**Chat bot with Artificial intelligence(AI)**

**Student. Ali N. Razook[[1]](#footnote-1)\***

**Student. Jalla N. Khoury[[2]](#footnote-2)\***

**ABSTRACT**

Chat bots are programs that mimic human conversation using Artificial Intelligence (AI). It is designed to be the ultimate virtual assistant, entertainment purpose, helping one to complete tasks ranging from answering questions, getting driving directions, turning up the thermostat in smart home, to playing one's favorite tunes etc. The use of chatbots evolved rapidly in numerous fields in recent years, including Marketing, Supporting Systems, Education, Health Care, Cultural Heritage, and Entertainment. Chatbot has become more popular in business groups right now as they can reduce customer service cost and handles multiple users at a time. In this project we will be creating a relatively simple chat bot that will be used to answer frequently asked questions to facilitate customer service.

**Keywords**: Chat bot, AI, Python, Machine Learning, HCI.

**روبوت الدردشة مع الذكاء الاصطناعي (AI)**

**الطالب: علي نزار رزوق [[3]](#footnote-3)\***

**الطالبة: جالا نصر خوري [[4]](#footnote-4)\***

**ملخّص**

روبوتات الدردشة هي برامج تحاكي المحادثة البشرية باستخدام الذكاء الاصطناعي (AI). تم تصميمه ليكون مساعداً افتراضياً، أو لأغراض ترفيهية، أو مساعدة الشخص على إكمال المهام التي تتراوح من الإجابة على الأسئلة ، والحصول على اتجاهات القيادة ، ورفع درجة الحرارة في المنزل الذكي ، إلى تشغيل الألحان المفضلة للفرد وما إلى ذلك. دخل استخدام روبوتات الدردشة في الكثير من المجالات في السنوات الأخيرة ، بما في ذلك التسويق وأنظمة الدعم والتعليم والرعاية الصحية والتراث الثقافي والترفيه. أصبحت روبوتات الدردشة أكثر شيوعاً في مجموعات الأعمال في الوقت الحالي حيث يمكنه تقليل تكلفة خدمة العملاء والتعامل مع العديد من المستخدمين في وقت واحد. في هذا المشروع ، سننشئ روبوت دردشة بسيطاً نسبياً سيتم استخدامه للإجابة على الأسئلة المتداولة لتسهيل خدمة العملاء.

**الكلمات المفتاحية:** روبوت الدردشة ، الذكاء الاصطناعي ، بايثون ، التعلم الآلي ، التفاعل بين الإنسان والحاسوب.

**مقدمة:**

دمج الذكاء الاصطناعي (ΑΙ) في حياتنا اليومية بشكل متزايد مع إنشاء وتحليل البرامج والأجهزة الذكية ، والتي تسمى العوامل الذكية. يمكن للوكلاء الأذكياء القيام بمجموعة متنوعة من المهام التي تتراوح من العمل العمالي إلى العمليات المعقدة. روبوت المحادثة هو مثال نموذجي لنظام الذكاء الاصطناعي وواحد من أكثر الأمثلة الابتدائية والأكثر انتشاراً للتفاعل الذكي بين الإنسان والحاسوب Human-Computer Interaction (HCI).(Bansal, H., Khan, R., 2018) .

إنه برنامج كمبيوتر ، يستجيب مثل الكيان الذكي عند التحدث معه من خلال النص أو الصوت ويفهم لغة أو أكثر من اللغات البشرية من خلال معالجة اللغة الطبيعية Natural Language Processing [NLP]. (Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Bhale, P., Das, T, 2015).

في المعجم ، يتم تعريف chatbot على أنه "برنامج كمبيوتر مصمم لمحاكاة المحادثة مع مستخدمين بشريين ، خاصة عبر الإنترنت". تُعرف روبوتات الدردشة أيضاً باسم الروبوتات الذكية أو الوكلاء التفاعليين أو المساعدين الرقميين أو كيانات المحادثة الاصطناعية. (Lexico Dictionaries)

**أهمية البحث وأهدافه:**

روبوت الدردشة هذا بسيط للغاية وسهل الاستخدام, وهو ليس معقداً جداً مثل روبوتات الدردشة الأخرى. كما أن عمل هذا البرنامج بسيط ويمكن لأي شخص فهمه بسهولة ويمكن استخدامه على سبيل المثال للرد الآلي على الزبائن في عمل ما.

**طرائق البحث ومواده:**

تم تنفيذ هذا المشروع باستخدام لغة البرمجة بايثون Python وباستخدام برنامج PyCharm وموقع <https://colab.research.google.com> كمحرر لكتابة الكود البرمجي وتنفيذه . تم استخدام الحزم التالية numpy , nltk, tensorflow ,tflearn .

**لماذا يستخدم المستخدمون روبوتات المحادثة؟**

يبدو أن روبوتات الدردشة تحمل وعوداً هائلة لتزويد المستخدمين بدعم سريع ومناسب للرد على أسئلتهم على وجه التحديد. يعتبر الدافع الأكثر شيوعاً لمستخدمي chatbot هو الإنتاجية ، بينما الدوافع الأخرى هي الترفيه والعوامل الاجتماعية والتواصل مع الحداثة. ومع ذلك ، لتحقيق التوازن بين الدوافع المذكورة أعلاه ، يجب بناء chatbot بطريقة تعمل كأداة ، ولعبة ، وصديق في نفس الوقت. Brandtzaeg, P.B., Følstad, A., 2017)).

يعد انخفاض تكاليف خدمة العملاء والقدرة على التعامل مع العديد من المستخدمين في وقت واحد من بعض الأسباب التي جعلت روبوتات المحادثة تحظى بشعبية كبيرة في مجموعات الأعمال. لم يعد يُنظر إلى روبوتات المحادثة على أنها مجرد مساعدين ، وطريقتهم في التفاعل تجعلهم أقرب إلى المستخدمين كرفاق ودودين. (Da-Costa, 2018)

وفقاً لدراسة ، فإن طلبات مستخدمي وسائل التواصل الاجتماعي على روبوتات المحادثة لخدمة العملاء هي طلبات عاطفية ومعلوماتية ، حيث يزيد معدل الفئة الأولى عن 40٪ ولا يعتزم المستخدمون أخذ معلومات محددة. (Xu, A., Liu, Z., Guo, Y., Sinha, V., Akkiraju, R, 2017). التعلم الآلي هو ما يعطي القدرة لروبوتات المحادثة الخاصة بخدمة العملاء لاكتشاف المشاعر وأيضاً القدرة على التواصل مع العملاء عاطفياً كما يفعل المشغلون البشريون. (Følstad, A., Nordheim, C.B., Bjørkli, C.A, 2018).

فيما يتعلق بثقة المستخدم في روبوتات المحادثة ، فإن ذلك يعتمد على عوامل متعلقة ببرنامج الدردشة نفسه ، مثل مدى استجابته كإنسان ، وكيف يتم تقديمه بنفسه ، ومدى احتراف مظهره. ومع ذلك ، فإنه يعتمد أيضاً على العوامل المتعلقة بسياقات الخدمة الخاصة به ، مثل العلامة التجارية لمضيف chatbot ، والخصوصية والأمان في chatbot ، وقضايا الخطر الأخرى حول موضوع الطلب. (Wallace, 2009). يمكن اقتراح التشابه البشري باستخدام الشخصيات البشرية (الإشارات المرئية) أو الأسماء المرتبطة بالبشر أو الهوية (إشارات الهوية) وتقليد اللغات البشرية (إشارات المحادثة). (Go, E., Sundar, S.S, 2019). لقد تمت بالفعل دراسة تأثير التجسيد والتفاعل في إفصاحات الأشخاص حول الموضوعات الحساسة ، مثل الضغوطات النفسية. (Sannon, S., Stoll, B., DiFranzo, D., Jung, M., Bazarova, N.N, 2018). من المهم أن نذكر أن روبوتات المحادثة لا تزال تفتقر إلى معنى فهم التعاطف وأنها ليست قادرة مثل البشر على فهم نغمات المحادثة. على الرغم من التقدم الذي تم إحرازه في هذا المجال ، وقريباً لن تتمكن الآلات من فهم ما يقوله شخص ما فحسب ، بل ستتمكن أيضاً من الشعور بما يقوله. (Fernandes, A).

ومع ذلك ، تم الكشف عن وجهة نظر متحيزة للجنس ، حيث تؤدي معظم روبوتات المحادثة مهاماً تعكس الأدوار الأنثوية تاريخياً وتوضح هذه الميزات بسلوكيات نمطية. وفقاً لذلك ، تعمل روبوتات المحادثة العامة أو المتخصصة على أتمتة العمل الذي يتم ترميزه على أنه أنثى ، نظراً لأنها تعمل بشكل أساسي في السياقات ذات الصلة بالخدمة أو المساعدة ، وتعمل كمساعدين شخصيين أو سكرتيرة . (da Costa, 2018)

مما لا شك فيه أننا سنعيش قريباً في عالم يكون فيه شركاء المحادثة بشراً أو روبوتات محادثة ، وفي كثير من الحالات ، لن نعرف ولن نهتم بما سيكون عليه شريكنا في المحادثة . (Dale, 2016)

**تصنيف روبوتات الدردشة :**

يمكن تصنيف روبوتات الدردشة باستخدام معلمات مختلفة: مجال المعرفة ، والخدمة المقدمة ، والأهداف ، وطريقة معالجة المدخلات وتوليد الاستجابة ، والمساعدات البشرية ، وطريقة البناء.

* التصنيف المستند إلى **مجال المعرفة** يأخذ بالاعتبار المعرفة التي يمكن أن يصل إليها برنامج الدردشة الآلي أو مقدار البيانات التي يتم التدريب عليها. يمكن أن تتحدث **روبوتات الدردشة ذات المجال المفتوح** عن الموضوعات العامة والرد بشكل مناسب ، بينما تركز **روبوتات الدردشة ذات المجال المغلق** على مجال معرفي معين وقد تفشل في الرد على أسئلة أخرى. (Nimavat, K., Champaneria, T., 2017).
* يأخذ التصنيف المستند إلى **الخدمة المقدمة** في الاعتبار القرب العاطفي لروبوت المحادثة من المستخدم ، ومقدار التفاعل الحميم الذي يحدث ، كما أنه يعتمد أيضاً على المهمة التي يؤديها روبوت المحادثة. تنشط **روبوتات الدردشة الشخصية Interpersonal** في مجال الاتصالات وتقدم خدمات مثل حجز المطاعم وحجز الرحلات الجوية وبوتات الأسئلة الشائعة. إنهم ليسوا رفقاء المستخدم ، لكنهم يحصلون على المعلومات ويمررونها إلى المستخدم. يمكن أن يكون لديهم شخصية ، ويمكن أن يكونوا ودودين ، وربما يتذكرون معلومات حول المستخدم ، لكنهم غير ملزمين أو غير متوقع منهم القيام بذلك. بينما توجد **روبوتات الدردشة الشخصية Intrapersonal** داخل المجال الشخصي للمستخدم ، مثل تطبيقات الدردشة مثل Messenger و Slack و WhatsApp. إنهم رفقاء للمستخدم ويفهمون المستخدم كما يفهمه الإنسان. أما **روبوتات المحادثة بين الوكلاء Inter-agent** فهي ضرورية الوجود حيث تتطلب جميع روبوتات المحادثة بعض إمكانيات الاتصال بين روبوتات المحادثة حيث ظهرت بالفعل الحاجة إلى بروتوكولات للاتصال بين روبوتات المحادثة. يعد تكامل Alexa-Cortana مثالاً على التواصل بين الوكلاء.(Nimavat, K., Champaneria, T., 2017) .
* يأخذ التصنيف المستند إلى **الأهداف** بالاعتبار الهدف الأساسي الذي تهدف روبوتات المحادثة إلى تحقيقه. تم تصميم **روبوتات المحادثة التثقيفية Informative** لتزويد المستخدم بالمعلومات المخزنة مسبقاً أو المتوفرة من مصدر ثابت ، مثل روبوتات المحادثة الشائعة. تتحدث **روبوتات المحادثة القائمة على الدردشة / المحادثة Chat-based/Conversational chatbots** إلى المستخدم ، مثل أي إنسان آخر ، وهدفها هو الاستجابة بشكل صحيح للجملة التي تم توجيهها إليها. تؤدي **روبوتات المحادثة القائمة على المهام Task-based** مهمة محددة مثل حجز رحلة طيران أو مساعدة شخص ما. روبوتات المحادثة هذه ذكية في سياق طلب المعلومات وفهم مدخلات المستخدم. روبوتات حجز المطاعم وروبوتات الدردشة الشائعة هي أمثلة على روبوتات الدردشة القائمة على المهام. (Nimavat, K., Champaneria, T., 2017 , Kucherbaev, P., Bozzon, A., Houben, G.-J, 2018).
* يأخذ التصنيف على أساس **معالجة المدخلات وطريقة توليد الاستجابة** في الحسبان طريقة معالجة المدخلات وتوليد الاستجابات. هناك ثلاثة نماذج تستخدم لإنتاج الاستجابات المناسبة: **النموذج القائم على القواعد** ، **والنموذج القائم على الاسترجاع** ، **والنموذج التوليدي** . (Hien, H.T., Cuong, P.-N., Nam, L.N.H., Nhung, H.L.T.K., Thang, L.D, 2018).
* روبوتات المحادثة النموذجية القائمة على **القواعد** هي نوع البنية التي تم بناء معظم روبوتات المحادثة الأولى بها ، مثل العديد من روبوتات المحادثة عبر الإنترنت التي تختار استجابة النظام بناءً على مجموعة قواعد ثابتة محددة مسبقاً ، بناءً على التعرف على الشكل المعجمي لنص الإدخال دون إنشاء أي إجابات نصية جديدة. المعلومات المستخدمة في روبوت المحادثة مشفرة يدوياً ويتم تنظيمها وتقديمها بأنماط محادثة. (Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K., 2017).
* يختلف النموذج القائم على القواعد اختلافًا طفيفًا عن النموذج القائم على **الاسترجاع** ، والذي يوفر مزيدًا من المرونة حيث يقوم بالاستعلام عن الموارد المتاحة وتحليلها باستخدام واجهات برمجة التطبيقات APIs. يسترد روبوت المحادثة المستند إلى الاسترداد بعض الاقتراحات للاستجابة من الفهرس قبل أن يطبق نهج المطابقة على اختيار الاستجابة. (Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.,2016).
* يولد النموذج **التوليدي** إجابات بطريقة أفضل من النماذج الأخرى ، بناءً على رسائل المستخدم الحالية والسابقة. روبوتات المحادثة هذه أكثر شبهاً بالبشر وتستخدم خوارزميات التعلم الآلي وتقنيات التعلم العميق. ومع ذلك ، هناك صعوبات في بنائها وتدريبها. . (Hien, H.T., Cuong, P.-N., Nam, L.N.H., Nhung, H.L.T.K., Thang, L.D, 2018).
* يعتبر تصنيف آخر لروبوتات الدردشة **مقدار المساعدة البشرية** في مكوناتها. تستخدم **روبوتات المحادثة بمساعدة الإنسان** الحساب البشري في عنصر واحد على الأقل من روبوت المحادثة. يمكن للمستقلين freelancer أو الموظفين بدوام كامل تجسيد ذكائهم في منطق روبوتات الدردشة لملء الفجوات التي تسببها قيود برامج الدردشة الآلية بالكامل. على الرغم من أن الحوسبة البشرية ، مقارنة بالخوارزميات المستندة إلى القواعد والتعلم الآلي ، توفر مزيدًا من المرونة والمتانة ، إلا أنها لا تستطيع معالجة جزء معين من المعلومات بنفس سرعة الجهاز ، مما يجعل من الصعب توسيع نطاقها لتلائم المزيد من طلبات المستخدمين. (Kucherbaev, P., Bozzon, A., Houben, G.-J., 2018).
* يمكن أيضًا تصنيف روبوتات الدردشة **وفقاً للأذونات التي توفرها منصة التطوير الخاصة بهم**. يمكن أن تكون منصات التطوير **مفتوحة المصدر** ، مثل RASA ، أو يمكن أن تكون **ذات رمز ملكية (مغلقة)** مثل منصات التطوير التي تقدمها عادةً الشركات الكبيرة مثل Google أو IBM. توفر الأنظمة الأساسية مفتوحة المصدر لمصمم روبوت الدردشة القدرة على التدخل في معظم جوانب التنفيذ. المنصات المغلقة، تعمل عادةً كصناديق سوداء ، والتي قد تكون عيباً كبيراً اعتماداً على متطلبات المشروع. ومع ذلك، يمكن اعتبار الوصول إلى أحدث التقنيات أكثر إلحاحاً بالنسبة للشركات الكبيرة. علاوة على ذلك ، يمكننا افتراض أن روبوتات الدردشة التي تم تطويرها بناءً على منصات الشركات الكبيرة قد تستفيد من كمية كبيرة من البيانات التي تجمعها هذه الشركات.

بالطبع ، لا تنتمي روبوتات المحادثة حصريًا إلى فئة أو أخرى ، ولكن هذه الفئات موجودة في كل روبوت محادثة بنسب متفاوتة.

**الجانب العملي:**

1. **تثبيت الحزم :**

قبل البدء في العمل نحتاج لتحميل بعض حزم البايثون:

* numpy
* nltk
* tensorflow
* tflearn

ببساطة يمكننا تنزيل هذه الحزم بكتابة التعليمة التالية في نافذة ال cmd:

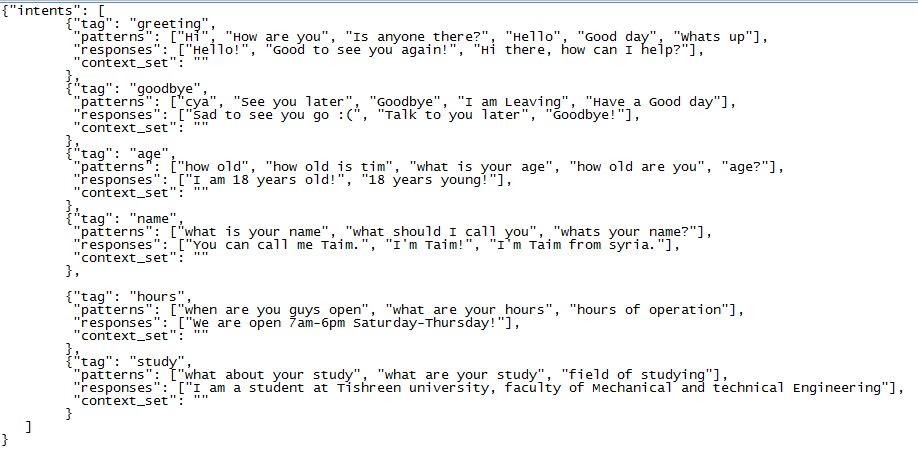
pip install "package name"

حيث يمكننا تبديل "package name" بأسماء الحزم السابقة.

1. **بيانات التدريب :**

الان سنحاول فهم ما هو نوع وطبيعة البيانات التي نحتاج أن نزودها ل روبوت الدردشة.

* لن نحتاج الى تنزيل مجموعات ضخمة من البيانات.
* سوف نستخدم فقط البيانات التي سنكتبها بأنفسنا
* سوف نحتاج لإنشاء JSON file الذي يحتوي نفس الصيغة الظاهرة في الشكل (1).



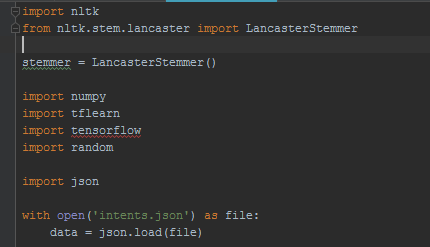
**الشكل(1): الملف intents.JSON الذي يحتوي بيانات التدريب.**

* ما سنفعله بملف ال JSON هو تشكيل حزمة من الرسائل التي من المرجح أن يستخدمها المستخدم وربط هذه الرسائل إلى مجموعة من الإجابات الملائمة.
* العلامة (TAG) لكل قاموس (dictionary) في الملف يحدد المجموعة التي تنتمي لها كل رسالة.
* باستخدام هذه البيانات سوف نقوم بتدريب الشبكة العصبونية لأخذ جملة من الكلمات وتصنيفها كواحدة من العلامات (tag) في الملف.
* بعدها يمكننا ببساطة أخذ جواب من هذه المجموعات وعرض ذلك للمستخدم.
* كلما زاد عدد العلامات (tags) والإجابات (responses) والأنماط (patterns) التي نزودها لروبوت الدردشة كلما كانت الدردشة أفضل ومعقدة أكثر.

1. **تحميل بيانات ال JSON :**

سوف نبدأ بإحضار بعض النماذج ونقوم بتحميلها في بيانات ال JSON .

يجب التأكد من أن ملف JSON يملك نفس الدليل (directory) كما هو في ال python script.



**الشكل(2): قراءة الملف intents.JSON الذي يحتوي بيانات التدريب وتخزين بياناته في المتحول data.**

بذلك تكون بيانات ال JSON الخاصة بنا خزنت في المتحول data.

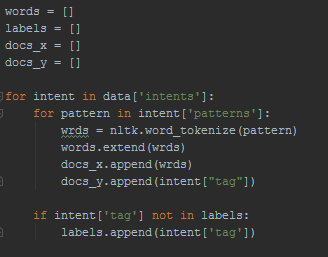
1. **استخراج البيانات (Extracting data):**

الآن سوف نقوم باستخراج البيانات التي نريدها من ملف ال json الخاص بنا.

نحتاج جميع الأنماط (patterns) وإلى أي صنف أو علامة(class/tag) تنتمي كل منها. كما نريد لائحة بجميع الكلمات المميزة في أنماطنا.

لذلك نقوم بما يلي:

* إعداد لوائح فارغة لتخزين هذه القيم.
* كل نمط سنقوم بتحويله الى لائحة (list) من الكلمات باستخدام nltk.word\_tokenizer بدلا من بقائها على شكل سلاسل (strings).
* بعدها سنقوم بإضافة كل نمط الى (docs\_x list) وعلامته الخاصة الى (docs\_y list).



**الشكل(3): استخراج البيانات.**

الآن سنبدأ بالمعالجة المسبقة للبيانات.

1. **اشتقاق الكلمة (Word stemming):**

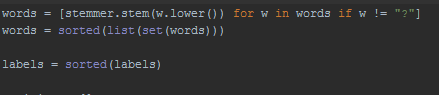
الاشتقاق (stemming) هو محاولة ايجاد جذر الكلمة.

مثال :

جذر كلمة “that’s” يمكن أن يكون “that”.

جذر كلمة “happening” يمكن أن يكون “happen”.

تفيد عملية الاشتقاق هذه بتقليل المفردات في نموذجنا ومحاولة ايجاد المعنى الأعم الذي تعنيه الجمل.



**الشكل(4): اشتقاق الكلمات.**

هذا الكود سيشكل ببساطة لائحة من الكلمات المشتقة لنستخدمها في المرحلة التالية من المعالجة المسبقة للبيانات.

1. **حقيبة الكلمات (bag of words):**

ان خوارزميات تعلم الالة والشبكات العصبية تتطلب ادخال رقمي. بالتالي نحتاج طريقة لتمثيل الجمل والعبارات بالأرقام لهذا أنشئت حقيبة الكلمات.

ما سنفعله هو تمثيل كل جملة بلائحة بطول كمية الكلمات في نموذج المفردات وكل موقع في هذه اللائحة سيمثل كلمة من المفردات الخاصة بنا.

• اذا كان الموقع في اللائحة هو 1 هذا يعني أن الكلمة موجودة في الجملة.

• واذا كان الموقع في اللائحة هو 0 هذا يعني أن الكلمة غير موجودة.

نسمي ذلك حقيبة الكلمات لأن الترتيب الذي تظهر به الكلمات في الجملة لا نعلمه, نحن نعلم فقط وجود الكلمات في نماذجنا من المفردات.

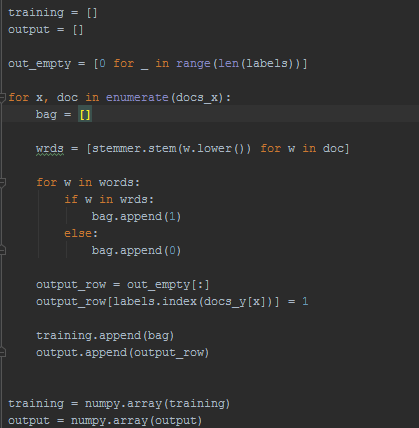
وكما قمنا بتشكيل الدخل أيضا نحتاج الى تشكيل الخرج لفهم الشبكة العصبونية الاصطناعية.

بشكل مشابه لحقيبة الكلمات نشكل لوائح خرج التي هي بطول كمية العلامات (tags) التي نملكها في مجموعة البيانات.

كل موقع في اللائحة سوف يمثل علامة واحدة ومحددة.

مثال: 1 في أي من هذه المواقع سيظهر العلامة التي يتم تمثيلها

في النهاية سنقوم بتحويل بيانات التدريب والخرج الناتج الى مصفوفات عددية.



**الشكل(5): حقيبة الكلمات.**

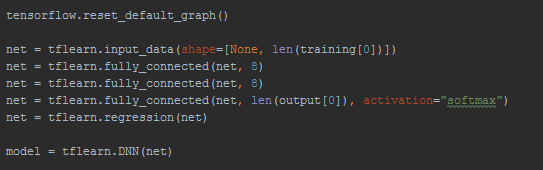
1. **تطوير النموذج:**

الان وبعد ما قمنا بمعالجة جميع البيانات نحن جاهزون لبدء انشاء وتدريب نموذج.

لغرضنا هذا سنقوم باستخدام شبكة عصبونية بدخل (تغذية) قياسية الى حد ما وهذه الشبكة تملك طبقتين مخفيتين.

الهدف من هذه الشبكة هو النظر الى حقيبة كلمات واعطاء الصنف الذي تنتمي له (واحدة من العلامات (tags) في ملف JSON )

سنبدأ بتحديد البنية لنموذجنا:



**الشكل(6): ضبط نموذج الشبكة العصبونية.**

1. **تدريب وحفظ النموذج:**

الان بعد أن انتهينا من اعداد النموذج حان وقت تدريبه على البيانات.

للقيام بذلك سنلائم بياناتنا مع النموذج. عدد ال (epoch) التي حددناها هو عدد المرات التي سيرى فيها النموذج نفس المعلومات أثناء التدريب.

عندما ننتهي من تدريب النموذج نستطيع حفظه إلى الملف model.tflearn لاستخدامه مع برامج أخرى.



**الشكل(6): تدريب وحفظ النموذج.**

1. **القيام بالتوقع:**

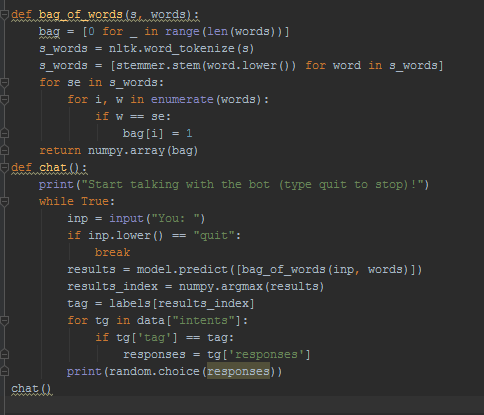
الان حان وقت الاستعمال الفعلي للنموذج.

بشكل مثالي نحتاج لتوليد إجابة لأي جملة يكتبها المستخدم.

للقيام بذلك يجب علينا تذكر أن نموذجنا لن يأخذ إدخال سلسله (string) بل يأخذ حقيبة من الكلمات.

يجب علينا أيضا لا يعطي جمل على خرجه بل يعطي لائحة من الاحتمالات لكل الاصناف, هذا يجعل إنشاء الإجابة على الشكل التالي:

* الحصول على دخل من المستخدم.
* تحويله الى حقيبة من الكلمات.
* الحصول على توقع من النموذج.
* إيجاد الصنف الأكثر ملائمة من الإجابات.
* اختيار إجابة من الصنف.



**الشكل(6): كود القيام بالتوقع والدردشة مع البوت.**

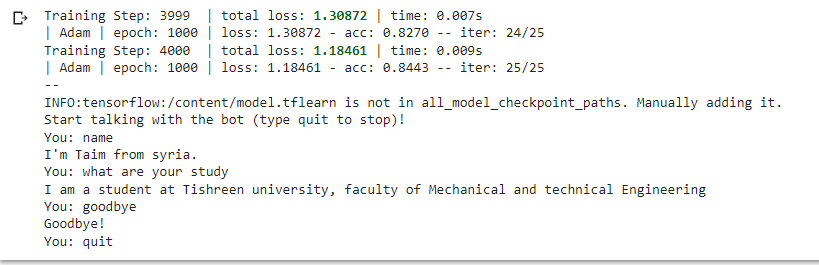
**الصعوبات التي واجهت العمل على المشروع:**

واجهتنا بعض المشاكل في تثبيت وتحميل حزمة tensorflow إلى برنامج pycharm , فقررنا استخدام موقع <https://colab.research.google.com> من أجل تنفيذ المشروع أونلاين حيث تكون مكتبة tensorflow مثبتة بشكل افتراضي .

**النتائج والمناقشة:**

تم الوصول إلى النتيجة المنشودة وهي تصميم بوت دردشة بسيط باستخدام الذكاء الاصطناعي , علما أن هذا البوت قابل للتطوير والتعديل لخدمة أي هدف سواء كان تجاري أو علمي أو ترويجي وما إلى هنالك.

إن نتيجة تنفيذ المشروع تظهر في الشكل(7).



**الشكل(7): تنفيذ الكود والدردشة مع البوت.**

**الاستنتاجات والتوصيات:**

تم التوصل إلى الاستنتاجات والمقترحات التالية:

* يمكن تغيير بارامترات الشبكة العصبونية المستخدمة للتدريب من أجل أداء أفضل.
* يمكن زيادة وتغيير قاعدة المعرفة للبوت من أجل خدمة مشروع ما (على سبيل المثال يمكن تصميم بوت طبي يمكنه وصف دواء مقترح لبعض الإصابات الشائعة التي نعاني منها).
* إن استخدام روبوتات الدردشة يمكن أن يقلل الحاجة إلى اليد العاملة في بعض المواقع في الشركات (مثل قسم call center).

**References**

1. Bansal, H., Khan, R.: A review paper on human computer interaction. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 8, 53 (2018).
2. Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Bhale, P., Das, T.: A study of today’s A.I. through chatbots and rediscovery of machine intelligence. Int. J. u- e-Serv. Sci. Technol. 8, 277–284 (2015).
3. chatbot | Definition of chatbot in English by Lexico Dictionaries. https://www.lexico.com/en/definition/chatbot.
4. Brandtzaeg, P.B., Følstad, A.: Why people use chatbots. In: Kompatsiaris, I., et al. (eds.) Internet Science, pp. 377–392. Springer, Cham (2017).
5. Da Costa, P.C.F.: Conversing with personal digital assistants: on gender and artificial intelligence. J. Sci. Technol. Arts 10, 59–72 (2018).
6. Xu, A., Liu, Z., Guo, Y., Sinha, V., Akkiraju, R.: A new chatbot for customer service on social media. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 3506–3510. ACM, New York (2017)
7. Følstad, A., Nordheim, C.B., Bjørkli, C.A.: What makes users trust a chatbot for customer service? An exploratory interview study. In: Bodrunova, S.S. (ed.) INSCI 2018. LNCS, vol. 11193, pp. 194–208. Springer, Cham (2018).
8. Wallace, R.S.: The anatomy of A.L.I.C.E. In: Epstein, R., Roberts, G., Beber, G. (eds.) Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer, pp. 181–210. Springer, Cham (2009).
9. Go, E., Sundar, S.S.: Humanizing chatbots: the effects of visual, identity and conversational cues on humanness perceptions. Comput. Hum. Behav. 97, 304–316 (2019).
10. Sannon, S., Stoll, B., DiFranzo, D., Jung, M., Bazarova, N.N.: How personification and interactivity influence stress-related disclosures to conversational agents. In: Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 285–288. ACM, New York (2018)
11. Fernandes, A.: NLP, NLU, NLG and how Chatbots work. https://chatbotslife.com/nlp-nlu-nlg-and-how-chatbots-work-dd7861dfc9df. last visit: 11/06/2022
12. Dale, R.: The return of the chatbots. Nat. Lang. Eng. 22, 811–817 (2016).
13. Nimavat, K., Champaneria, T.: Chatbots: an overview types, architecture, tools and future possibilities. Int. J. Sci. Res. Dev. 5, 1019–1024 (2017).
14. Kucherbaev, P., Bozzon, A., Houben, G.-J.: Human-aided bots. IEEE Internet Comput. 22, 36–43 (2018).
15. Hien, H.T., Cuong, P.-N., Nam, L.N.H., Nhung, H.L.T.K., Thang, L.D.: Intelligent assistants in higher-education environments: the FIT-EBot, a chatbot for administrative and learning support. In: Proceedings of the Ninth International Symposium on Information and Communication Technology, pp. 69–76. ACM, New York (2018).
16. Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K.: A survey of design techniques for conversational agents. In: Kaushik, S., Gupta, D., Kharb, L., Chahal, D. (eds.) ICICCT 2017. CCIS, vol. 750, pp. 336–350. Springer, Singapore (2017).
17. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots (2016).

**الملحق (1): ملف قاعدة المعرفة "intents.json" التي درّبنا عليها الشبكة العصبونية**

{**"intents"**: [  
 {**"tag"**: **"greeting"**,  
 **"patterns"**: [**"Hi"**, **"How are you"**, **"Is anyone there?"**, **"Hello"**, **"Good day"**, **"Whats up"**],  
 **"responses"**: [**"Hello!"**, **"Good to see you again!"**, **"Hi there, how can I help?"**],  
 **"context\_set"**: **""** },  
 {**"tag"**: **"goodbye"**,  
 **"patterns"**: [**"cya"**, **"See you later"**, **"Goodbye"**, **"I am Leaving"**, **"Have a Good day"**],  
 **"responses"**: [**"Sad to see you go :("**, **"Talk to you later"**, **"Goodbye!"**],  
 **"context\_set"**: **""** },  
 {**"tag"**: **"age"**,  
 **"patterns"**: [**"how old"**, **"how old is tim"**, **"what is your age"**, **"how old are you"**, **"age?"**],  
 **"responses"**: [**"I am 18 years old!"**, **"18 years young!"**],  
 **"context\_set"**: **""** },  
 {**"tag"**: **"name"**,  
 **"patterns"**: [**"what is your name"**, **"what should I call you"**, **"whats your name?"**],  
 **"responses"**: [**"You can call me Taim."**, **"I'm Taim!"**, **"I'm Taim from syria."**],  
 **"context\_set"**: **""** },  
  
 {**"tag"**: **"hours"**,  
 **"patterns"**: [**"when are you guys open"**, **"what are your hours"**, **"hours of operation"**],  
 **"responses"**: [**"We are open 7am-6pm Saturday-Thursday!"**],  
 **"context\_set"**: **""** },  
 {**"tag"**: **"study"**,  
 **"patterns"**: [**"what about your study"**, **"what are your study"**, **"field of studying"**],  
 **"responses"**: [**"I am a student at Tishreen university, faculty of Mechanical and technical Engineering"**],  
 **"context\_set"**: **""** }  
 ]  
}

**الملحق (2): كود المشروع كاملاً الذي تم تنفيذه على Google Colab**

import nltk

from nltk.stem.lancaster import LancasterStemmer

nltk.download('punkt')

stemmer = LancasterStemmer()

!pip install tflearn

import numpy

import tflearn

import tensorflow

import random

import json

with open('intents.json') as file:

    data = json.load(file)

words = []

labels = []

docs\_x = []

docs\_y = []

for intent in data['intents']:

    for pattern in intent['patterns']:

        wrds = nltk.word\_tokenize(pattern)

        words.extend(wrds)

        docs\_x.append(wrds)

        docs\_y.append(intent["tag"])

    if intent['tag'] not in labels:

        labels.append(intent['tag'])

words = [stemmer.stem(w.lower()) for w in words if w != "?"]

words = sorted(list(set(words)))

labels = sorted(labels)

training = []

output = []

out\_empty = [0 for \_ in range(len(labels))]

for x, doc in enumerate(docs\_x):

    bag = []

    wrds = [stemmer.stem(w.lower()) for w in doc]

    for w in words:

        if w in wrds:

            bag.append(1)

        else:

            bag.append(0)

    output\_row = out\_empty[:]

    output\_row[labels.index(docs\_y[x])] = 1

    training.append(bag)

    output.append(output\_row)

training = numpy.array(training)

output = numpy.array(output)

from tensorflow.python.framework import ops

ops.reset\_default\_graph()

net = tflearn.input\_data(shape=[None, len(training[0])])

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, len(output[0]), activation="softmax")

net = tflearn.regression(net)

model = tflearn.DNN(net)

model.fit(training, output, n\_epoch=1000, batch\_size=8, show\_metric=True)

model.save("model.tflearn")

def bag\_of\_words(s, words):

    bag = [0 for \_ in range(len(words))]

    s\_words = nltk.word\_tokenize(s)

    s\_words = [stemmer.stem(word.lower()) for word in s\_words]

    for se in s\_words:

        for i, w in enumerate(words):

            if w == se:

                bag[i] = 1

    return numpy.array(bag)

def chat():

    print("Start talking with the bot (type quit to stop)!")

    while True:

        inp = input("You: ")

        if inp.lower() == "quit":

            break

        results = model.predict([bag\_of\_words(inp, words)])

        results\_index = numpy.argmax(results)

        tag = labels[results\_index]

        for tg in data["intents"]:

            if tg['tag'] == tag:

                responses = tg['responses']

        print(random.choice(responses))

chat()

1. \***5th year Student- Faculty Of Mechanical and Electrical engineering- Tishreen University .** [↑](#footnote-ref-1)
2. \***5th year Student- Faculty Of Mechanical and Electrical engineering- Tishreen University .** [↑](#footnote-ref-2)
3. **\*طالب في السنة الخامسة- كلية الهندسة الميكانيكية والكهربائية- جامعة تشرين - سورية.** [↑](#footnote-ref-3)
4. **\*طالبة في السنة الخامسة- كلية الهندسة الميكانيكية والكهربائية- جامعة تشرين - سورية.** [↑](#footnote-ref-4)