

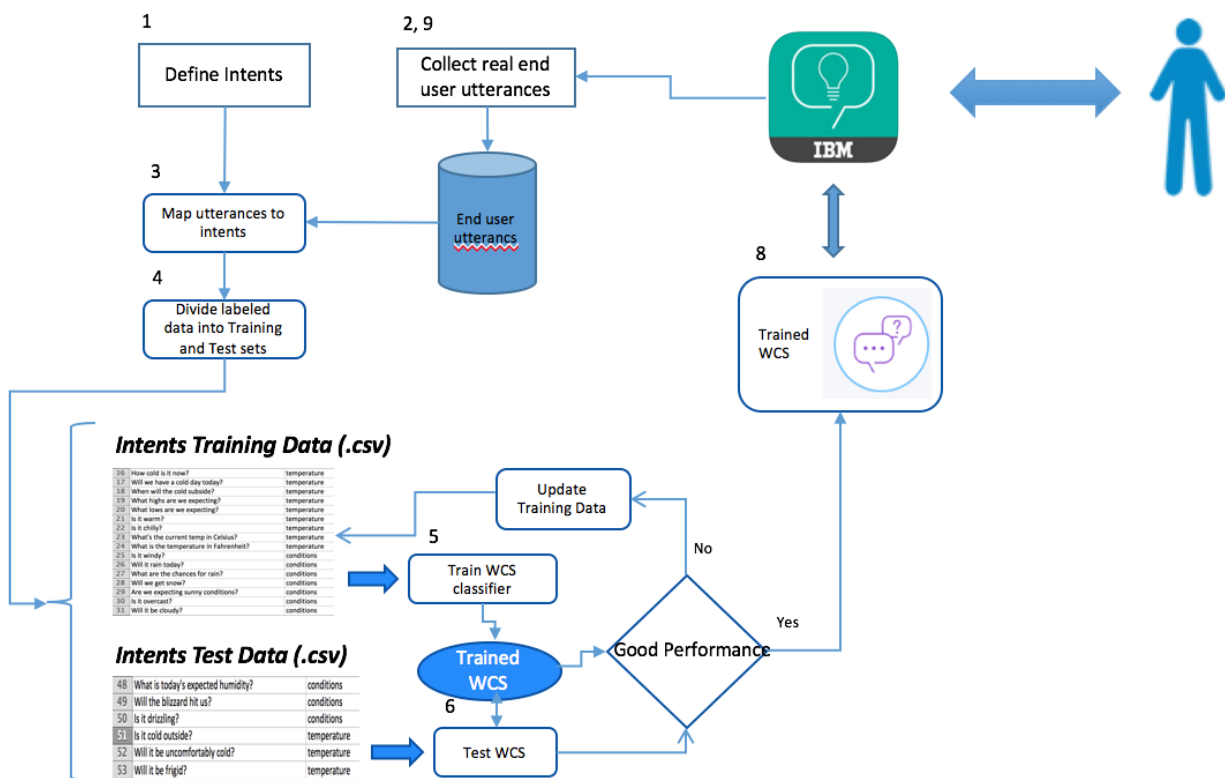
# How to train your chatbot

- Difference between **utterance** (something the user might say), **intent** (what the utterance means, what the user intends to get from the chatbot) and **entity** (a keyword that makes the user's intent more clear, for example: openingHours, borrowBook, etc. Basically Lisa's tags in the database)
- Define many utterances the user might use to get a certain intent, aka expand the database. Train the chatbot with this data. The more data we have, the more helpful the chatbot will be. Make sure each intent contains many utterances.
- Continue training the chatbot after it has been deployed (which is what we are planning now. Based on whether the user was satisfied or not, we will add their entry to the database and train the chatbot again at a set time)
- The more we can plan for, the less we will have to rely on AI to do the heavy lifting.  
-> <https://www.hubtype.com/blog/how-to-train-a-chatbot-2/>

## Step-by-step guide to training your chatbot

1. Define the intents (also known as classes or categories) you'd like your chat bot to extract from natural language utterances. While you can define a large number of intents for a variety of reasons, it is best to focus your intent definition on the purpose of your chat bot. In our chatbot, common intents would include *OpeningHours*, *BorrowBooks*, *Printing*, *CoronaPrinting*, and so on. However, if your bot takes the same action for *Printing* and *CoronaPrinting*, then you probably want to define one intent *PrintingIssues* instead.
2. Collect real end-user utterances that you'd want Watson Assistant to map to intents. It is important that the utterances come from end-users. Trying to guess what end-users would say may be acceptable for initial setup but you should plan to collect and leverage real end-user utterances. The performance of the system is strongly a function of how accurately it captures real end-user utterances. The more realistic the training data is, the better is the performance of the system. We are doing this by collecting the data from TopDesk from the client. For example, when training the Watson Business Coach application, we interviewed sellers, partners, and clients to collect questions and utterances like: "show me a Watson demo in Healthcare", "how can I use cognitive to improve customer service", "how is cognitive different from analytics", etc. You can use various techniques for collecting real end-user utterances such as crowd-sourcing or leveraging historical chat logs.
3. Assign the utterances collected in step 2 to the different intents defined in step 1. This step will most likely require subject matter experts (SMEs) to help with this mapping. For utterances that don't clearly map to any of the intents defined, either leave those empty (no intent) or map them to "other" (or "offtopic") intent. It is important to be able to capture intents that are "offtopic" so the application can handle them adequately. We could forward these "offtopic" utterances to the Ask Your Librarian page, for example.
4. Randomly divide the utterances in step 3 into two sets, a training set and a test set. A 70% training and 30% test is a typical split.

- Train your chatbot using the training set from step 4 (or step 7). This training set would constitute the “groundtruth” for the system.
- After training is complete, run the test set against the trained classifier and collect performance metrics such as accuracy, precision, and recall. For details and sample code on this, check out this blog. (I unfortunately haven’t found this sample code yet..)
- Perform Error Analysis: review the results in step 6 to understand why the classifier missed certain utterances. Update your training data accordingly and go back to step 5.
- After you’re satisfied with the results produced by the trained system, the system is now ready to be released (alpha/beta). Instrument in your solution a mechanism to collect end-user feedback. Some examples of collecting end-user feedback include prompting the user with “thumbs-up”/” thumbs-down” or with a star-rating for relevance of returned results. We do this by asking the user whether they were satisfied with their answer.
- When your chatbot is in use, continue to collect end user utterances, the intents returned by trained Watson Assistant service as well as end-user feedback.
- Map results collected in step 9 to new training/test data. Go back to step 4 and iterate.



➔ <https://developer.ibm.com/technologies/artificial-intelligence/blogs/10-steps-to-train-a-chatbot-and-its-machine-learning-models-to-maximize-performance/>

### Inspiration for coding the NN:

➔ <https://towardsdatascience.com/how-to-create-a-chatbot-with-python-deep-learning-in-less-than-an-hour-56a063bdfc44>

