Entropy-Driven Dialog for Topic Classification: Detecting and Tackling Uncertainty

Manex Serras, Naiara Perez, María Inés Torres and Arantza Del Pozo

Abstract A frequent difficulty faced by developers of Dialogue Systems is the absence of a corpus of conversations to model the dialog statistically. Even when such a corpus is available, neither an agenda- nor a statistically-based dialog control logic are options if the domain knowledge is broad. This paper presents a module that automatically generates system-turn utterances to guide the user through the dialog. These system-turns are not established beforehand, and vary with each dialog. The module is valid for agenda-based and statistical approaches, being applicable in both types of corpora. Particularly, the task defined in this paper is the automation of a call-routing service. The proposed module is used when the user has not given enough information to route the call with high confidence. Doing so, and using the generated system-turns, the obtained information is improved through the dialog. The paper focuses on the development and operation of this module.

1 Introduction

Developing a Dialog System (DS) may be a harsh task, especially if there is a lack of dialog-based corpus and/or the domain to model within the dialog is too broad. The agenda-based approach is commonly adopted for dialog modeling [1] in the first case. However, devisal of plan-based conversation streams requires that developers have full knowledge of the domain [4], and/or dialog streams defined beforehand for every possible situation. For this reason, this approach has been mostly used for fixed domains and formulary tasks, such as booking airline tickets and consulting

Manex Serras, Naiara Perez and Arantza Del Pozo
HSLT department, Vicomtech-IK4 research centre, Donostia-San Sebastián, Spain,
e-mail: {mserras,nperez,adelpozo}@vicomtech.org

María Inés Torres
Speech Interactive Research Group, Universidad del País Vasco UPV/EHU, Spain,
e-mail: manes.torres@ehu.eus

bus schedules or fares [3, 7, 9]. But, even when it is possible to use a Markov Decision Process (MDP) - or Partially Observable MDP-based approach, developers may find problems when the application domain is too vast to cover.

This paper proposes a novel module to enhance the common Dialog Manager (DM) logic: the Suggestion Generator (SG) module. SG generates a dialog system turn to retrieve more information from the user to achieve the task set when the information given by the user is not enough. In order to measure the amount of information, the entropy score is used in a similar way as [12], but to quantify a Topic Classifier's (TC) decision quality. TC bases its decisions in language units extracted from the user turn by the Natural Language Understanding (NLU) module.

The specific project task of this paper is the automation of a call-routing service. The corpus available consists of written records and customer e-mails, which describe a vast range of problems –and not necessarily all the possible problems: new ones are arising all the time. The main contribution of the paper lies in embedding within a Dialog System (DS) a module that allows the user-machine interaction to be guided through questions not established beforehand, generating a dialog stream automatically without a corpus of dialogs nor having to design manually an agenda-based strategy for each possible situation. SG grants the system enough flexibility to adapt to and route unseen situations despite the domain being too broad.

The paper is structured as follows: Section 2 provides an overview of the DS in terms of architecture and interaction between modules; Section 3 explains how SG and its algorithm work; Section 4 describes the task corpus, and explains the automatic selection of language units for SG and TC; Section 5 evaluates SG and its impact on uncertainty detection; finally, in Section 6, the conclusions reached and future guidelines are presented.

2 Overview: the Dialog System

The system consists currently of 5 modules, as shown in Figure 1: a NLU module for semantic parsing and language unit extraction, an agenda-based DM, the Topic Classifier (TC), the Suggestion Generator (SG), and the Natural Language Generator (NLG). Automatic Speech Recognition (ASR) and Text-to-Specch (TTS) systems have not been implemented still. An internal server is responsible for the communication between the existing modules.

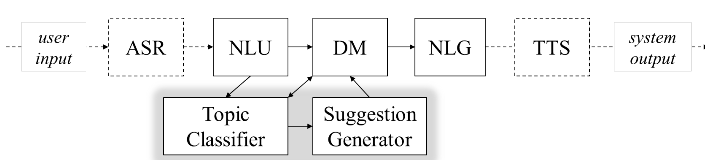

Fig. 1 Proposed system architecture

The dialog is plan-based in the general thread and most simple subtasks controlled by the DM. When the point of the conversation is reached when the user issue must be categorized, however, the DM delegates most of the decision making burden to the statistical components of the system: TC and SG. In very few words, this is how the two modules come into play:

1. The user turn is processed with the parser. Its output is passed to TC.
2. TC assigns the class of highest probability to the observed output. The set of classes is composed of the departments of the company providing the service.
3. The entropy of the taken decision is measured. If uncertainty remains low, TC's decision is accepted. Otherwise, the decision is revoked and SG suggest a language unit for the NLG to generate a system turn, in an attempt to retrieve more information from the user. TC is re-evaluated with the user's answer.

3 The Suggestion Generation Module

The Suggestion Generator (SG) module serves the purpose of retrieving information from the user when the DM considers that the classification confidence is not high enough.

The call-routing procedure is explained as follows: first, the utterance representation $Y = (u_{Y_1}, u_{Y_2}, \cdots, u_{Y_{|Y|}})$ is extracted from the user turn by the NLU module, where each u_{Y_i} corresponds to a language unit (a concept, word, lemma, ...). Y is passed to the Topic Classifier (TC). Then, having defined the set of departments as the class set Ω, where each class ω_j is one department, the call is routed to the assigned department if the classification confidence is high. Otherwise –if the confidence is low–, the DM uses SG to choose a language unit unseen in Y, $\hat{u}$, about which to ask the user in the next system turn. SG obtains $\hat{u}$ from a graph structure that contains the informational relationships of the corpus, the Information Graph (IG), exploring it with the Suggestion Generation Algorithm (SGA) designed. When $\hat{u}$ is chosen, the NLG module generates the question with framed prompts prepared beforehand. It simply picks one of the prompts, and merges $\hat{u}$ in the empty frame.

The classification confidence to determine whether SG has to be activated is obtained using the entropy measure when the TC assigns a class ω_j to the utterance representation Y:

$$H(Y) = -\sum_{j=1}^{|\Omega|} P(Y \mid \omega_j) log_{|\Omega|}(P(Y \mid \omega_j)) \tag{1}$$

where $P(Y \mid \omega_j)$ is the probability of Y given the class ω_j. In this stage, the uncertainty evaluation is done: a utterance representation Y is labeled as an Uncertain Case (UC) if and only if:

$$H(Y) \geq \theta_e \tag{2}$$

where θ_e is the entropy threshold. When Y is labeled as UC, the DM activates SG.

Figure 2 depicts an example where the user does not grant enough information and the system draws on SG to choose the language unit for the next system turn:

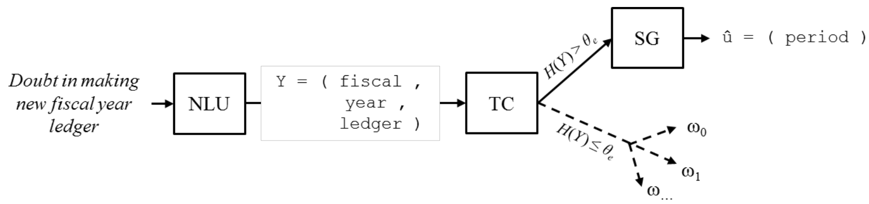

Fig. 2 Processing of a low information user utterance through the system

This section explains how IG is modeled and explored by the SGA.

3.1 The Information Graph

The Information Graph is the structure that captures the knowledge of the corpus. In this structure, those combinations of language units that grant information to achieve the proposed goal (here, call-routing) are represented as directed edges. The entropy measure of the language units is adopted as the graph connection criterion.

Being U the set of selected language units from the training set S, $H(u_i)$ is the entropy of the language unit $u_i \in U$:

$$H(u_i) = -\sum_{j=1}^{|\Omega|} \hat{P}(u_i|\omega_j) log_{|\Omega|}(\hat{P}(u_i|\omega_j)) \tag{3}$$

$H(u_i)$ measures the amount of information given by u_i in Ω: the lower $H(u_i)$ is, the higher information it grants.

For each $u_i,\ u_m \in U :\ u_i \neq u_m$ the joint probability given the class $\hat{P}(u_i, u_m|\omega_j)$ is estimated for each class, in order to calculate the joint entropy measure:

$$H(u_i, u_m) = -\sum_{j=1}^{|\Omega|} \hat{P}(u_i, u_m|\omega_j) log_{|\Omega|}(\hat{P}(u_i, u_m|\omega_j)) \tag{4}$$

When the entropy for each u_i and $(u_i, u_j) : u_i \neq u_j$ in U is calculated, the Information Graph of S is defined. Let IG be a directed graph represented as a set of nodes and edges, $IG = (U, E)$, where each language unit $u \in U$ is a node of the graph. Two units $u_i, u_m \in U$ are connected by a directed edge $e_{u_i,u_m} \in E$ starting from u_i and heading to u_m, if and only if

$$H(u_i) \geq H(u_i, u_m) + \theta_1 \tag{5}$$

that is, if the entropy measure is reduced above a threshold θ_1 –set in a tuning phase– when combining these two units.

Definition 1. A language unit u_j is a child of another unit u_i in $IG = (U,E)$ if and only if $e_{u_i,u_j} \in E$ exists.

Definition 2. The set of children of u_i in the $IG = (U,E)$ is the union of all the children of u_i, and is denoted as $ch(u_i) = \{u_j \in U : \exists\, e_{u_i,u_j} \in E\}$.

3.2 The Suggestion Generation Algorithm

The Suggestion Generation Algorithm (SGA) is responsible for picking a language unit $\hat{u}$ about which to ask the user in the next system turn. In order to achieve the goal defined in the DM, the suggested unit $\hat{u}$ must provide additional information regarding the task. To this end, SGA explores IG to choose $\hat{u}$.

3.2.1 Association Rules

The graph is explored using Association Rules, common in Data Mining [8, 17] because of their flexibility and real-time results, even in domains with a huge amount of features. As the suggestion generated needs to be related with the user's utterance representation Y and, at the same time, have the highest occurrence rate possible, the association rules support and confidence are used. Subsets are used instead of sequences in these rules, so let R be the set of units observed in the sequence Y where $R \subseteq U$.

$$Y = (u_{Y_1}, \cdots, u_{Y_{|Y|}}) \Rightarrow R = \{u_{Y_i}\} \;:\; i = 1, \cdots, |Y| \tag{6}$$

Being $S = \{s_j\}$ $j = 1, \cdots, |S|$ the corpus where each s_j is a sample represented as a set of language units, the support count of R is:

$$\sigma(R) = |\{s_j \in S \mid R \subseteq s_j\}| \tag{7}$$

Being Z another set of language units, the support and confidence rules are defined as follows:

$$Support:\; s(R \to Z) = \frac{\sigma(R \cup Z)}{|S|} \qquad Confidence:\; c(R \to Z) = \frac{\sigma(R \cup Z)}{\sigma(R)} \tag{8}$$

The support represents the probability of observing the sets R and Z together in the training set. The confidence rule measures how likely subset Z is to appear once R has been observed.

3.2.2 Graph Exploration Algorithm

The SGA obtains a language unit $\hat{u} \notin R$ that maximizes the confidence score $c(R \rightarrow \hat{u})$ using the connections defined in the IG.

When searching for the new language unit $\hat{u}$ related with R, it may happen that the set R or $(R \cup \hat{u})$ have not been seen in the training set S. This is why the search is done through the subsets $r' \subseteq R$, and the aim is reformulated as obtaining the $\hat{u} \notin R$ which maximizes $c(r' \rightarrow \hat{u})$ being r' the biggest subset possible.

Because the number of subsets of R are $\sum_{k=1}^{|R|} \binom{|R|}{k}$ and they grow exponentially depending on $|R|$, the A-Priori pruning algorithm is used to reduce the search space.

Theorem 1. ***A-Priori principle:*** *If a set of elements is frequent, then all of its subsets must also be frequent.*

Setting a K pruning-parameter, K amount of language units of highest support are chosen from R. Using these units, the subsets $r' \subseteq R$ are generated. The set of these subsets r' is denoted as R'. As depicted in Figure 3, when choosing the language units `fiscal` and `ledger`, the number of subsets to search through is decreased to $|R'| = \sum_{k=1}^{K} \binom{K}{k}$.

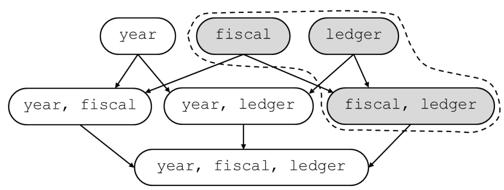

Fig. 3 A-Priori Pruning with K=2

When the most frequent subsets $r' \in R'$ are generated, the Information Graph is used to retrieve the semantic unit $\hat{u}$ which maximizes $c(R \rightarrow \hat{u})$ for the biggest r' where $\sigma(r' \cup \hat{u}) \neq 0$.

Summarizing, the suggested unit $\hat{u}$ has to meet the next conditions:

1. $\hat{u}$ must be a child of every element of a set $r' \in R'$ in the IG.
2. The support count of $(r' \cup \hat{u})$ must be non-zero: $\sigma(r' \cup \hat{u}) \neq 0$.
3. The set r' must contain as many language units as possible, up to K.

This is the actual algorithm designed to explore the graph; R' is sorted from the largest to the smallest subset due to condition 3:

Algorithm 1: Suggestion Generation Algorithm

input : R', IG, K, R
initialization: size = K, valid_units = $\{\}$, c_score = $\{\}$
for r' **in** R' **do**
  for *each unit* u_i **in** r' **do**
    Obtain the children of u_i in the IG, $ch(u_i)$;
  Set $ch(r') = \cap_{i=1}^{|r'|} ch(u_i)$;
  if $ch(r') \neq \emptyset$ **and** *size* = $|r'|$ **then**
    Append all the $u_i \in ch(r')$ to valid_units;
    Append confidence $c(r' \rightarrow u_i)$ to c_score;
  if $valid\_units = \emptyset$ **and** $|r'| = size - 1$ **then**
    size $\leftarrow$ $|r'|$;
if $valid\_units \neq \emptyset$ **then**
  Set $\hat{u}$ as the valid unit with highest confidence score unseen in R;

In short, the algorithm finds through the IG those children of the largest possible sets of R' and tag them as valid units. From those valid units, the one with highest confidence score and unseen in R is chosen.

3.3 Answer evaluation

Once the unit $\hat{u}$ is suggested through a system turn, three scenarios have been defined for the next user turn:

1. If the user confirms that the problem is related to $\hat{u}$, the classifier is re-evaluated with the updated sequence $Y_{new} = (Y, \hat{u})$.
2. If the user rejects $\hat{u}$, the SGA is re-evaluated omitting from the training set every sample $s_j \in S$ that contains $\hat{u}$.
3. If the user does not know whether $\hat{u}$ is related with the issue, the next valid unit with highest confidence is used to generate an additional system turn.

Those user calls that remain unclassified after some suggestions of SG are routed to a technician for further analysis. When this cannot be made, TC is forced to route the call to a department by setting $\theta_e = 1$.

4 The Task and Feature Selection

The input of the Topic Classifier (TC) and the Suggestion Generator (SG) module is the utterance representation of the user turn, Y, as extracted by the NLU module. The representation is a sequence of language units: $Y = (u_{Y_1}, u_{Y_2}, \cdots, u_{Y_{|Y|}})$ (see Figure 2), where $u_{Y_j} \in U$, U being the set of language units selected from the training set S. The units chosen to evaluate the system proposed in this paper are lemmas automatically selected from the corpus available. As proven in [16], words tend to overfit the task and are computationally too demanding, whereas concepts are lighter but do not lend that easily to automatic selection and update. Lemmas seek the balance between the two.

In the best-case scenario, the corpus from which to extract the lemmas should consist of recorded calls of the service to be automatized. However, the corpus actually available is composed of 25568 written technical records of issues consulted. They are in Spanish and organized in 5 classes, each corresponding to one of the departments to which calls must be routed: ω_0 - Finance, ω_1 - Human Resources, ω_2 - Information Technologies, ω_3 - Logistics, and ω_4 - Software & Hardware. They include e-mails sent by customers explaining their problems and/or quick notes taken by the technicians that attended to them. Thus, a normalization pre-processing has been necessary to make records as similar as possible to the output of an automatic speech recognizer: typographic and orthographic errors have been corrected, shortened forms expanded, and digits converted to character sequences.

The proposed architecture requires that the lemmas selected from the corpus be encoded in grammar-rules for the NLU module to extract the language units from user turns. In this light, automatic selection of lemmas comes down to automatic generation of lemma-grammars. The procedure designed for this purpose draws on the language analyzer toolkit Freeling [2] and consists on the following steps: *a)* tokenize the corpus and discard stopwords; *b)* lemmatize and analyze morphologically the words in the resulting list; *c)* keep the words labeled Noun, Verb or Adjective with a confidence score higher than 0.3 –a threshold fixed in a tuning phase–; and, *d)* group together the words with the same lemma. Each resulting cluster <lemma + words> is a rule of the grammar that tells the parser to return the lemma whenever the user turn contains one of the words associated to it. Figure 4 shows part of a real example of a grammar-rule automatically generated:

```
[Recuperar]            \\ Lemma     To recover
  (recuperar)          \\ Word 1    to recover
  (recuperados)        \\ Word 2    recovered
  (recuperas)          \\ Word 3    you recover
  ...
;
```

Fig. 4 Partial lemma-grammar rule example

The grammar generated applying the procedure explained to the entire normalized corpus has 5882 rules (i.e., lemmas), reducing the dimension of U by 60.96% as compared to using the whole vocabulary of the corpus as language units (see Table 1). More significantly, parsing the corpus with this grammar has revealed that the utterance representation dimension diminishes 76.69% on average (see Table 2).

Table 1 Language unit set (U) dimension reduction

Vocabulary size (i.e., amount of distinct words in the corpus)	15070
Amount of lemmas selected	5882

Table 2 Utterance representation (Y) dimension reduction

Amount of distinct words per record	26.26 $\{820.40 \pm\}$
Amount of lemmas extracted per record	6.12 $\{8.60 \pm\}$

5 Experimental Evaluation

The experimental section is divided in two sections. The first one evaluates the impact of the entropy threshold θ_e in the classifier's metrics. In the second section, the SGA is used to recover information for Uncertain Cases, making suggestions throughout the dialog.

5.1 Classifier Evaluation

Once the user has explained their issue, the utterance representation Y is passed to TC to choose the target department of call-routing. As stated in Section 3, the entropy of Y is used to determine whether the user is giving enough information to route the call with high confidence or not. The call is routed if:

$$H(Y) \leq \theta_e \tag{9}$$

The representations Y which are above the threshold θ_e should not to be routed, so the classifier's decision is revoked.

In our particular case, the classification algorithm Multinomial Naive Bayes (M-NB)[1] is used in the Dialog System, as it has proven to be well-suited for the task [16]. In order to observe the impact of the threshold in the classification metrics, the obtained F_1 score is depicted along with the ratio of classified instances in Figure 5: As can be observed, the lower θ_e is set, the lower is the ratio of calls routed –they are labeled as UC– and the higher is the F_1 score. Since an automatic algorithm has been designed to retrieve more information from the user –the SGA–, the main focus is set on seeking balance between the scores and the percentage of UC samples. For

[1] The implementation can be found in the Scikit-learn Python package [14].

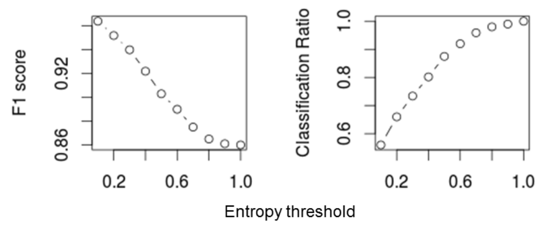

Fig. 5 Entropy threshold adjustment

this reason, the threshold θ_e has been set to 0.5.

Table 3 shows the results of classification excluding the UC records obtained with the new threshold, as compared to having $\theta_e = 1$: excluding the 12.5% UC from classification improves the metrics of call-routing by 4 points; these results suggest that entropy is useful to detect the cases at highest risk of being wrongly categorized and, thus, routed to an incorrect department.

Table 3 Results of MNB classifier with θ_e threshold set to 1 and 0.5

MNB	Precision	Recall	F_1	UC %
$\theta_e = 1$	0.86 ± 0.01	0.86 ± 0.01	0.86 ± 0.01	0.0
$\theta_e = 0.5$	$\mathbf{0.90} \pm 0.01$	$\mathbf{0.90} \pm 0.01$	$\mathbf{0.90} \pm 0.01$	12.5

5.2 Recovering from Uncertainty

A second set of experiments has been carried out in order to evaluate SGA and its reliability at recovering from Uncertain Cases (UC). First, an assessment has been done in terms of utterance appropriateness, as defined in [5]:

> An utterance is considered appropriate when [...] it asks for additional information which is essential to respond to the user's request [...]. A[ppropriate] U[tterance] evaluates whether the DM provides a coherent response at each turn according to its input.

We have measured the metric manually, taking the 12.5% UC records of one of the folds in the previous experiments –322 in total– to emulate user turns as input for SG module. K is set to 5. Table 4 shows the results of the evaluation locally (i.e., taking as reference the amount of suggestions generated) and globally (taking as reference all the instances labeled as UC).

The second experiment aims at checking that the information retrieved with SG helps improve the classification confidence. To this end, the same 322 UC-input/system turn pairs of the previous experiment have been used as the initial state. A user turn has been emulated that always accepts the suggestion offered by the system (the first scenario of answer evaluation, Section 3.3). Up to 4 additional suggestion and acceptance rounds have been emulated whenever the classifier returned UC after its update. Table 5 shows the results:

Table 4 Appropriateness of system turns generated with SG (K = 5)

	Total Instances	Total Suggestions	AU	Non-AU
SGA	322	318	267	51
Global %	—	**98.7**	**83**	15.7
Local %	—	—	84	16

Table 5 Classification results using SG to resolve low information user turns

	Suggestions	Turn 1	Turn 2	Turn3	Turn 4	Turn 5	Total Instances = 322	
SGA	318	159	68	33	17	17	Local	Global
					Classified %		92.5	**91.3**
					Not Classsified %		7.5	**8.7**

The results obtained indicate that the suggested language unit $\hat{u}$ reduces the entropy of the sample, improving the classification confidence and helping in the intended call-routing task.

6 Conclusions and Future Work

This paper proposes a strategy that reinforces the common Dialog Manager logic. More specifically, it generates dialog systems turns automatically to retrieve information from the user when the latter does not provide enough information to complete the task set. This is done introducing an entropy measure criterion to detect uncertainty and low classification confidences, plus the the Suggestion Generation (SG) module, which chooses, using a graph representation of the domain knowledge –the Information Graph (IG)–, a language unit related with high confidence to the current dialog.

The proposed SG Algorithm has proven to be effective, rendering an appropriate suggestion 82.91% of the times. The suggestions generated are meaningful for the classification task 91.3% of the times. These results prove that IG is a valid structure to represent the informational relationships of the task domain. In conclusion, the inclusion of the entropy-based uncertainty detection and the SG module in a Spoken Dialog System improves the Dialog Manager logic, obtaining a more flexible and adaptive system.

Regarding future work, full implementation of the Dialog System in a real-user environment is planned, to face real and new situations, and thus evaluating SG in a wider range of situations and obtaining new data directly from the users. Also, other ways to explore the IG are being developed, taking into account the amount of information provided by each language unit of the graph.

References

1. Bohus, D., Rudnicky, A.: The RavenClaw dialog management framework: architecture and systems. Computer Speech and Language XXIII(3), 257–406 (2008)
2. Carreras, X., Chao, I., Padró, L., Padró, M.: Freeling: An Open-Source Suite of Language Analyzers. Proceedings of the 4th Language Resources and Evaluation Conference (LREC 2004) 4, 239–242 (2004), http://hnk.ffzg.hr/bibl/lrec2004/pdf/271.pdf
3. Chu-Carroll, J., Carpenter, B.: Vector-based natural language call routing. Comput. Linguist. 25(3), 361–388 (Sep 1999)
4. Denecke, M., Waibel, A.: Dialogue strategies guiding users to their communicative goals. In: Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997 (1997)
5. Ghigi, F., Torres, M.I., Justo, R., Benedí, J.M.: Evaluating spoken dialogue models under the interactive pattern recognition framework. In: INTERSPEECH 2013. pp. 480–484. Lyon (2013)
6. Gorin, A.L., Riccardi, G., Wright, J.H.: How may i help you? Speech Commun. 23(1-2), 113–127 (Oct 1997)
7. Griol, D., Torres, F., Hurtado, L.F., Grau, S., Sanchis, E., Segarra, E.: Development and evaluation of the DIHANA project dialog system. In: Proceedings of Interspeech-06 Satellite Workshop Dialogue on Dialogues. Multidisciplinary Evaluation of Advanced Speech-based Interactive Systems. Pittsburgh (2006)
8. Gosain, A., Bhugra, M.: A comprehensive survey of association rules on quantitative data in data mining. In: Information and Communication Technologies (ICT). pp. 1003–1008 (Apr 2013)
9. Gupta, N., Tur, G., Hakkani-Tur, D., Bangalore, S., Riccardi, G., Gilbert, M.: The at&t spoken language understanding system. Audio, Speech, and Language Processing, IEEE Transactions on 14(1), 213–222 (Jan 2006)
10. Leistensnider, J., Wildling, J.: Semantic smoothing the multinomial naive bayes for biomedical literature classification 2, 750–754 (Nov 1997)
11. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI/ICML-98 Workshop on Learning for Text Categorization. pp. 41–48 (1998)
12. Misu, T., Kawahara, T.: Dialogue strategy to clarify users queries for document retrieval system with speech interface. Speech Communication 48(9), 1137 – 1150 (2006)
13. Morchid, M., Dufour, R., Bousquet, P.M., Bouallegue, M., Linares, G., De Mori, R.: Improving dialogue classification using a topic space representation and a gaussian classifier based on the decision rule. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. pp. 126–130 (May 2014)
14. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
15. Serras, M.: Automatizando la atencin post-venta de los bienes de equipo (Sep 2015)
16. Serras, M., Perez, N., Torres, M.I., del Pozo, A., Justo, R.: Topic Classifier for Customer Service Dialog Systems. In: 18th International Conference on Text, Speech and Dialogue. Springer (2015)
17. S.J., Y., A.L.P., C.: An efficient data mining technique for discovering association rules. In: Workshop on Database and Expert Systems Applications. pp. 664–669 (Sep 1997)
18. Ward, W.: Extracting information in spontaneous speech. In: Spoken Language Processing (ICSLP '94), Third International Conference on. pp. 83–87 (1994)
19. Wen, J., Li, Z.: Semantic smoothing the multinomial naive bayes for biomedical literature classification. In: IEEE International Conference on Granular Computing (Nov 2007)