

Discovering Dialog Rules by means of an Evolutionary Approach

David Griol¹, Zoraida Callejas²

¹Dept. of Computer Science, Universidad Carlos III de Madrid, Spain

²Dept. of Languages and Computer Systems, University of Granada, Spain

dgriol@inf.uc3m.es, zoraida@ugr.es

Abstract

Designing the rules for the dialog management process is one of the most resources-consuming tasks when developing a dialog system. Although statistical approaches to dialog management are becoming mainstream in research and industrial contexts, still many systems are being developed following the rule-based or hybrid paradigms. For example, when developers require deterministic system responses to keep total control on the decisions made by the system, or because the infrastructure employed is designed for rule-based systems using technologies currently used in commercial platforms. In this paper, we propose the use of evolutionary algorithms to automatically obtain the dialog rules that are implicit in a dialog corpus. Our proposal makes it possible to exploit the benefits of statistical approaches to build rule-based systems. Our proposal has been evaluated with a practical spoken dialog system, for which we have automatically obtained a set of fuzzy rules to successfully manage the dialog.

Index Terms: Dialog Management, Spoken Dialog Systems, Dialog Rules, Evolving Classifiers.

1. Introduction

Spoken Dialog Systems (SDSs) are computer programs that receive speech as input and generate synthesized speech as output, engaging the user in a dialog that aims to be similar to that between humans [1, 2, 3, 4].

The design practices of conventional commercial dialog systems are currently well established in industry. In these practices, voice user interface (VUI) experts [5, 6] handcraft a detailed dialog plan based on their knowledge about the specific task and the business rules. In addition, designers commonly define the precise wording for the system prompts according to the dialog state and context, and also the expected types of user's utterances for each turn. As described in [7, 8], this approach is well-documented [9] and has been used to develop hundreds of successful commercial dialog systems.

The design of dialog strategies is a challenging task in the development of spoken dialog systems given two intertwined challenges: the variety of factors that must be considered to select the next system response (e.g., user goals and preferences, dialog history, results provided by the data repositories, external context, etc.) and the uncertainties derived from ambiguous inputs, partially observable environments, recognition errors, etc.

Thus, a great effort is currently employed in commercial systems to design the described set of rules for dialog management and find empirical evidence of their appropriateness. This design is usually carried out in industry by hand-crafting dialog strategies tightly coupled to the application domain in order to optimize the behavior of the dialog system in that context, which is a very time-consuming process and has the disadvantage of lack of portability and adaptation to new contexts.

This has motivated the research community to find ways for automating dialog learning by using statistical models trained with real conversations [10, 11, 12, 13]. Statistical approaches can model the variability in user behaviors and allow exploring a wider range of strategies. Although the construction and parameterization of the model depends on expert knowledge of the task, the final objective is to develop dialog systems that have a more robust behavior, better portability, and are easier to adapt to different user profiles or tasks.

In this paper, we present a hybrid approach to dialog management that seeks to combine the benefits of rule-based and statistical techniques in a single framework. To do this, we propose a statistical technique to automatically extract the set of rules for dialog management from a labeled dialog corpus. Our overall approach is related to the main statistical dialog management approaches, including those that use clustering to construct dialog graphs from dialog data [14], optimize dialog strategies using reinforcement learning (RL) [11], combine RL with information state update rules [15], use discriminative classification models to learn information state updates [16], use probabilistic rules whose values are estimated from dialog data using Bayesian inference [17], or uses procedural dialog management methodologies based on defining dialog trees [18].

However, our proposal differs from the aforementioned approaches in several main aspects. Our proposal models the dialog by means of a classification process, which considers the complete history of the dialog as input. We model this classification process by means of fuzzy-rule-based evolving classifiers [19]. This allows obtaining a set of fuzzy rules that can be directly employed to develop a rule-based dialog manager, then making possible to obtain new generation interfaces without the need for changing the already existing commercial infrastructures. These rules are designed to operate under partially observable settings.

Our proposal has been evaluated with a practical dialog system providing railway information, for which a set of fuzzy rules has been obtained to manage the dialog. The results of the evaluation show that the fuzzy-rule-based dialog manager provides satisfactory results in terms of correct responses selected and dialog success rate.

2. Our proposal for automatically defining the dialog rules

As described in the previous section, we propose the use of fuzzy-rule-based evolving classifiers to discover the set of rules that are implicit in a dialog corpus. Specifically, we propose the use of the *eClass* (evolving Classifier) family. During the training of these classifiers, a set of fuzzy rules that describes the most important observed features for the classification of each class (i.e., system prompt) is formed. These rules can also be constantly adjusted to the available training data.

One of the main advantages of *eClass* is that it does not require parameter optimization as its only parameter 'scale' can be directly inferred from the training data. This technique [19] is based on partitioning the data space into overlapping local regions through *Recursive Density Estimation* (RDE) and associating clusters (respectively fuzzy sets) to them. The *eClass* family includes two different architectures and on-line learning methods: i) *eClass0* with the classifier consequents representing class label [19]; ii) *eClass1* for regression over the features using the first order eTS fuzzy classifier.

2.1. Codification of the input of the classifier

Our proposal is not only focused on slot-filling dialog systems, for which dialog managers use a structure comprised of one slot per piece of information that the system can gather from the user, but can also consider additional information pieces related to the responses provided by the system or the context of the interaction (e.g., user's previous interactions with the system, user's preferences, users' emotional state, location, etc.).

We model the complete set of input features by means of a data structure, which we call *Interaction Register* (*IR*). The *IR* keeps the information provided by the user (e.g., slots) throughout the previous history of the dialog and the described additional information sources. The modeled dialog systems can capture several data at once and the information can be provided in any order (more than one slot can be filled per dialog turn and in any order), thus supporting mixed-initiative dialogs.

To extract the set of dialog rules and employ them to take the next system decision, we have assumed that the exact values of the task-dependent entities are not significant. They are important for accessing databases and for constructing the output sentences of the system. However, the only information necessary to predict the next action by the system is the presence or absence of dialog entities. Therefore, the codification we use for each dialog act provided by the SLU module is in terms of three values, $\{0, 1, 2\}$, according to the following criteria:

- (0): The value for the slot has not been provided;
- (1) The value is known with a confidence score that is higher than a given threshold;
- (2): The value of the slot has a confidence score that is lower than the given threshold.

2.2. Obtaining the set of fuzzy rules

Fuzzy-rule-based evolving classifiers are trained from a labeled corpus of training dialogs. In this paper, we focus on the *eClass0* classifier, given that it possesses a zero-order Takagi-Sugeno consequent, so a fuzzy rule in the *eClass0* model has the following structure:

$$\begin{aligned} \text{Rule}_i &= IF(\text{Feature}_1 \text{ is } P_1) \text{ AND } \dots \\ &\dots \text{ AND } (\text{Feature}_n \text{ is } P_n) \\ &\text{ THEN } \text{Class} = c_i \end{aligned} \quad (1)$$

where i represents the number of rule; n is the number of input features (observations corresponding to the different slots defined for the semantic representation of the user's utterances); the vector *Feature* stores the observed features, and the vector *P* stores the values of the features of one of the prototypes (coded in terms of three possible values, $\{0, 1, 2\}$) of the corresponding class $c_i \in \{\text{set of different classes}\}$. Each class is then associated to a specific system action (response).

The *eClass0* model consists of several fuzzy rules per class (the number of rules depends on the heterogeneity of the input data of the same class). During the training process, a set of rules is formed from scratch using an evolving clustering approach to decide when to create new rules. The inference in *eClass0* is produced using the "winner takes all" rule and the membership functions that describe the degree of association with a specific prototype are of Gaussian form.

The *potential* (Cauchy function of the sum of distances between a certain data sample and *all* other data samples in the feature space) is used in the partitioning algorithm. However, in these classifiers, the potential (*P*) is calculated recursively, which makes the algorithm faster and more efficient. The potential of the k^{th} data sample (x_k) is calculated by means of equation 2 [20]. The result of this function represents the *density* of the data that surrounds a certain data sample.

$$P(x_k) = \frac{1}{1 + \frac{\sum_{i=1}^{k-1} \text{distance}(x_k, x_i)}{k-1}} \quad (2)$$

where the function *distance* provides the distance between two samples in the data space.

The potential can be calculated using the euclidean or the cosine distance. In this case, cosine distance (*cosDist*) is used to measure the similarity between two samples; as described in Equation 3.

$$\text{cosDist}(x_k, x_p) = 1 - \frac{\sum_{j=1}^n x_{kj} x_{pj}}{\sqrt{\sum_{j=1}^n x_{kj}^2 \sum_{j=1}^n x_{pj}^2}} \quad (3)$$

where x_k and x_p represent the two samples to measure its distance and n represents the number of different attributes in both samples.

The resolution of Equation 2 requires all the accumulated data sample available to be calculated, which contradicts to the requirement for real-time and on-line application needed in the proposed problem. For this reason, a recursive expression for the cosine distance is proposed in [20]:

$$\begin{aligned} P_k(z_k) &= \frac{1}{2 - \frac{1}{(k-1) \sqrt{\sum_{j=1}^n (z_k^j)^2}} B_k}; k = 2, 3, \dots \\ \text{where : } B_k &= \sum_{j=1}^n z_k^j b_k^j; b_k^j = b_{(k-1)}^j + \sqrt{\frac{(z_k^j)^2}{\sum_{l=1}^n (z_l^j)^2}} \\ \text{and } b_1^j &= \sqrt{\frac{(z_1^j)^2}{\sum_{l=1}^n (z_l^j)^2}}; j = [1, n+1]; P_1(z_1) = 1 \end{aligned} \quad (4)$$

where z_k represents the k^{th} data sample (x_k) and its corresponding label ($z = [x, \text{Label}]$). Using this expression, it is only necessary to calculate $(n+1)$ values where n is the number of different subsequences obtained; this value is represented by b , where $b_k^j, j = [1, n]$ represents the accumulated value for the k^{th} data sample.

In this case, a specific system action can be represented by several rules, depending on the heterogeneity of the samples that represent the same action. Thus, a class could be represented by one or several prototypes. The different prototypes that represent a system action are obtained from the input data and they are updated constantly. However, an initial rule-based model can be defined (if necessary) by hand as start point of the classifier. In this sense, new prototypes are created or existing prototypes are removed if necessary.

The *eClass0* evolving classifier is trained by means of the set of steps that are described in [21]:

1. Classify each new sample in a group represented by a prototype (i.e., data sample that groups several samples which represent a specific system action). To do this, the sample is compared with all the prototypes previously created.
2. Calculate the potential of the new data sample to be a prototype. Based on the potential of the new data sample to become a prototype, it could form a new prototype or replace an existing one.
3. All the existing prototypes are updated considering the new data sample. A new prototype is created if its value is higher than any other existing prototype. Existing prototypes could also be removed.

2.3. Practical application: a spoken dialog system providing railway information

We have applied our proposal to extract the dialog rules for a mixed-initiative spoken dialog system providing railway information system using spontaneous speech in Spanish [13]. As in many other conversational agents, the semantic representation chosen for dialog acts of the SLU module is based on the concept of frame [22]. This way, one or more dialog acts represent the intention of the utterance, and a sequence of attribute-value pairs contains the information about the values given by the user. For the task, we defined four entities related to the queries users can ask for (*Hour*, *Fares*, *Train-Type*, *Trip-Time*, and *Services*) and eleven entities, which can be divided into two groups:

1. *Task-dependent entities*: they represent the information pieces that are required to complete the previously described (*Origin*, *Destination*, *Departure-Date*, *Arrival-Date*, *Class*, *Departure-Hour*, *Arrival-Hour*, *Train-Type*, *Order-Number*, and *Services*).
2. *Task-independent entities*: they represent typical interactions in a dialog used after a confirmation required by the system (*Acceptance*, *Rejection*, and *Not-Understood*).

A total of 51 system responses were defined for the task (classified into confirmations, questions to require data from the user, and answers obtained after a query to the database).

Using the previously described codification for the entities, when a dialog starts (in the greeting turn) all the values in the Interaction Register are initialized to “0”. The information provided by the users in each dialog turn is employed to update the previous values and obtain the current ones, as Figure 1 shows.

This figure shows the semantic interpretation and confidence scores (in brackets) for a user’s utterance provided by the SLU module. In this case, the confidence score assigned to the entity *Date* is very low. Thus, a “2” value is added in the corresponding position for this entity. The query (*Hour*) and the entity *Destination* are recognized with a high confidence score, adding a “1” value in the corresponding positions.

The set of features for the classifier includes the codification of the different queries and task-dependent entities that can be provided by the user and the task-independent queries provided in the last user turn (none in this case).

An initial corpus of 900 dialogs (10.8 hours) was acquired for the task by means of the Wizard of Oz technique with 225 real users, for which an initial dialog strategy was defined by

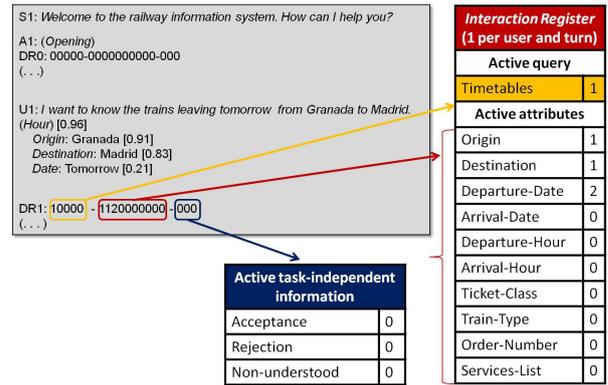


Figure 1: Excerpt of a dialog for the railway task with its correspondent representation of the task-dependent and active task-independent information for one of the dialog turns

experts [13]. A set of 20 scenarios was used to carry out the acquisition. Each scenario defined one or two objectives to be completed by the user and the set of entities that they must provide. The corpus consists of 6,280 user turns, with an average number of 7.7 words per turn. The corpus was split into a training subset of 4,928 samples (80% of the corpus) and a test subset of 1,232 samples (20% of the corpus).

A total of 49 rules for the task were obtained with *eClass0*. Figure 2 shows the structure of these rules. Using them, the dialog rule corresponding to the class '*SystemResponse23*' (confirmation of the departure date) would be selected for the dialog turn described in Figure 1.

```
FRB – RailwayTask(eClass0) :

IF (Hour is 1) AND (Fares is 0) AND ... AND (Not –
Understood is 0)
THEN Class = ' Ask – Date'

IF (Hour is 2) AND (Fares is 0) AND ... AND (Not –
Understood is 1)
THEN Class = ' Confirm – Hour'

IF (Hour is 0) AND (Fares is 1) AND ... AND (Not –
Understood is 0)
THEN Class = ' Provide – Fares'

IF (Hour is 1) AND (Fares is 2) AND ... AND (Not –
Understood is 1)
THEN Class = ' Close – Dialog'
...
```

Figure 2: Set of fuzzy rules obtained with the *eClass0* classifier for the railway task

3. Evaluation

We have completed an evaluation of the practical application of our proposal covering three main objectives. Firstly, we have compared the responses selected applying the learned fuzzy rules with the ones that would be selected using very well-known alternatives for the definition of the classification function. Secondly, we have asked several VUI experts whether the dialog rules automatically obtained were coherent with the corresponding dialog situations. Finally, we have evaluated a dialog manager applying the set of fuzzy rules with recruited users.

3.1. Evaluation of the classification function

We have firstly assessed the behavior of our proposal comparing it with different definitions of the classification function used to determine the next system response. Three approaches for the definition of this function were used: a multilayer perceptron (MLP), a multinomial naive Bayes classifier, and finite-state classifiers (bigram models, trigram models, and Morphic Generator Grammatical Inference (MGGI) models [23]). A 5-fold cross-validation process was used to carry out this evaluation. The corpus was randomly split into five subsets of 1,232 samples (20% of the corpus).

We propose three measures to evaluate the obtained set of rules, which are calculated by comparing the response automatically generated by applying this set for each input in the test partition with regard to the reference answer annotated in the corpus (the answer provided by the WOz). This way, the evaluation is carried out turn by turn. These three measures are: i) *Matching*: percentage of responses selected by means of the rules that match exactly the initial dialog strategy defined by the experts; ii) *Coherence*: percentage of responses that are coherent with the current dialog state but do not match the initial strategy defined to acquire the training corpus; and iii) *Error*: percentage of responses that could cause a dialog failure.

Table 1 shows the results obtained. As it can be observed, the Fuzzy-rule-based classifier provides satisfactory results in terms of the percentage of correct responses selected (*Matching* and *Coherence* measures) and responses that could cause the failure of the dialog (*Error* measure). With regard the rest of classifiers, the MLP classifier is the one providing the closest results to our proposal. The table also shows that among the finite-state model classifiers, the bigram and trigram classifiers are worse than the MGGI classifier, this is because they cannot capture long-term dependencies. The renaming function defined for the MGGI classifier seems to generate a model with too many states for the size of the training corpus, therefore, this classifier could be underestimated.

Table 1: Results of the evaluation of the classification functions

Dialog manager	Matching	Coherence	Error
Fuzzy-rule-based (FRB) classifier	76.7%	89.2%	5.6%
MLP classifier	76.8%	88.8%	5.8%
Multinomial classifier	63.4%	76.7%	10.6%
Bigram classifier	28.8%	37.3%	42.2%
Trigram classifier	31.7%	42.1%	44.1%
MGGI classifier	46.6%	67.2%	24.8%

3.2. Evaluation of the quality of the obtained rules

Secondly, we asked 6 VUI experts from 4 different universities to develop rule-based dialog models for the task and also evaluate the quality and coherence of each one of the automatically learned fuzzy rules. The following measures were considered: percentage of rules included in all the developed rule-based dialog models (*Included*), rules that are considered coherent for the current state of the dialog (*Coherent*), and rules that could be removed (*Removed*). Table 2 shows the results of this evaluation. As it can be observed, there was a satisfactory agreement with regard the number of rules included in the manually defined models and considered coherent with the conditions required. Only the 5.8% of rules was considered to be covered by other rules.

Table 2: Results of the comparative evaluation with manually defined rule-based dialog models

Included	83.7%
Coherent	97.2%
Removed	5.8%

3.3. Evaluation of our proposal in real time operation

Finally, we evaluated our proposal with the acquisition of 150 dialogs by means of 15 recruited users. We considered the following measures: i) Dialog success rate (*Success*); ii) Average number of turns per dialog (*nT*); iii) Confirmation rate (*Confirmation*); and iv) Error correction rate (*ECR*). The confirmation rate was computed as the ratio between the number of explicit confirmation turns and the total number of turns in the dialog. The *ECR* was computed as the number of errors detected and corrected by the dialog manager divided by the total number of errors.

The results presented in Table 3 show that in most cases the automatically learned dialog model has the capability of correctly interacting with the user. The dialog success depends on whether the system provides the correct data for every objective user's query. All of the objectives defined are achieved in 93.5% of the dialogs. The analysis of the main problem detected in the acquired dialogs shows that, in some cases, the system did not detect the introduction of data with a high confidence value due to errors generated by the ASR that were not detected by the dialog manager. However, the evaluation confirms a good operation of the approach since the information is correctly given to the user in the majority of cases. The confirmation and error correction rates have also a remarkable impact on the described system performance.

Table 3: Results of the evaluation with recruited users

Success	nT	Confirmation	ECR
93.5%	13.8	22%	0.87%

4. Conclusions and Future Work

In this paper, we have presented a proposal that employs evolving classifiers to automatically obtain a set of fuzzy rules that can be directly employed to develop a rule-based dialog manager, thus reducing the considerable effort and time that is required to manually define the dialog strategy. We have applied our proposed technique to develop a dialog manager for a system that provides railway information. The evaluation results show that the proposed technique can predict coherent system answers in most of the cases.

For future work we are interested in applying our proposal to multi-domain tasks in order to measure the capability of our methodology to adapt efficiently to contexts that vary dynamically. We also want to combine our proposal to facilitate the interaction using also additional input and output modalities.

5. Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823907 (MENHIR project: <https://menhir-project.eu>).

6. References

- [1] M. McTear, Z. Callejas, and D. Griol, *The Conversational Interface: Talking to Smart Devices*. Springer, 2016.
- [2] M. McTear and Z. Callejas, *Voice Application Development for Android*. Packt Publishing, 2013.
- [3] R. Pieraccini, *The Voice in the Machine: Building Computers That Understand Speech*. MIT Press, 2012.
- [4] Z. Callejas, K. Benghazi, M. Noguera, M. Torres, and R. Justo, “MENHIR: Mental health monitoring through interactive conversations,” *SEPLN Journal*, vol. 63, 2019.
- [5] D. Suendermann, *Advances in Commercial Deployment of Spoken Dialog Systems*. Springer-Verlag, 2011.
- [6] E. Barnard, A. Halberstadt, C. Kotelly, and M. Phillips, “A Consistent Approach To Designing Spoken-dialog Systems,” in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU’99)*, Keystone, Colorado, USA, 1999, pp. 1173–1176.
- [7] R. Pieraccini and J. Huerta, “Where do we go from here? Research and commercial spoken dialog systems,” in *Proc. 6th SIG-dial Workshop on Discourse and Dialog*, Lisbon, Portugal, 2005, pp. 1–10.
- [8] J. Williams, “The best of both worlds: Unifying conventional dialog systems and POMDPs,” in *Proc. International Conference on Spoken Language Processing (InterSpeech’08)*, Brisbane, Australia, 2009, pp. 1173–1176.
- [9] M. Cohen, J. Giangola, and J. Balough, *Voice User Interface Design*. Addison Wesley, 2004.
- [10] D. Griol, Z. Callejas, R. López-Cózar, and G. Riccardi, “A domain-independent statistical methodology for dialog management in spoken dialog systems,” *Computer Speech and Language*, vol. 28, no. 3, pp. 743–768, 2014.
- [11] S. Young, M. Gasic, B. Thomson, and J. Williams, “POMDP-based Statistical Spoken Dialogue Systems: a Review,” *Proc IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.
- [12] L. Hurtado, J. Planells, E. Segarra, E. Sanchis, and D. Griol, “A Stochastic Finite-State Transducer Approach to Spoken Dialog Management,” in *Proc. 11th Annual Conference of the International Speech Communication Association (InterSpeech’10)*, Makuhari, Chiba, Japan, 2010, pp. 3002–3005.
- [13] D. Griol, L. Hurtado, E. Segarra, and E. Sanchis, “A Statistical Approach to Spoken Dialog Systems Design and Evaluation,” *Speech Communication*, vol. 50, no. 8–9, pp. 666–682, 2008.
- [14] C. Lee, S. Jung, K. Kim, and G. G. Lee, “Automatic agenda graph construction from human-human dialogs using clustering method,” in *Proc. of Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder (USA), 2009, pp. 89–92.
- [15] P. Heeman, “Combining reinforcement learning with information-state update rules,” in *Proc. 8th Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL’07)*, Rochester, New York, USA, 2007, pp. 268–275.
- [16] D. Hakkani-Tur, G. Tur, L. Heck, A. Fidler, and A. Celikyilmaz, “A discriminative classification-based approach to information state updates for a multi-domain dialog system,” in *Proc. of 13th Annual Conference of the International Speech Communication Association*, Portland, Oregon (USA), 2012, pp. 330–333.
- [17] P. Lison, “A hybrid approach to dialogue management based on probabilistic rules,” *Computer Speech & Language*, vol. 34, no. 1, pp. 232 – 255, 2015.
- [18] S. Volkova, P. Choudhury, C. Quirk, B. Dolan, and L. Zettlemoyer, “Lightly Supervised Learning of Procedural Dialog Systems,” in *Proc. of 51st Annual Meeting of the Association for Computational Linguistics*, Sofia (Bulgaria), 2013, pp. 1669–1679.
- [19] P. Angelov and D. Filev, “An approach to online identification of takagi-sugeno fuzzy models,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 1, pp. 484–498, feb. 2004.
- [20] P. Angelov and X. Zhou, “Evolving fuzzy-rule-based classifiers from data streams,” *IEEE T. Fuzzy Systems*, vol. 16, no. 6, pp. 1462–1475, 2008.
- [21] F. Ordóñez, J. Iglesias, P. de Toledo, A. Ledezma, and A. Sanchis, “Online activity recognition using evolving classifiers,” *Expert Systems with Applications*, vol. 40, no. 4, pp. 1248–1255, 2013.
- [22] M. Minsky, *The Psychology of Computer Vision*. McGraw-Hill, 1975, ch. A Framework for Representing Knowledge, pp. 211–277.
- [23] E. Segarra and L. Hurtado, “Construction of Language Models using Morfic Generator Grammatical Inference MGGI Methodology,” in *Proc. Eurospeech’97*, 1997, pp. 2695–2698.