HAPe: Optimizing Customer Relation by Automatic Task Distribution using Constrained Optimization and Natural Language Processing

Romain Gemignani, Eunice Akani, Jean-Pierre Delrieux, Abdoulaye Sayouti Souleymane

HAPE: OPTIMIZING CUSTOMER RELATION BY AUTOMATIC TASK DISTRIBUTION USING CONSTRAINED OPTIMIZATION AND NATURAL LANGUAGE PROCESSING

Romain GEMIGNANI
Enedis – France
romain.gemignani@enedis.fr

Eunice AKANI
Enedis – France
adueni-adjoba-eunice.akani@enedis.fr

Jean-Pierre DELRIEUX
Enedis - France
jean-pierre.delrieux@enedis.fr

Abdoulaye SAYOUTI SOULEYMANE[1]
Avignon University – France
abdoulaye.sayoutisouleymane@gmail.com

ABSTRACT

Since 37 million customers use Enedis services daily, the company must manage massive requests. To affect these requests as fast as possible to the agent, the company needs several advanced planning models. Therefore, the problem is to distribute tasks to the agents automatically while respecting the time available to them to carry out the tasks and their competence. To this end, we present the different models embedded in HAPe, an application that allows us to address the problem. This paper proposes a constrained linear optimization model for automatic distribution, a text classification model to help predict agent skills, and an automatic entity recognition model that extracts essential domain-specific information. These two NLP models are based on our language model, which we implemented. The results obtained from our different studies are promising and allow us to increase the agents' efficiency. This study is the first study on this subject for DSO activities.

INTRODUCTION

The biggest challenge in business is planning. It is crucial to have an optimal schedule to optimize available resources. Most of the work on this theme used constrained optimization algorithms. The idea is to minimize or maximize an objective function under multiple constraints. The implementation of these constraints is still tedious if we want to have a linear problem to solve. Enedis, a French Distribution System Operator, faces this challenge. Indeed, having an application that centralizes tasks that several agents must handle, optimizing the task assignment is a significant challenge. Thousands of tasks must be assigned to thousands of agents according to their skills and available time. On the one hand, this problem involves hundreds or even thousands of variables and constraints. On the other hand, matching agents' skills to tasks according to their description requires advanced NLP models. This paper introduces the different models we have implemented in our back-office request centralization application: HAPe ("Hypervision des Activités par Peignage"). The optimization was based on the agent's skills, so we need them. The skills depend on different requests. A skill can group several types of requests. Thus, we had to predict the different requests to have the groups. To do this, we used a natural language processing model in front of a constrained optimization system. We also introduce different natural language processing models to help agents focus on tasks with high benefit priority and process them quickly and efficiently.
Our contribution can be summarized as follows:

- We made a text classification system to gather the different types of requests as a skill.
- We defined and implemented constrained optimization algorithms for automatic job distribution.
- We implemented a named entity recognition system specific to Enedis and its activities.

OPTIMIZATION OF TASK DISTRIBUTION

Definition of the problem

The objective is to optimize the distribution of tasks to the agents according to their competence and the time they are given. We are faced with a constrained optimization problem. Thus, the continuation of this part will define the objective function to be maximized (the number of tasks assigned to an agent) as well as the various constraints that govern the problem. It is essential to consider that each task has a different priority. We define the objective function to be maximized as the sum of each task assigned to an agent multiplied by its priority. Given m as the number of agents, n as the number of tasks, x_{ij} that means task i is assigned to agent j, and p_i the priority of task i, we defined the objective function mathematically as follows:

$$f(x) = \sum_{i=0}^{n} \sum_{j=0}^{m} x_{ij} * p_i \quad (1)$$

The constraints governing the problem are as follows:

- Several agents must not process a task, i.e., it can be distributed to at most one agent
- An agent cannot process tasks if he does not have enough time available for these tasks, i.e., the distributed tasks must be less than the total activity time of the agent.

1 Was at Enedis during work

Once the various constraints have been stated, the problem must be defined mathematically. Given m as the number of agents, n as the number of tasks, x_{ij} that means task i is assigned to agent j, p_i the priority of task i, t_{ij} as the processing time of task i by agent j, and h_j the planned processing time for agent j, the mathematical definition of the problem is as follows:

$$max \sum_{i=0}^{n}\sum_{j=0}^{m} x_{ij} * p_i \quad (2)$$

s.t. $\sum_{j=0}^{m} x_{ij} \leq 1, \forall\, i \in \{0..m\}$ (3)

$\sum_{i=0}^{n} t_{ij}x_{ij} \leq h_j, j \in \{0..m\}$ (4)

$x_{ij} \in \{0,1\}, \forall\, i,j$ (5)

The problem thus defined, we proceeded to verification on a reduced example. The following subsection presents the details of this assessment.

Example

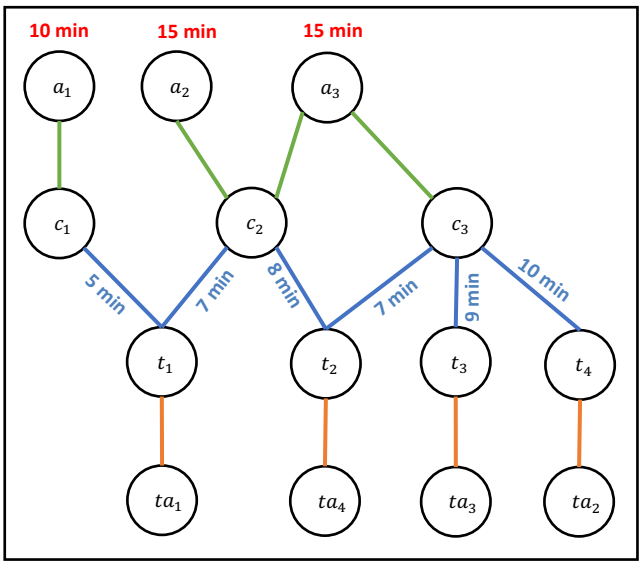

Figure 1 - Link between the different entities involved in the optimisation. a_i : the agents, c_j : the skills, t_k : task types, ta_l : tasks. In red the planned time of each agent, in green the link between skills and agents, in blue the link between skills and task types as well as the unit processing time of the corresponding task type and in orange the link between task types and tasks.

Let's assume we have three agents, three different skills, and four tasks with the same priority 1. Figure 1 shows the different links between entities. For example, agent 1 has skill 1 with a planned time of 10 minutes and can perform the type 1 task, which requires 5 minutes each for processing.

This data allows us to deduce the table corresponding to the matrix t_{ij} of equation (4). Since the agents can only handle the tasks they are competent to handle, we set the time of the tasks they cannot handle to 0. Some agents may have several skills that allow them to handle a task. In this case, only the minimum time required to process the task will be taken. The resulting matrix is therefore:

$$t_{ij} = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 7 & 0 & 0 & 8 \\ 7 & 7 & 10 & 8 \end{pmatrix}$$

All tasks have the same priority, so we can define the objective function to be maximized as follows:

$$f(x) = x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + x_{31} + x_{32} + x_{33} + x_{41} + x_{42} + x_{43}$$

Therefore, we can deduce the problem's different constraints first, which involves the agents' planned time (see equation 4):

$$\begin{cases} 5x_{11} + 0x_{21} + 0x_{31} + 0x_{41} \leq 10 \\ 7x_{12} + 0x_{22} + 0x_{32} + 8x_{42} \leq 15 \\ 7x_{13} + 7x_{23} + 10x_{33} + 8x_{41} \leq 15 \end{cases}$$

Then the constraint of the uniqueness of the task (eq. 3):

$$\begin{cases} x_{11} + x_{12} + x_{13} \leq 1 \\ x_{21} + x_{22} + x_{23} \leq 1 \\ x_{31} + x_{32} + x_{33} \leq 1 \\ x_{41} + x_{42} + x_{43} \leq 1 \end{cases}$$

And finally, the constraint on the values of the variables (see equation 5):

$$x_{ij} \in \{0,1\}, \forall\, i \in \{1,2,3,4\}\ et\ j \in \{1,2,3\}$$

To solve the problem, we used the solver HiGHs [6] and obtained the following solution matrix:

$$x_{ij} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

It is easy to check that the result obtained maximizes the number of tasks assigned to the agents while respecting the problem's constraints.

The following section presents the NLP model used to predict the types of tasks that form an agent's skills.

TEXT CLASSIFICATION

Enedis receives requests from suppliers called "miscellaneous requests". The agents analyze each task, process the ones they are competent at, or leave them in the processing bin with the risk of taking them up again at another time. This processing generates a lot of wasted time because requests are processed randomly from the list, not according to their urgency or importance. In this section, we present our natural language processing model, which allows us to classify various requests automatically.

Method

Since the breakthrough of the transformer [1], many language models have been created. BERT [2] and RoBERTa [4] are some illustrations of this breakthrough

in English. For French, models like CamemBERT [3], based on the architecture of RoBERTa, and FlauBERT [5], based on the architecture of BERT, were introduced. These models give high results for downstream tasks like text classification or information extraction. We introduce EneBERT, a French language model for DSO activities for the technical language of the company to be understood by the model. EneBERT is based on the architecture of CamemBERT. Thus, we finetuned EneBERT on the dataset to perform the text classification task. We do not present in detail We do not present the details of the model in this article, but for more information, refer to [3], as they have the same architecture.

Dataset

As this is a classification of various requests, we have collected and asked the agents to label them. This way, we obtained about 30,000 samples for 14 classes (types of requests). One of the disadvantages of this dataset is its size; the other is that the data is not balanced within the different classes. To remedy the problem, we divided our dataset into two datasets: one for training and another for evaluation. Then we increased the training dataset to minimize the number of errors in the classes with the least amount of data.

	F1-Score	F1-score AUG	Nb samples
Class_0	0.98	0.98	1 333
Class_1	0.94	**0.96**	303
Class_2	0.84	**0.89**	527
Class_3	0.97	0.97	263
Class_4	0.92	**0.94**	185
Class_5	0.89	**0.91**	114
Class_6	0.83	**0.92**	189
Class_7	0.98	**0.99**	113
Class_8	0.90	**0.93**	335
Class_9	0.92	**0.95**	293
Class_10	0.64	**0.79**	83
Class_11	0.80	**0.83**	127
Class_12	0.76	**0.79**	57
Class_13	0.56	**0.80**	47

Table 1 - F1 score on the evaluation dataset for the models trained with and without training data augmented. Nb samples is the number of samples of each class in the evaluation dataset.

Training parameters

Using CamembertForSequenceClassification from the transformers library of Hugging Face [7], we trained our model for 5 epochs. We kept only the epoch's weight which minimizes the loss function and maximizes the score. We used a batch of 24, a learning rate 5.10^{-6} with a weight decay of 10^{-2}. The results obtained are given in the next section.

Results

We evaluated our method on the evaluation dataset using F1 Score, which is the harmonic mean between precision and recall. We computed the F1 score for each class and compared the result with another model we trained without using the training augmented dataset. Table 1 shows the results obtained. We can see that the model trained on augmented data gives better results with the F1 score being relatively good for all classes. Thus, the model has been integrated to predict the different types of tasks to help optimize the automatic task distribution.

NAMED ENTITY RECOGNITION

From the dataset of the company, agents can extract numerous critical pieces of information. Thus, helping the agents by creating an automatic system that can extract this information automatically will be a real-time save for them. Many systems of named entity recognition exist and perform well for the task. However, the information extracted is general, not specific to the domain. So, it can be useless for our domain. It was, therefore, essential to create a named entities recognition system specific to extract the information that can help agents process their tasks. This named entity recognition system can also help to increase our text classification system. Suppose we have the two sentences “Merci d'annuler l'affaire X66832” and “Merci d'annuler l'affaire C45545”. The two sentences are mostly the same; only the case number differs, and the classification does not depend on the case number. Thus, replacing the case number with the named entity tag as a token will help the model to focus on the critical information during training. This section presents the named entity recognition system we created and integrated into HAPe.

Dataset

We used a dataset that contains 286000 samples. Unfortunately, the dataset was not labeled for named entity recognition. Thus, we decided to annotate 6000 samples manually. First, we processed the dataset to clean it and put it into the correct format. And then, we annotated following the BIO (Begin, Inside, Outside) annotation. Since entities can be divided into multiple tokens, this format allows us to know the boundaries of the named entity to be annotated; where it starts (B) and where it ends (the last I), what is an entity (B or I) and what is not (O). Thus, we have 12 entity tags: address, supplier's name, hour, date, electricity meter index, off-peak electricity meter index, electricity meter index in full hour, power, phone number, case number, card number, and reference point measurement (PRM). Using the BIO format, the different labels are: <O> <B-Adress> <I-Adresse> <B-Supplier> <I-Supplier> <B-Hour> <B- Date> <I-Date> <B-Indx> <B-Puissance> <B-Num-Tel> <B-Indx-HC> <B-PRM> <B-Case-Num> <B-Card> < B-Indx-HP>.
For the remaining 280.000 samples, we used some regular expressions to annotate them automatically to train our NER model. We divided the dataset annotated manually into three parts: a training dataset to train the model, a validation dataset to perform an evaluation during training, and the testing dataset to report the performance of the NER model.

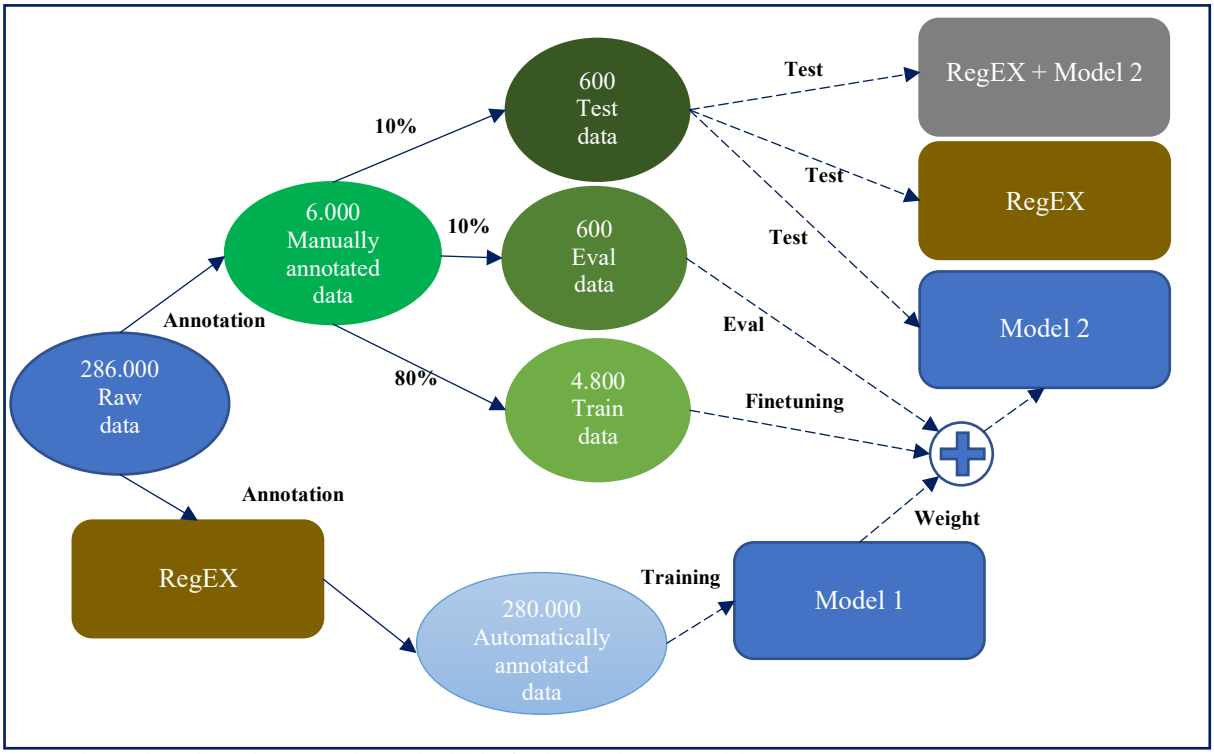

Figure 2 - Schema describing the steps taken to build the final named entity model

Method

We used EneBERT, a French language model for DSO activities that we created to train our NER model. Hugging face provides an excellent template for the training step, so we used it to train the model. Due to the need for more data annotated by humans, we used a strategy to compensate for this lack. We decided to train our model on the 280.000 samples labeled automatically using regular expressions, and then we finetuned the model on the training set of the 6.000 labeled data. We also evaluate a regular expression system on the test set to compare it to the named entity recognition system trained using a deep learning approach. Finally, we create another approach that merges the regular expression system with the one trained. Figure 2 gives the different steps we perform to have the three approaches. In Figure 2 we can see that model 2 obtained was initialized with the weight of model 1 that was trained on the automatically annotated data. The following section presents the results obtained from the different approaches.

Result

We evaluate the different approaches on the test dataset and report the result in Table 2. We compute the F1 Score for all the tags. The approach that combined the regex (regular expression) and the model gave the best result most of the time. Also, for entities that contain a number, the regex system performs well. Figure 3 gives an example of named entity detection by our model.

	Regex	Model	Regex + Model	Support
<O>	**1.0**	0.99	**1.0**	44693
<B-Adress>	0.83	0.84	**0.85**	80
<I-Adress>	0.74	0.75	**0.8**	500
<B-Supplier>	**1.0**	0.89	**1.0**	127
<I-Supplier>	0.95	0.90	**0.97**	79
<B-Hour>	**1.0**	0.95	**1.0**	100
<B-Date>	**0.98**	0.94	**0.98**	513
<I-Date>	0.0	0.53	**0.53**	13
<B-Indx>	0.87	0.80	**0.90**	169
<B-Power>	0.98	0.93	**0.99**	82
<B-Num-Tel>	1.0	0.82	**1.0**	61
<B-Indx-HC>	**0.75**	0.5	**0.75**	27
<B-PRM>	**1.0**	0.79	**1.0**	43
<B-Case-Num>	**1.0**	0.86	**1.0**	360
<B-Card>	**1.0**	0.96	**1.0**	220
<B-Indx-HP>	**0.84**	0.51	**0.84**	28

Table 2 - F1 Score of the 3 approaches on test dataset.

Bonjour. Merci de confirmer l'index de sortie HP 2120 HC 81675 car consommation négative très important et index inférieurs aux index de MES du 02.01.2023. Cdt

---

Bonjour, Merci d'annuler l'affaire X66832. Cdt

Figure 3 - Examples of named entity recognition system apply on two sentences. In blue and orange an electricity meter index in off-peak and full hour respectively. In green, a date. And in red, a case number.

CONCLUSION

In this paper, we present our application HAPe, which was made thanks to the constrained optimizing method and natural language processing. We showed that we introduced a first brick named entity recognition specific to the company with no labeled data and gave a good result for the task. Our automatic task distribution system performs well and is a real-time save.
The industrial version of HAPe has been tested, with an estimated return on investment of around 3M€ per year. It manages about 100 000 customs requests per day dispatched in 1500 operators' planning. This fully manual task is now entirely done by the application and the ai models, which continuously learn data generated by HAPe used itself.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and A. N. Gomez and L. K. and I. Polosukhin, 2017, "Attention Is All You need", *arXiv.*

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, 2019, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, ACL, vol.1, 4171-4186.

[3] M. Louis, B. Muller, P. J. Ortiz Suarez, Y. Dupont, L. Romary, E. de la Clergerie, D. Seddah, and B. Sagot, 2020, "CamemBERT: a Tasty French Language Model", *Proceedings Association for Computational Linguistics (ACL)*, 7203-7219.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. "Roberta: A robustly optimized BERT pre-training approach.", *ArXiv.*

[5] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L.Besacie, and D. Schwab, 2020, "FlauBERT: Unsupervised Language Model Pre-training for French", *Proceedings Conference on Language Resources and Evaluation, LREC, ELRA*, 2479-2490.

[6] Q. Huangfu, and J. A. J. Hall, 2020. "Parallelizing the dual revised simplex method", *Proceedings Mathematical Programming Computation*, 119-142.

[7] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, 2018. "Transformers: State-of-the-Art Natural Language Processing", *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, ACL, 38-45.