Hybrid Syntactic-Semantic Reranking for Parsing Results of ECAs Interactions Using CRFs

Enzo Acerbi[1], Guillermo Pérez[1], and Fabio Stella[2]

[1] Julietta Research Group, University of Seville
{enzoace,gperez}@us.es
[2] University of Milano-Bicocca
stella@disco.unimib.it

Abstract. Reranking modules of conventional parsers make use of either probabilistic weights linked to the production rules or just hand crafted rules to choose the best possible parse. Other proposals make use of the topology of the parse trees and lexical features to reorder the parsing results. In this work, a new reranking approach is presented. There are two main novelties introduced in this paper: firstly, a new discriminative reranking method of parsing results has been applied using Conditional Random Fields (CRFs) for sequence tagging. Secondly, a mixture of syntactic and semantic features, specifically designed for Embodied Conversational Agents (ECAs) interactions, has been used. This approach has been trained with a Corpus of over 4,000 dialogues, obtained from real interactions of real users with an online ECA. Results show that this approach provides a significant improvement over the parsing results of out-of-domain sentences; that is, sentences for which there is no optimal parse among the candidates given by the baseline parse.

Keywords: Embodied conversational agents, natural language processing, dialogue systems, sequence tagging, CRFs.

1 Introduction

1.1 Embodied Conversational Agents

Conversational Agents (CAs) can be defined as "*communication technologies that integrate computational linguistics techniques with the communication channel of the Web to interpret and respond to statements made by users in ordinary natural language*" [1]. Embodied Conversational Agents (ECAs) are empowered with a human representation that shows some degree of empathy (smiling, showing sadness, disgust) with the user as the dialogue goes on.

The fact of adding explicit anthropomorphism in Conversational Agents has some effects over the solution designed:

- A number of the user interactions are actually social dialogue or "small-talk", where the users interact with the ECA informally [2]

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 15–26, 2010.

– Users may perceive the combination of embodied characters with advanced natural language processing techniques and social dialogue strategies positively. But on the other hand, if the language understanding performance or the social dialogue strategies behave poorly, users perceive the solution worse than the equivalent text-only chatbot without any character [3], [4].

Natural language processing for commercial ECAs applications shows some peculiarities. Usually, customers and service providers come to an agreement on the set of questions and services that the final users may request to the ECA. Customers demand optimal performance and fast reaction time over the previously agreed domain. This implies that these in-domain utterances from the user have to be accurately parsed, while some degree of flexibility can be tolerated in the rest of the sentences. A common approach to cope with these requirements is to divide the lexical items into two groups: those that belong to the agreed Corpus and the rest of the words. The first group is configured using domain specific semantic labels while the second one is assigned common syntactic categories.

Similarly, the production rules at grammar level are semantically oriented for the sentences included in the ECA's Corpus and syntactically oriented for utterances that don't belong to the ECA's Corpus.

This work has been trained over a set of 4,000 sentences from real users to an online ECA. The application domain is a Corpus of common questions asked to the customer service of a furniture retail company. Examples of these questions are:

1. What are your opening hours?
2. How much does a sofa cost?

Along with the retail specific questions, there is a wide coverage of general questions included. These questions include flirting interactions, insults, compliments and general knowledge (politics, sport, etc.). This coverage is treated as part of the domain configuration and is known as the "social configuration" or "personality" of the ECA.

1.2 Related Work

The idea of discriminative reranking of parsing results is not new. In [5], [6] the authors propose a reranking process over the parsing results using a Maximum Entropy approach. Also Collins [7] propose a similar strategy making use of Markov Random Fields and boosting approaches, achieving significant improvement on error rate over the baseline system performance.

The approaches detailed in those papers are based on lexical and syntactic features describing the components of the parse tree and their syntactic relationship. The reranking layer is applied over a set of candidates which are obtained with a classical generative parser.

In [8] an application of the previous proposals for semantic parsing is described. In addition to the purely syntactic features, the authors include semantic features on the reranking process, obtaining partial improvements.

In this paper radical a different strategy is proposed: all parse tree structure is ignored and only terminal symbols are taken into account. To our knowledge, there is no previous work on reranking parsing results making use of sequence labeling as reranking method.

1.3 Generative Parser

The approach hereby described relies on a set of candidate parsing results provided by a generative parser. The parser used in the experiments was [9], [10], a unification grammar based context free parser inspired in the Lexical Functional Grammar formalism [11]. The parsing results are therefore provided by means of two different structures: the F-structure and the C-structure. The first one is a set of language independent attribute–value pairs while the second one is the language–dependent parse tree.

Regarding the parsing strategy, the previously described mixture syntactico-semantic approach has been followed: semantically oriented lexical and grammatical description for the domain and personality Corpora, and a syntactically oriented configuration for the other utterances. When the parser provides a parse with plain syntactic labels of an incoming sentence, the ECA uses it to look up the customer's web site for pages where the mentioned terms are included. On the other hand, when the parser provides a parse with semantic labels, the ECA returns the appropriate preconfigured answer with that representation or engages in a subdialogue with the user.

The baseline system make use of a set of heuristic domain-independent rules to choose the best candidate. These rules take into account the tree structure of the parsing results. Some of the rules are specifically designed for ECAs interactions.

2 Conditional Random Fields

CRFs are probabilistic undirected graphical models. Each vertex of the CRF represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. $\mathbf{X}$ is the sequence of observations and $\mathbf{Y}$ is the sequence of unknown state variables that needs to be inferred based on the observations. In the application hereby described, $\mathbf{X}$ is formed by all the words of the sentence, while $\mathbf{Y}$ is the sequence of their corresponding labels. CRFs are especially suitable for sequence labeling problems since independence assumptions are made among $\mathbf{Y}$ but not among $\mathbf{X}$. Thats is, CRFs allow the use of strong interdependent and overlapping features, as required by sequence labeling problems.

Formally, CRFs can be defined as follows [12]: *Let $G = (V, E)$ be a graph such that $\boldsymbol{Y} = (\boldsymbol{Y}_v)_{v \in V}$, so that $\boldsymbol{Y}$ indexed by the vertices of G. Then ($\boldsymbol{X}, \boldsymbol{Y}$) is a conditional random field in case, when conditioned on $\boldsymbol{X}$, the random variables $\boldsymbol{Y}_v$ obey the Markov property with respect to the graph: $p(\boldsymbol{Y}_v|\boldsymbol{X}, \boldsymbol{Y}_w, w{\neq}v) = p(\boldsymbol{Y}_v|\boldsymbol{X}, \boldsymbol{Y}_w, w{\sim}v)$, where $w \sim v$ means that w and v are neighbors in G.* The likelihood probability for the CRF model θ is calculated in this way: $p_\theta(\mathbf{y} \mid \mathbf{x}) =$

$$exp\left\{\sum_{e\in E,k}\lambda_k f_k(e,\mathbf{y}\mid_e,\mathbf{x})+\sum_{v\in V,k}\mu_k g_k(v,\mathbf{y}\mid_v,\mathbf{x})\right\} \tag{1}$$

Notice that λ and μ represent the model's parameters and f and g represent the feature functions that are provided to the model in the training phase.

3 A New Approach

3.1 Parse Trees as Lexical Sequences

A key observation that allows this new approach is the fact that no pair of alternative trees provided by the generative parser share the same sequence of lexical categories. This statement is true because the syntactic ambiguity is locally solved by the baseline parser before providing the alternative trees to the reranking module. In other words, to distinguish one parse tree from another, one can just look at the categories assigned to each word in the sentence. Therefore, the problem of finding the optimal parse boils down to finding the optimal assignment of the lexical category for each word in the sentence, among those given by the parser. Thus, a new parse tree sequence representation is proposed. The problem of reranking parsing results is therefore reduced to a word-category assignment: the new problem is to find the best assignment for the whole sentence, which is a typical sequence labeling problem.

The sequence labeling problem is faced with up to 223 different labels:

- 13 classical syntactic lables (noun, verb, etc.)
- 210 domain specific semantic labels (furniture, price, etc.)
- 2 additional labels:
  - One to describe the lexical items not included in the best alternative.
  - One to identify the lexical items not included in the best alternative abut located in the middle of two partial sequences.

3.2 New Problem Characteristics

The reranking approach described in this paper is conditioned by the following issues:

- The parser handles a mixture of syntactic and semantic lexical categories and grammar production rules, with overlapping syntactic-semantic alternative trees.
- The reranking algorithm must face an elevated tagset dimension with 223 different labels. High dimensional tagsets like this one could make the problem intractable.

4 The Proposed Solution

4.1 Theory

The strategy to keep the problem tractable despite the tagset dimension is based on helping the model in two major ways. The first one is through the introduction of highly informative features in order to reduce the tagset dimension for every specific word. This goal is achieved by exploiting a priori knowledge about a term. Secondly, the model prediction is driven; the model is not asked to directly predict the correct label sequence: instead, the likelihood of every sequence is used for optimal selection. Additionally, the training set size is high enough to ensure the presence of "past cases" for every label in the tagset.

Since words in a sentence are strongly interdependent, the solution has to be able to model dependencies between entities; moreover, words can be linked to a big set of features that can help classification, but dependencies may exist also between features.

One of the most well-known approaches to sequence labeling is Hidden Markov Models [13]. The potential problem using HMMs is that they calculate p($x \mid y$), where x is the word, and y is the label. The point is that what really needs to be modeled is p($y \mid x$). A solution can be Maximum Entropy Markov Models (MEMM), where p($y \mid x$) is calculated using a maximum entropy model. But MEMM can suffer the label bias problem.

CRFs are a suitable model for the task at hands, since they do not suffer the label bias problem; they are not per-state normalized like MEMMs: instead of being trained to predict each label independently, they are trained to get the whole sequence correctly.

4.2 Implementation

Offline. Due to the specificity of the problem, the creation of an *ad hoc* training set has been necessary in order to take into account domain-dependent semantic categories. The training set is formed by a Corpus of over 4,000 dialogues of Human-ECAs interactions and all the alternative parse trees for every sentence.

During the offline phase, the correct alternative has been manually tagged for every sentence. The tagging process was done by choosing among a set of sequence-like representation of the parse trees. The tagger application graphically shows, for each sentence, all the possible sequences of lexical categories and allows to select the best sequence or the best combination of sequences. Figure 2 shows a screenshot of the application.

If the sequence selected does not include all the words in the sentence, the excluded words are labelled as *Not_Used*. Sometimes the correct parse tree of a sentence is captured by a combination of two or more partial sequences. In order to prevent the bad tendency of the model to predict too many words as *Not_Used*, words between two partial sequences are classified as *Link*. Thus the model learns to distinguish between words that can be ignored, namely *Not_Used*, and words that are functioning as a bridge between partial sequences, namely *Link*. Figure 3 shows the merging of the two partial sequences selected in Figure 2.

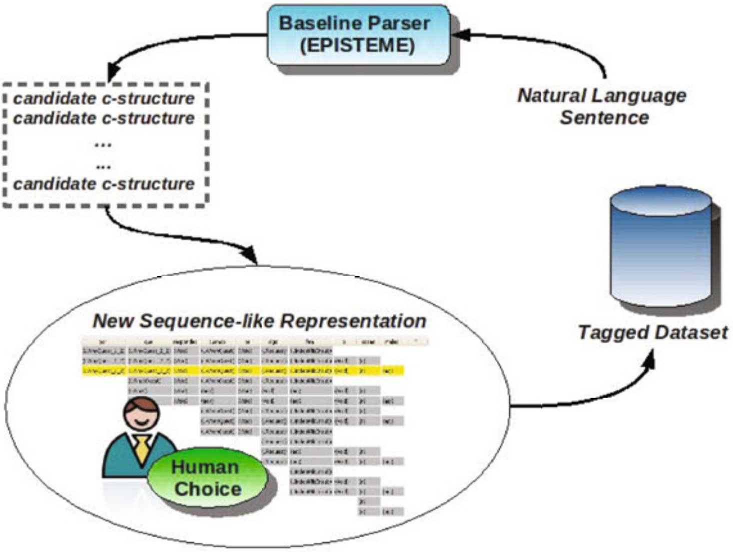


Fig. 1. Offline processing of the proposed approach

Sentences can be classified in two main categories:

- **In-domain sentences:** Sentences for which an optimal full-parse sequence or an optimal combination of partial sequences can be obtained among the candidates given by the baseline parser.
- **Out-of-domain sentences:** Sentences for which an optimal full-parse sequence or an optimal combination of partial sequences can not be obtained among the candidates given by the baseline parser.

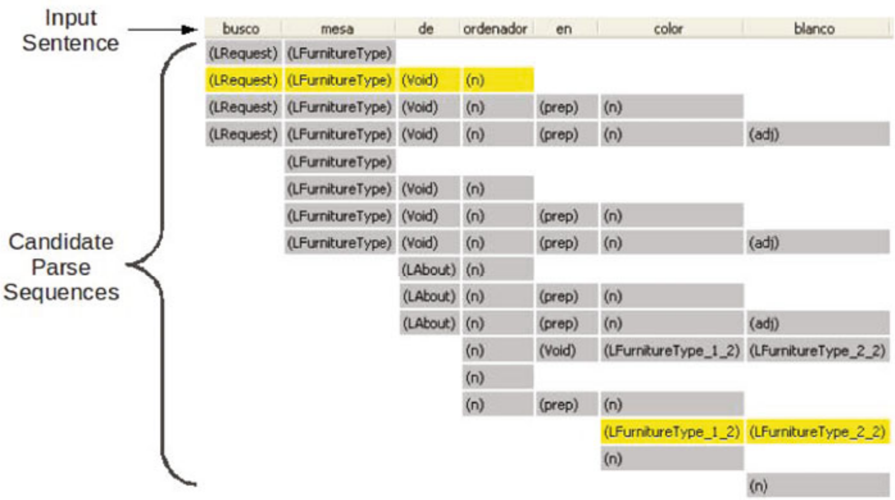
Both kinds of sentences were tagged, but only in-domain sentences were used to train the model.

The following table provides some data about the distribution of in-domain and out-of-domain sentences in the dataset:

	Number of Sentences	Number of Words	Average length
In-domain	4,096	32,134	8.2
Out-of-domain	1,011	15,712	13.8

Besides these two groups, extremely bad formed sentences were classified as *No Parse* and discarded in the training phase (5% of the total amount of analyzed sentences).

The tagging application allows the user to choose the correct sequence in a time between 5 and 15 seconds approximately, depending on the sentence complexity. The final average tagging rate was 100 sentences per hour; the entire tagging process took over 50 hours.



busco	mesa	de	ordenador	en	color	blanco
(LRequest)	(LFurnitureType)					
(LRequest)	(LFurnitureType)	(Void)	(n)			
(LRequest)	(LFurnitureType)	(Void)	(n)	(prep)	(n)	
(LRequest)	(LFurnitureType)	(Void)	(n)	(prep)	(n)	(adj)
	(LFurnitureType)					
	(LFurnitureType)	(Void)	(n)			
	(LFurnitureType)	(Void)	(n)	(prep)	(n)	
	(LFurnitureType)	(Void)	(n)	(prep)	(n)	(adj)
		(LAbout)	(n)			
		(LAbout)	(n)	(prep)	(n)	
		(LAbout)	(n)	(prep)	(n)	(adj)
			(n)	(Void)	(LFurnitureType_1_2)	(LFurnitureType_2_2)
			(n)			
			(n)	(prep)	(n)	
					(LFurnitureType_1_2)	(LFurnitureType_2_2)
					(n)	
						(n)

Fig. 2. A detail of the application created for corpus tagging

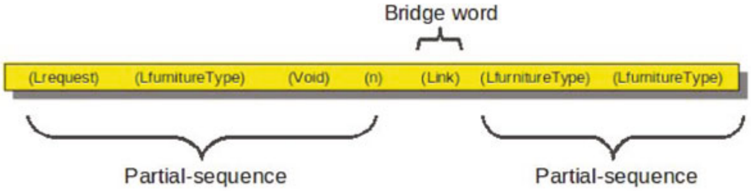


Fig. 3. Merging sequences in a unique global sequence

The MALLET library [14] was used for building the CRF model and has been modified to obtain the likelihood associated with each candidate. The model was validated with a 5-fold cross validation; details about the dataset are provided in the Experimental Results section.

Online. The online phase refers to the real interactions between the ECA and a user. Figure 4 shows the way the CRF model is used at running time: the natural language sentence provided by the user is analyzed by the baseline parser and a set of candidate sequences is returned. At this point, all possible combinations of partial sequences have to be generated and added to the original set of candidates. The CRFs model trained in the offline phase returns the likelihood probability associated with each candidate sequences. The highest likelihood sequence is identified as the optimal one, and the related c-structures (one or more) chosen.

Features. As previously mentioned, a set of highly informative features was introduced in order to limit the number of possible labels for a specific word.

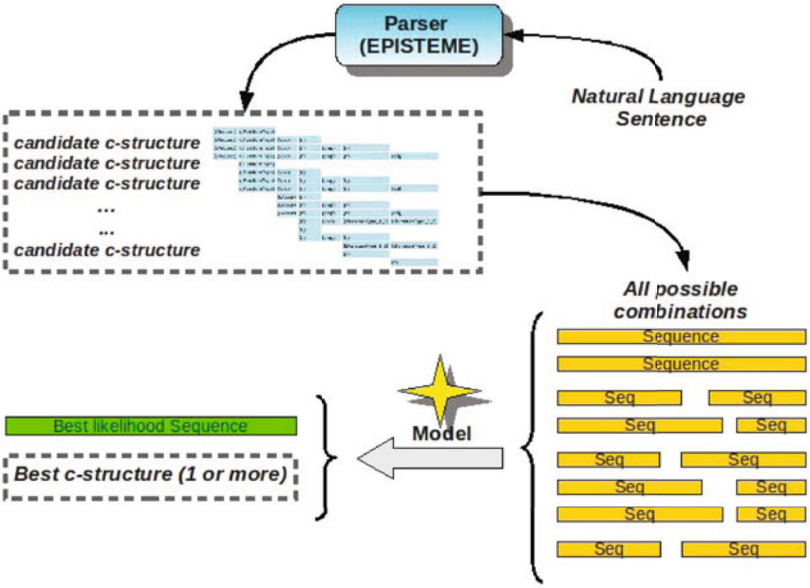


Fig. 4. Online processing

If it is known that word x can be classified only as tag_1, tag_2, tag_3 and tag_4 and this information is properly introduced into the model, it allows the model to focus the prediction on the specific subset, ignoring remaining label assignments.

Three kind of features were used:

- The root or lemma of the word.
- Word related features: the highly informative features described above. They consist of a large set of binary features indicating if the word belongs or not to a specific subdictionary. For example, if the word *beautiful* appears in the *nouns*, *adjectives* and *compliments* dictionaries, the corresponding binary features are set to true. This implies that the word can be classified as *noun*, *adjective* or as the semantic tag *compliments*.
- Sentence related features: introduced to support the diffusion of relevant pieces of information along the whole sentence. This kind of features is used to identify some potentially relevant semantic elements in the sentence. CRFs natively promote information flow through the graph, however performance improvements were experienced using this kind of features.

5 Experimental Results

As previously explained, only in-domain sentences were used to train the model, but both in-domain and out-of-domain sentences were used to test. For the out-of-domain inputs, the model is expected to choose a correct syntactic parse tree which includes all the relevant terms in the sentence. The presence of the main concepts in the syntactic tree is a key factor since the ECA will use them to search in the host web page.

To reduce risk of overfitting, a 5-fold cross validation was applied on the in-domain dataset. The in-domain dataset, consisted of 4,096 propositions and was divided into 5 subset of approximately 820 sentences each. Each time, one of the 5 subsets was used to test while the remaining 4 subsets were used to train. In this way, each sentence in the dataset was used to test exactly once and 4 times to train. The k-fold technique for performance estimation is computationally very expensive but allows to obtain a more accurate estimate of true accuracy than classical hold-out methods. Out-of-domain sentences were tested using a model trained with the whole in-domain dataset. The best results have been obtained by setting the CRF's window size to 7; the number of tokens representing context surrounding a word.

An Intel Quad Core Q9400 2.66 GHz machine with 4,096 MB of RAM was used to train the model. Training required a very long time to converge, before introduction of phrase-based features, about 90 hours were necessary to train with the whole in-domain dataset. After the introduction of this kind of features, traning time decreased to about 40 hours. During the experiments, a real risk of local minima was detected.

Performance of both rule based (baseline) and CRFs based reranking systems were evaluated in terms of accuracy, F-measure, precision and recall. The Table 1 shows the baseline rule system performance: rules perform well on in-domain sentences, while on out-of-domain sentences, the performance dramatically drops by losing 13,18% on accuracy and 8,89% on F-measure. Accuracy was calculated among the whole set of 223 categories, while precision, recall and F-measures were calculated only for those categories that occurred more than 20 times in the dataset.

Table 1. Baseline rule-based system performance

	Accuracy	F-measure	Precision	Recall
In-domain	86.59%	92.77%	95.32%	90.35%
Out-of-domain	73.41%	83.88%	85.78%	82.06%
Mixed	80.00%	88.32%	90.55%	86.21%

The CRFs based reranking performance is depicted in Table 2; CRFs perform worse than the baseline system when in-domain sentences are considered, while they perform better than the baseline system when out-of-domain sentences are considered. In this case CRFs significantly improve the baseline system by obtaining a 5.21% increase on accuracy and a 4.98% increase on F-measure.

Table 3 shows the performance evolution in relation to the training set size. Due to considerable computational costs, 5-fold cross validation was applied only for the biggest training set; the remaining were tested with a simple hold-out technique. It is worthwhile that each training set in the table isn't a new training

Table 2. CRFs-based Reranking performance

	Accuracy	F-measure	Precision	Recall
In-domain	80.64%	87.21%	90.49%	84.15%
Out-of-domain	78.62%	88.86%	89.76%	87.98%
Mixed	79.63%	88.03%	90.12%	86.07%

Table 3. Performance evolution on different training set sizes

Training set size	Mixed Accuracy	Mixed F-measure
800	51.74%	59.16%
1,756	64.43%	76.12%
2,480	72.09%	81.70%
3,600	77.09%	87.14%
4,096	79.63%	88.03%

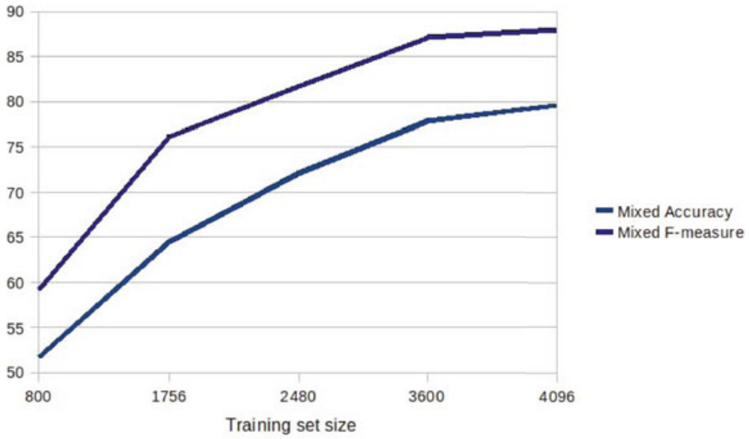


Fig. 5. Performance evolution on different training set sizes

set, but only an extended version of the previous one. Figure 5 shows how the performance improvements are slowly getting smaller as the training set size increase and is essentially stable around 4,000 sentences.

6 Conclusions

The performances achieved by the baseline system and the new proposal are quite mirrored: rule-based performance results are better for the in-domain sentences, while CRFs are better for the out-of-domain ones. The reason why CRFs outperforms the baseline system on out-of-domain sentences is mainly because they learn which terms are relevant for this particular domain, even if they are to be parsed within a syntactic tree. On the other hand, the baseline system has no semantic knowledge when trying to rerank syntactic parse trees.

The CRFs approach on its own would provide similar results to the baseline system in terms of the overall performance. However, the baseline approach is still more suitable for this particular ECAs application since, as previously explained, in-domain parsing failures are more harmful than out-of-domain ones.

But the work hereby described is not useless. Actually the results presented in the previous section clearly suggest that a combination of both approaches (rule-based for the in-domain sentence and CRFs for the out-of-domain ones) would very much increase the overall performance of the system.

Moreover, the relative importance of both approaches depends on the particular domain evaluated. In section 4.2 a division in-domain versus out-of-domain sentences of 80/20 was detailed. This percentage is very much dependant on the domain and the particular coverage of the ECA application. CRFs based approach would be more suitable for applications with higher out-of-domain input sentences percentage.

7 Future Work

The best way to make use of this approach is by combining it with the baseline rule–based one. There are two alternative approaches to accomplish this:

- Placing a filtering module before both models. This module will decide if the input sentence is an in–domain one, therefore calling the rule based model, or an out–of–domain one, calling the CRFs model.
- Calling the CRFs model always and defining a likelihood threshold *above* which, the CRF solution is discarded and the rule based model is used.

A major concern of the CRFs model described in this paper is the need of a big corpus of input sentences and the man hours needed to tag them. These elements are particularly relevant in the case where the model is to be used for real world applications. Future research directions should focus not only on performance improvement but also on these practical issues.

Acknowledgments. This work has been founded by the Spanish Ministry of Education under grant No TIN2009-14659-C03-03, and the Regional Government of Andalucia under grant No P09-TIC-5138.

References

1. Lester, J., Branting, K., Mott, B.: Conversational Agents. The Practical Handbook of Internet Computing. Chapman and Hall, Boca Raton (2004)
2. Robinson, S., Traum, D., Ittycheriah, M., Henderer, J.: What would you ask a Conversational Agent? Observations of Human-Agent Dialogues in a Museum Setting. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)

3. De Angeli, A., Johnson, G., Coventry, L.: The Unfriendly User: Exploring Social Reactions to Chatterbots. In: Proceedings of International Conference on Affective Human Factor Design, pp. 257–286 (2001)
4. Schulman, D., Bickmore, T.: Persuading users through counseling dialogue with a conversational agent. In: Proceedings of the 4th International Conference on Persuasive Technology (2009)
5. Charniak, E., Johnson, M.: Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. ACL (2005)
6. Riezler, S., King, T.H., Kaplan, R.M., Crouch, R., Maxwell, J.T., Johnson, I.M.: Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (2002)
7. Collins, M., Koo, T.: Discriminative Reranking for Natural Language Parsing. In: Computational Linguistics, pp. 175–182. Morgan Kaufmann, San Francisco (2003)
8. Ge, R., Mooney, R.J.: Discriminative Reranking for Semantic Parsing. In: Proceedings of the COLING/ACL-2006 Main Conference Poster Sessions (2006)
9. Quesada J. F., Amores J. G.: Diseño e implementacin de sistemas de traduccin automtica. In: Universidad de Sevilla, Secretariado de publicaciones (2002)
10. Amores, J.G., Quesada, J.F.: Episteme. In: Procesamiento del Lenguaje Natural (1997)
11. Bresnan, J.: The mental representation of grammatical relations. The MIT Press, Cambridge (1982)
12. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data (2001)
13. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE (1989)
14. McCallum, A.K.: Mallet: A machine learning for language toolkit (2002)