Chapter 1: Part 1

**Introduction To**

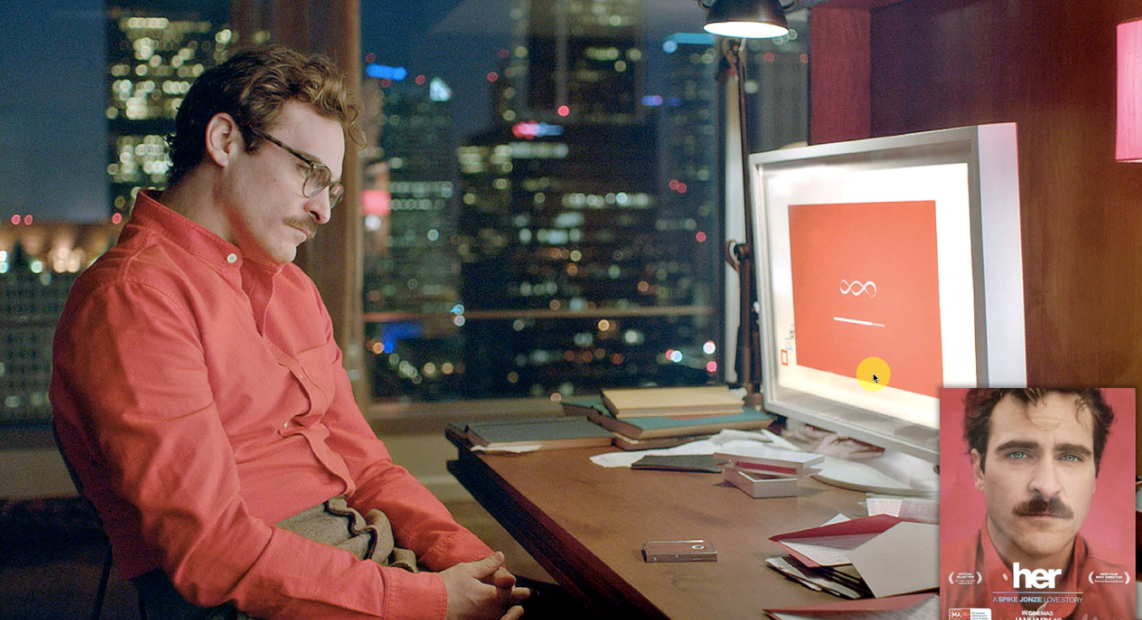
**Deep NLP**

**1.1 Why ChatBot**

Apple Microsoft Google and Amazon and all of them have invested major R&D funds and time and effort into ChatBot. Apple has ***Siri***, Microsoft has ***Cortana***. Google has ***Google assistant***. And finally Amazon has Amazon ***ALEXA***.

* These are predominantly voice ChatBot. All of thealgorithms that you will learn, all of the methodologies, all of the Deep Natural Language Processing techniques, they're all applicable to both text and voice. Here we'll learn how to create the texting ChatBot, same exact principles can be used for voice ChatBot.
* Usage:
* You just say hey Cortona, What's the weather like for today? That's the fastest way and you get an answer.
* And so that's the way where the world is going. We don’t have to type on our keyboards all the time to get information.
* This is a progression of user interface.
* So we need computers which can understand our voices or if we in some cases we need type our messaging app they need to understand human language.
* Facebook is doing something even more radical. They created their own factory of ChatBot. Called ***itsalive.io***
* There you can build a ChatBot for your Facebook if you have a Facebook community. And program it to communicate with your users to help.
* This important for businesses and for communities and so on. Because
* They are Autonomous
* Fast
* Drive Conversations
* Relevant
* Also chat boards are scalable. They're the most saleable approach a business can take in order to interact with users. If you have ***100000*** *customers* you don’t need a thousand *customer-supporter* you just need one ChatBot.

Movie Recommendation: Her.



**1.2 ChatBot Applications**

Here are the areas where you can apply your new skills. We can use the same skills methods and models in order to create many more things. Following are ordered by importance:

* Speech recognition: Taking in audio form and putting it into text.
* To understand the *wave forms* and some of the *sound* we can apply a *deep* *NLP* and
* Then put in that in the second part (the chatbot) in order to *understand* what to do with that *text/instructions*.
* Neural Machine Translation or NMT: Used in ***Google Translate***. Previously, Google Translate used phrase based translation algorithms. Now they have switched to NMT which is powered by exactly the same models as you'll see in this course.
* ChatBots: Will be talking about in depth in the course and we create one ourselves.
* Q&A: Its similar to ChatBots but it's not in-depth. For example you have a model which has access to a resource of the Harry Potter books and you can ask a question like: What is the name of Harry Potter's mother and it'll spit out an answer. So question-answer and no conversation going on.
* Here you can get very quickly information from volume volumes sources such as books or other sources on the Internet like Wikipedia.
* Text summarization: Similar *ChatBots* & *Q&A* but *less interactive*. If you have a lot of text and you need to create a summary. Eg: If you have a book and you just create a summary of the book or if you have lots of medical documents and journals and you need to write a synopsis for them then you can use Text summarization as well.
* Image captioning: It is a bit trickier. The models were going to look at an image and observing the dependencies in the image to create a caption. For instance you have a Black Dog Jumping Over A Fence. So the model needs to understand that:

1. there's a dog
2. it's a black dog
3. There is a fence

* And then put all together the dogs above the fences or jumping over the fence.
* Basically the model is able to assess what's going on in the image and then spit out a text transcription or caption. It can help the Blind Peoples.
* Video captioning: It's very similar to the image. Here you have information/interrelations inside each frame like: what's happening with the characters.
* The model can understand the conversation between characters, what was going on in the previous episodes etc.

**1.3 Objectives**

Our goal is to apply the most advanced *Sequence To Sequence* (**seq2seq**) *NLP* model to our *Chatbot*. We'll follow the below steps:

1. Types of Natural Language Processing: Before starting the actual project we'll talk about the *types of natural language* *processing*. We will discuss those types in a Venn diagram which demonstrates ***Deep Learning*** vs ***NLP*** vs ***Deep-NLP*** which will show us where the *Sequence To Sequence* lies in that diagram. It will be very helpful for us to keep track of how we're progressing into the world of NLP.
2. Classical vs Deep Learning Models: We'll talk about *classical* versus a *deep learning models*. Look at some actual examples of applications of *NLP*. To show us that *NLP* is not limited simply to *Chatbots* but covers a huge range of different applications and we'll see a couple of them here.
3. End-to-end Deep Learning Models: There will talk about End-to-End deep learning models and how they're different to *non- End-to-End deep learning models* and what are the advantages
4. Bag-of-words model: We'll talk about the *Bag Of Words NLP* *Model*. We'll see two of its variations and this will help us gradually progress to the most advanced *sequence-to-sequence* model.
5. Seq2Seq Architecture: We will discuss the ***seq2seq*** ***architecture***. This is the model that we're actually after. This is what we'll implement to our Chatbot.
6. Seq2Seq Training: Here we'll talk about the seq2seq training. We'll actually see it in action as if it was trained. It will help us better understand what is desired from the training and then it will be easier to cover all the training afterwards.
7. Beam Search Decoding: We'll discuss about *Beam Search Decoding* which is the way that the *seq2seq architecture/model* comes up with the *outputs* that we want from *ChatBot*.
8. Attention Mechanisms: Finally we'll talk about *Attention mechanisms* this is an additional *augmentation* of the *seq2seq model* which helps with a *long term memory* for the *algorithm*.

Note: We have to review ***ANN***, ***RNN (LSTM)*** before starting ***seq2seq*** architecture.

* Note: This project is most similar to following projects. Links:
* Suriyadeepan Ram: Suriyadeepan Ramamoorthy, from Puducherry, India. An Independent Researcher.

<http://suriyadeepan.github.io/>

<http://suriyadeepan.github.io/aboutme/>

<https://suriya.mystrikingly.com/>

* Github profiles:

[https://github.com/suriyadeepan](https://github.com/suriyadeepan?fbclid=IwAR1lRR7DxxgmnFTkzErye1eWfV0eu4dlDWGONzHplwSiOoZxZS6xJmq3uTA)

[https://github.com/suriyadeepan/practical\_seq2seq](https://github.com/suriyadeepan/practical_seq2seq?fbclid=IwAR3-aoDETUIQhRfsRk2nDYH5aRpNHU72GcmTw98cCU9Sa-6hpZTezo6rMP8)

[https://github.com/suriyadeepan/rnn-from-scratch](https://github.com/suriyadeepan/rnn-from-scratch?fbclid=IwAR0CXom-zdTnj2GtXuAoWgLITcfTVtkfc6nYKbc6G10dTA_w1pDK5gZrFOg)

[https://github.com/suriyadeepan/easy\_seq2seq](https://github.com/suriyadeepan/easy_seq2seq?fbclid=IwAR2Hl1dFrAn9gGvOhQ0WysX9KDymseKTad3B51_CT6XN_nrunB723yY4Txk)

**1.4 Types of NLP**

|  |  |
| --- | --- |
| * In the Venn diagram *NLP* is *green* area, *Deep-NLP* is *purple* area, a small portion of Deep-NLP is *Seq2Seq* (which is cutting edge NLP technology) represented by orange color, Deep-Learning is represented by blue area. * *Deep-NLP* are models which are *NLP* but also are *deep learning* with *neural networks*. * The sub section of the Deep-NLP is Seq2Seq model, the most cutting edge and powerful models.   Note: The size of these diagrams is not reflective of the importance or the volumes of these different fields. It's not scaled at all. |  |

**1.5 Classical vs Deep-Learning Models**

We'll look at a couple of examples from NLP.

Then we look an example from Deep-NLP and finally we'll see the Seq2Seq.

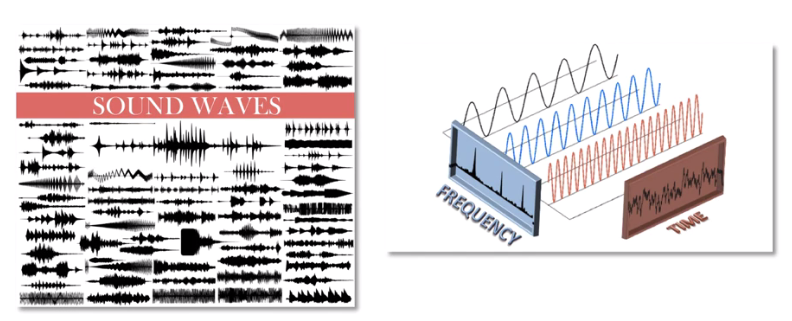
* NLP examples:



1. ***if/else rules:*** This is a way that we used to create chat-bots in the past. It is basically a huge list of possible questions and answers to those questions.

* Such a mechanical approach to answering questions or Chat-bot does not result in anything "human like" or anything realistic.

1. ***Audio frequency components analysis (speech recognition):*** NLP can be applied to speech recognition. There are algorithms that use non deep learning methods for *speech recognition*. And one of them is Audio Frequency Components Analysis. It's another *classical version of NLP* is in the *green area* of our *Venn Diagram*.

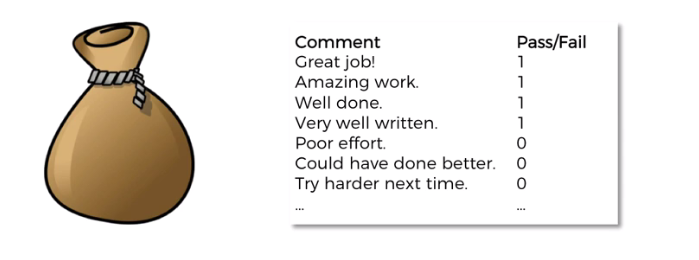


* We look at the *sound waves* of somebody's *speech*/*sound* then we try to identify what *waveforms* (for example using ***Fourier*** ***Transformation***) w.r to Time.
* After that, we *compare* them to *pre recorded frequencies* (certain combination of frequencies) means different type of word/sound. Simply we look at the frequencies using a certain *mathematical approach*.
* The key point is, we're not doing any Neural-Network computation, we're not creating neural networks just doing *mathematical calculations* around the *frequencies*.
* Then comparing them to the *mathematical calculations* in our *library* all *pre analyzed frequencies* and then we're *matching* it up with the *word* a person is saying/ or what *question* they're asking/ or what the *sentence* is meaning.

This is how we Recognize Speech.

1. **Bag Of Words:** Next one is the Bag Of Words model which is used for classification of good review/bad review. It is used for *text* *analysis* .

* Basically it creates a bag of words from the comment/review and counts the frequency of good/bad using 1 or 0. For example ***Good-job***, ***Nice, Great*** these words are indicate *1* - *positive review*. Words like ***Poor*** or ***Harder*** will be associate ***0- negative rivew***.
* Next time something comes up. For instance somebody says ***Good***, ***Good job*** or ***keep it up*** or something. It'll ***analyze*** the words that are in that ***new sentence*** by ***pulling*** them ***out of the bag*** and looking at them and ***understanding*** are they mostly associated ***ones*** or ***zeros***.

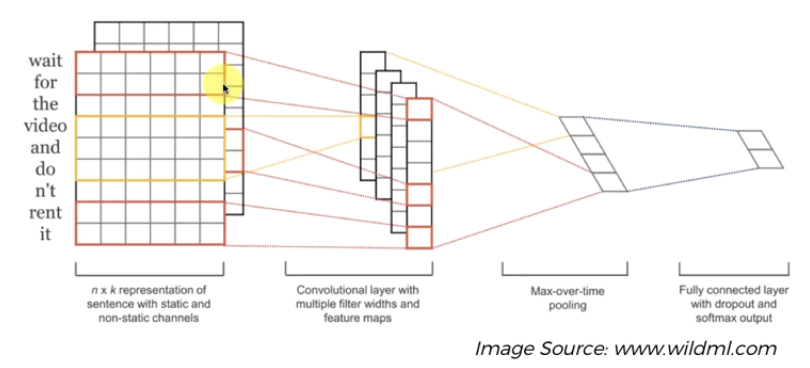


* This Bag Of Word model will put all these words into a bag and it will remember how often does the word grace come up with a ***1***. And how often does the word **Great** come up with a ***0***.
* Then it'll try to classify these words, to determine when the *sentiment* is *negative/positive*

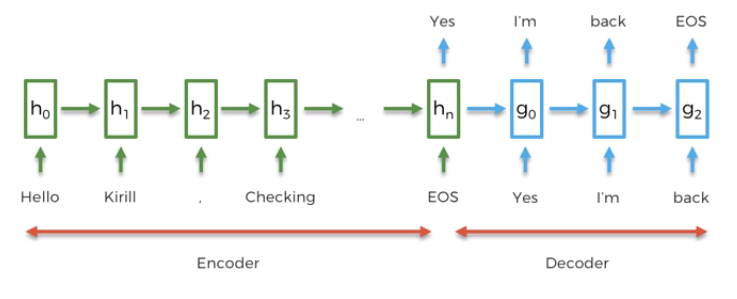
That's how it'll be able to predict/classify the new comments.

1. **CNN for text Recognition (Classification):** Convolutional neural networks for text recognition. CNN is mostly used for image recognition for self-driving cars/ videos.

* The way it works for text is:
* Words are *transformed* into a *matrix* through an operation called ***Embedding Of Words***.
* And then once they're in a *matrix*, we apply the same principles of *Convolution* that were applied for images (*Max-pooled/Mean-Pooled/Sum-pooled* ).
* And then they're flattened and then we have the prediction.
* Note: In this project we don’t use CNN, mostly we use RNN.
  + You can use *CNN* for *text recognition* just as you would do with *images*.



1. **Seq2Seq:** Finally the main model that we'll be looking for working with Sequence To Sequence (Seq2Seq). It is based on RNN and looks like following:



We'll discuss it in the following chapter and sections.

**1.6 End To End deep learning models**

*End To End deep learning* models will help us understand why ***Seq2Seq models*** are actually so good because Seq2Seq models are a type of End To End Deep-Learning model.

* Let's say we want to contact a customer support center of the company that we have for a product. What we normally do is we call them up and we talk to the Customer Support Representer. The problem is we're looking here is the problem of communication.
* Did we get our message across.
* Did they understand the message.
* Did they apply some sort of model (in this case the Human-brain) to come up with a solution to help us.

Now that’s work well for Human Customer Support Representer.

|  |  |
| --- | --- |
| * Case-1: So what happened if the company replace their support representatives with automated systems?   As usual we get a per-recorded voice something like:   1. press 1 for customer support 2. press 2 for billing. 3. Press 3 to check your account balance press. E   Etc   * This is one of those Heuristical if else if – model/rules. If your queries are quite easy this approach is good for use. But lot of the time we have very specific queries in mind that don't get solved through a decision tree or a flow chart (if else if – model/rules) like that. |  |

* In this case, we cannot communicate with it.
* Case-2: In the next case, we can talk with the machine, we use *Deep-learning* for *voice recognition* but with *classical NLP* algorithm for *processing* the data.
* It could be done with *audio waveforms analysis* (mathematically) with *classical NLP*.
* But let's assume the *voice recognition* is with *Deep Learning* to understand what you're *saying/your query*. That’s smart. That's a step forward.
* But there is still a problem because even though it allows you to say things in a *human voice* instead of using *dial-pad* it still passes your **result** into the **flowchart/decision-tree** and is like pretty much the same thing as you would use a *dial pad*. So still not good communication.
* This actually happened!! After using this path, some companies got even more frustrated customers than in the previous *dial-pad* version, so they reverted back to the *dial-pad* .
* Case-3: Next case is, we use Deep-Learning algorithm for both Speech Recognition and *Deep NLP* algorithm for *processing* the data.
* Listen: So there's a deep learning algorithm trained to analyze your *sound forms* that you're making with your *vocal chords*.
* Take-action: And there's another deep learning algorithm that is trained to take that information (data in text-form/machine-form) and it understand the meaning.
* So now we're fully in the deep natural language processing. But there is still some problems.
* Why: Because we have two separate Deep-Learning models. In some cases there are 3 or 4 Deep-Learning models that are playing.
* They *train separately* and *works separately* that’s why in some cases these DL models cannot *connect with each other* *properly*.

It would be better if we had one Deep-Learning algorithm.

* Case-4 (End to End Deep-NLP): To solve the above problem, we need an End to End Deep-NLP model, so that one brain (one Deep-Learning model) is taking care of the whole thing and that is the next stage (or final step?).
* So one Deep-NLP, one model that recognizes what you say, analyzes it and takes the actions. Very similar to a Human Representative in a customer care.
* The concept is that there's one neural network running the whole show and we have more chances of it being successful and delivering great customer experience.