Balancing Data-driven and Rule-based Approaches in the Context of a Multimodal Conversational System

Srinivas Bangalore
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932
srini@research.att.com

Michael Johnston
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932
johnston@research.att.com

Abstract

Moderate-sized rule-based spoken language models for recognition and understanding are easy to develop and provide the ability to rapidly prototype conversational applications. However, scalability of such systems is a bottleneck due to the heavy cost of authoring and maintenance of rule sets and inevitable brittleness due to lack of coverage in the rule sets. In contrast, data-driven approaches are robust and the procedure for model building is usually simple. However, the lack of data in a particular application domain limits the ability to build data-driven models. In this paper, we address the issue of combining data-driven and grammar-based models for rapid prototyping of robust speech recognition and understanding models for a multimodal conversational system. We also present methods that reuse data from different domains and investigate the limits of such models in the context of a particular application domain.

1 Introduction

In the past four decades of speech and natural language processing, both data-driven approaches and rule-based approaches have been prominent at different periods in time. In the recent past, rule-based approaches have fallen into disfavor due to their brittleness and the significant cost of authoring and maintaining complex rule sets. Data-driven approaches are robust and provide a simple process of developing applications given the data from the application domain. However, the reliance on domain-specific data is also one of the significant bottlenecks of data-driven approaches. Development of a conversational system using data-driven approaches cannot proceed until data pertaining to the application domain is available. The collection and annotation of such data is extremely time-consuming and tedious, which is aggravated by the presence of multiple modalities in the user's input, as in our case. Also, extending an existing application to support an additional feature requires adding additional data sets with that feature.

In this paper, we explore various methods for combining rule-based and in-domain data for rapid prototyping of speech recognition and understanding models that are robust to ill-formed or unexpected input in the context of a multimodal conversational system. We also investigate approaches to reuse out-of-domain data and compare their performance against the performance of in-domain data-driven models.

We investigate these issues in the context of a multimodal application designed to provide an interactive city guide: MATCH. In Section 2, we present the MATCH application, the architecture of the system and the apparatus for multimodal understanding. In Section 3, we discuss various approaches to rapid prototyping of the language model for the speech recognizer and in Section 4 we present two approaches to robust multimodal understanding. Section 5 presents the results for speech recognition and multimodal understanding using the different approaches we consider.

2 The MATCH application

MATCH (Multimodal Access To City Help) is a working city guide and navigation system that enables mobile users to access restaurant and subway information for New York City (NYC) (Johnston et al., 2002b; Johnston et al., 2002a). The user interacts with a graphical interface displaying restaurant listings and a dynamic map showing locations and street information. The inputs can be speech, drawing on the display with a stylus, or synchronous multimodal combinations of the two modes. The user can ask for the review, cuisine, phone number, address, or other information about restaurants and subway directions to locations. The system responds with graphical callouts on the display, synchronized with synthetic speech output. For example, if the user says *phone numbers for these two restaurants* and circles two restaurants as in Figure 1 [a], the system will draw a callout with the restaurant name and number and say, for example *Time Cafe can be reached at 212-533-7000*, for each restaurant in turn (Figure 1 [b]). If the immediate environment is too noisy or public, the same command can be given completely in pen by circling the restaurants and writing *phone*.

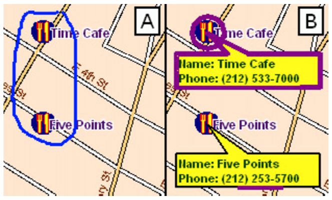

Figure 1: Two area gestures

2.1 MATCH Multimodal Architecture

The underlying architecture that supports MATCH consists of a series of re-usable components which communicate over sockets through a facilitator (MCUBE) (Figure 2). Users interact with the system through a Multimodal User Interface Client (MUI). Their speech and ink are processed by speech recognition (Sharp et al., 1997) (ASR) and handwriting/gesture recognition (GESTURE, HW RECO) components respectively. These recognition processes result in lattices of potential words and gestures. These are then combined and assigned a meaning representation using a multimodal finite-state device (MMFST) (Johnston and Bangalore, 2000; Johnston et al., 2002b). This provides as output a lattice encoding all of the potential meaning representations assigned to the user inputs. This lattice is flattened to an N-best list and passed to a multimodal dialog manager (MDM) (Johnston et al., 2002b), which re-ranks them in accordance with the current dialogue state. If additional information or confirmation is required, the MDM enters into a short information gathering dialogue with the user. Once a command or query is complete, it is passed to the multimodal generation component (MMGEN), which builds a multimodal *score* indicating a coordinated sequence of graphical actions and TTS prompts. This score is passed back to the Multimodal UI (MUI). The Multimodal UI coordinates presentation of graphical content with synthetic speech output using the AT&T Natural Voices TTS engine (Beutnagel et al., 1999). The subway route constraint solver (SUBWAY) identifies the best route between any two points in New York City.

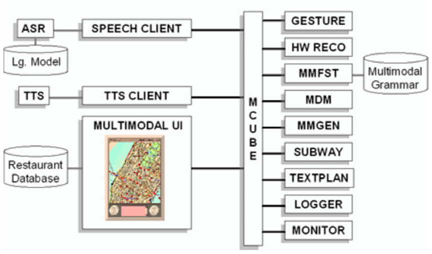

Figure 2: Multimodal Architecture

2.2 Multimodal Integration and Understanding

Our approach to integrating and interpreting multimodal inputs (Johnston et al., 2002b; Johnston et al., 2002a) is an extension of the finite-state approach previously proposed (Bangalore and Johnston, 2000; Johnston and Bangalore, 2000). In this approach, a declarative multimodal grammar captures both the structure and the interpretation of multimodal and unimodal commands. The grammar consists of a set of context-free rules. The multimodal aspects of the grammar become apparent in the terminals, each of which is a triple *W:G:M*, consisting of speech (words, *W*), gesture (gesture symbols, *G*), and meaning (meaning symbols, *M*). The multimodal grammar encodes not just multimodal integration patterns but also the syntax of speech and gesture, and the assignment of meaning. The meaning is represented in XML, facilitating parsing and logging by other system components. The symbol *SEM* is used to abstract over specific content such as the set of points delimiting an area or the identifiers of selected objects. In Figure 3, we present a small simplified fragment from the MATCH application capable of handling information seeking commands such as *phone for these three restaurants*. The epsilon symbol (ϵ) indicates that a stream is empty in a given terminal.

CMD	→	ϵ:ϵ:<cmd> INFO ϵ:ϵ:</cmd>
INFO	→	ϵ:ϵ:<type> TYPE ϵ:ϵ:</type>
		for:ϵ:ϵ ϵ:ϵ:<obj> DEICNP ϵ:ϵ:</obj>
TYPE	→	phone:ϵ:phone | review:ϵ:review
DEICNP	→	DDETPL ϵ:area:ϵ ϵ:sel:ϵ NUM HEADPL
DDETPL	→	these:G:ϵ | those:G:ϵ
HEADPL	→	restaurants:rest:<rest> *SEM*:*SEM*:ϵ ϵ:ϵ:</rest>
NUM	→	two:2:ϵ | three:3:ϵ ... ten:10:ϵ

Figure 3: Multimodal grammar fragment

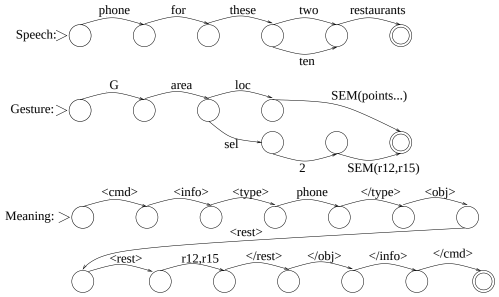

Figure 4: Multimodal Example

In the example above where the user says *phone for these two restaurants* while circling two restaurants (Figure 1 [a]), assume the speech recognizer returns the lattice in Figure 4 (Speech). The gesture recognition component also returns a lattice (Figure 4, Gesture) indicating that the user's ink is either a selection of two restaurants or a geographical area. The multimodal grammar (Figure 3) expresses the relationship between what the user said, what they drew with the pen, and their combined meaning, in this case Figure 4 (Meaning). The meaning is generated by concatenating the meaning symbols and replacing *SEM* with the appropriate specific content: *<cmd> <info> <type>* phone *</type> <obj> <rest> [r12,r15] </rest> </obj> </info> </cmd>*. For the purpose of evaluation of concept accuracy, we developed an approach similar to (Boros et al., 1996) in which computing concept accuracy is reduced to comparing strings representing core contentful concepts. We extract a sorted flat list of attribute value pairs that represents the core contentful concepts of each command from the XML output. The example above yields the following meaning representation for concept accuracy.

$$\texttt{cmd}:\texttt{info type}:\texttt{phone object}:\texttt{selection} \quad (1)$$

The multimodal grammar can be used to create language models for ASR, align the speech and gesture results from the respective recognizers and transform the multimodal utterance to a meaning representation. All these operations are achieved using finite-state transducer operations (See (Bangalore and Johnston, 2000; Johnston and Bangalore, 2000) for details). However, this approach to recognition needs to be more robust to extra-grammaticality and language variation in user's utterances and the interpretation needs to be more robust to speech recognition errors. We address these issues in the rest of the paper.

3 Bootstrapping Corpora for Language Models

The problem of speech recognition can be succinctly represented as a search for the most likely word sequence (w) through the network created by the composition of a language of acoustic observations (O), an acoustic model which is a transduction from acoustic observations to phone sequences (A), a pronounciation model which is a transduction from phone sequences to word sequences (L), and a language model acceptor (G) (Pereira and Riley, 1997). The language model acceptor encodes the (weighted) word sequences permitted in an application.

$$\underset{w}{argmax}\ \pi_2(O \circ A \circ L \circ G)(w) \quad (2)$$

Typically, G is built using either a hand-crafted grammar or using a statistical language model derived from a corpus of sentences from the application domain. While a grammar could be written so as to be easily portable across applications, it suffers from being too prescriptive and has no metric for relative likelihood of users' utterances. In contrast, in the data-driven approach a weighted grammar is automatically induced from a corpus and the weights can be interpreted as a measure for relative likelihood of users' utterances. However, the reliance on a domain-specific corpus is one of the significant bottlenecks of data-driven approaches, since collecting a corpus specific to a domain is an expensive and time-consuming task.

In this section, we investigate a range of techniques for producing a domain-specific corpus using resources such as a domain-specific grammar as well as an out-of-domain corpus. We refer to the corpus resulting from such techniques as a *domain-specific derived corpus* in contrast to a *domain-specific collected corpus*. The idea is that the derived domain-specific corpus would obviate the need for in-domain corpus collection. In particular, we are interested in techniques that would result in corpora such that the performance of language models trained on these corpora would rival the performance of models trained on corpora collected specifically for a specific domain. We investigate these techniques in the context of MATCH.

We use the notation C_d for the corpus, λ_d for the language model built using the corpus C_d, and G_{λ_d} for the language model acceptor representation of the model λ_d, which can be used in Equation 2 above.

3.1 Language Model using in-domain corpus

In order to evaluate the MATCH system, we collected a corpus of multimodal utterances for the MATCH domain in a laboratory setting from a set of sixteen first time users (8 male, 8 female). We use this corpus to establish a point of reference to compare the models trained on derived corpora against models trained on an in-domain corpus. A total of 833 user interactions (218 multimodal / 491 speech-only / 124 pen-only) resulting from six sample task scenarios involving finding restaurants of various types and getting their names, phones, addresses, or reviews, and getting subway directions between locations were collected and annotated. The data collected was conversational speech where the users gestured and spoke freely. We built a class-based trigram language model (λ_{MATCH}) using the 709 multimodal and speech-only utterances as the corpus (C_{MATCH}). The performance of this model serves as the point of reference to compare the performance of language models trained on derived corpora.

3.2 Grammar as Language Model

The multimodal CFG (a fragment is presented in Section 2) encodes the repertoire of language and gesture commands allowed by the system and their combined interpretations. The CFG can be approximated by an FSM with arcs labeled with language, gesture and meaning symbols, using well-known compilation techniques (Nederhof, 1997). The resulting FSM can be projected on the language component and can be used as the language model acceptor (G_{gram}) for speech recognition. Note that the resulting language model acceptor is unweighted if the grammar is unweighted and suffers from not being robust to language variations in user's input. However, due to the tight coupling of the grammar used for recognition and interpretion, every recognized string can be assigned an interpretation (though it may not necessarily be the intended interpretation).

3.3 Grammar-based N-gram Language Model

As mentioned earlier, a hand-crafted grammar typically suffers from the problem of being too restrictive and inadequate to cover the variations and extra-grammaticality of user's input. In contrast, an N-gram language model derives its robustness by permitting all strings over an alphabet, albeit with different likelihoods. In an attempt to provide robustness to the grammar-based model, we created a corpus (C_{gram}) of k sentences by randomly sampling the set of paths of the grammar ($L(M)$) and built a class-based N-gram language model(λ_{gram}) using this corpus. Although this corpus might not represent the true distribution of sentences in the MATCH domain, we are able to derive some of the benefits of N-gram language modeling techniques. This technique is similar to Galescu et.al (1998).

3.4 Combining Grammar and Corpus

A straightforward extension of the idea of sampling the grammar in order to create a corpus is to select those sentences out of the grammar which make the resulting corpus "similar" to the corpus collected in the pilot studies. In order to create this corpus, we choose the k most likely sentences as determined by a language model (λ_{MATCH}) built using the collected corpus. A mixture model (λ_{mix}) with mixture weight (α) is built by interpolating the model trained on the corpus of extracted sentences (λ_{close}) and the model trained on the collected corpus (λ_{MATCH}).

$$\begin{aligned} C_{close} &= \{S_1, \ldots S_k | S_i \in L(M) \qquad (3) \\ &\quad S_i \; ordered \; by \; Pr_{\lambda_{MATCH}}(S_i)\} \\ \lambda_{mix} &= \alpha * \lambda_{close} + (1-\alpha) * \lambda_{MATCH} \qquad (4) \end{aligned}$$

3.5 Class-based Out-of-domain Language Model

An alternative to using in-domain corpora for building language models is to "migrate" a corpus of a different domain to the MATCH domain. The process of migrating a corpus involves suitably generalizing the corpus to remove information specific only to the out-of-domain and instantiating the generalized corpus to the MATCH domain. Although there are a number of ways of generalizing the out-of-domain corpus, the generalization we have investigated involved identifying linguistic units, such as noun and verb chunks in the out-of-domain corpus and treating them as classes. These classes are then instantiated to the corresponding linguistic units from the MATCH domain. The identification of the linguistic units in the out-of-domain corpus is done automatically using a supertagger (Bangalore and Joshi, 1999). We use a corpus collected in the context of a software helpdesk application as an example out-of-domain corpus. In cases where the out-of-domain corpus is closely related to the domain at hand, a more semantically driven generalization might be more suitable.

3.6 Adapting the SwitchBoard Language Model

We investigate the performance of a large vocabulary conversational speech recognition system when applied to a specific domain such as MATCH. We used the Switchboard corpus (C_{swbd}) as an example of a large vocabulary conversational speech corpus. We built a trigram model (λ_{swbd}) using the 5.4 million word corpus and investigated the effect of adapting the Switchboard language model given k in-domain untranscribed speech utterances ($\{O^i_M\}$). The adaptation is done by first recognizing the in-domain speech utterances and then building a language model (λ_{adapt}) from the corpus of recognized text (C_{adapt}). This bootstrapping mechanism can be used to derive an domain-specific corpus and language model without any transcriptions. Similar techniques for unsupervised language model adaptation are presented in (Bacchiani and Roark, 2003; Souvignier and Kellner, 1998).

$$\begin{aligned} C_{adapt} &= \{S_1, S_2, \ldots, S_k\} \qquad (5) \\ S_i &= \underset{S}{argmax} \; \pi_2(O^i_M \circ A \circ L \circ G_{swbd})(S) \end{aligned}$$

3.7 Adapting a wide-coverage grammar

There have been a number of computational implementations of wide-coverage, domain-independent, syntactic grammars for English in various formalisms (XTAG, 2001; Clark and Hockenmaier, 2002; Flickinger et al., 2000). Here, we describe a method that exploits one such grammar implementation in the Lexicalized Tree-Adjoining Grammar (LTAG) formalism, for deriving domain-specific corpora. An LTAG consists of a set of elementary trees (Supertags) (Bangalore and Joshi, 1999) each associated with a lexical item. The set of sentences generated by an LTAG can be obtained by combining supertags using substitution and adjunction operations. In related work (Rambow et al., 2002), it has been shown that for a restricted version of LTAG, the combinations of a set of supertags can be represented as an FSM. This FSM compactly encodes the set of sentences generated by an LTAG grammar.

We derive a domain-specific corpus by constructing a lexicon consisting of pairings of words with their supertags that are relevant to that domain. We then compile the grammar to build an FSM of all sentences upto a given length. We sample this FSM and build a language model as discussed in Section 3.3. Given untranscribed utterances from a specific domain, we can also adapt the language model as discussed in Section 3.6.

4 Robust Multimodal Understanding

The grammar-based interpreter uses composition operation on FSTs to transduce multimodal strings (gesture,speech) to an interpretation. The set of speech strings that can be assigned an interpretation are exactly those that are represented in the grammar. It is to be expected that the accuracy of meaning representation will be reasonable, if the user's input matches one of the multimodal strings encoded in the grammar. But for those user inputs that are not encoded in the grammar, the system will not return a meaning representation. In order to improve the usability of the system, we expect it to produce a (partial) meaning representation, irrespective of the grammaticality of the user's input and the coverage limitations of the grammar. It is this aspect that we refer to as robustness in understanding. We present below two approaches to robust multimodal understanding that we have developed.

4.1 Pattern Matching Approach

In order to overcome the possible mismatch between the user's input and the language encoded in the multimodal grammar (λ_g), we use an edit-distance based pattern matching algorithm to coerce the set of strings ($\mathcal{S}$) encoded in the lattice resulting from ASR ($\lambda_{\mathcal{S}}$) to match one of the strings that can be assigned an interpretation. The edit operations (insertion, deletion, substitution) can either be word-based or phone-based and are associated with a cost. These costs can be tuned based on the word/phone confusions present in the domain. The edit operations are encoded as an transducer (λ_{edit}) as shown in Figure 5 and can apply to both one-best and lattice output of the recognizer. We are interested in the string with the least number of edits ($argmin$) that can be assigned an interpretation by the grammar. This can be achieved by composition ($\circ$) of transducers followed by a search for the least cost path through a weighted transducer as shown below.

$$s^* = \underset{s \in \mathcal{S}}{argmin}\ \lambda_{\mathcal{S}} \circ \lambda_{edit} \circ \lambda_g \tag{6}$$

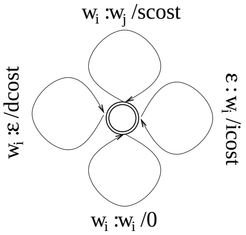

Figure 5: Edit transducer with insertion, deletion, substitution and identity arcs. w_i and w_j could be words or phones. The costs on the arcs are set up such that `scost < icost + dcost`.

This approach is akin to example-based techniques used in other areas of NLP such as machine translation. In our case, the set of examples (encoded by the grammar) is represented as a finite-state machine.

4.2 Classification-based Approach

A second approach is to view robust multimodal understanding as a sequence of classification problems in order to determine the *predicate* and *arguments* of an utterance. The meaning representation shown in (1) consists of an predicate (the command attribute) and a sequence of one or more argument attributes which are the parameters for the successful interpretation of the user's intent. For example, in (1), `cmd` : `info` is the predicate and `type` : `phone` `object` : `selection` is the set of arguments to the predicate.

We determine the predicate (c^*) for a N token multimodal utterance (S_1^N) by maximizing the posterior probability as shown in Equation 7.

$$c^* \;=\; \underset{c}{argmax}\ Pr(c \mid S_1^N) \tag{7}$$

We view the problem of identifying and extracting arguments from a multimodal input as a problem of associating each token of the input with a specific tag that encodes the label of the argument and the span of the argument. These tags are drawn from a tagset which is constructed by extending each argument label by three additional symbols I, O, B, following (Ramshaw and Marcus, 1995). These symbols correspond to cases when a token is inside (I) an argument span, outside (O) an argument span or at the boundary of two argument spans (B) (See Table 1).

Given this encoding, the problem of extracting the arguments is a search for the most likely sequence of tags (T^*) given the input multimodal utterance S_1^N as shown in Equation (8). We approximate the posterior probability $Pr(T \mid S_1^N)$ using independence assumptions as shown in Equation (9).

User Utterance	cheap thai upper west side
Argument Annotation	<price> cheap < /price> <cuisine> thai </cuisine> <place> upper west side </place>
IOB Encoding	cheap_price<B> thai_cuisine<B> upper_place<I> west_place<I> side_place<I>

Table 1: The $\{$I,O,B$\}$ encoding for argument extraction.

$$T^* = \underset{T}{argmax}\ Pr(T \mid S_1^N) \tag{8}$$

$$\approx \underset{T}{argmax} \prod_i Pr(t_i \mid S_{i-n}^i, S_{i+1}^{i+n+1}, t_{i-1}, t_{i-2}) \tag{9}$$

Owing to the large set of features that are used for predicate identification and argument extraction, we estimate the probabilities using a classification model. In particular, we use the Adaboost classifier (Freund and Schapire, 1996) wherein a highly accurate classifier is build by combining many "weak" or "simple" base classifiers f_i, each of which may only be moderately accurate. The selection of the weak classifiers proceeds iteratively picking the weak classifier that correctly classifies the examples that are misclassified by the previously selected weak classifiers. Each weak classifier is associated with a weight (w_i) that reflects its contribution towards minimizing the classification error. The posterior probability of $Pr(c \mid x)$ is computed as in Equation 10.

$$Pr(c \mid x) = \frac{1}{(1 + e^{-2*\sum_i w_i * f_i(x)})} \tag{10}$$

It should be noted that the data for training the classifiers can be collected from the domain or derived from an in-domain grammar using techniques similar to those presented in Section 3.

5 Experiments and Results

We describe a set of experiments to evaluate the performance of the speech recognizer and the concept accuracy of speech only and speech and gesture exchanges in our MATCH multimodal system. We use word accuracy and string accuracy for evaluating ASR output. All results presented in this section are based on 10-fold cross-validation experiments run on the 709 spoken and multimodal exchanges collected from the pilot study described in Section 3.1.

5.1 Language Model

Table 2 presents the performance results for ASR word and sentence accuracy using language models trained on collected in-domain corpus as well as on corpora derived using the different methods discussed in Section 3. For the class-based models mentioned in the table, we defined different classes based on areas of interest (eg. riverside park, turtle pond), points of interest (eg. Ellis Island, United Nations Building), type of cuisine (eg. Afghani,

Scenario		ASR Word Accuracy	Sentence Accuracy
Grammar Based	Grammar as Language Model	41.6	38.0
	Class-based N-gram Language Model	60.6	42.9
In-domain Data	Class-based N-gram Model	73.8	57.1
Grammar+In-domain Data	Class-based N-gram Model	**75.0**	59.5
Out-of-domain	N-gram Model	17.6	17.5
	Class-based N-gram Model	58.4	38.8
	Class-based N-gram Model with Grammar-based N-gram Language Model	**64.0**	45.4
SwitchBoard	N-gram Model	43.5	25.0
	Language model trained on recognized in-domain data	55.7	36.3
Wide-coverage Grammar	N-gram Model	43.7	24.8
	Language model trained on recognized in-domain data	55.8	36.2

Table 2: Performance results for ASR Word and Sentence accuracy using models trained on data derived from different methods of bootstrapping domain-specific data.

Indonesian), price categories (eg. moderately priced, expensive), and neighborhoods (eg. Upper East Side, Chinatown).

It is immediately apparent that the hand-crafted grammar as language model performs poorly and a language model trained on the collected domain-specific corpus performs significantly better than models trained on derived data. However, it is encouraging to note that a model trained on a derived corpus (obtained from combining migrated out-of-domain corpus and a corpus created by sampling in-domain grammar) is within 10% word accuracy as compared to the model trained on the collected corpus. There are several other noteworthy observations from these experiments.

The performance of the language model trained on data sampled from the grammar is dramatically better as compared to the performance of the hand-crafted grammar. This technique provides a promising direction for authoring portable grammars that can be sampled subsequently to build robust language models when no in-domain corpora are available. Furthermore, combining grammar and in-domain data as described in Section 3.4, outperforms all other models significantly.

For the experiment on migration of out-of-domain corpus, we used a corpus from a software helpdesk application. Table 2 shows that the migration of data using linguistic units as described in Section 3.5 significantly outperforms a model trained only on the out-of-domain corpus. Also, combining the grammar sampled corpus with the migrated corpus provides a further improvement.

The performance of the SwitchBoard model on the MATCH domain is presented in Table 2. We built a trigram model using a 5.4 million word SwitchBoard corpus and investigated the effect of adapting the resulting language model on in-domain untranscribed speech utterances. The adaptation is done by first recognizing the training partition of the in-domain speech utterances and then building a language model from the recognized text. We observe that although the performance of the SwitchBoard language model on the MATCH domain is poorer than the performance of a model obtained by migrating data from a related domain, the performance can be significantly improved using the adaptation technique.

The last row of Table 2 shows the results of using the MATCH specific lexicon to generate a corpus using a wide-coverage grammar, training a language model and adapting the resulting model using in-domain untranscribed speech utterances as was done for the SwitchBoard model. The class-based trigram model was built using 500,000 randomly sampled paths from the network constructed by the procedure described in Section 3.7.

5.2 Multimodal Understanding

In this section, we present results on multimodal understanding using the two techniques presented in Section 4. We use concept token accuracy and concept string accuracy as evaluation metrics for the entire meaning representation in these experiments. These metrics correspond to the word accuracy and string accuracy metrics used for ASR evaluation. In order to provide a finer-grained evaluation, we breakdown the concept accuracy in terms of the accuracy of identifying the predicates and arguments. Again, we use *string accuracy* metrics to evaluate predicate and argument accuracy. We use the output of the ASR with the language model trained on the collected data (word accuracy of 73.8%) as the input to the understanding component.

The grammar-based multimodal understanding system composes the input multimodal string with the multimodal grammar represented as an FST to produce an interpretation. Thus an interpretation can be assigned to only those multimodal strings that are encoded in the grammar. However, the result of ASR and gesture recognition may not be one of the strings encoded in the grammar, and such strings are not assigned an interpretation. This fact is reflected in the low concept string accuracy

	Predicate String	Argument String	Concept Token	Concept String
	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)
Baseline	65.2	52.1	53.5	45.2
Word-based Pattern-Matching	73.7	62.4	68.1	59.0
Phone-based Pattern-Matching	73.7	63.8	67.8	61.3
Classification-based	84.1	59.1	73.5	56.4

Table 3: Performance results of robust multimodal understanding

for the baseline as shown in Table 3.

The pattern-matching based robust understanding approach mediates the mismatch between the strings that are output by ASR and the strings that can be assigned an interpretation. We experimented with word based pattern matching as well as phone based pattern matching on the one-best output of the recognizer. As shown in Table 3, the pattern-matching robust understanding approach improves the concept accuracy over the baseline significantly. Furthermore, the phone-based matching method has a similar performace to the word-based matching method.

For the classification-based approach to robust understanding we used a total of 10 predicates such as *help, assert, inforequest*, and 20 argument types such as *cuisine, price, location* . We use unigrams, bigrams and trigrams appearing in the multimodal utterance as weak classifiers for the purpose of predicate classification. In order to predict the tag of a word for argument extraction, we use the left and right trigram context and the tags for the preceding two tokens as weak classifiers. The results are presented in Table 3.

Both the approaches to robust understanding outperform the baseline model significantly. However, it is interesting to note that while the pattern-matching based approach has a better argument extraction accuracy, the classification based approach has a better predicate identification accuracy. Two possible reasons for this are: first, argument extraction requires more non-local information that is available in the pattern-matching based approach while the classification-based approach relies on local information and is more conducive for identifying the simple predicates in MATCH. Second, the pattern-matching approach uses the entire grammar as a model for matching while the classification approach is trained on the training data which is significantly smaller when compared to the number of examples encoded in the grammar.

6 Discussion

Although we are not aware of any attempts to address the issue of robust understanding in the context of multimodal systems, this issue has been of great interest in the context of speech-only conversational systems (Dowding et al., 1993; Seneff, 1992; Allen et al., 2000; Lavie, 1996). The output of the recognizer in these systems usually is parsed using a handcrafted grammar that assigns a meaning representation suited for the downstream dialog component. The coverage problems of the grammar and parsing of extra-grammatical utterances is typically addressed by retrieving fragments from the parse chart and incorporating operations that combine fragments to derive a meaning of the recognized utterance. We have presented an approach that achieves robust multimodal utterance understanding using the edit-distance automaton in a finite-state-based interpreter without the need for combining fragments from a parser.

The issue of combining rule-based and data-driven approaches has received less attention, with the exception of a few (Wang et al., 2000; Rayner and Hockey, 2003; Wang and Acero, 2003). In a recent paper (Rayner and Hockey, 2003), the authors address this issue by employing a decision-list-based speech understanding system as a means of progressing from rule-based models to data-driven models when data becomes available. The decision-list-based understanding system also provides a method for robust understanding. In contrast, the approach presented in this paper can be used on lattices of speech and gestures to produce a lattice of meaning representations.

7 Conclusion

In this paper, we have addressed how to rapidly prototype multimodal conversational systems without relying on the collection of domain-specific corpora. We have presented several techniques that exploit domain-specific grammars, reuse out-of-domain corpora and adapt large conversational corpora and wide-coverage grammars to *derive* a domain-specific corpus. We have demonstrated that a language model trained on a derived corpus performs within 10% word accuracy of a language model trained on collected domain-specific corpus, suggesting a method of building an initial language model without having to collect domain-specific corpora. We have also presented and evaluated pattern-matching and classification-based approaches to improve the robustness of multimodal understanding. We have presented results for these approaches in the context of a multimodal city guide application (MATCH).

8 Acknowledgments

We thank Patrick Ehlen, Amanda Stent, Helen Hastie, Candy Kamm, Marilyn Walker, and Steve Whittaker for their contributions to the MATCH system. We also thank Allen Gorin, Mazin Rahim, Giuseppe Riccardi, and Juergen Schroeter for their comments on earlier versions of this paper.

References

J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. 2000. An architecture for a generic dialogue shell. *JNLE*, 6(3).

M. Bacchiani and B. Roark. 2003. Unsupervised language model adaptation. In *In Proc. Int. Conf. Acoustic,Speech,Signal Processing*.

S. Bangalore and M. Johnston. 2000. Tight-coupling of multimodal language processing with speech recognition. In *Proceedings of ICSLP*, Beijing, China.

S. Bangalore and A. K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2).

M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal. 1999. The AT&T next-generation TTS. In *In Joint Meeting of ASA; EAA and DAGA*.

M. Boros, W. Eckert, F. Gallwitz, G. Gŏrz, G. Hanrieder, and H. Niemann. 1996. Towards Understanding Spontaneous Speech: Word Accuracy vs. Concept Accuracy. In *Proceedings of ICSLP*, Philadelphia.

Stephen Clark and Julia Hockenmaier. 2002. Evaluating a wide-coverage CCG parser. In *Proceedings of the LREC 2002 Beyond Parseval Workshop*, Las Palmas, Spain.

J. Dowding, J. M. Gawron, D. E. Appelt, J. Bear, L. Cherny, R. Moore, and D. B. Moran. 1993. GEMINI: A natural language system for spoken-language understanding. In *Proceedings of ACL*, pages 54–61.

D. Flickinger, A. Copestake, and I. Sag. 2000. Hpsg analysis of english. In W. Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 254–263. Springer–Verlag, Berlin, Heidelberg, New York.

Y. Freund and R. E. Schapire. 1996. Experiments with a new boosting alogrithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156.

L. Galescu, E. K. Ringger, and J. F. Allen. 1998. Rapid language model development for new task domains. In *Proceedings of the ELRA First International Conference on Language Resources and Evaluation (LREC), Granada, Spain*.

M. Johnston and S. Bangalore. 2000. Finite-state multimodal parsing and understanding. In *Proceedings of COLING*, Saarbrücken, Germany.

M. Johnston, S. Bangalore, A. Stent, G. Vasireddy, and P. Ehlen. 2002a. Multimodal language processing for mobile information access. In *In Proceedings of ICSLP*, Denver, CO.

M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. 2002b. MATCH: An architecture for multimodal dialog systems. In *Proceedings of ACL*, Philadelphia.

A. Lavie. 1996. *GLR*: A Robust Grammar-Focused Parser for Spontaneously Spoken Language*. Ph.D. thesis, Carnegie Mellon University.

M-J. Nederhof. 1997. Regular approximations of CFLs: A grammatical view. In *Proceedings of the International Workshop on Parsing Technology*, Boston.

Fernando C.N. Pereira and Michael D. Riley. 1997. Speech recognition by composition of weighted finite automata. In E. Roche and Schabes Y., editors, *Finite State Devices for Natural Language Processing*, pages 431–456. MIT Press, Cambridge, Massachusetts.

Owen Rambow, Srinivas Bangalore, Tahir Butt, Alexis Nasr, and Richard Sproat. 2002. Creating a finite-state parser with application semantics. In *In Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.

Lance Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, MIT, Cambridge, Boston.

M. Rayner and B. A. Hockey. 2003. Transparent combination of rule-based and data-driven approaches in speech understanding. In *In Proceedings of the EACL 2003*.

S. Seneff. 1992. A relaxation method for understanding spontaneous speech utterances. In *Proceedings, Speech and Natural Language Workshop*, San Mateo, CA.

R.D. Sharp, E. Bocchieri, C. Castillo, S. Parthasarathy, C. Rath, M. Riley, and J.Rowland. 1997. The Watson speech recognition engine. In *In Proceedings of ICASSP*, pages 4065–4068.

B. Souvignier and A. Kellner. 1998. Online adaptation for language models in spoken dialogue systems. In *Int. Conference on Spoken Language Processing (ICSLP)*.

Y. Wang and A. Acero. 2003. Combination of cfg and n-gram modeling in semantic grammar learning. In *In Proceedings of the Eurospeech Conference*, Geneva, Switzerland.

Y.Y. Wang, M. Mahajan, and X. Huang. 2000. Unified Context-Free Grammar and N-Gram Model for Spoken Language Processing. In *Proceedings of ICASSP*.

XTAG. 2001. A lexicalized tree-adjoining grammar for english. Technical report, University of Pennsylvania, http://www.cis.upenn.edu/ xtag/gramrelease.html.