحاوره ثلاثة ماهية وتعليلية وتدليلية ويحتوي على الكلمة والمفهوم [وهو مركب ذهني من قواعد و مؤثرات]

وأما الأنطلوجيا: فهو مصطلح إغريقي يعني علم الوجود [وقد عرّبته للهيكلة المعنوية وغيري لـ النماذج المعرفية]

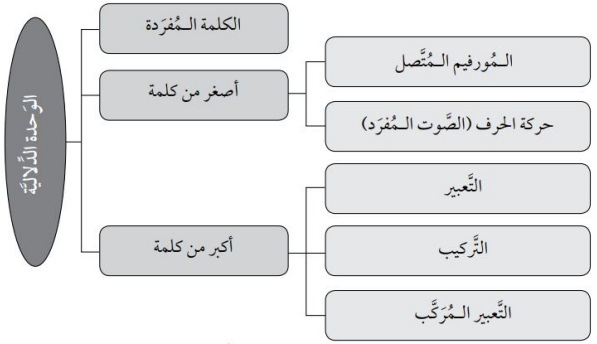
وأما هندسة [وهي قياس الواقع على الأصول أو المفاهيم] الأنطولوجيا: فهي فن بناء هياكل مساعدة على ترتيب البيانات وسهولة فهمها واستخدامها واسترجاعها

كتاب: علم الدلالة والانطولوجيا من منظور حوسبة اللغة العربية

الوحدة الدلالية

تلخيص لبعض المصطلحات المنتشرة:

- مورفيم: وهي وحدة المعالجة البنيوية أي أصغر وحدة لها معنى
- الجذع: هو مجرَّدُ السوابق واللواحق
- والجذر: أصل الكلمة المُشتَقَّةِ منه
- والزوائد سوابق ولواحق [أوائل وأواخر][1]



الشَّكل ٦: أنماط الوَحدة الدَّلاليَّة في اللُّغات الطَّبيعيَّة

م	أنماط الوحدات		نماذج الوَحدة الدَّلاليَّة
١	الكلمة الـمُفرَدة (الـمُورفيم الـمُفرَد)		إنسان
٢	أصغر من كلمة	الـمُورفيم الـمُتَّصل	(سـ) يعمل = التَّسويف
		حركة الحرف	كَتَبَ (ـتَ) = خطاب الـمُفرَد الـمُذكَّر
٣	أكبر من كلمة	التَّعبير	ضَرَبَ أخماسًا في أسداس = تحيَّر
		التَّركيب	تأبَّطَ شرًّا = العلميَّة
		التَّعبير الـمُرَكَّب	الحافلة النَّهريَّة

الجدول ٩: أنماط الوحدات الدَّلاليَّة في اللُّغة العربيَّة

(١): مصطلحيات هذا هو المشهور ولكني وجدت أحسن كلمةً لها لاصِبة: لاحقة أو إضافة حرفية تُزاد في أوّل الكلمة أو آخرها أو تُدرج في وسطها لتغيّر من دلالتها اللغوية، أو لتخصّص معناها في منظومة اصطلاحية معيّنة، فإذا كانت في أول الكلمة تُسمّى **الرَاعِفة**، بمعنى السابقة، فالرَّعْف في اللغة التقدّم والسَّبْق، و إذا كانت في آخر الكلمة فهي **الكاسِعة**، والكَسْع في اللغة الإتباع، والأصل أن يُقال القافية إلاّ أنّ هذا المصطلح أُستهلك في الشعر، امّا في وسط الكلمة فهي القاحِمة من القَحْم وهو الدخول في الشيء.

شرح النموذج

عادة الآن صارت النماذج عصبونية شديدة العمق والتعقيد كثير عدد المتغيرات جدا ولكن هذا لا يعني أن نترك ما يُعطي نتائج جيدة وتكلفة تدريبه أقل بكثير

- نموذج إحصائي: مثل حدْسِ بايز و المتجهات المعتمدة و الانحدار المنطقي الرمزي وسلاسل ماركوف وغيرها
- نموذج عميق: الشبكة العصبونية بأنواعها تلافيفية أو ارتدادية أو بدائية أو تنبهية وغير ذلك

[تنبيه على الأصل في الهندسة التفاضلي]

لمن أحب مراجعة مصدرٍ صعب ولكنه مفيد في تبيين أصول التعلم العميق معمما ومؤصلا كل النماذج لباب واحد فلينظر المحاضرة و ورقتها

ولمن أحب البدء في تعلم الآلة فعليه بفهم علم البيانات أولا من numpy, pandas, matplot ثم البداية في الخوارزميات الرئيسية ثم ينطلق الشخص إلى التعلم العميق هناك كتاب بالعربي للأستاذ طعيمة (رابط) وله أيضا كتاب في التعلم العميق وآخر في مشاريع علوم البيانات (رابط مجموع الكتب) [ملاحظة كل كتب الدكتور مترجمة]
وهناك كتب ومحاضرات كثيرة بالكلزية (eng) منشورة على الشابكة لمن أحب

كيفَ المجالُ الآن ؟

انزاح المجال تماما نحوَ نماذج المحولات من ٢٠١٧ فما أن يخرج نموذجٌ حتى يَتبعهُ آخر فظهر الآن نماذج لكل فرع من معالجة اللغة

عليكم بـ hugging face

في اكتشاف **المُسمّيات الكيانية**

NER ⟵ XLM-Roberta

لتوليد النصوص

Text Gen: GPT, LLAMA, Falcon, BERT, T5 (full)

موضوع الفعالية في الإستخدام من أمثلته **التكميم** (تصغير عدد الفواصل)

دخول المحولات في المجالات الاخرى كمعالجة الصور والصوتيات والحمض النووي

لمزيد من التفاصيل انظر كتاب Natural Language Processing with Transformers

تطبيق عملي

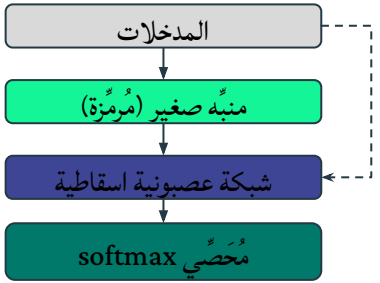
1. بناء شبكة انتباه صغيرة
2. باحث معنوي
3. استعمال متمم (chat GPT) مع هندسة المُدخلات

إضافات إذا سمح الوقت: نموذج احصائي ساذج و انحدار منطقي رمزي (لوجستي)

لخّص ما يحسُن من دروس cs124 عند كتابة مسودة الكتاب

النُبيه الصغير

لمزيد من التفاصيل حول حسابيات المحولات انظر الورقة

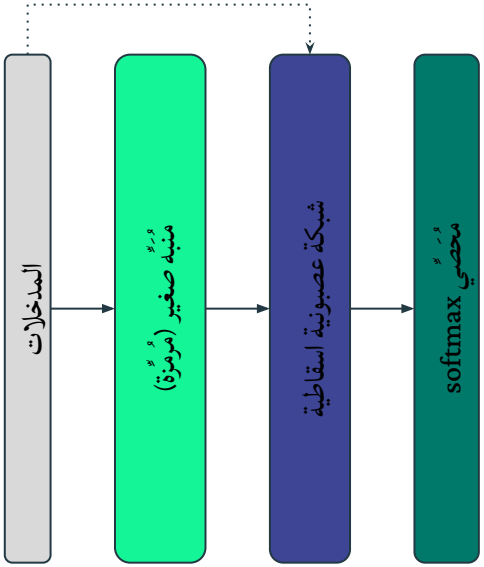


يمكننا أيضا أن نضيف طبقة ناظمية norm لمشكلة تلاشي المُشتق

حواشٍ توضيحية:

- سين مدخل والهمزة تُبدل بحسب اسم المصفوفة (كلها نفس الأبعاد ولكن الإسم يختلف) حيث ميم هي المفتاحية وقاف هو القيمية والالف هي الاستردادية (انظر شرحها في أخر)
- فجُداء س مع ء يعطينا ق,ا×م حيث أبعاد كل واحدةٍ (ن×ك)
- نتيجة الإنتباه مصفوفة خ$_{ن×ك}$
- ثم الإسقاطية مصفوفة ش ك×م
- عملية التحصية على الأعمدة م في النهاية
- هذه العملية بدون نظام المؤازيات المُدخلية (بدون مرصوصات)
- ج متجهة عمودي فيه احتمال الكلمة التالية وهو المُستمثل

السهم المُنقَّط اضافة ممكنة بأن تقفزَ للطرف التالي وقد استُعملت في الورقة



معادلات النموذج

$$س_{ن×ل} ص^{(ء)}_{ل×ك} \text{ حيث ء} = \{م,ق,ا\}$$

$$\text{انتباه}(م,ق,ا) = \text{مُحَصِّ}(ا.م^{\text{مقلوبة}} \setminus \sqrt{\text{أبعاد}}).ق = خ_{ن×ك}$$

$$خ.ش = ن_{ن×م}$$

$$\text{مُحَصِّ}(ن) = ج$$

الباحث المعنوي

معادلة جاكارد يسيرة جدا ان شاء الله

ب(جـ١، جـ٢) = تقاطع(جـ١، جـ٢) \ اجتماع (جـ١، جـ٢)

ويمكن تحسين هذه المعادلة بعمل تقطيع نوني (كلنوني) (كلمتان أو ثلاثة الخ) وكذا يمكن تحسينها بتنظيف المدخل **تجذيعا** أو **تجذيرا**

أما المعادلة الأخرى فهي تكرار الكلمة و معكوس التكرار المستندي (تكٌ متم)

تك(كلمة, مستندها) = تكرارها الى كلماته

[ت\ك]

متم(الكلمة, المستندات) = لوغ (عدد المستندات \ عدد ما جاءت فيه)

[لـ(ع\ظ)]

استعمال متمم (مدردم [نحت من مدردش ومتمم])

الصورة مربوطة
فاضغطها

أصول التعامل معه أربعة:

- محددات إحصائية (عادة لا نستعملها) مثل وُسعَ أو مدى العشوائية وكمية الإختيارات البحثية وغيرها
- تدريبية: كإدخال قطعة مع سؤال عنها أو سياق لفهمه وغير ذلك
- نوع المخرج نصا أو برمجة أو غير ذلك
- تأطير المخرج

أمور قد تفيدك في الدراسة:

- اطلب نظام ٨٠-٢٠ باريتو
- اطلب خطة لدراسة موضوع أو أكثر تداخلا
- خطة للمراجعة التكرارية للضبط
- تصور المفاهيم المعقدة (طلب تصوير منطقي)
- اطلب أسئلة للتأكد من الفهم
- حول المفهوم الى قصة
- اطلب الشرح خذي خمسٍ

هناك bard من جوجل وكذا موقع https://www.perplexity.ai/ ولمن أحب فهذا تلخيصي لمحاضرات الدكتور أنيس وهواري عن الموضوع رابط

خاتمة

كيف يبدأ شادي المجال فيه:

1. تعلم البرمجة ببايثون وأتقنها ليسهل عليك برمجة النماذج
2. تعلم أساسيات الجبر الخطي والإحصاء لتفهم أصول عمل النماذج (طبعا هنا لا حَدَّ أعلى ولكن أدنى)
3. تعلم مكتبات معالجة اللغات مثل camel-tools, nltk ومكتبات الذكاء الصناعي ويكفيك: pytorch, sklearn
4. طبق ثم طبق ثم طبق وابنِ ما استطعت من المشاريع وتجعلها تتابعًا بل تآزيا (مع بعض)
5. اقرأ إذا أحببت ما تيسر من الكتب وبالعربي كتب مركز الملك عبد الله
6. كمقدمات مع تطبيقات ومشاريع مفيدة انظر كتب الأستاذ علام طعيمة (انظر الشريحة رقم ٤٠)

- مكتبات شديدة النفع والإفادة Useful libraries
  - Camel tools
  - farasa (py)
  - spacy
  - nltk
  - gensim
  - hugging face
    - transformers and tokenizers
    - datasets
  - spark NLP
  - Regex
  - pytorch

لمن أحب البدء في المجال فعليه بدروس وكتاب جرفاسكي وصاحبه مارتن رابط وهذا رابط مقاطعها

مصادر أوراق بحثية عربية

https://github.com/iwan-rg/ArabicSurvey

رابط لمنتقياتي من كتب مركز الملك عبد الله لمن أحب

من أهم المصادر في الترجمة "معجم مصطلحات الرياضيات" طبعة مجمع دمشق (رابط)

مقالة فيها كتب لرياضيات التعالؤ رابط

جميع الشفرات البرمجية في هذه المُسودة لمن أحب تشغيلها ودراستها (رابط) ولمن يجد خطأً فليُكرمنا به ولم يبقَ ناقصا منها إلا برمجةِ النُّبيه لعل الله ييسر ذلك قريبا

الشرائح من إعداد سري السباعي

Email: serrymrss@gmail.com
github: https://github.com/serrysibaee
linkedin: https://www.linkedin.com/in/serry-sibaee/