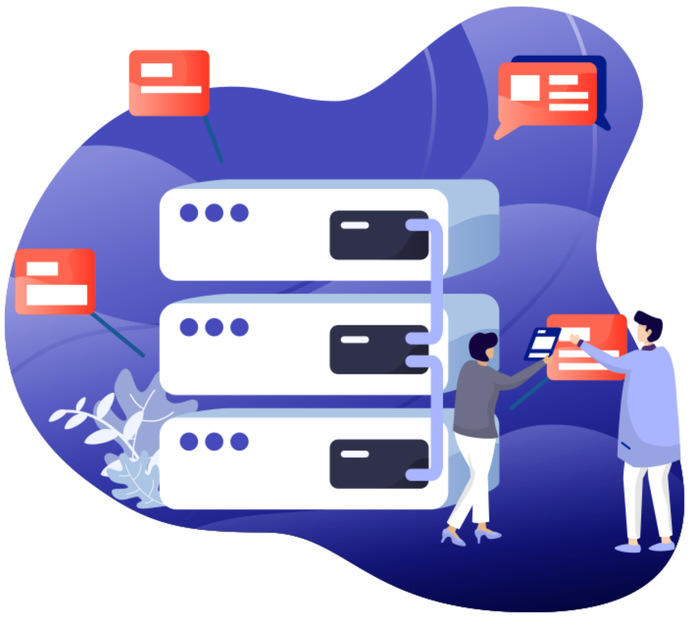
TERRANOHA

How to do NLP, episode 2: Construct and train NLP models

9/4/2022

Introduction

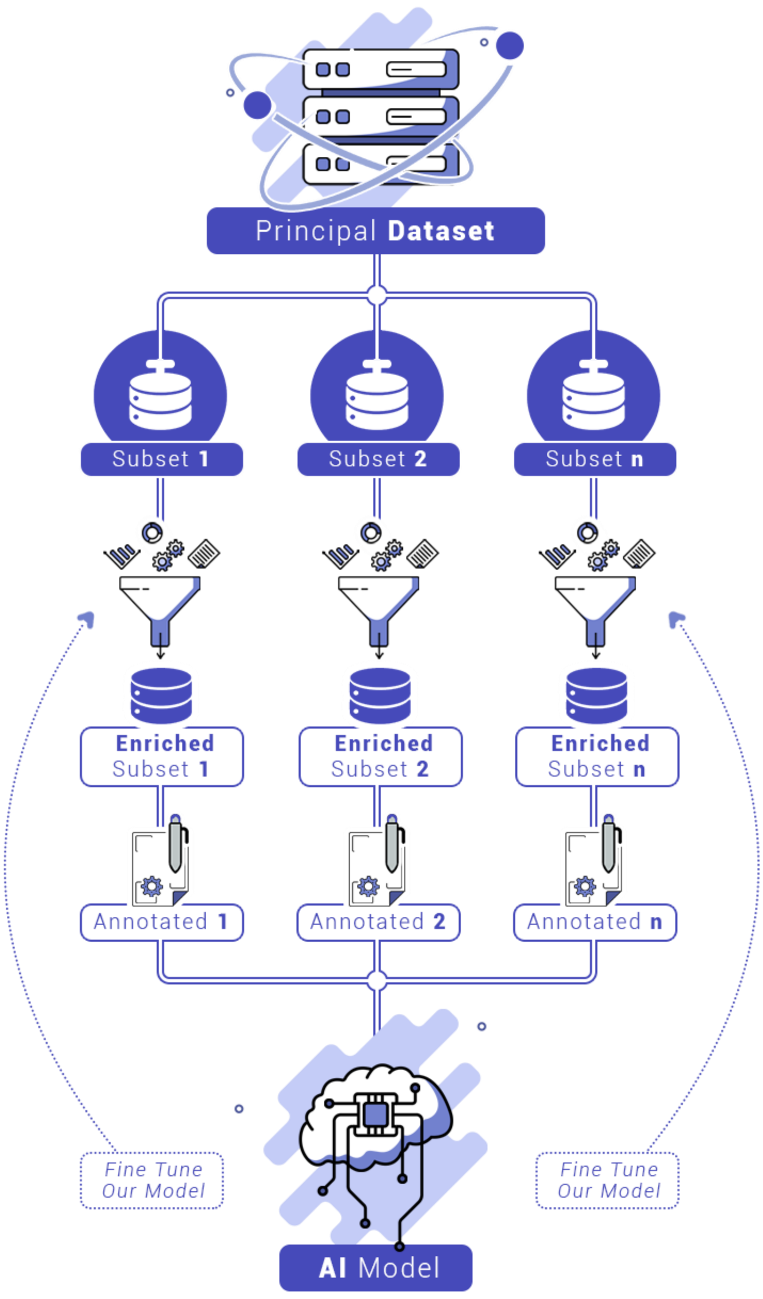


As we have seen in episode 1, good datasets are the foundation of any successful artificial intelligence, especially when it comes to NLP. Once our database is properly implemented and structured, the next steps are much simpler.

Today, we will see how our database will allow us to build sets of specific queries, which we will then be able to annotate and enrich, from other sources or from variations of known examples. Finally, we will see how to train and refine our model, to perfect it.

Objective

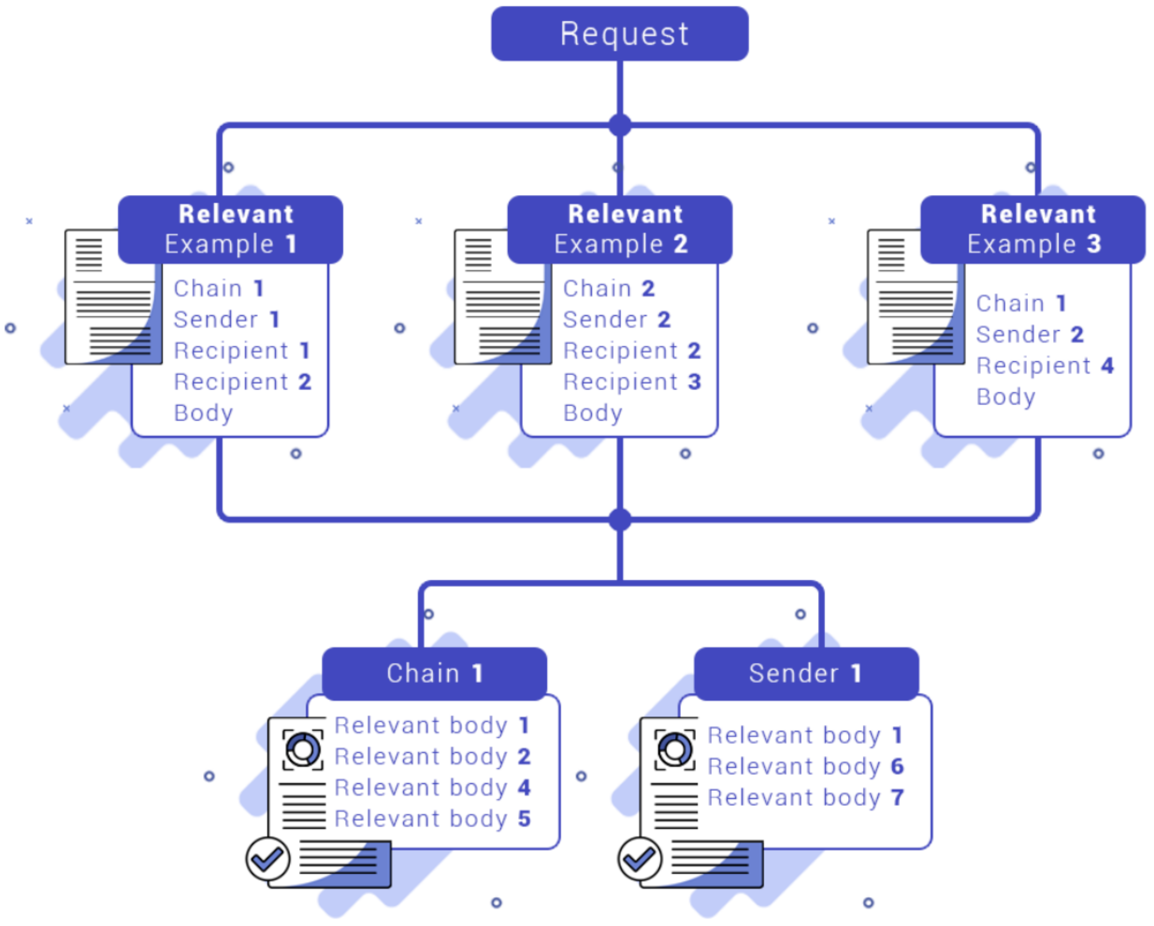
We now have an exceptionally large amount of data at our disposal, properly classified and structured. The aim is to use this data to build artificial intelligence capable of analyzing business requests. This data will then have to be annotated, enriched, and integrated into a model that we will then have to evaluate, adjust, and improve.



Building relevant subset from our database

Thanks to the classification of our data (described in S01-E01), we can extract from our index, data related to many business areas with one precise query. Obviously, this requires a detailed knowledge of the business areas concerned. In the case of our dataset, you cannot just type in a keyword to obtain the full list of all e-mails related to this domain. Indeed, you may end up with unrelated emails that contain the right word, used in another context or which may be, for example, the sender's name! It is then necessary to think smartly about your request.

For example, if we take the case of RFQs (Request for Quotes) like equity prices requests, we are sure to find in emails containing such requests references to equity related terms, such as “stock value” or “shares”. A query and quick analysis of the results on Kibana will find some relevant examples. Thanks to our structure, it is then extremely easy to identify the sender, the recipient, or the conversation in which the mail is, and to be able to retrieve, with this method, many linked mails, and thus many other examples.



To build a relevant subset on a given business domain it is necessary to find at least ten different examples of it. Obviously, the more varied the examples, the easier it will be to work on building a high-performance model, and the less it will need to be enriched. Therefore, it is mandatory to go through all the sources of emails mentioned above, to make sure that we have extracted the maximum potential from our dataset.

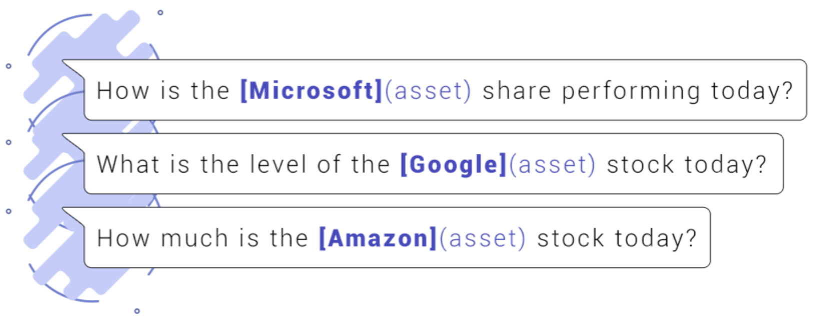
Once these data were extracted, they had to be annotated to allow the training of our AI. This part will consist in separating our different examples from the dataset into different "intentions".

- First, we will have to separate the emails we have retrieved into relevant requests and "decorative" parts. The latter will allow us to train our AI on introductory phrases, politeness, small talk, etc.
- Then, we can separate our requests into “intentions” that correspond to specific actions, for example: "equity request”, “bond request”, and so on.
- Then, in each of our intentions, we will have to determine "entities". These entities are key values for recognizing the relevant elements that belong to one or more intentions. For example, with the intention "equity request", we can imagine retaining the entities “asset”, "quantity", "book”, etc.
- After that, we must annotate our data. For this part, we could rely on this tool, which simplified our work a lot: https://spacy.io/universe/project/prodigy. After annotating the text, a simple transformation script, using regular expressions, allows the annotations to be modified to suit the chosen learning tool.

The need to enrich our subsets

Once done, you can theoretically train your first models. However, you will find that some of your subsets are not complete enough or do not cover enough ground. Even though you have exhausted the batch of examples available in your dataset, there is still a solution: create variations on examples you already have.

We will look at the specific example of requests about equities. Let us imagine that we have taken a few examples from our dataset, which we have annotated:

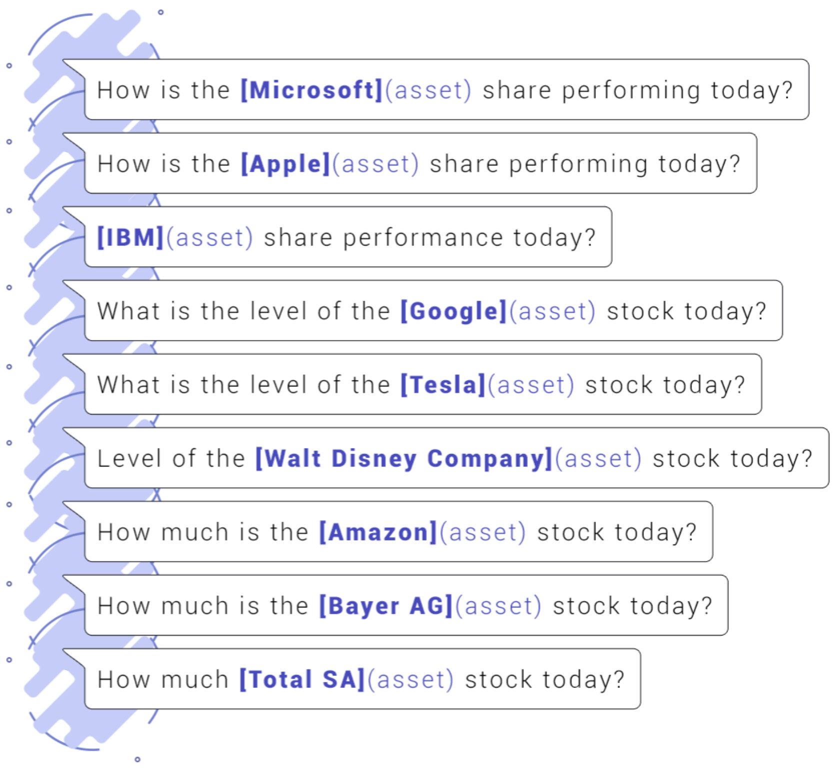


How is the **[Microsoft]**(asset) share performing today?

What is the level of the **[Google]**(asset) stock today?

How much is the **[Amazon]**(asset) stock today?

Duplicate the examples written previously, modify the annotated entities or their order, without changing the general meaning of the request. This way of proceeding will allow us to better integrate which notions are necessary to define a particular intention, without introducing bias.



How is the **[Microsoft]**(asset) share performing today?

How is the **[Apple]**(asset) share performing today?

[IBM](asset) share performance today?

What is the level of the **[Google]**(asset) stock today?

What is the level of the **[Tesla]**(asset) stock today?

Level of the **[Walt Disney Company]**(asset) stock today?

How much is the **[Amazon]**(asset) stock today?

How much is the **[Bayer AG]**(asset) stock today?

How much **[Total SA]**(asset) stock today?

Once we have enough annotated data, we are ready to train our model.

Train and adjust our model

To build our model, we relied on standard models. Building on an existing model has the advantage of simplifying the recognition of entities and intentions, as the existing model will be able to associate the entities of our model with its own entities.

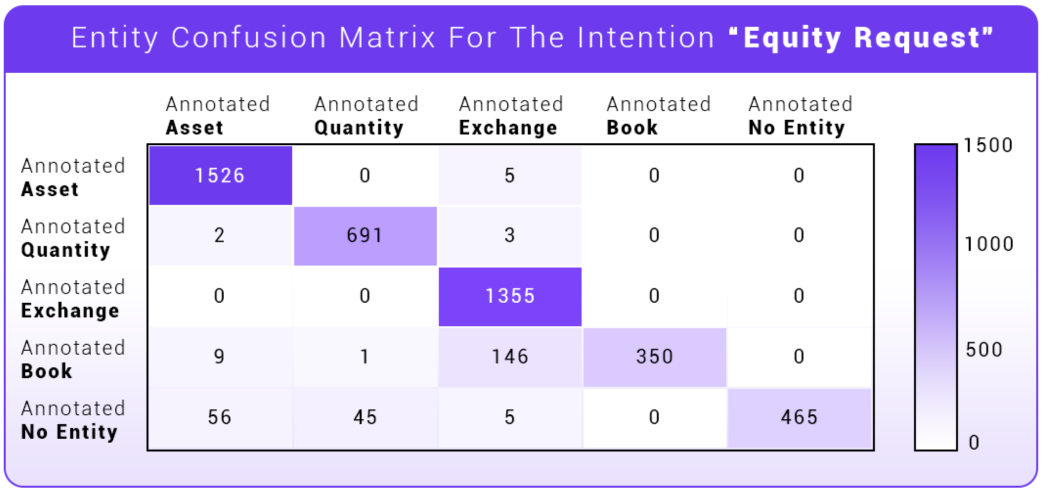
Hi want to **[Buy]**(side) **[10]**(quantity) **[Microsoft]**(asset) shares at **[260$]**(price)

Standard Model →	**VERB**	**CARDINAL**	**ORG**	**AMOUNT**
Customized Model →	SIDE	QUANTITY	ASSET	PRICE

This was particularly relevant in our case, with the intention related to equity transactions. For example, it is important to be able to differentiate between the query "What is the price of Microsoft shares?" and the query "What is the price of a guitar?", and therefore to be able to implicitly differentiate between company names and musical instruments.

We have therefore trained our first models from the previously annotated data. We now need to evaluate these results to see if they meet our needs. For this specific need, we will generate confusion matrices for our model.

Confusion matrixes are a fundamental tool for evaluating the performance of an AI on familiar ground on familiar ground: they give us an idea of the AI's capabilities on the data we have annotated. The results that are expected must therefore be perfect. If this is not the case, it will be necessary to rework our model to remove confusions, even before testing it on unknown elements.


Entity Confusion Matrix For The Intention **"Equity Request"**

	Annotated **Asset**	Annotated **Quantity**	Annotated **Exchange**	Annotated **Book**	Annotated **No Entity**
Annotated **Asset**	1526	0	5	0	0
Annotated **Quantity**	2	691	3	0	0
Annotated **Exchange**	0	0	1355	0	0
Annotated **Book**	9	1	146	350	0
Annotated **No Entity**	56	45	5	0	465

1500
1000
500
0

The matrixes show which entities have been correctly predicted and, if not, which ones they are confused with. We have the same thing for the intentions. We see here an example of this on the "equity request" intention described above.

It may happen that some entities or intentions are confused with each other. Adjustments will then be necessary. Sometimes the definition of intentions and entities will need to be rethought to remove ambiguities.

Once your model has passed all these tests perfectly, it is time to test it in real conditions. But that is for the next episode!

Conclusion

In this second episode, we have seen:

- How to prepare our model from the dataset created earlier
- How to enrich it
- And finally, how to train and adjust it

These 3 steps are especially important: only a bulletproof model will be able to do NLP efficiently. Indeed, you will have human users facing your model, capable of expressing a request in many ways.

Building an AI able to autonomously answer requests generated by human people is a gradual process. On our side, the development required rigorous and meticulous work of data mining, enrichment, annotation, and adjustment before being able to deploy our AI in production and process real transactions.

It is precisely this last step that we will address in the next episode! We will look at how to approach your first phases of testing in real conditions, what to expect from your AI and how to adapt it.

Stay tuned for episode 3!

Web

www.terranoha.com

Mailbox

sales@terranoha.com

Address

Terranoha SA, Route de Pré-Bois 29,
1215 Geneva - Switzerland

TERRANOHA
INTELLIGENT FINANCIAL BRIDGE