Grammatical and contextual analysis based on NLP in loss reports.

Stage 2, training and implementation of the NER model

Eduardo Santos, Alberto de Obesso, Rómulo Troncoso,

***Abstract*—The Loss Runs Reports provide enough contextual clues to enable a grammar-driven approach for information extraction. Therefore, it is necessary to train a specialized NER model in the task of recognizing typical entities in loss reports. Using previously trained language models such as *Word2vec* and tools such as *Gensim* and *Spacy*, it is possible to extract topics of interest automatically, for this; deep learning methods are applied in the word embedding paradigm. In this way, it will be possible to robustly identify, format-independent, the entities related to the topics (i.e., *MONEY* entity to TOTAL INCURRED topic). So the use of GPUs is recommended for the training stage in order to reduce development time.**

***Index Terms*—Loss run, NER, Spacy, Gensim, Python, EC2 cloud computing.**

I. Project plans

LOSS reports contain information in context from which entities related to topics of interest can be extracted using a name entity recognition system. This through the application of contextual analysis models specialized in language and loss reports with programming tools with wide support and scalability such as Python, Gensim and Spacy. One way to optimize development time is cloud computing with services like AWS EC2 with graphics card hosting. The cloud processing service requires the configuration of the development environment and charges the user only per hour of virtual machine use.

The purpose of this document is to describe the scope, objectives and milestones for Loss Runs grammar-driven approach.

A. Scope

This document will be used as a reference document throughout the project to ensure that all parties are aligned in their understanding of the project's objectives, activities and ways of working. It incorporates the:

- project scope,
- objectives,
- project milestones,
- responsibilities.

B. Intended audience

This document is intended to provide information to:

- management,
- staff involved in implementing the project,
- project scope and objectives.

II. Background

The training stage in a neural network (NN) increases according to the amount of training data and the topology of the NN itself, so deep architectures represent a significant computational load that translates into better performance of the recognition system. Using specialized processing cores optimizes the machine learning stage. The use of specialized processing cores optimizes the machine learning stage, which is why a graphics card is required in artificial intelligence applications at an industrial level. Graphics processing units (GPUs) can speed up computational processes for deep learning, when you train a deep learning model, two main operations are performed, the forward pass and backward pass, both of these operations are essentially matrix multiplications, operations can be performed at the same time instead of doing it one after the other when we use GPU. Using gpu instead of cpu can speed up the training stage 2-3 times [1]. GPUs has hundreds of simpler cores and can handle thousands of concurrent threads also maximize floating-point throughput.

Therefore, a text string is processed as shown in the figure 1 with the space tool, resulting in an object with defined attributes such as NER, TOKEN, SENTENCE, etc.

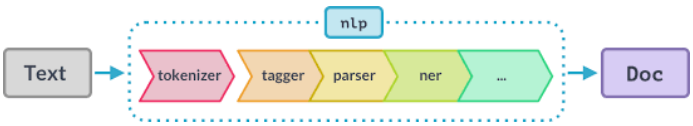

Fig. 1. Spacy pipeline NLP model for a defined text [2]

III. Objectives

To automatically extract relevant information in loss reports, it is necessary to create a robust and stable NER model for several report formats. For this, the following general stages have been identified:

- tokenize,
- position tag,
- supervised tagging,
- chunking with grammars,
- info extraction.

In the current state of development, the last 3 objectives are worked on in an *online* scheme. To optimize the training of the NER model, it is necessary to use graphics card accelerated computing, so one development option lies in cloud computing. With what particularly the objectives are (assuming AWS service):

- define EC2 service,
- configure EC2 services,
- configure VM on EC2,
- train final NER model.

In order to identify the list of entities related to the topics as shown in table III:

TABLE I
ENTITIES RELATED TO EACH TOPICS OF INTEREST IN LOSS RUNS REPORTS

Entities	Topics
DATE	report_date
ALPHANUM	policy_num
MONEY	total_inc, alae, loss_r, loss_e
ORGED	insured
ORGIN	insurer

Use GPU to train the machine learning model with large datasets so it can be more accurate, take advantage of GPUs parallel architecture. This will allow to reduce training time to hours instead of days. Generally speaking, accelerated computing is applied in the final stage of machine learning as shown in the figure 2.

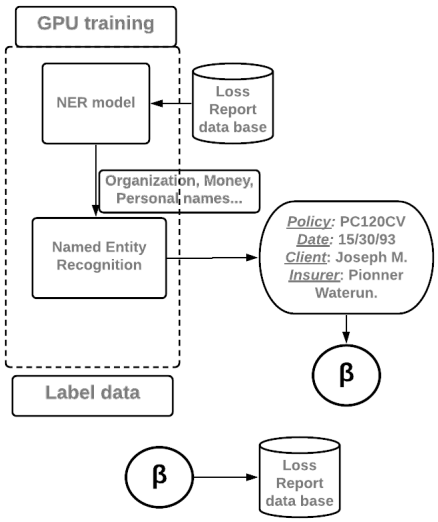

Fig. 2. Stage 2, NER model training for specific entities in loss reports.

IV. JUSTIFICATION

Training models for image classification and language processing involves compute-intensive matrix multiplication and other operations for that reason is recommendable the use of GPU, to select a GPU that suits for the problem we focus on the following aspects.

- Memory bandwidth,
- number of cores (Determines the speed at which the GPU can process data),
- video RAM size,
- processing power.

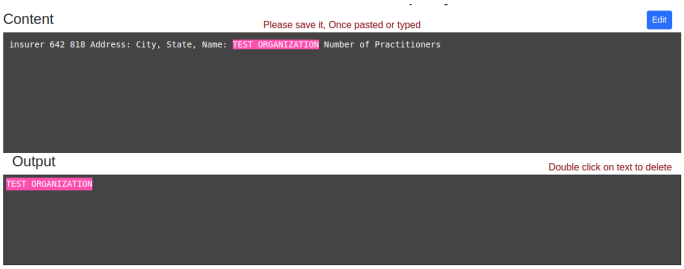

Fig. 3. Supervised tag toolkit

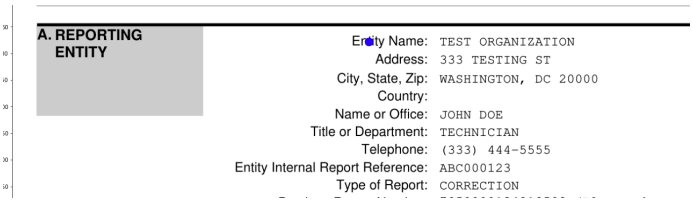

Fig. 4. NPDB report where only the topic of interest is selected

A. Scope

The scope of this document is to show the benefits of use GPU to train the model [3].

- **Select AWS Deep Learning Container**: Image pre-installed with deep learning framework, skips the complicated process of building an optimizing environment from scratch.
- **Configure and connect to AWS instance**: Choose an Amazon instance type, (service to create and run virtual machines in the cloud)
- **Configure instance with AWS credentials**: to store, manage and deploy docker container images
- **Train the model with Deep Learning Container**: Start the process of training

V. RESULTS.

Using the toolkit shown in the figure 3, 160 reports were labeled in a supervised way for the temporal NER model, where each document has an average of 6 entities. For this amount of data, the training stage takes approximately 2 hours mounted on the CPU RYZEN 7 3700X, S-AM4, 3.60GHz, 8-Core. Recognition rate is over 95% for Gren Hill, Pionner, and National Professional Database (NPDB) templates, consult figure 4.

VI. DISCUSSION

Given the several types of formats used in loss reports, it is necessary to train the NER identification model with a greater number of examples, which implies an increase in time in the learning stage, for which the use of GPUs. to determine the optimal hyperparameters of the model is recommended in order to increase the general performance of the automatic recognition system of relevant information in loss reports.

For more information see the **GitHub** repository [4]: Assym github repo

REFERENCES

[1] (Jan. 7, 2020). "spaCy training using GPU," mc.ai, [Online]. Available: https://mc.ai/spacy-training-using-gpu/ (visited on 10/13/2020).

[2] (Jul. 2, 2019). "Language processing pipelines · spaCy usage documentation," Language Processing Pipelines, [Online]. Available: https://spacy.io/usage/processing-pipelines (visited on 10/13/2020).

[3] (). "Deep learning benchmarks comparison 2019: RTX 2080 ti vs. TITAN RTX vs. RTX 6000 vs. RTX 8000 selecting the right GPU for your needs," Exxact, [Online]. Available: https://blog.exxactcorp.com/whats-the-best-gpu-for-deep-learning-rtx-2080-ti-vs-titan-rtx-vs-rtx-8000-vs-rtx-6000/ (visited on 10/13/2020).

[4] (Jan. 7, 2020). "Asymm-developers/nowinsurance-loss-runs," GitHub, [Online]. Available: https://github.com/Asymm-Developers/nowinsurance-loss-runs (visited on 10/13/2020).