Combining text categorization and dialog modeling for speaker role identification on call center conversations

Rémi Lavalley\textsuperscript{1,2}, Chloé Clavel\textsuperscript{2}, Patrice Bellot\textsuperscript{1}, Marc El-Bèze\textsuperscript{1}

\textsuperscript{1}Laboratoire Informatique d’Avignon, University of Avignon, France
\textsuperscript{2}Electricité de France R&D, France

\{remi.lavalley, patrice.bellot, marc.elbeze\}@univ-avignon.fr, chloe.clavel@edf.fr

Abstract

In this paper, we address the problem of speaker role identification on a corpus of manually transcribed call center conversations. We first tackle it as a text categorization task. Then, we combine these categorization results with a dialog modeling approach. We achieve 93% of correct role assignment with the least method. Our method also offers the possibility to extract text spans specific to each role. These strings slightly improve the role identification results and are an interesting element for conversation analysis.

Index Terms: speaker role identification, text categorization, dialog modeling, category-specific strings

1. Introduction

Analysis of the speakers involved in broadcast news or call center conversations is interesting for many applications: information retrieval, summarization, interaction analysis, dialog analysis. Several tasks are concerned with this issue: speaker segmentation, speaker identification, speaker verification, speaker diarization, speaker role identification. These tasks can be performed by using acoustic or linguistic clues.

To our knowledge, identification of speaker role has rarely been addressed in the literature. Both \cite{1} and \cite{2} have proposed methods for identifying speaker role in broadcast speech. Considering broadcasts or call center corpora leads to different approaches. In broadcast news, speech turns are generally longer than in call center conversations. In \cite{1} the authors process specifically the boundary sentences of speech turns as they provide more information. On call center corpora, few speakers are involved in one conversation, but there are more interactions and speech turns are shorter (they rarely contain several sentences).

Speaker role identification is different from speaker identification or diarization, where one has to assign the precise speaker (e.g., his name). These last tasks are mainly based on acoustic features. Even if each role can be played by several people, voice may nevertheless have some interest (prosody can be different whether the speaker is professional or not) for role identification. We focused here on linguistic clues which we think are more crucial for role identification. This is mainly due to the fact that call center operators are trained to use some specific vocabulary and brief on how to interact with the clients. Hence, text clues (terms, turns of phrases, ways of expression) are important. A previous work done on that corpus \cite{3} led to the conclusion that Clients’ vocabulary is different from Operators’ one.

In this paper, we address the problem of role (Client or Operator) identification in a call center conversation analysis context. This work is motivated by the fact that EDF (“Electricité de France” - French power supply company) call center conversations are recorded on a 1-channel device (there is no differentiated channels for clients and operators). Knowing these informations about conversations’ structures is interesting in many ways. First, it will allow the company to analyse Client-Operator interactions and define conversation profiles, which are useful for marketing analysis and for improving call center efficiency. Then, we think that building language models specific to each role and topic will be more efficient than having only topic models for a further topic categorization task planned on this corpus. Furthermore, the method we present offers the possibility to automatically extract role-specific strings (i.e. text spans often employed by the clients or the operators).

We first choose to tackle the problem of role identification as a text categorization task (each speech turn has to be assigned to a role-category). Then a second step, based on the outputs of the categorization task, adjusts the results by taking into account the observed role sequences. In this paper we only consider the specific problem of speaker role identification. Then we use a corpus of manually transcribed call center conversations. Additionally, speech turns have been manually segmented and annotated. This corpus is described in Section 2. In Section 3, we present the methods we have tested for categorization and dialog modeling and the results obtained in this way.

2. Corpus

The corpus used is part of the EDF CallSurf \cite{4} corpus. The CallSurf corpus was built by recording Customer-Operator conversations in an EDF Pro call center during the summer of 2006. Conversations involving dozen agents were recorded for 2 months, compiled into a corpus of almost 5800 calls (620 hours). The experiments presented below have been conducted on 910 calls of the CallSurf corpus. These 910 conversations have been manually transcribed and anonymized (names, phone numbers, Client references were replaced with generic labels). Speech turns and speaker roles have been manually defined. We can observe some overlap between speakers, because of the fact that the Client and the Agent have been recorded on the same audio channel. The mean length of a conversation is approximately 6 minutes.

A call consists of a conversation between a Client, generally calling EDF Pro to outline a problem, or to ask a question, and an Operator. Sometimes, the Operator can call another one for obtaining more information. Less often, some conversations start with an Operator calling a Client. Several roles are annotated per conversation: Operator, Client, Client-Operator\textsuperscript{1},

\textsuperscript{1}corresponding to overlapping speech - generally these speech turns

PREPRESS PROOF FILE
Our first approach consists in considering the speaker role identification as a text categorization task. We first build one model for each role (Operator, Client and Client-Operator). Then, each speech turn has to be classified in one of these three categories without taking into account the previous or following speech turns.

The categorization is made with a cosine normalization of $\text{TF} \times \text{IDF}$ (“Term Frequency - Inverse Document Frequency”) [5], adjusted with the Gini criterion, as it has been proposed in [6]. This adjustment variable reduces the influence of the terms with a bad discriminative power (words occurring in several categories without major discrepancies in their frequencies).

The similarity measure between a speech turn $s$ and a category (a role) $r$ is computed as presented in Formula 1. $\cos(s, r)$ contains no text, because it was inaudible and could not be transcribed is the cosine normalization of the document vector $W_s$ and the role vector $W_r$, where the features are the words $i$:

$$\cos(s, r) = \frac{\sum_i (W_s(i) \times W_r(i))}{\sqrt{\sum_i W_s(i)^2 \times \sum_i W_r(i)^2}}$$

where: $W_s = \text{TF}_s(i) \times \text{IDF}_s(i) \times p\text{Gini}(i)$

$$W_r = \text{TF}_r(i) \times \text{IDF}_r(i) \times p\text{Gini}(i)$$

$p\text{Gini}(i)$ is the Gini purity criterion, i.e. the discriminative power of the term $i$, with respect to its distribution into the categories $k$:

$$p\text{Gini}(i) = \sum_k P(k| i)^2$$

$$\text{IDF}(i) = -\log \left( \frac{\text{number of speech turns containing } i}{\text{total number of speech turns}} \right)$$

This method yields to 66% of speech turns correctly labeled.

3.3. Dialog modeling

Our second approach consists in adjusting the categorization results by taking into account the observed role sequences. We then add a second step which combines the categorization outputs (i.e., the Cosine scores between each speech turn and the different roles - these scores have been normalized to sum to 1) and the probability of having each role knowing the role assigned to the previous speech turns. We tested two methods: by considering a local or a global context.

3.3.1. Local context

We first use the training corpus to estimate the n-turns probabilities of role sequences (for instance, the probability that a speech turn is assigned to Operator, knowing that the two previous speech turns were assigned to Operator - it happens in a case of a conversation between two Operators). To finally assign a role to a speech turn, we combine this information with the categorization scores, as presented in Formula 5.

$$\text{Sim}(s, r) = \lambda_s \times \cos(s, r) + \lambda_d \times (\lambda_a \times U_s + \lambda_b \times B_s + \lambda_t \times T_s)$$

where:

$$U_s = P(r(s)) \times B_r = P(r(s)) \times P(s - 1)$$

$$T_s = P(r(s)|r(s - 1), r(s - 2)), \lambda_a = 1, \lambda_b + \lambda_c + \lambda_d = 1$$

$P(r(s)|r(s - 1))$ is the probability of having $r$ to the speech turn $s$, knowing the role assigned to the previous speech turn.

We compare the results obtained with a 2-turns (biturns) dialog modeling and a 3-turns (triturns) dialog modeling. Biturns corresponds to the case where $\lambda_t = 0$.

We empirically set: $\lambda_a = \lambda_b = 0.5$. Thus, both the categorization and the dialog modeling have the same importance. In case of biturns, the role assigned to a speech turn is chosen regarding: the categorization results, the probability of having that role (i.e. the relative frequency of this role) and the probability of having this role knowing the role we assigned to the previous speech turn: $\lambda_a = 0.1; \lambda_b = 0.9; \lambda_t = 0$. Biturn

<table>
<thead>
<tr>
<th>number of characters</th>
<th>5127380</th>
</tr>
</thead>
<tbody>
<tr>
<td>total number of words</td>
<td>995893</td>
</tr>
<tr>
<td>number of unique words</td>
<td>12716</td>
</tr>
<tr>
<td>mean number of words per conversation</td>
<td>1094.4</td>
</tr>
<tr>
<td>mean number of words per speech turn</td>
<td>9.8</td>
</tr>
<tr>
<td>number of speech turns per role</td>
<td></td>
</tr>
<tr>
<td>- Operator</td>
<td>55848</td>
</tr>
<tr>
<td>- Client</td>
<td>45386</td>
</tr>
<tr>
<td>- Client-Operator</td>
<td>513</td>
</tr>
</tbody>
</table>
probability is much more important than uniturn one (corresponding to the observed distribution of the speech turns into the different roles). In the triturn case, we also consider the probability of having a role knowing the ones we assigned to the two previous speech turns. The optimal parameters we found are: \( \lambda_2 = 0.1; \lambda_0 = 0.45; \lambda_1 = 0.45 \).

The biturn method achieves a 76.6\% good role assignment, the triturns achieves 78.7\%. We do not think that using a larger context (4-turns) would yield to significative improvements. Triturns help the system to detect parts of discussion between two Operators (the two previous speech turns have been assigned to Operator) or one Operator and one Client (Client Operator alternation). In fact, the main errors produced by this method come from the snowball effect: when the categorization step assigns a wrong role with a very high score, it confuses the dialog model. If this mistake is made on the first speech turn of the conversation, it can lead to misclassify all the speech turns of the conversation (in the case where none of the following speech turns will be linked to his actual category with a strong probability by the categorization step to set things right).

3.3.2. Global context

The second method we test for dialog modeling consists in using the Viterbi algorithm to combine categorization results and role sequences probabilities. The aim of the Viterbi algorithm is to find the most likely states sequence (roles sequence) considering the whole conversation (hence avoiding the snowball effect). Here a state is referring to a role, knowing the previous one. Hence, for each speech turn, we compute the probability of having each role, for every possible origin (see Formula 6).

\[
\text{State}(r(s), r(s-1)) = \arg \max_{r(s-2)} \left( \lambda_s \times \log \left( \frac{P(r(s)|r(s-1), r(s-2))}{\epsilon} \right) + \lambda_c \times \log \left( \text{Cos}(s, r) \right) + \text{State}(r(s-1), r(s-2)) \right)
\]

where: \( r(s) \) is the role of the current speech turn
\( s - 1 \) is the previous speech turn
\( P(r(s)|r(s-1), r(s-2)) \) is the probability of having \( r \) under the hypothesis the roles \( r(s-1) \) and \( r(s-2) \) have been assigned to the previous speech turns (triturn frequency computed on the training corpus)

The lambda coefficients have been empirically set to \( \lambda_s = 0.2 \) and \( \lambda_c = 0.8 \). This configuration yields to 91.3\% of correct role assignment. This method outperforms the previous one (simple n-turn dialog modeling), because all the speech turns roles assignments do not depend on the first one.

3.4. Finding role-specific strings

We present here a method for extracting text spans specific to each role. Following the idea introduced by [7] (who proposed to extract domain-oriented pairs of words from texts), our method consists in considering only the training speech turns assigned to a specific role (Client or Operator), and in extracting collocations from these texts, by applying the log-likelihood [8]. By doing this, we obtain one list of collocations per role. Then, we merge these collocations (called ‘agglomerates’ in the following) in the whole corpus, regardless of the category: if two words \( w_1 \) and \( w_2 \) (taken from any of the lists we have) have a sufficient log-likelihood value (> 75) for being considered as an interesting collocation and occur more than 10 times in the training corpus, we consider them as a unique term by replacing each occurrence of the two isolated words “\( w_1, w_2 \)” by an unique pseudo-word “\( \text{agl}_{w_1-w_2} \)” in the training and in the testing corpus. Then, we iterate this procedure in order to obtain agglomerates of more than two words.

By doing so, we have one list of agglomerates per role; such a list can be presented to someone interested in viewing text spans specific to each role (for instance, customer relationship management employees can have an insight on main remarks made by Clients or points often tackled by Operators). Furthermore, our corpus contains agglomerates that are expected to be more discriminating than isolated words for the categorization step.

Here are some of the 1825 agglomerates extracted from the corpus. From the Operators speech turns, we find typical introductions: “sylvie-xx-edf-pro” (Sylvie X, EDF Pro), “xx-edf-pro-bonjour” (X, EDF Pro, good morning), “edf-pro-jérôme-xx-à-votre-service-bonjour” (EDF Pro, Jerome X at your service, good morning) some formal parting phrases “bonne-journée-merci-de-votre-appel-au-revoir” (Thank you for calling, have a nice day, good bye), “vous-remercier-patienter-an-petit-instant” (thank you for holding the line) and some technical expressions “pénalités-de-retard” (late payment penalty), “votre-historique-réel-de-consommation” (your actual consumption history), “vous-entreprise-autorisation-de-prélèvement” (send you a direct debit authority). From the Clients speech turns, we find several expressions used to explain the reason of the call: “vous-appelle-pas” (call you because), “me-permet-de-vous-appeler” (am [taking the liberty of] calling you), “c’est-pour-ca-que-je-vous-appelle” (that is why I am calling you), “le-problème-c’est-que” (the problem is that); some interesting spans for customer relationship management could be the ones related to unsolved problems, for instance “un-de-vos-problèmes” (one of your colleague), generally found in sentences like “The last time I called, one of your colleagues told me that...”; if we find lots of this kind of strings, we can infer that we have a problem to correctly answer to customers’ issues (or by studying their occurrences at different periods, we can analyse the evolution of the call center efficiency). In the same way, we can see what are the recurring customers’ questions at different periods.

We evaluated the influence of these 1825 agglomerates on the categorization results. The use of these agglomerates in the case of categorization only (as presented in Section 3.2) yields to a 68.1\% of correct assignments (a 2.1\% gain). The use of these agglomerates on the whole process (categorization and dialog modeling with Viterbi algorithm) yields to a 92\% of correct assignments (a 0.7\% gain compared with the same process without agglomerates, as presented in Section 3.3.2).

3.5. Text-length weighted Viterbi

From the previous experiments, we observed that:

- the categorization step is really efficient when the speech turns have a sufficient length (but often unsure when there are only one or two words);
- if long speech turns are well categorized with a high confidence, the Viterbi algorithm will easily deduce the roles that should be assigned to the short ones.

Noting this, we have transformed the Formula 6 to take into account the length of speech turns. The categorization result is now weighted by the sentence length, as presented in Formula 7.
\[ \text{State}(r(s), r(s-1)) = \arg\max_{r(s-2)} \left( \lambda_r \times \log \left( P(r(s)|r(s-1), r(s-2)) \right) + \mu \times \log \left( \cos(s, r) \right) + \text{State}(r(s-1), r(s-2)) \right) \]

where: \( \mu = \lambda_r + \frac{\text{length}(s)}{W} \)

\( \text{length}(s) \) is the number of characters of the speech turn \( s \) (without spaces)

After experiments, \( W \) has been empirically set to 12, and the lambdas to \( \lambda_r = 0.88 \) and \( \lambda_c = 0.1 \). This system achieves 93\% of correct role assignment.

### 3.6. Baselines

We compared our approaches to three systems:

- for the categorization step, we tested a Support Vector Machines approach with linear kernel: we used liblinear [9], optimal SVM parameters (\( C=1 \) and \( \epsilon = 0.0625 \)) have been found by grid search, with \( C \) varying from \( 2^{-5}, 2^{-4}, \ldots \) to \( 2^15 \) and \( \epsilon \) in \( \{2^{-15}, \ldots, 2^3\} \). It achieves 69.1\% of correct role assignment, i.e. 3.1\% better than the simple Cosine and 1\% better than Cosine using agglomerates, which confirms that the simple categorization score achieves a rate of about 70\%.

- for the dialog modeling step, we tested a trivial decision rule assigning alternatively ‘Operator’ and ‘Client’ to the speech turns (beginning with an ‘Operator’ since it is the case of most of the conversations). This rule achieves 71.1\% of correct assignment. We then can infer that a more sophisticated dialog modeling is necessary and the categorization step is useful.

- third, we tested a SVM, adjusted with the Viterbi algorithm. This system achieves 90.1\% of correct assignment. It is 1.2\% less than Cosine + Viterbi and 2.9\% less than Cosine categorization using agglomerates + length weighted Viterbi. The reason why SVM+Viterbi is worse than Cosine+Viterbi whereas SVM categorization is better than Cosine categorization relies on the fact that SVM make wrong decisions with high confidence scores that can not be readjusted by the dialog modeling.

### 3.7. Results summary

**Table 2:** Percentage of well role-assigned speech turns.

<table>
<thead>
<tr>
<th>Methods</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosine categorization (TFxIDFxpGini)</td>
<td>66</td>
</tr>
<tr>
<td>Cosine + agglomerates</td>
<td>68.1</td>
</tr>
<tr>
<td>SVM categorization</td>
<td>69.1</td>
</tr>
<tr>
<td>Cosine + bigram dialog modeling</td>
<td>76.6</td>
</tr>
<tr>
<td>Cosine + trigram dialog modeling</td>
<td>78.7</td>
</tr>
<tr>
<td>Cosine + Viterbi</td>
<td>91.3</td>
</tr>
<tr>
<td>SVM + Viterbi</td>
<td>90.1</td>
</tr>
<tr>
<td>Cosine + agglomerates + Viterbi</td>
<td>92</td>
</tr>
<tr>
<td><strong>Cosine + agglomerates + length weighted Viterbi</strong></td>
<td><strong>93</strong></td>
</tr>
<tr>
<td><strong>trivial assignment: Operator/Client alternation</strong></td>
<td><strong>71.1</strong></td>
</tr>
</tbody>
</table>

In Table 2, we summarize all the results (percentage of correct role assignments) of the experiments presented in the previous subsections. In all of them, about 0.5\% of error rate is due to the misclassification of “Client-Operator” speech turns, which are under-represented and generally contain no text. Thus, they have bad models and are difficult to retrieve.

### 4. Conclusions

We have presented several methods for speaker role identification. We first tackled the problem as a text categorization task, yielding to about 70\% of well identification. We then tried different methods to combine the categorization results with a dialog structure modeling (speech turns sequences). The use of a Viterbi algorithm thus allowed us to correctly identify more than 90\% of speech turns’ speaker role. Finally, we added a role-specific string extraction module. The strings thus extracted slightly improved speaker role identification, but they have a huge interest in conversations analysis. Our method, which correctly assigns 93\% of the speech turns of our call center corpus could be tested in the case of more roles (where dialog modeling should be less efficient) and with an automatically transcribed corpus (where categorization models are expected to be noisy).

We consider that this method achieves good results enough to reuse them in order to improve other tasks. We then plan to use this speaker role identification for a topic categorization task on the same corpus, by building role-specific topics models. We consider that using language models specific to each role and topic will be more efficient than having only topic models, in the way that clients and operators do not use the same expressions when referring to a subject.

### 5. References


