Topic-Specific Parser Design in an Air Travel Natural Language Understanding Application

Chaitanya J.K. Ekanadham[1] *and Juan M. Huerta*

IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598
huerta@us.ibm.com

Abstract

In this paper we contrast a traditional approach to semantic parsing for Natural Language Understanding applications in which a single parser captures a whole application domain, with an alternative approach consisting of a collection of smaller parsers, each able to handle only a portion of the domain. We implement this topic-specific parsing strategy by fragmenting the training corpus into subject specific subsets and developing from each subset a corresponding subject parser. We demonstrate this procedure on the Darpa Communicator task, and we observe that given an appropriate smoothing mechanism to overcome data sparseness, the set of subject-specific parsers performs as effectively (in accuracy terms) as the original parser. We present experiments both under supervised and unsupervised subject selection modes.

1. Introduction

Extraction of semantic constituents (attributes) in a Natural Language Understanding (NLU) system is the typical role of the parser/Attribute-Value (AV) extractor modules. A traditional approach to such systems consists of a configuration in which the output of a speech recognition system (for a spoken language system) is processed by a parser engine which, regardless of the state of the conversation, produces a parse tree of the speech data. This system allows the user to effectively "say anything at any time," or in other words, it doesn't restrict the domain of the parser in any state of the dialog (*i.e.* a parser will produce the same parse tree for a given sentence regardless of the point in the dialog/transaction in which the sentence was observed). The Attribute-Value pairs produced by this system are then passed to the Dialog Manager which is responsible for keeping track of the state and evolution of the dialog. This type of system employs a single parser typically trained on a large corpus. In this paper, we refer to this type of system as a Type I implementation (fig. 1).

Some systems maintain a single parser modality (Type I) but pre-annotate the output of the speech recognizer (*i.e.* data prior to parsing) with a tag reflecting the state of the dialog. While this mechanism still permits the use of a single state-independent parser, this unique set of parser models needs to be trained with data that includes these dialog state annotations. The advantage of this modality is that the state of the dialog, as reflected in the tags, might provide the parser with information that might help it to resolve ambiguities. It also has the characteristic of not restricting the span of the parser at every dialog state. Instead, the system is *biased* towards a certain parsing outcome through the dialog state tags, as opposed to forcing it into a certain interpretation. This modification of the Type I configuration produces reasonable results in task-oriented applications like the Darpa Communicator [6].

An alternative approach to parsing utilizes an additional mechanism designed to first determine the "topic" or general intention of the utterance, and based on this topic, a sub-selection of the adequate parser is made. The parser and the dialog manager then assume that the user will limit the input utterances to the pre-established topic, until at least one additional change of topic is made. This configuration is useful in applications like call routing or steering in which, after the specific determination of the subject is made, the user remains in a specific domain [2, 3]. In this paper we refer to this type of system as Type II (fig. 1).

Given the existence of alternative approaches to topic determination (*e.g.*, [7]) and the relative simplicity of the topic specific parsers compared to a complete domain parser, it seems attractive to explore the possibility of implementing a Type II system, departing from a Type I system. In this paper, we explore the issues related to such a transition. Specifically we based our experiments on the IBM Darpa Communicator system in the presence of a subsystem to determine the primary topic of a sentence.

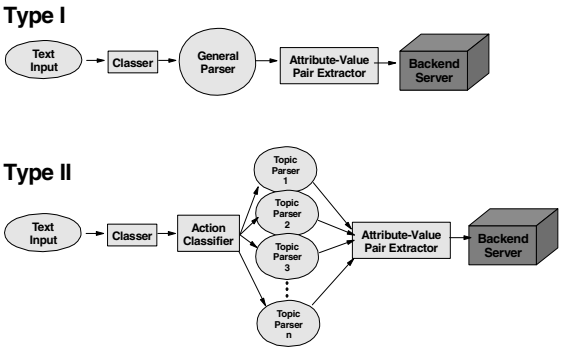

Figure 1: Schematic diagram of a Type I (single parser system) and a Type II (topic-specific parser) system.

2. The Air Travel Reservation system

We depart from the the IBM Darpa Communicator project [6], which is an NLU based flight information and booking application derived from the Air Travel Information

[1] This work was made during Mr. Ekanadham's Summer 2002 student Internship at IBM Watson Lab.

System (ATIS) project (an example of an ATIS system can be found in [8]). The basic modules making up this system are: (a) speech recognition system, (b) a word classer, whose goal is to identify tokens corresponding to semantic entities (*i.e.*, dates, quantities, names, airports), (c) a parser, that produces a semantic annotation in the form of a parse tree, (d) an attribute value extractor, (e) a dialog manager, and possibly a natural language generator. The system utilizes a single set of parser modules and tracks the state of the dialog utilizing the dialog state tags (known as feedback tags) to aid the parser in resolving ambiguities. For an in-depth description, please refer to [6].

3. Development of the topic-specific parsers and classification components

In this section we briefly describe the methods and techniques that were used to develop the components and topic clusters employed by our NLU system. We first describe the general approach to parsing used by our system.

3.1. Decision-Tree based parsing

Our parsing engine is based on statistical decision-trees [1, 4, 5]. The process of developing a decision-tree-based statistical parser mainly involves (a) feature identification (*i.e.* identifying which features in the corpus are relevant when making a decision), (b) assigning an outcome based on feature observations and (c) weighting outcomes probabilistically. The parser hypothesis is the parse tree with the highest likelihood, given the observed sentence. The probability of a parse tree *T,* given a sentence *S,* is the sum over all possible derivations δ of such tree, which in turn is the product of the probability of each active node *N* in *T* and all the feature value assignments N_x at node *N* made in that derivation:

$$P(T \mid S) = \sum_{\delta} P(T, \partial \mid S) = \sum \prod P(active = N \mid context(\partial_i)) P(N_x \mid context(\partial_i))$$

Training a decision-tree-based parser thus implies collecting an annotated corpus, defining the universe of features from which the subset of features employed by the tree is selected, and training the probabilistic models $P(N_x \mid context(\partial_i))$ which models the probability of a feature value at node *N* given its context.

A parser that spans a complete domain will include in the set of possible questions a broader array of questions than the sets available for topic-specific parsers. However, the same broad set of questions can be used in developing each topic specific-parser, because although the set of questions to choose from might contain more elements than needed, the training algorithm automatically selects the most useful questions and ignores the less useful ones.

3.2. Supervised fragmentation of the training and testing corpora

The set of topics that make up the Air Travel domain was obtained from repeated manual analysis of sentences in the training, smoothing and testing corpora. The sentence ordering, in terms of dialog flow, was not necessarily preserved in these corpora. Thus we assumed that the topic can only be inferred from each sentence's words as opposed to the context in which the sentence occurred. The following topics were deemed to be the central topics of the domain and were used to manually label the general pool of data: flight availability, price checking, flight information, flight booking, ground transportation, airline inquiries, and general information. The rest of the input sentences that did not fit precisely into any of these classifications were collected into an "unclassified" group. These sentences were mostly short and one-word sentences, including cues, such as "Yes." and "No."

After having determined the set of topics that constitute the domain, we labeled, with the aid of a rule-based script, the training, smoothing and testing corpora. We iterated on the rules of the script until the data was adequately clustered in the matching topics. It is important to clarify at this point that the rules that were developed to partition the data into topics made use of word and phrases in the data as well as manually derived parser and classer annotations. In a real-time application, a topic selection module that classifies the input of an NLU system would not be expected to have access to the parser annotations.

The partition of the corpora into the conforming topics produced subsets of the size shown in figure 2 below:

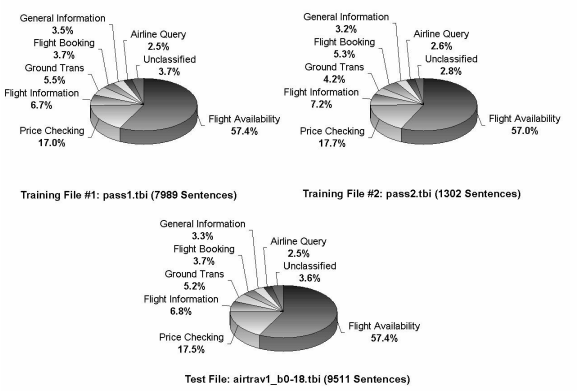

Figure 2: Composition of the training, smoothing and testing sets based on topic.

3.3. Training and smoothing of the parser models

After the training, smoothing, and testing corpora were partitioned, we trained topic-specific parser models. One of the risks involved in partitioning the development corpora is having topic data subsets with small numbers of sentences. Some models for such systems can therefore be under-trained. To avoid such problems, we mixed training data sets for the topics with little data, combining topic-specific sentences with sentences belonging to any of the other topics. To ensure that the trained parser was biased towards the desired topic we weighted by means of sentence duplication the topic-relevant data.

Each of the topic-specific parser models was smoothed through deleted interpolation [4] using held-out topic-specific data. Similar to the parser module training procedure, we mixed and weighted the topic-specific smoothing datasets with portions of the overall data in order to avoid model under-training of the interpolation parameters. We chose for each topic-specific parser the configuration that resulted in the best parsing accuracy of the corresponding test set.

3.4. Action classification

The classification of unparsed sentences is done by the action classification module. Similarly to call-steering systems, the role of this module is to assign the observed sentence to a topic or class [2, 3]. To do this, the action classification module computes the probability of a sentence belonging to a given class based on the TFIDF algorithm [7] operating on n-gram features. During classification, the algorithm computes the class likelihood for the observed sentence. The class with the highest likelihood is hypothesized as the generating class of the sentence. However, if the likelihood of the second most likely class is not smaller than a certain percent value from the hypothesis's likelihood, the sentence is flagged "*unclear,*" and a possible disambiguation or clarification mechanism might ensue.

We stress the fact that the action classifier acts on sentences that have not been parsed, and thus classify based only on the word distributions, making the classification less accurate than when parser annotations are available. Figure 3 shows an example of the classification of a sentence and the output likelihoods of each of the 8 classes. Evidently, for this example, Flight Availability has the highest likelihood and Price Checking is behind by a 20%, and the class Airline Inquiries trails by 64.45% etc. In the next section, we present results on the accuracy of the classifier on the training data.

User Input	
I want to fly to New York tomorrow .	
Action Classifier Output	
Flightavail 0.034557725775	pricecheck 0.0273149283946
Airline 0.00971049872622	flightinfo 0.00577112216645
Flightbook 0.00461688016831	unclassified 0.00182740895163
Generalinfo 0.00173280951267	gtrans 0.00110223515104
flightavail 20.96 pricecheck 64.45 airline 40.57 flightinfo	

Figure 3: The topic is listed with its corresponding likelihood. The top four matches are then given with their relative percent difference from the previous topic.

4. Experiments and results

In this section we describe the parsing experiments we performed on the topic-partitioned test set. In section 4.1, we present the parsing experiments based on supervised classification (*i.e.* when the true topic is used to choose the topic-specific parser). In section 4.2, we present the parsing experiments for the unsupervised case, including the results of the action classification module. For each topic a parser was developed as described in section 3.3. We chose the best training and smoothing configuration for each parser based on the test set accuracy. Results report the Attribute-Value pair error rate, which includes insertions and deletions.

4.1. Parsing experiments: supervised topic classification

Figure 4 below shows experimental results on topic-specific parsing when the topic is known. The figure shows the AV sentence error rate for each topic. A parsed sentence was deemed correct if the set of AV pairs extracted from it matched exactly those of the reference. If there were any insertions, deletions or substitutions of the comprising attributes, or differences in their values with respect to the original hypothesis, the sentence was counted as incorrect.

The baseline results are computed in a similar fashion, the only difference being that the parser employed for these sentences is the general parser. We can see that for most of the topics, the error rates of the topic-specific parsing scheme are better than the rates of the general parsing ones. In an actual implementation, the total system error rate would be expected to be higher, as the accuracy of the topic classification algorithm is factored in.

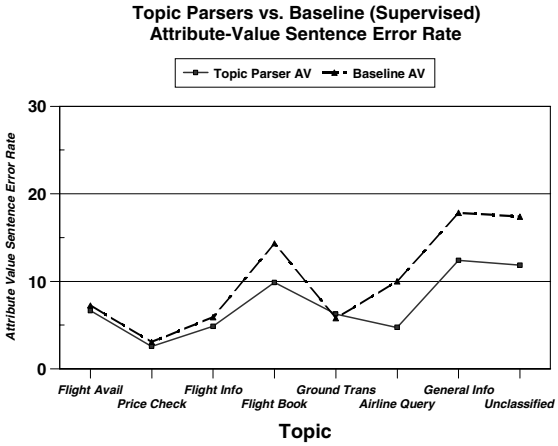

Figure 4: Sentence AV error rate on supervised classification mode for each topic.

4.2. Action Classification and Parsing experiments: unsupervised topic classification

Figure 5 below shows the accuracy of the action classifier for each specific topic data set. With the exception of the General Information and the Unclassified categories, the parser has above 90% accuracy in each topic, with most topic accuracies being above 95%.

Figure 6 below, shows experimental results for parsing in unsupervised modality. This means that the action classifier specified which parser to use for each sentence. For all the sentences in a given topic, the action classification mapped them to new categories, followed by the corresponding parser analysis. Intuitively, one would expect the accuracy to be lower than the corresponding accuracy in supervised conditions; however, as we can see, most of the topics are almost identical to the corresponding supervised results with the exception of the "unclassified" category. This seems to reflect the fact that the accuracy of the action classifier is high, and when classification errors are committed, the mismatched parser is able to produce parses accurate enough to result in correct Attribute-Value pairs. The results shown in figure 7 show the AV accuracy for each of the topics after the classification was made. In contrast to figure 6, these results are not sentence error rate, but rather token error rate.

4.3. Implementation of a "type-to" system

We implemented the system with topic-specific parsers into a "type-to" system that simulated the NLU system but replaced the ASR component for a keyboard. We departed from the full "type-to" Darpa Communication System and replaced the parser with the topic-specific ones. We then included the action classifier, and the most likely action was then used to select the parser. Anecdotically, our system performed with as good accuracy as the original one, but because of the comparative size of the parsers, there was a perceivable increase in system response time due to the substitution of the parsers.

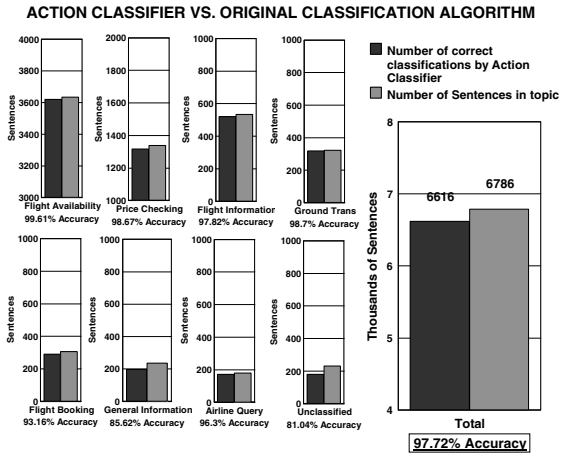

Figure 5: Analysis of the accuracy of the action classification module for each of the topics.

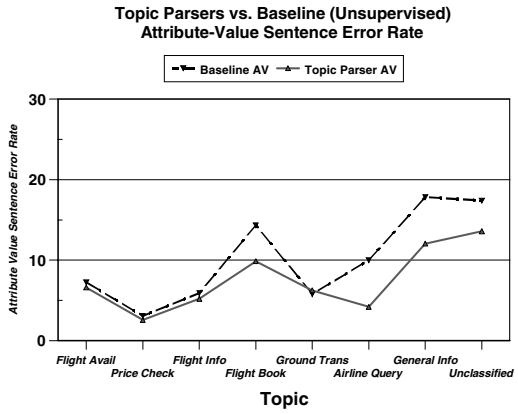

Figure 6: Sentence AV error rate on unsupervised classification mode for each topic

5. Conclusions

In this paper we showed that without loss in parser accuracy it is possible to fragment a single large parser into smaller topic specific parsers. Even after factoring the effect in accuracy due to action classification, the accuracy obtained by these subject specific parsers is essentially as good as the baseline system. In order to reduce the effect that data fragmentation might have in accuracy, we proposed the smoothing of the data by weighting the comprising corpora. We have demonstrated the incorporation of the new parsers into a "type-to" system, which essentially replicates the functionality of the telephony-based original IBM Darpa Communicator system with improved system response time.

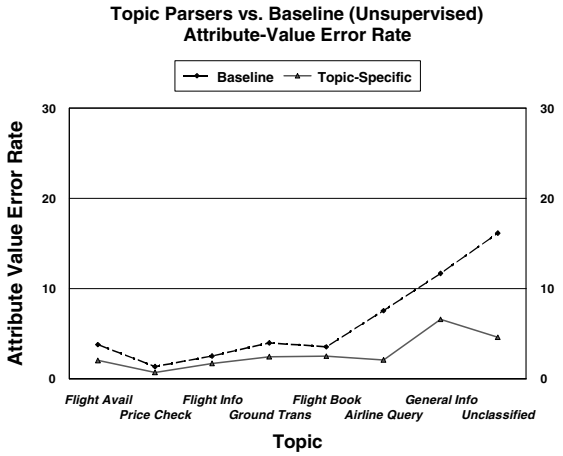

Figure 7: Token AV error rate on supervised classification mode for each topic.

6. Acknowledgements

The authors would like to thank Dr. Rajesh Balchandran for providing the Action Classifier module.

7. References

[1] Bahl, Brown, de Souza, Mercer "A tree-based statistical language model for natural language speech recognition" IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(7).

[2] Balachandran, Rajesh. Personal Communication

[3] Chou, W., Zhou, Q., Kuo, H.-K. J., Saad, A., Attwater, D., Durston, P., Farrell, M., Scahill, F., "Natural Language Call Steering for Service Applications," in Proceedings of the International Conference on Spoken Language Processing, Beijing, China, Oct. 2000

[4] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press, 1997.

[5] Magerman, David M., "Statistical Decision-Tree Models for Parsing," Proceedings of the ACL Conference, 1995.

[6] Luo, X. and Papineni, K. "IBM DARPA Communicator v1.0," DARPA Communicator Principle Investigators Meeting, Philadelphia, PA, USA, 2000.

[7] G. Salton, 1989, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, 1989

[8] Ward, W. and Issar, S., "The CMU ATIS system." Proceedings of the Human Language Technologies Workshop, Austin, TX, USA, January 1995.