TERRANOHA

How to do NLP, episode 1:

How to transform an open-source data for NLP training

16/06/2022

Introduction

Good datasets are the foundation of any successful artificial intelligence, especially when it comes to NLP. A poorly mastered, biased, or too light dataset will result in a failing AI. On the other hand, well-controlled data sets will improve the results obtained. Therefore, working on the data upstream is a fundamental element of any development.

We will see here, through a detailed example, the different steps that allow obtaining relevant and easily buildable and accessible datasets.

Objective

Our goal here was to succeed in obtaining datasets allowing to train our AI to automate answers to bids and asks requests on P&G expressed by mail. There are many points on which it is necessary to train it to obtain an efficient AI: it should not only be able to correctly recognize RFQ (Request for Quotes) –though it is its main purpose-, but also recognize what is not RFQ, e.g., introductory formula, signature (automatic or not), chit-chat, e-mails sent by mistake, or which are not correctly formulated.

This implies the possibility of learning and training on specific notions, and therefore numerous and rich data sets. This is where our work is fundamental.

About the dataset

To develop an AI capable of analyzing threads of information and requests, from emails, it was necessary to obtain a relevant dataset to train it. Datasets of this kind are not easily provided, as companies in the financial sector are quite cautious with data disclosure.

There is however one of considerable size, which gathers email exchanges from the early 2000s in a large American company in the energy sector.

The company having gone bankrupt, all the non-sensitive mails were gathered and made public in a famous dataset which has already found many applications in the field of NLP, especially in sentiment analysis.

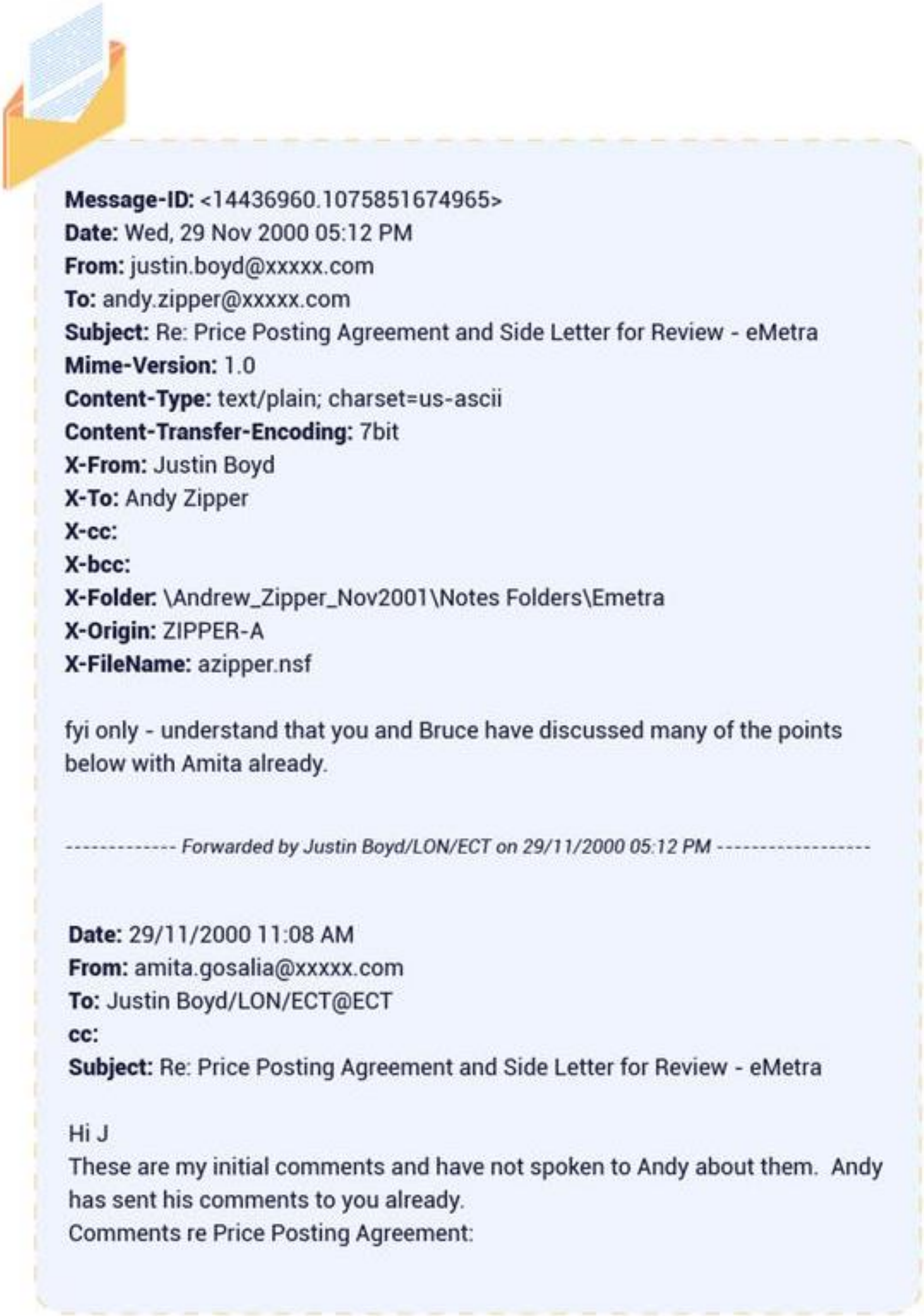
The dataset is a compressed folder containing the mailboxes of 150 employees of the company: their received, sent, and deleted mails, their outbox, etc. It contains a total of 517.401 mails.

These mails are both professional mails and informal mails, as can be exchanged between colleagues or even unwanted mails from outside.

The result is an extraordinarily rich dataset, but also filled with data that is *a priori* uninteresting from our point of view (classic employee chit-chat, for example). Therefore, it is important to have a very rigorous classification method, so that we can make the most of all this data.

More precisely, the aim here is to manage, after having processed this data, to easily constitute learning and training data sets. It was therefore essential to recover the essential elements allowing to easily identify the mails, to be able to classify them easily.

Once the dataset unzipped (1.5Giga), and browsed through manually to perform a first analysis, we found out a steady pattern in the email format: some of its attributes were always found at the same place:



Message-ID: <14436960.1075851674965>
Date: Wed, 29 Nov 2000 05:12 PM
From: justin.boyd@xxxxx.com
To: andy.zipper@xxxxx.com
Subject: Re: Price Posting Agreement and Side Letter for Review - eMetra
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Justin Boyd
X-To: Andy Zipper
X-cc:
X-bcc:
X-Folder: \Andrew_Zipper_Nov2001\Notes Folders\Emetra
X-Origin: ZIPPER-A
X-FileName: azipper.nsf

fyi only - understand that you and Bruce have discussed many of the points below with Amita already.

------------- *Forwarded by Justin Boyd/LON/ECT on 29/11/2000 05:12 PM* -----------------

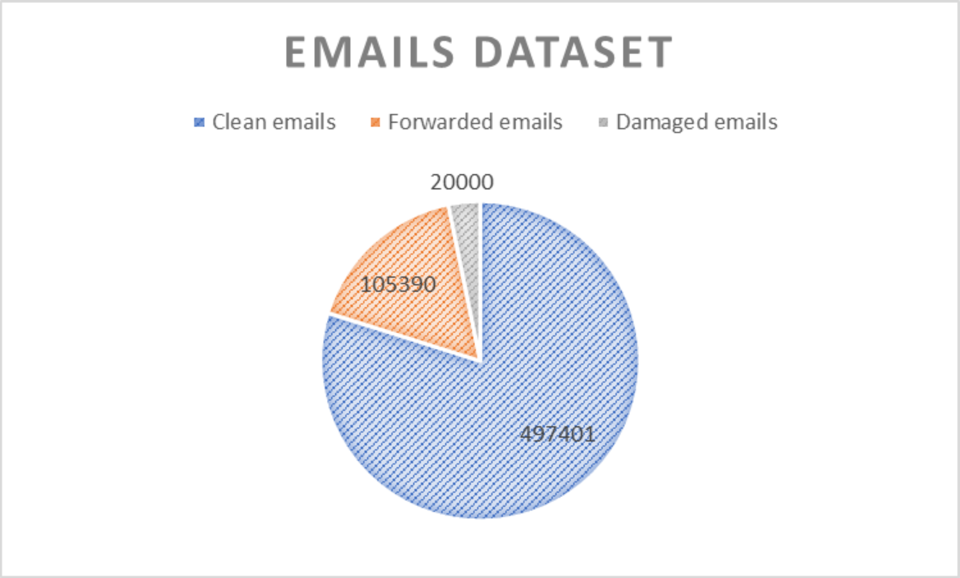
Date: 29/11/2000 11:08 AM
From: amita.gosalia@xxxxx.com
To: Justin Boyd/LON/ECT@ECT
cc:
Subject: Re: Price Posting Agreement and Side Letter for Review - eMetra

Hi J
These are my initial comments and have not spoken to Andy about them. Andy has sent his comments to you already.
Comments re Price Posting Agreement:

Two points in caught our attention:

- A mail migration went wrong and "spoiled" some mails which are filled with unwanted characters. These mails should be identified and treated to be correctly exploited;
- Thousands of mails that are mail chains. Decomposing them would allow us to improve the volume and quality of our dataset

The dataset is thus presented with the following volumes:



After processing, this would give us a total of 622.791 emails.

Data processing

The main goal was to be able to create subsets, according to our punctual needs, to train our AI on customed sets. What can be done with such a vast dataset? We have imagined numerous business-related uses, like identification of company names, equities, or recognition of entities related to the trading business. We also had in mind usages that are more generic: we can train our AI to recognize salutations, email signatures, and everything that is not relevant for the business. This is not an exhaustive list: other uses can be determined later, according to our needs.

Data with clear meta-data is easily understandable and accessible. It was then necessary to obtain and isolate as much information as possible from emails, and to parse them efficiently, to enable fast and efficient querying.

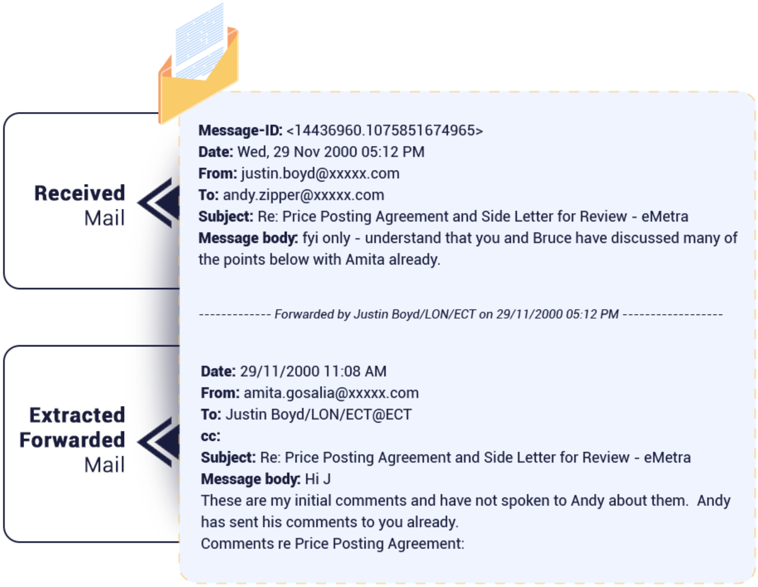
The parsing was easy, because, as we saw before, in each of these e-mails, we were assured of easily recovering an identifier associated with a sender, a subject, a recipient, etc. By isolating these elements, we can perform a relevant classification, making data easy to manage.

All the elements presented here will be stored as full-fledged e-mail attributes and allow us to easily link mails together and to reconstruct the chronology of a conversation. We can focus on the interest of the Message-ID. The Message-ID is a unique identifier generated by the company's email management tool. It allows us to reconstitute a chain of mails from one of them, for example to give an interesting mail a context from the dataset.

The bulk of the work, therefore, consisted of two key tasks, both of which were aimed at improving the quality of our data:

- The correction of e-mails damaged during a migration, and now difficult to read for an AI: about 20.000 mails were "repaired" out of the 517.401 that make up the dataset;
- The processing chains of transferred e-mails, and splitting them into independent mails, allowed us to add 2663 emails (with update information) to the dataset

After processing the data, we get something like this for each email:



Message-ID: <14436960.1075851674965>
Date: Wed, 29 Nov 2000 05:12 PM
From: justin.boyd@xxxxx.com
To: andy.zipper@xxxxx.com
Subject: Re: Price Posting Agreement and Side Letter for Review - eMetra
Message body: fyi only - understand that you and Bruce have discussed many of the points below with Amita already.

------------- *Forwarded by Justin Boyd/LON/ECT on 29/11/2000 05:12 PM* -----------------

Date: 29/11/2000 11:08 AM
From: amita.gosalia@xxxxx.com
To: Justin Boyd/LON/ECT@ECT
cc:
Subject: Re: Price Posting Agreement and Side Letter for Review - eMetra
Message body: Hi J
These are my initial comments and have not spoken to Andy about them. Andy has sent his comments to you already.
Comments re Price Posting Agreement:

Storage solution

Once this work was done, we had to choose a storage solution that would allow us to efficiently build relevant datasets from our original dataset.

Several solutions were offered to us, especially in the field of NoSQL. We can note two main ones, which are MongoDB and Elasticsearch.

Here is a comparative table, giving the best of these two solutions in different fields:

	Elasticsearch	MongoDB
Distributed search	☑	✕
Distributed storage	✕	☑
Full text search	☑	✕
Search analyzers	☑	✕
CRUD operations	✕	☑
Visualization tool	☑	✕

We chose Elasticsearch. What sealed our choice, besides its document indexing, was the vast range of searches, infinitely adaptable and extremely flexible. In addition to the notion of "score", which gives a value to the relevance of the result found by Elasticsearch following a query, Elastic search has a powerful JSON-based DSL, which allows development teams to construct complex queries and fine tune them to receive the most precise results from a search. It also provides a way of ranking and grouping results.

In addition, we could use Kibana, Elasticsearch's visualization tool. Easy to use and configure, because dedicated to Elasticsearch, it allows us to easily get an overview of our data, and to intuitively formulate queries to build and evaluate datasets. This made dataset construction easier by giving us better understanding of our data. This is the first cornerstone to build our datasets efficiently. Finally, since we were not planning to amend the database, one of MongoDB's strengths was mitigated.

To reinforce this solution and ensure that we did not skip some relevant data, several complementary solutions were put in place. This is where all the other data we have recovered comes into play: identifier, date, sender, recipient. All this data allows us to reconstruct a conversation. For example, relying on the Message-ID of an email allows us to recover all the emails that may have been transferred or linked to this email. Also, precisely knowing the protagonists and the date of a conversation allows us to reconstruct it from a single email.

The work done upstream combined with this solution allows us to take advantage of a colossal dataset, detailed and classified in a relevant way, which allows us to obtain any kind of dataset adapted to our needs.

The work upstream combined with this solution allows us to take advantage of a colossal dataset, incredibly detailed and classified in a relevant way, which allows us to obtain any kind of dataset adapted to our needs.

Thanks to Kibana, we can visualize our data easily. Thus, there was only to add a pinch of human intelligence to build datasets adapted to our multiple needs: recognition of entities related to the trading business, company names, signatures, and so on.

Conclusion

As we have seen, a lot of preparation work is necessary to obtain quality results with NLP:

- First, it is essential to obtain a source of raw data relevant to the work you wish to accomplish, which can be very time-consuming. Indeed, it is quite rare to find a dataset that perfectly matches your needs. It is therefore sometimes necessary to find datasets that do not correspond exactly to the initial wish, and to rework them afterwards. This is what we did.

- It is also imperative to think about intelligent integration of this data. This has two folds:

  - ✓ The first is to work on this raw data to get the most out of it: this can mean extracting meta-data (like we have done here), working on data to make it more in line with our needs, or even deleting superfluous data, for example.
  - ✓ The second aspect of this integration relates to the storage and access to this data. As we saw earlier, there are many solutions available to achieve this. Ranging from the most classic and proven solutions (Oracle SQL...), to the more recent and flexible NoSQL solutions, such as MongoDB or Elasticsearch. All of them have their advantages, and it is important to choose a solution in line with what you want to achieve.

Once done, you have the foundation to build a robust NLP.

The next challenge is to construct your datasets and prepare them for NLP. This will be covered in our episode 2, stay tuned!

Web

www.terranoha.com

Mailbox

sales@terranoha.com

Address

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

TERRANOHA
INTELLIGENT FINANCIAL BRIDGE