Automating Customer Experience Agents' Evaluation with Natural Language Processing

Nicolás García Aramouni[1], Nahuel Pelli[1], Lucas Soules[1], Julieta De Antonio[1], Ines Nosetti[1], Juan Cazorla[1], and Pedro Fraguas[1]

[1]Accenture

The importance and relevance of online commerce in society have grown uninterruptedly over the last years, with people changing their purchasing habits. The COVID-19 pandemic significantly accelerated this adoption, as quarantines forced people to stay in their homes. Under this scenario, online shopping became a more convenient option, considering product availability, the possibility to compare prices and shipping options, and door-to-door delivery. With this, e-commerce retailers and marketplaces must give excellent customer service to their customers, especially in this context when most activities are returning to their pre-COVID levels and restrictions. To manage customers that have any kind of problem, digital marketplaces and retailers usually have a team of Customer Experience (CX) agents that interact, talk and answer clients' questions and inquiries, avoiding any potential negative situations on three different channels: voice calls, e-mails, and chat messages. However, to correctly assess customers' experience on these sites, it is also important to evaluate CX agents' performance, checking if they are correctly solving the clients' problems, if they are communicating correctly, etc. In a particular Latin-American e-commerce site, this evaluation has multiple points to be covered and is done manually, by another team (that we will call "auditors"). Considering that the agents' team is bigger than the auditors' team (which is logical considering also that auditors have a greater level of experience), the share of CX interactions that end up being evaluated is rather small: around 1%-2%. Natural language processing (NLP), machine learning (ML) and analytics, in general, can help us in this problem, as multiple classification models can be built to automatically evaluate a particular interaction. Our initial approach to this problem was to design a system that would download chat and e-mail conversations between agents and customers and evaluate automatically, with the help of ML models, the agent's performance. In terms of scope, we just checked the communication style, which represented approximately 16% of the time for assessments. However, it is important to mention that the communication style evaluation had several dimensions:

- Orthography: checking accents and if words are spelled correctly
- Punctuation: checking if punctuation marks are used correctly
- Typographical errors: checking if there are no typos in the text
- Formatting: checking if bold, italic and other style parameters are used correctly
- Redundancy: checking the agent isn't repeating terminology in the text
- Concordance: checking if the gender of pronouns, adjectives and nouns match
- Non-professional language: checking the agent isn't using colloquial language

- Technical terms: checking the agent using terminology referred to the e-commerce internal systems
- Negativity: checking the agent isn't focusing on the problem, as he/she should focus on the solution and its benefits
- Personal pronouns: checking the agent is using the correct pronouns when talking to the customer
- Reception: checking the agent has correctly greeted the client
- Finalization: checking the agent finished the conversation with the proper words

From the list aforementioned we can obtain our first and most important insight: *no single model can tackle all twelve dimensions simultaneously.* Thus, an experimental approach is necessary to achieve the best possible results. Therefore, we designed a three-step system to tackle each evaluation:

1. **ETL:** Download of historic interactions, connecting to multiple database systems and APIs to obtain previous conversations and evaluations. Here, what we obtain is, by interaction, the complete text of the e-mails and chat messages written by the agents and sent to the customers. In this sense, the output of the ETL stage is a table of n rows, one per interaction, and m columns: one indicating the case number, multiple columns that help us characterize the interaction (language, country, channel, etc.), and one column that has the concatenation of all the messages sent by the agent, which is the main column that will be used as an input for the machine learning models. The only processing that we did for this column was add special characters to help signaling the start/end of a new message.

2. **Hard Rules Evaluation:** For a subset of dimensions (orthography, punctuation, concordance, and typographical errors), using the library language-tool-python we automatize a subset of the grammar checks. The package is the foundation of what Open Office uses for spellchecking. However, it is important to mention that it usually overestimates mistakes as, for example, names are not always dealt properly. Therefore, we defined a set of rules that had to be taken into consideration to properly label a mistake. For instance, when considering orthography or typographical errors, the original word and its correction couldn't start with a capital letter and the difference in length of both words should not be greater than one.

3. **Vectorization and classification:** For the remaining analysis (and the cases that werent classified as mistakes for the previous dimensions), we built a model that classifies each text as correct or incorrect. To achieve this, we leverage two approaches: A Word2Vec based vectorization connected with a classification layer based on SVM or LightGBM algorithm and, on the other hand; a transformed-based approach using a custom modeled Multilingual BERT architecture with a binary classification layer as output. In both scenarios, several experiments were performed to tune the hyperparameters. Logically, the hyperparameters that were optimized changed per algorithm: for SVM we optimized the kernel and the gamma values doing random search, while for LightGBM we optimized the number of estimators, the depth of each tree, the learning rate and regularization parameters for alpha and lambda. Lastly, with the BERT transformer, a masked language modeling was performed to ensure the model understand the variability of the agents' locales. In our final models, Word2Vec was implemented to solve 2 dimensions, while the remaining 10 used BERT. After the final models were trained, we tuned the decision threshold of each model, to satisfy specific recall requirements, evaluating each threshold on a validation set.

4. Additionally, each day these models are used to evaluate the previous day's interactions, so an additional process that downloads new cases, vectorizes them and classifies them was also designed. Considering the daily execution of this process, we had to build it in a way that would satisfy time-execution requirements.

Graphically, the training process can be illustrated by the following Figure:

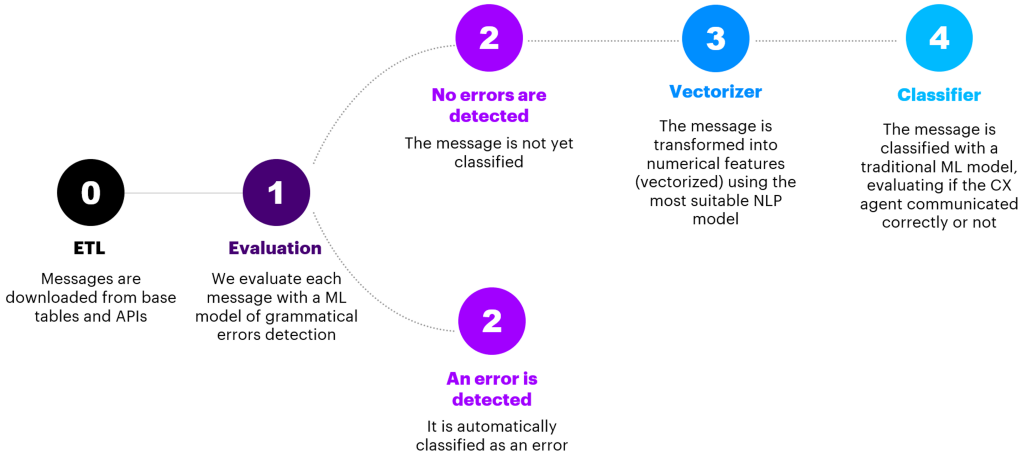

Figure 1: Training Process Diagram

When the experimentation process was performed, given the volume of data to process and the complexity of the tasks for the production environment we expected around 2-3 days of execution for an end-to-end (E2E) analysis. In this sense, using a Directed Acyclic Graph (DAG) for parallel execution, we manage to atomize the steps involved in the processes and reduce the E2E execution time to 16 hours, representing a 70% reduction in overall execution time.

The following picture illustrates the old and new DAG:

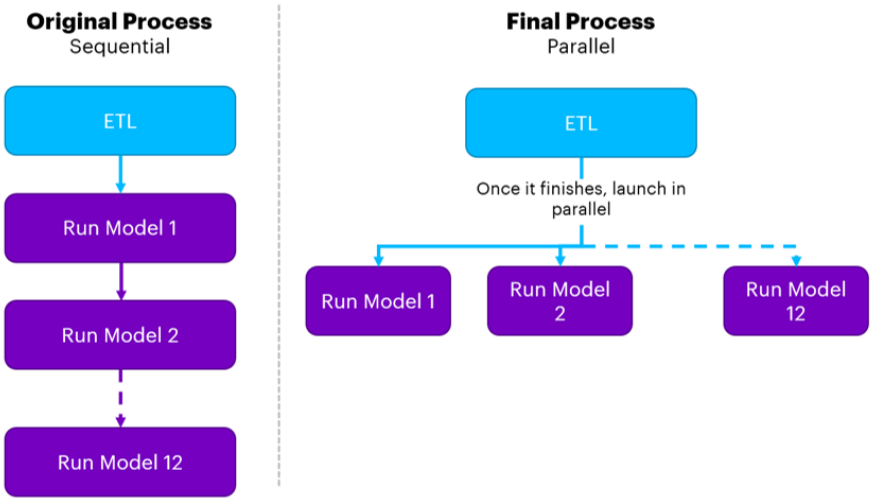

Figure 2: Directed Acyclic Graph for the original and the final process

After a four-week pilot, we proved that our system had a dual benefit. On one hand, we achieved an 86% level of automation for the communication evaluation, that reduced overall time by 13.7%, which results in an increased overall capacity of the assessment process. As evaluations come from a sample of the total number of interactions, this lets us automatically increase the sample number due to the reduction of the time required to complete an assessment. Therefore, the global number of cases rises 12%, while the number of cases whose communication is assessed increases 6522% as a result of the automation. On the other hand, we also increased overall accuracy in dimensions where the hard rules evaluation is possible. Considering the additional mistakes that are being now captured, we increased overall accuracy by approximately 20%, obtaining a 71% total accuracy and 77% level of recall, which was 11% higher than our requirement. These key performance indicators show the value of our solution and how we can improve the quality of evaluations with a solution that intelligently leverages on artificial intelligence and analytics.