April 2021

SIMPLIFIED AI POWERED DIALING PLAN HANDLER IN UNIFIED COMMUNICATIONS MANAGER

Anupam Mukherjee

SIMPLIFIED AI POWERED DIALING PLAN HANDLER IN UNIFIED COMMUNICATIONS MANAGER

AUTHORS:
Anupam Mukherjee

ABSTRACT

In a unified communications (UC) environment, the Call Manager (UCM) application contains a call control layer in which a digit analysis module supports the transformation of called and calling numbers and the determination of a routing decision. Such activities are typically complex and computationally intensive. To address such challenges techniques are presented herein that leverage artificial intelligence (AI) models (such as Artificial Neural Network (ANN) models) that are, among other things, customized and trained (based on, for example, customer requirements) in a cloud to, among other things, augment the number transformation and routing decision processes. Under aspects of the techniques presented herein a dialing plan may be offloaded from a UCM configuration workflow and moved to a collaboration cloud for centralized dialing plan management. Such an approach will reduce the complexity of the call control layer which will, in turn, make the UCM faster and computationally cheaper.

DETAILED DESCRIPTION

In a UC environment the UCM application has a call control layer where, among other things, a digit analysis module analyzes a called party number and a calling party number (both inward and outward) in a digit-by-digit fashion according to, for example, a translation pattern, a transformation pattern, a route pattern, etc. to either route a call to a specific destination – externally (to, for example, a T1 or E1 trunk, a Session Initiation Protocol (SIP) trunk, a Media Gateway Control Protocol (MGCP) gateway to the public switched telephone network (PSTN), a border element to a SIP service provider, etc.) or internally (to, for example, the same site or branch, etc.) – or invoke a feature in the Call Manager. During such a process, the module will apply the configured dialing plan to the numbers and transform the numbers to be able to execute the rules to mask the number or route properly. The Call Manager transforms the called party number and the calling party number as specified by a dialing plan and either selects the destination to route the call or invokes a feature. A digit analysis module in the call control layer plays a significant role in transforming the digits and aggregator layer components play a major role in determining the destination depending upon the output digits from the digit analysis.

The overall process that was described above is quite complicated and costly from a computational perspective due to, for example, its iterative nature. Additionally, the complete digit analysis module is typically a legacy module which requires a substantial degree of expertise to properly maintain or debug. Further, the overall dialing plan configuration typically is very complex and if the customer has multiple clusters, he must then configure the same information in every cluster, which is quite time consuming and requires dedicated effort for a flawless setup.

To address the types of challenges that were described above, techniques are presented herein that leverage AI models (that are, among other things, customized and trained based on, for example, customer requirements) to, among other things, augment the number transformation and routing decision processes. Under aspects of the techniques presented herein, a dialing plan may be offloaded from the UCM configuration workflow and moved to a collaboration cloud for centralized dialing plan management and machine learning (ML) purposes. As a result, there is no dialing plan configured in the actual system and instead of a dialing plan the AI models will handle the call routing decision (classification of destination) process along with the transformation of called and calling party numbers.

In the UCM, the process of digit analysis revolves around two key principles:

1. The transformation of a called number (e.g., the dialed digits) and a calling number to match routing rules for both outgoing and incoming calls (including call-backs).
2. A decision to either route to a specific destination – externally (to, for example, a T1 or E1 trunk, a SIP trunk, a MGCP gateway to the PSTN, a border element to a SIP service provider, etc.) or internally (to, for example, the same site or branch, etc.) or invoke a feature.

As depicted in Figure 1, below, both of the key principles of digit analysis, as described above, may be accomplished with the help of AI which may be utilized to provide intelligent transformation along with routing destination classification for a calling use case.

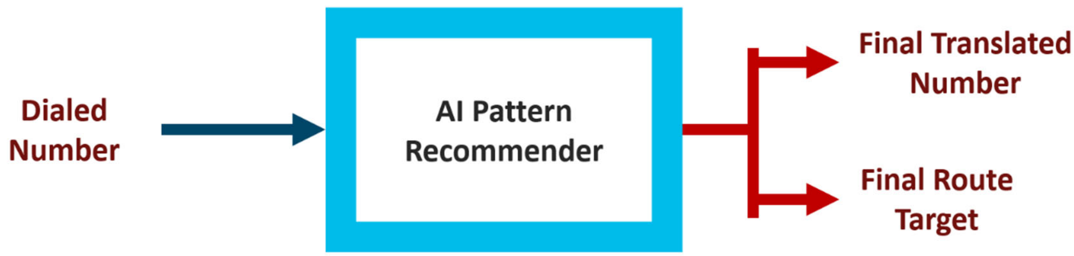

Figure 1: AI Augmented Digit Analysis

For purposes of explication, in the narrative that follows aspects of the techniques presented herein will be discussed in conjunction with elements of the exemplary architecture that is depicted in Figure 2, below.

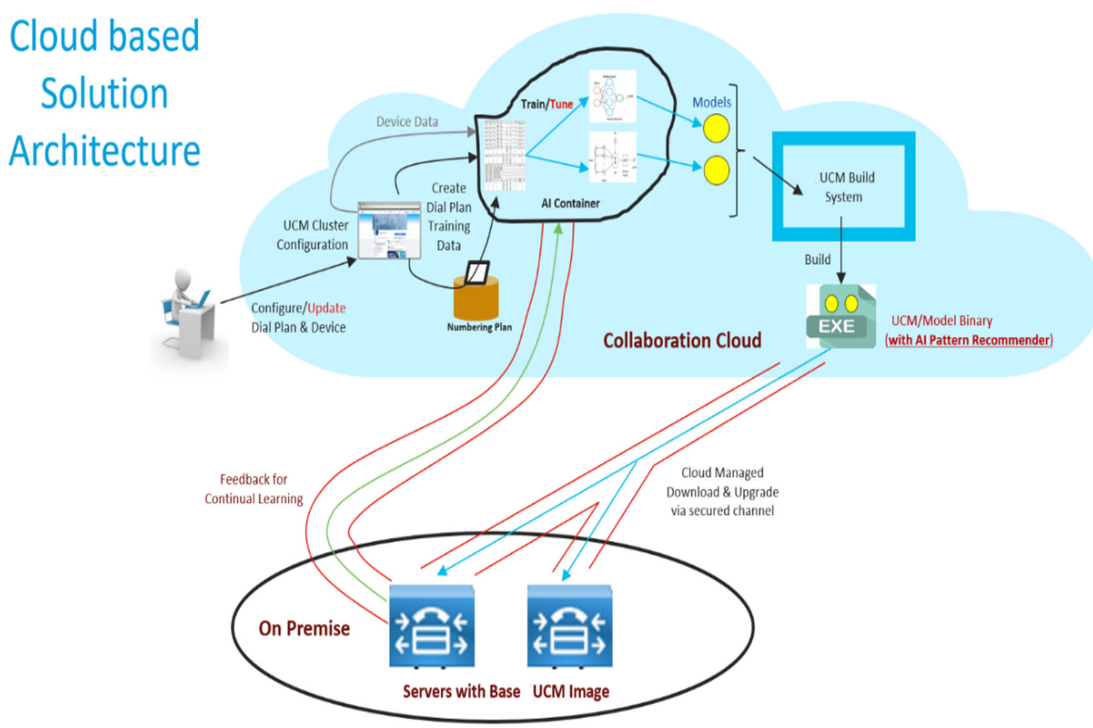

Figure 2: Exemplary Solution Architecture

Under aspects of the techniques presented herein, and as illustrated in Figure 2, above, a collaboration cloud may host a UCM cluster configuration user interface (UI) through which a customer may configure, among other things, the devices and the dialing plan(s) for the cluster(s).

A proposed cluster configuration utility on the cloud may generate a simplified dialing plan based on, for example, the configuration information. The dialing plan for a customer may be utilized to generate different training datasets (which should include cluster information).

A first training dataset will have as input sample enumerated (from the dialing plan) called number/digit-set details and calling number details and as output a final transformed called number and calling number. A second training dataset will contain as input the final called number and as output the destination decision (e.g., a trunk, a gateway, an internal directory number, etc.).

Two ANN models may be built and trained in the cloud. The ANN models will be pre-trained with different country specific dialing plans as default. An ANN is simply a deterministic algorithm that statistically approximates functions. Once the training of an ANN is completed, the internal workings of a neural network are always deterministic and are not stochastic, making them well suited for the instant tasks.

For a specific customer, the dialing plan-based training datasets will be utilized to train the two ANN models (in accordance with Transfer Learning mechanism with very low learning rate) offline in the cloud. The models will be trained based on the routed call patterns and destinations for both incoming and outgoing calls. Thus, based on the dialing plan that is specified by the customer, the final trained model-duo will be totally customized according to the customer's requirements. Thus, the entire dialing plan-specific data will be learnt by the ANN models and there will no longer be a need to keep the dialing plan data in the Call Manager database.

Based on the dialing plan that is specified by a customer on the collaboration cloud, the AI models can be totally customized and trained on the cloud according to the requirements of that specific customer. Additionally, as noted previously dialing plan data may be offloaded from the UCM configuration workflow and moved to a collaboration cloud for centralized dialing plan management and ML exploitation. Thus, a customer will configure the dialing plan in the cloud and then receive in return tailor-made AI models which will have the knowledge of the dialing plan that was defined by that specific customer.

Regarding the dialing rules that are specified by national numbering plans and private numbering plans, the base AI models may be pre-trained with different country specific dialing plans as a default. As well, there may be country-specific base AI models. As per a customer's location, appropriate base AI models may be chosen and trained (e.g., through transfer learning process) on the private numbering plan through the specific training data of that customer.

In brief, under aspects of the techniques presented herein AI is leveraged to accomplish and simplify the two key digit analysis principles that were described previously:

1. The translation and transformation of a called number (e.g., the dialed digits) and a calling number to match routing rules for both outgoing and incoming calls (including call-backs).
2. A decision to either route to a specific destination – externally (to, for example, a T1 or E1 trunk, a SIP trunk, a MGCP gateway to the PSTN, a border element to a SIP service provider, etc.) or internally (to, for example, the same site or branch, etc.) or invoke a feature

As mentioned above, since the AI models are ANN models, they will be completely deterministic and hence will always yield the same result no matter how many times they are used for translations or transformations and routing decisions.

As will be described and illustrated below, the two ANN models will work as an AI Pattern Recommender (cascaded mode) in the UCM stack. The cloud will also host a UCM build system which will include both of the ANN models as part of the UCM binary. The UCM binary may then be pushed to customer premise-based UCM nodes.

As depicted in Figure 3, below, an AI Pattern Recommender (as described above) will reside in the call control layer of the Call Manager and process a called number and a calling number to convert them into the required format and then select the appropriate routing destination without consulting the dialing plan or any other configuration information in the Call Manager.

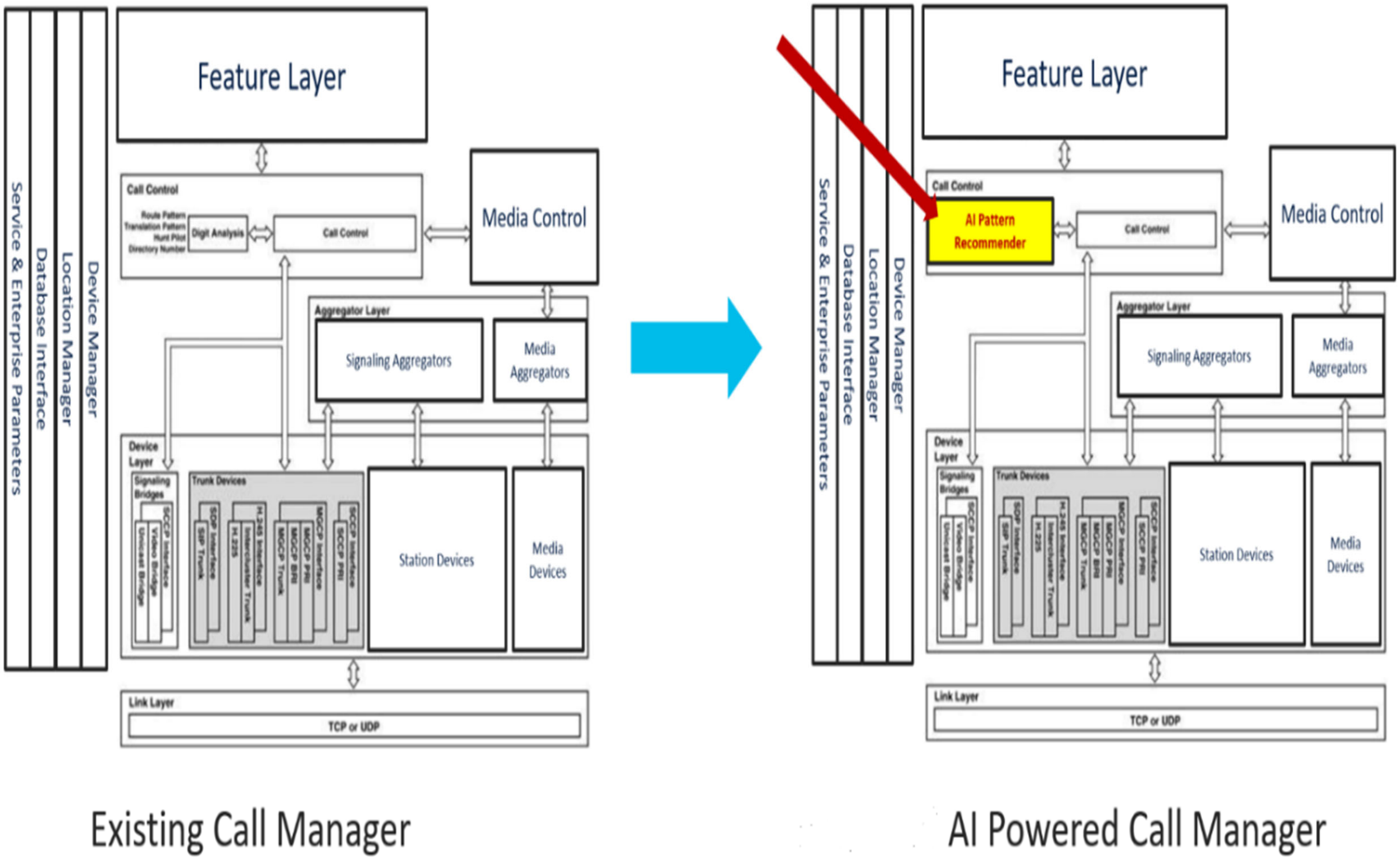

Figure 3: Exemplary Call Manager

The approach that was described and illustrated above will reduce the complexity of the call control layer which will, in turn, make the Call Manager faster and computationally cheaper. The AI powered digit analysis module will be very easy to maintain and debug because like any AI model it will be completely data driven (through training data). Moreover, every cluster-specific UCM may now use the same AI Pattern Recommender (unlike the existing cluster-specific dialing plan handling in the Call Manager) since cluster information is included in both of the training datasets.

Once the models are part of the call control stack, any unforeseen call handling scenario will create a record for the ANN models which may be transmitted to the cloud as feedback for continual learning where, after a prespecified time, the models may be retrained (with a very low learning rate) and then updated.

Aspects of the techniques presented herein support the generation of a number list from the configured dialing plan of a customer first and then using such a list as the customer-specific number corpus or vocabulary, similar to how a text corpus may be employed for natural language processing (NLP). Then a recurrent neural network (RNN) model (such as, for example, long short-term memory (LSTM), gated recurrent unit (GRU), Bidirectional Long Short-Term Memory (BLSTM), etc.) may learn the complete number vocabulary. Some number embedding (e.g., a number vector) may be created for better training purposes. The RNN model does not memorize the data, rather it learns the interdependency of the numbers and the respective output. Here the paradigm follows a language translation approach (such as, for example, translation from English to German).

The training data must be intelligently generated using some efficient utility. The training data generator utility, working offline in the cloud, will ingest the configured dialing plan of a customer and follow various rules (such as, for example, "create a source number and apply all the translation rules one after another to generate the final number," "don't consider the wild character portions of the numbers and instead take the static part of the rule and compute the respective final translated or transformed portions of the digits," etc.) to generate the training data.

In the existing dialing plan implementation, the same final translated numbers may be obtained after multiple rounds of translations whereas here the first model will directly, and importantly offline, learn the same and then during runtime recommend the final translated number (which may be a partial number pattern) from the source number. The same recommended number will then be sent to the second model for the routing decision.

The training data generator will also determine the routing decision from the configured dialing plan and create a supervised model specific training data set for the routing decision. The second model will be of supervised classifier type (ANN based) which will learn the routing decision specific training data offline and then during runtime obtain the number from the first model and then directly suggest the routing destination.

The actual per-customer training data will likely be vast. The AI models will learn in the cloud offline and then apply the developed knowledge during runtime in the Call Manager. Note that this approach may be applied to any call control solution (e.g., cloud-based, enterprise, etc.).

It is important to note as an alternative to the approach that was described above both of the AI models may be clubbed into a single model of multilabel type.

Consider as one example a model where the number 89025555 routes to San Jose, the number 89025666 routes to San Jose, and then an administrator adds the number 89025777 which also routes to San Jose. If the training data that is generated encompasses:

```
<------------features------------><--output-->
prefix postfix_number_length       routeclass
8902   4                           SJ
```

then any number that is characterized by the same will be classified as San Jose (SJ). Consider also what will happen if a number starts with "890", has 5 digits after it, and has corresponding `routeclass` of "Dallas". This case will never arise as the training generator will never generate any overlapping data. Overlapping data is strictly prohibited in an AI world; data must always be distinct.

Under another option all of the numbers corresponding to the patterns and the respective final translated numbers may be generated as a number corpus (as described above) so that instead of generating the numbers intelligently, the model is left to learn by itself the pattern and the interrelationships between the digits (input and output). The instant approach mirrors the standard NLP paradigm which too works on the interrelationships of words in different sentences in multiple text corpus. Accordingly, any absent number like "89025777" may be treated like an "OUT_OF _VOCAB" (OOV) situation. For OOV-like use cases, multiple options may exist. One option is quite obvious

– that is, retrain the model. Alternatively, another OOV-specific model may exist that may work in such situations (i.e., on failure of the primary model) and decide the most appropriate location by detecting its prefix and respective `routeclass` as explained above (for the first option on training data).

Consider also the example where dialing 8888 in a store, in a large universe of chain stores, reaches a security desk within the same store. Aspects of the techniques presented herein promote customized AI models, along with training data, per customer. The base model will be trained on the respective national numbering plan. It will be extended through transfer learning for every customer to learn the customer-specific private numbering plan. The dialing plan will be configured on the cloud by the customer (e.g., through a new dialing plan wizard) which will capture the private numbering plan for the customer and then at the backend the intelligent training data generator will generate the actual training data from the customer-specific private numbering plan. The same data will be learnt by the AI models for that customer so that it can now tell the final translated number and respective routing decision if 8888 is dialed.

Now suppose there are two customers – Customer-A and Customer-B. In such a case there will be two set of models – AI_Models_A and AI_Models_B. Customer-A may want to reach the security desk if 8888 is dialed whereas Customer-B may want to reach another employee in the office. AI_Models_A and AI_Models_B will have this knowledge and they will behave accordingly per each respective customer's specification. Thus, there is no need to partition any training data as everything may be controlled through data-related features.

Unlike other approaches, the proposed model set must be very accurate and because it needs to be somewhat overfitted (optionally as opposed to regularized) its training accuracy will also be very high (e.g., 100%). Additionally, the models need to be tested thoroughly with over- and under-sampled data based on different features of the data (e.g., AI specific dimensions). Ideally the proposed model set should not have any scope of error on known sets of data (e.g., training data may contain either a complete enumeration or prefix/postfix/length type of features). In the case of an OOV type of error, it will not transform or route the call and there it can either use an OOV specific flow or an existing manual flow which will eliminate the need for administrative intervention.

As discussed above, training data will never be incomplete. Of importance is what training data is generated from a customer-specific dialing plan. The data must be complete (in any form) for that customer. Additionally, even if a number is transformed multiple times of importance to the training data is just the source number and the corresponding final number irrespective of any intermediate translation or transformation details. Those two numbers (complete or partial) are part of the training corpus that is generated by the training data generator. If it is a Called Party Number (CdPN), based on the final number (as recommended by the first AI model) the respective routing decision will be made by the second AI model. If it is a Calling Party Number (CgPN), then the masking is happening by a transformation pattern and that is what will be recommended by the first AI model.

It is important to note that under aspects of the techniques presented herein there is no change in UCM routing logic. The above-described AI models will obtain the source number in order to suggest the final number and then predict (with complete confidence) the routing destination (as a class).

In the case of a change in requirements (e.g., as described above) a customer may go to the cloud and specify the new requirements. Next the training generator will generate the required set of new data and train the AI models to learn it. After the learning process is completed the AI models may be pushed to the customer's UCM and then the new dialing plan will be activated. Meanwhile, if the new requirements are completely new then under an OOV scenario (as described above) the alternative flow may be activated until the new models become available.

Form a customer perspective, the dialing plan may be simplified (preferred, with the particulars being part of an implementation) or may be the same (but on the cloud). The training data generator will know how to interpret the dialing plan and generate the training data for the training of the AI models.

As noted previously above, ANN based models are completely deterministic and not probabilistic following training. Due to a slight overfitting (optional) and a corpus-based NLP styled modelling, the chance of error is eliminated unless the data is OOV.

In order to perform an accurate job of call control-based recommendation, AI Pattern Recommender based models should:

1. Be complex and slightly overfitting (optional). Alternatively, Content-Addressable Memory or an Associate Memory Network may be employed.
2. Have very low bias. Variance may be somewhat higher and the same must be contained presently by the feedback based continual learning and update mechanism.

Generative adversarial networks (GAN) or Softmax-based ANN models may optionally be used for final routed call patterns. Additionally, Softmax-based ANN models may be used for routing decisions.

Based on the dialing plan that is specified by a customer on a collaboration cloud, the AI model-duo will be totally customized and trained on the cloud per the Customer's requirements.

As noted previously, under aspects of the techniques presented herein a dialing plan may be offloaded from a UCM configuration workflow and moved to a collaboration cloud for centralized dialing plan management and machine learning (ML) exploitation. Among other things, through aspects of the techniques presented herein it may become possible to minimize the size of a UCM so as to fit in, for example, an embedded environment. Further through aspects of the techniques presented herein debugging will become easier as the rules that are directing the flow of a call are now part of the training data. Thus, once a source number becomes known it is easy to find the destination number and the corresponding routing decision.

In summary, techniques have been presented that leverage AI models (such as ANN models) that are, among other things, customized and trained (based on, for example, customer requirements) in a cloud to, among other things, augment the number transformation and routing decision processes. Under aspects of the techniques presented herein a dialing plan may be offloaded from a UCM configuration workflow and moved to a collaboration cloud for centralized dialing plan management. Such an approach may reduce the complexity of the call control layer which will, in turn, make the UCM faster and computationally cheaper.