WILDML
Artificial Intelligence, Deep Learning, and NLP

APRIL 6, 2016 BY DENNY BRITZ

Deep Learning for Chatbots, Part 1 – Introduction

Chatbots, also called Conversational Agents or Dialog Systems, are a hot topic. Microsoft is making big bets on chatbots, and so are companies like Facebook (M), Apple (Siri), Google, WeChat, and Slack. There is a new wave of startups trying to change how consumers interact with services by building consumer apps like Operator or x.ai, bot platforms like Chatfuel, and bot libraries like Howdy's Botkit. Microsoft recently released their own bot developer framework.

Many companies are hoping to develop bots to have natural conversations indistinguishable from human ones, and many are claiming to be using NLP and Deep Learning techniques to make this possible. But with all the hype around AI it's sometimes difficult to tell fact from fiction.

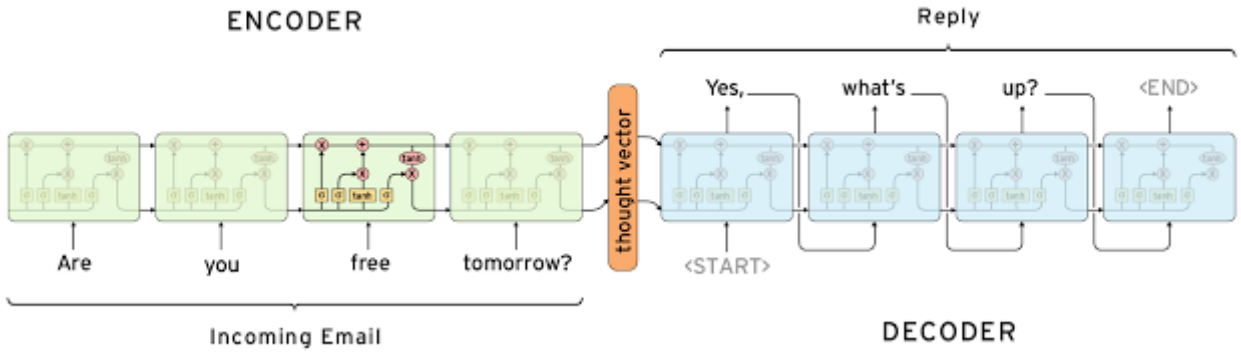
In this series I want to go over some of the Deep Learning techniques that are used to build conversational agents, starting off by explaining where we are right now, what's possible, and what will stay nearly impossible for at least a little while. This post will serve as an introduction, and we'll get into the implementation details in upcoming posts.

A taxonomy of models

Retrieval-Based vs. Generative Models

Retrieval-based models (easier) use a repository of predefined responses and some kind of heuristic to pick an appropriate response based on the input and context. The heuristic could be as simple as a rule-based expression match, or as complex as an ensemble of Machine Learning classifiers. These systems don't generate any new text, they just pick a response from a fixed set.

Generative models (harder) don't rely on pre-defined responses. They generate new responses from scratch. Generative models are typically based on Machine Translation techniques, but instead of translating from one language to another, we "translate" from an input to an output (response).



Both approaches have some obvious pros and cons. Due to the repository of handcrafted responses, retrieval-based methods don't make grammatical mistakes. However, they may be unable to handle unseen cases for which no appropriate predefined response exists. For the same reasons, these models can't refer back to contextual entity information like names mentioned earlier in the conversation. Generative models are "smarter". They can refer back to entities in the input and give the impression that you're talking to a human. However, these models are hard to train, are quite likely to make grammatical mistakes (especially on longer sentences), and typically require huge amounts of training data.

Deep Learning techniques can be used for both retrieval-based or generative models, but research seems to be moving into the generative direction. Deep Learning architectures like Sequence to Sequence are uniquely suited for generating text and researchers are hoping to make rapid progress in this area. However, we're still at the early stages of building generative models that work reasonably well. Production systems are more likely to be retrieval-based for now.

Long vs. Short Conversations

The longer the conversation the more difficult to automate it. On one side of the spectrum are **Short-Text Conversations (easier)** where the goal is to create a single response to a single input. For example, you may receive a specific question from a user and reply with an appropriate answer. Then there are **long conversations (harder)** where you go through multiple turns and need to keep track of what has been said. Customer support conversations are typically long conversational threads with multiple questions.

Open Domain vs. Closed Domain

In an **open domain (harder)** setting the user can take the conversation anywhere. There isn't necessarily have a well-defined goal or intention. Conversations on social media sites like Twitter and Reddit are typically open domain – they can go into all kinds of directions. The infinite number of topics and the fact that a certain amount of world knowledge is required to create reasonable responses makes this a hard problem.

In a **closed domain (easier)** setting the space of possible inputs and outputs is somewhat limited because the system is trying to achieve a very specific goal. Technical Customer Support or Shopping Assistants are examples of closed domain problems. These systems don't need to be able to talk about politics, they just need to fulfill their specific task as efficiently as possible. Sure, users can still take the conversation anywhere they want, but the system isn't required to handle all these cases – and the users don't expect it to.

Common Challenges

There are some obvious and not-so-obvious challenges when building conversational agents most of which are active research areas.

Incorporating Context

To produce sensible responses systems may need to incorporate both *linguistic context* and *physical context*. In long dialogs people keep track of what has been said and what information has been exchanged. That's an example of linguistic context. The most common approach is to embed the conversation into a vector, but doing that with long

conversations is challenging. Experiments in Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models and Attention with Intention for a Neural Network Conversation Model both go into that direction. One may also need to incorporate other kinds of contextual data such as date/time, location, or information about a user.

Coherent Personality

When generating responses the agent should ideally produce consistent answers to semantically identical inputs. For example, you want to get the same reply to "How old are you?" and "What is your age?". This may sound simple, but incorporating such fixed knowledge or "personality" into models is very much a research problem. Many systems learn to generate linguistic plausible responses, but they are not trained to generate semantically consistent ones. Usually that's because they are trained on a lot of data from multiple different users. Models like that in A Persona-Based Neural Conversation Model are making first steps into the direction of explicitly modeling a personality.

message	Where do you live now?
response	I live in Los Angeles.
message	In which city do you live now?
response	I live in Madrid.
message	In which country do you live now?
response	England, you?

Evaluation of Models

The ideal way to evaluate a conversational agent is to measure whether or not it is fulfilling its task, e.g. solve a customer support problem, in a given conversation. But such labels are expensive to obtain because they require human judgment and evaluation. Sometimes there is no well-defined goal, as is the case with open-domain models. Common metrics such as BLEU that are used for Machine Translation and are based on text matching aren't well suited because sensible responses can contain completely different words or phrases. In fact, in How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation researchers find that none of the commonly used metrics really correlate with human judgment.

Intention and Diversity

A common problem with generative systems is that they tend to produce generic responses like “That’s great!” or “I don’t know” that work for a lot of input cases. Early versions of Google’s Smart Reply tended to respond with “I love you” to almost anything. That’s partly a result of how these systems are trained, both in terms of data and in terms of actual training objective/algorithm. Some researchers have tried to artificially promote diversity through various objective functions. However, humans typically produce responses that are specific to the input and carry an intention. Because generative systems (and particularly open-domain systems) aren’t trained to have specific intentions they lack this kind of diversity.

How well does it actually work?

Given all the cutting edge research right now, where are we and how well do these systems actually work? Let’s consider our taxonomy again. A retrieval-based open domain system is obviously impossible because you can never handcraft enough responses to cover all cases. A generative open-domain system is almost Artificial General Intelligence (AGI) because it needs to handle all possible scenarios. We’re very far away from that as well (but a lot of research is going on in that area).

This leaves us with problems in restricted domains where both generative and retrieval based methods are appropriate. The longer the conversations and the more important the context, the more difficult the problem becomes.

In a recent interview, Andrew Ng, now chief scientist of Baidu, puts it well:

Most of the value of deep learning today is in narrow domains where you can get a lot of data. Here’s one example of something it cannot do: have a meaningful conversation. There are demos, and if you cherry-pick the conversation, it looks like it’s having a meaningful conversation, but if you actually try it yourself, it quickly goes off the rails.

Many companies start off by outsourcing their conversations to human workers and promise that they can “automate” it once they’ve collected enough data. That’s likely to happen only if they are operating in a pretty narrow domain – like a chat interface to call an Uber for example. Anything that’s a bit more open domain (like sales emails) is beyond

what we can currently do. However, we can also use these systems to assist human workers by proposing and correcting responses. That's much more feasible.

Grammatical mistakes in production systems are very costly and may drive away users. That's why most systems are probably best off using retrieval-based methods that are free of grammatical errors and offensive responses. If companies can somehow get their hands on huge amounts of data then generative models become feasible – but they must be assisted by other techniques to prevent them from going off the rails like Microsoft's Tay did.

Upcoming & Reading List

We'll get into the technical details of how to implement retrieval-based and generative conversational models using Deep Learning in the next post, but if you're interested in looking at some of the research then the following papers are a good starting point:

- Neural Responding Machine for Short-Text Conversation (2015-03)
- A Neural Conversational Model (2015-06)
- A Neural Network Approach to Context-Sensitive Generation of Conversational Responses (2015-06)
- The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems (2015-06)
- Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models (2015-07)
- A Diversity-Promoting Objective Function for Neural Conversation Models (2015-10)
- Attention with Intention for a Neural Network Conversation Model (2015-10)
- Improved Deep Learning Baselines for Ubuntu Corpus Dialogs (2015-10)
- A Survey of Available Corpora for Building Data-Driven Dialogue Systems (2015-12)
- Incorporating Copying Mechanism in Sequence-to-Sequence Learning (2016-03)
- A Persona-Based Neural Conversation Model (2016-03)
- How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation (2016-03)

CONVERSATIONAL AGENTS, DEEP LEARNING, NEURAL NETWORKS, NLP, RNNS

So it seems to me that Wit.ai is a rule based system + some sort of mechanic for "learning", as Denny hypotize.

2. A system very very similar to Wit.ai is www.API.ai

3. About Luis.ai, the Microsoft Understanding Intelligent Service (LUIS), maybe just an annocement now, seems very similar to both above :-D

giorgio

2 ^ v • Reply • Share ›

Emilio Srougo ➚ Giorgio Robino • a year ago

Thanks for the info!
Yes, it seems that api.ai and wit are quite similar, their power lies in making the dev of a ml bot as easy as a rules based one (by rules I think they mean approaches like AIML).
For example, in their abilities to teach the bot to recognise an intent based on a few examples, but, more remarkably, the ability to extract common entities in the user messages, like dates (tomorrow, next week, etc) and cities.

I'm really curious about the details of those kind of systems, so let's see what **@Denny Britz** comes up with in the next part!

2 ^ v • Reply • Share ›

zihaolucky ➚ Emilio Srougo • a year ago

Both of them use 'examples' and 'context' to begin a development and dialog. Begin creating intent with a few examples, so I think it's not possible to incorporate DL in intent identification. I try to use kNN with a EMD behind as sentence distance as my intent classification model, but I'm not sure if it works well.

^ v • Reply • Share ›

Denny Britz Mod ➚ Emilio Srougo • a year ago

I'm not familiar with Luis and I've only briefly looked at wit.ai. Yes, they are retrieval-based models. It's hard to tell exactly what tech they are using, but probably something quite similar to what's in the paper http://arxiv.org/abs/1506.0.... Maybe they are not using Deep Learning models and pick responses using some other classifier (SVMs, Logistic regression, etc). Who knows. In the next part I'll build something quite similar - it's actually pretty easy. It may be fun to compare to accuracy of that to wit.ai ;)

I haven't used them personally so I can't say much about their effectiveness. One additional feature of wit.ai is that you can extract entities (like names) and re-use them in your responses, but for most entity types that's a pretty easy thing to add. Another aspect is that wit.ai may achieve better performance of analyzing

aggregate data across users and conversations and training on that. So they have a potentially huge training data set. But I don't know if they actually do that.

Reply • Share ›

Emilio Srougo ➚ Denny Britz • a year ago

Looking forward to that next part!
But I wonder, why do you say that entity extraction is pretty easy, I mean, how do you extract things like dates and locations, which can be represented in many ways and formats?

1 Reply • Share ›

Denny Britz Mod ➚ Emilio Srougo • a year ago

Dates, locations, people, organizations, etc are common entity types and many software packages (like Stanford CoreNLP) do a very good job at extracting those. People have been working on this for decades. You can just use one of these libraries and call the right function.

What's challenging is domain-specific entity types, like drug names, legal entities, product names, and so on. You need to train your own system for that, and getting the right training data isn't easy.

Reply • Share ›

Neel Shah • a year ago

Great post !! I am designing a chatbot for displaying various statistics a client asks for about his company. There are around 1500-2000 such clients and their data is available to us. Currently there are around 25-30 statistics with different variations including few parameters. The types of statistics will increase further and also it is important to remember the usage pattern of each user. What do you recommend to use and how to structure the model?

Thanks.

1 Reply • Share ›

Jia Zhang • a year ago

Great Post!
I think there is another big challenge for generative model: the chatbot may be misguided by malicious user. As far as I know, some chat systems will refine their models by utilizing conversations with users and as there is no efficient method to judge good or evil, the system will go off rails. I don't know whether there are some research subject to this problem.
Thanks.

1 Reply • Share ›

Denny Britz Mod ➚ Jia Zhang • a year ago

I agree, good point. I haven't seen any research on this, but I'm sure we will soon ;)

1 • Reply • Share ›

Sazi Suber ➚ Denny Britz • 3 months ago

Microsoft had launched a bot that learnt from Twitter users and it was trained by users to literally become a nazi in a matter of hours forcing Microsoft to take it down and mellow down the next version !!

• Reply • Share ›

Abben • a year ago

Excellent! Looking forward to the upcoming.

1 • Reply • Share ›

Avi Gaur • 3 months ago

Thanks ! for the post a real eye opener.

• Reply • Share ›

CHANN • 6 months ago

Thank you for your article. I learned you.

• Reply • Share ›

Hunir Bhullar • 10 months ago

Hey man. Thanks you for taking the time to write this.
Thanks especially for putting up the reading list! Much appreciated.

• Reply • Share ›

Parry • a year ago

I have just added tensorflow based implementation of the model on github .

https://github.com/pbhatia2...

Only thing about seq2seq models is sometimes they get too generic with lot of data . I observed model performing better with few million tweets and became generic with increasing data .

• Reply • Share ›

Qian Hong • a year ago

Thanks for the nice post!

I'd like to share that the conversations from Stack Exchange are published in CC license, updated every months, with lots of contents from hundreds of sub sites of stackexchange, which might be useful for researcher working on chatbots as training data:

https://archive.org/details...

• Reply • Share ›

Omar Shehata • a year ago

Very much looking forward to the next post! That was a super clear overview. I feel like I

learned more reading this than days of searching around the internet.

Your comment that implementing something like what Wit.ai does being easy has gotten me really excited! It does some seem like magic, but I'm really eager to see how that could be implemented.

Reply • Share ›

raj • a year ago

Hey,

We want to build a chatbot that could answer queries based on a community support forum(Ex: Q:XYZ router not working and red light blinking. A:Reset the router).I think it involves only certain technical domains like networking , internet ,and other IT related problems.What should be the technology we use to solve this problem via a ChatBOt?

I came across AIML based solutions but that involve making the AIML Files manually and for such a large database I think it would not be possible or if its , then can you please specify how?What could be another alternative to this?Should we go for Deep Learning or AIML will be enough?

Kindly reply as soon as possible.

Thanks,

Raj Miglani

Reply • Share ›

Giorgio Robino ➚ raj • a year ago

using AIML is possible... but it's a nightmare, immo. long story.

I'd split solution in two possible steps, without resort any complex machinelearning algo:

1- create a database populated by info in forum; how to make up this DB is the creative/tricky task, maybe really need some NLP & clustering ;-)

2- create a dialog script with a good rule based chatbot language as Bruce Wilcox's ChatScript (or Successor) to understand questions and translate to DB queries.

Reply • Share ›

raj ➚ Giorgio Robino • a year ago

Hey,
Thanks for a quick reply.

But creating the DB for lets say 40000 Q&A poses a problem.Could you be more specific on how to do that?Lets say we have Q&A arranged in different categories, then according to you the system should just make a Db query and reply?

• Reply • Share ›

Vlad • a year ago

Awesome article! When are you planning to publish next part?

• Reply • Share ›

Denny Britz Mod → Vlad • a year ago

Probably in 1-2 weeks. Working on it but I was traveling for a while.

• Reply • Share ›

Madhawa Vidanapathirana → Denny Britz • 10 months ago

Hi,
Is the next article published?

• Reply • Share ›

Mike Chung • a year ago

Hey can you discuss how Parsey Mc Parseface would be intergrated to work in a chat bot system? for your next talk
thanks!

• Reply • Share ›

Denny Britz Mod → Mike Chung • a year ago

Haven't looked at it in detail, but I don't think it has *direct* relevance to chatbots. Sure, you can use parse trees as features for building bots, but you could always do that.

• Reply • Share ›

Mike Chung → Denny Britz • a year ago

Hey is there any more information about parsing trees as features for a building bots? I cannot seem to find anything on it on the web. Thanks!

• Reply • Share ›

Anantharaman Narayana • a year ago

Excellent post, thanks Denny! Specifically, I liked 3 aspects of this post:

(a) A taxonomy that helps us to categorize and position our own products avoiding mixing up issues (b) A neat mention of third party frameworks to build these bots and (c) A great list of references that include high quality papers. Here are my couple of observations/questions:

1. How much of the chat bot development the third party frameworks (such as Chatfuel or the Microsoft one) automate? Do they expose deep learning engines on which we could train our custom building blocks like NER? Given that each chat system would have its own goals, is it possible to build a generic engine that supports bulk of the tasks required for any application?

2. I just bumped on another informative site that may be of interest to readers of your blog:

2. I just bumped on another informative site that may be of interest to readers of your blog: https://www.chatbots.org/

3. How close these systems are to a human-like conversation, at least in a closed domain?

Thanks again and looking forward to the next post!

▲ | ▼ • Reply • Share ›

Denny Britz Mod ➚ Anantharaman Narayana • a year ago

Hey,

1. I haven't really looked at them in detail and many of these platforms are in an early beta stage so it's hard to stay. From what I've seen most services allow you to upload training conversations and then learn retrieval-based models from that. Most of these probably won't allow you to add custom code since they want you to use their service exclusively. It's definitely possible to build a "generic" engine and that's probably what the long-term plan for many of these companies is.

2. Thanks, will check it out.

3. If the domain is small enough (like, ordering an uber) you can probably build a retrieval-based model that comes pretty close to a human conversation. You'd just need to make sure that you can handle all the cases. For generative models we are very far away from human-like conversation.

▲ | ▼ • Reply • Share ›

Anantharaman Narayana ➚ Denny Britz • a year ago

Thanks much!

▲ | ▼ • Reply • Share ›

jun zhang • a year ago

Hi Denny,
Great post!
I am interested in automatic summarization, and I will be very grateful If you can write something about the automatic summarization on deep learning.
Thanks a lot.

▲ | ▼ • Reply • Share ›

Denny Britz Mod ➚ jun zhang • a year ago

I'll put it on my list of topics :)

▲ | ▼ • Reply • Share ›

Jordy Van Landeghem ➚ Denny Britz • a year ago

http://nlp.seas.harvard.edu... This paper might suit your appetite ;)

▲ | ▼ • Reply • Share ›

Giorgio Robino • a year ago

Hi Denny!

Hi Denny!

I really enjoyed your article and chatbots taxonomy! Here below some notes and questions:

1. I associate dialog systems you call "retrieval-based" with languages as AIML, ChatScript (beloved Bruce Wilcox's creature) and some recent portings of above, like RiveScript, SuperScript. That's correct ? (see my modest articles below: [1][2][3]).

2. You say: "The heuristic could be as simple as a rule-based expression match, or as complex as an ensemble of Machine Learning classifiers."

Interesting point: There is any implementation of a rule-based + machine learning classifier you can give me as example (any commercial / open-source sw) ?

3. You say: "Generative models are typically based on Machine Translation techniques, but instead of translating from one language to another, we “translate” from an input to an output (response)"

The generative models (that I presume are implemented with some sort of neural network

see more

• Reply • Share ›

Denny Britz Mod ➚ Giorgio Robino • a year ago

1. Kind of, but not really. By retrieval-based methods I meant that you train a system to give some response from a set of pre-defined responses. I'm not familiar with the languages you mentioned, but it seems like they are a way to write "rules" according to which the system behaves, like a higher-level programming language. I guess that's a third approach that doesn't really fall under Machine Learning.

2. See for example http://arxiv.org/abs/1510.0.... I'm not sure about production system, since they don't usually talk about their implementation details. I'm actually guessing Tay was one of those.

3. I don't think there are production systems out there yet because as I mentioned in the post these don't work "well enough" yet, e.g. often make grammatical mistakes.

• Reply • Share ›

Giorgio Robino ➚ Denny Britz • a year ago

1. AIML or ChatScript implement "machine learning" in a partial sense, I agree.

Quoting your definition: "By retrieval-based methods I meant that you train a system to give some response from a set of pre-defined responses." ... "train" is the keyword :)

ChatScript (http://brilligunderstanding... by example behave as a sort of old fashioned rule-based "expert" systems:

1. A dialog flow author write "precoded" collection of rules, where "rule" is a question/answer match ("volley" in ChatScript parlance).
In my opinion The great point is that dialog scripting is contained in text files, with a simple syntax, cohomprensible by a linguistic expert (non necesseraly a "programmer"). Decoupling fact knowledge (on a certain domain) from the engine and nackend integration logic is a big plus.

2. ChatScript engine afterward digest the dialog flow script, selecting at runtime the next rule using a pattern matching (refined) "mechanic".

see more

• Reply • Share ›

Usama Yaseen • a year ago

Thank you for the post, it's awesome !
Which dataset you will be using in the upcoming post ?

• Reply • Share ›

Denny Britz Mod ➚ Usama Yaseen • a year ago

Haven't decided yet, but probably the Ubuntu Dialog Corpus (https://github.com/rkadlec/...

1 • Reply • Share ›

Usama Yaseen ➚ Denny Britz • a year ago

Thanks, 'Ubuntu Dialog Corpus' looks Interesting.

I have one question regarding architectures like 'Sequence to Sequence Learning':

In a typical encoder-decoder framework; encoder encodes the input sequence to a fixed length vector (also called context), the decoder then takes the context, last output word and one keeps on sampling from decoder till end of sentence etc is generated, in this case the output of decoder can be of different 'length' then the 'target label/sequence', so how can this be handled in cost function (how to compute cost of two sequences of different length) ?

• Reply • Share ›

Denny Britz Mod ➚ Usama Yaseen • a year ago

During training you only compute the cost of length up to the target sequence, you don't care what the decoder generates after that.

• Reply • Share ›