First Steps towards Dialogue Modelling from an Un-annotated Human-Human Corpus

Sudeep Gandhe and **David Traum**
Institute for Creative Technologies
University of Southern California
13274 Fiji Way, Suite 600, Marin Del Ray, CA 90292
gandhe@ict.usc.edu, traum@ict.usc.edu

Abstract

Virtual human characters equipped with natural language dialogue capability have proved useful in many fields like simulation training and interactive games. Generally behind such dialogue managers lies a complex knowledge-rich rule-based system. Building such system involves meticulous annotation of data and hand autoring of rules. In this paper we build a statistical dialogue model from roleplay and wizard of oz dialog corpus with virtually no annotation. We compare these methods with the traditional approaches. We have evaluated these systems for perceived appropriateness of response and the results are presented here.

1 Introduction

Virtual human characters equipped with natural language dialogue capability have proved useful in many fields like simulation, training and interactive games. These dialogue capabilties are the essential part of their human-like persona. This interface has to be good enough to engage the trainee or the gamer in the activity.

Natural language dialogue systems come in many different flavors. Chatterbot systems like Eliza [Weizenbaum, 1966] or Alice [Wallace, 2003] have to operate in an unrestricted domain with an aim of being human-like. The user input can be about any topic he/she can think of. On the other hand, task-oriented dialogue systems such as pizza-ordering, ATIS [Seneff *et al.*, 1991] or Trains [Allen, 1995] restrict the user quite severely in the topics and ways of talking about them that are allowed.

In casual conversation, even without specific domain knowledege, one can always find reasonable things to say, e.g., “I don’t want to talk about that”, or “Why do you say that?”. Moreover, it is often sufficient to talk about topics at a fairly shallow level, without requiring a lot of detailed task knowledge or knowledge of how some parts of a task relate to others. On the other hand, for a task oriented dialogue in which the system is expected to perform a task or provide task-relevant information, a detailed understanding of the progression of the task and which information has been expressed is often crucial. There are some domains that fall between these extremes, for instance negotiation about whether or not to adopt a proposal. In this case, there is definitely a task or set of tasks involved, but one does not necessarily require as detailed knowledge as is required to actually perform the task. One could agree or disagree for partial or even hidden reasons. This can allow much more flexibility in the type of dialogue interaction, including more varied levels of initiative and dialogue moves, as well as more general arguments and assessments.

There are also various methods for dialogue management. Chatbots typically follow Eliza in operating at a textual level, with pattern matching and substitution to compute a response from an initiative. This can provide a degree of generality, as a single pattern may produce a large range of responses to different initiatives. On the other hand, they can be fairly brittle if the pattern is not appropriately constrained and match inappropriately, producing sometimes uncomprehensible results. Corpus-based retrieval approaches (e.g., [Chu-Carroll and Carpenter, 1999; Leuski *et al.*, 2006]) have an advantage of robust selection, with a more limited set of responses.

Task oriented dialogue generally operates at a concept or dialogue act level. This allows reasoning at more of a meaning than form level and easy integration with other kinds of knowledge-based reasoning, but also more kinds of processing to translate from the surface level to the meaning level and back again.

All of these methods require either extensive writing of rules or other symbolic processing methods, or extensive corpus annotations, both of which serve to introduce a high cost in the development of a dialogue system for a new domain.

In this work we take a look at unsupervised corpus based methods to bootstrap dialogue bots. They don’t have sophasticated cognitive models, but they can be built instantly from a dialogue corpus without annotation or rule-writing. We compare these methods with the more traditional approach of building a information-state based dialogue system.

In the next section we will introduce our first case study system for an annotation-less virtual human dialogue manager. In the next section we will elaborate more on the motivtion for using corpus based methods for such systems. In section 4 we describe the chat-bot systems we have implemented. Section 5 presents the evaluation of the implemented systems and we conclude with discussion and future work.

2 SASO-ST

At Institute for Creative Technologies, USC researchers have developed prototype virtual human characters used for simulation training. SASO-ST [Traum *et al.*, 2005] is one such environment, involving a prototype of a training environment for learning about negotiating with people from different cultures and with different beliefs and goals. In the first scenario, the trainee acts as an army Captain negotiating with a simulated doctor. The goal is convince him to move his clinic to another location. The captain can offer help in moving the clinic and some other perks like medical supplies and equipments.

In order to investigate this domain, and build resources for the system, we collected a corpus of roleplay dialogues and Wizard of Oz (WoZ) dialogues. Roleplay dialogues feature more free-form human face to face interaction whereas the WoZ interactions are constrained by allowing the wizard playing the role of doctor to choose from a limited set of replies. Fig 1 shows a typical roleplay dialogue.

3 Motivation

A typical lifecycle of the dialogue modelling process for virtual humans begins with defining the domain of interaction which follows from the story line. The process includes defining the beliefs and goals of all the parties involved. It is followed by conducting roleplays where volunteers carry out conversations with these goals in mind. This gives a better idea about the behavior of participants that would be expected in real simulation. Experts can then formalize the task structure based on these sample interactions. Additional speech and language data can be gatherd by carring out Wizard of Oz studies and transcribing it. This gathered data can be used for training speech recognition acoustic and language models.

In an information-state based [Traum and Larsson, 2003] approach as used in SASO-ST, the dialogue model has to maintain the information-state — a description of the current state of information that is important for participating in the dialogue. This is done by applying a set of update-rules which are used to change the information-state based on the new input as the dialogue proceeds. Generally the input to information-state is a set of dialogue acts and semantic interpretation about an utterance.

In order to use corpus dialogue data for this kind of system, one must either write parsing or translation rules, or annotate sufficient quantities to train statistical systems. Fig 2 shows an example of the semantic annotation for an utterance in the SASO-ST system. It includes information like speech-acts, modality and case-roles. Based on pairs of sentences with annotated reporesentations like this, a Natural Languge Understanding module can be trained in a supervised fashion which maps the utterance to its semantic meaning. Rule-based processing is then used by the dialogue manager to compute resulting information state components and system utterances.

Producing training data for speech recognition langauge models makes it worthwhile to collect roleplay/WoZ data. But to make further use of this data, significant human effort is required either to write rules or annotate data. Alleviating this human-effort requirement is the main motivation behind the idea of using corpus-based methods to bootstrap dialogue systems without any annotation required. The shallow task structure and the constrained scenario of the negotiation domain make it viable to model dialogue as a sequence of tokens, a language. These modelling techniques are inspired from Information Retrieval field and try to predict the next utterance given the context of the dialogue. They work at the lexical level which does not need the dialogue act or semantic annotaion.

doctor		
	0.0	yes what is it
	1.063	i've got a lot of patients in the back .
	3.03	what can i do for you .
captain		
	4.217	how are you doing sir ,
	5.175	uh my name's captain (xx) ,
	6.748	how are you today ?
doctor		
	7.78	uh well ,
	8.905	
	9.623	i could be better ,
	10.44	i've got a lot of patients in the back ,
	12.061	uh we just had uh FIVE of them come in from the LAST bombing ?
	15.718	so ,
	16.311	what can i do for you .
captain		
	17.342	okay i know you're very busy so i'll get straight to what i came here to talk to you about .
	22.983	right now ,
	24.185	with our estimate ,
	25.077	this is a very unsecure area .
	26.827	and what we'd like to do sir is uh secure and stabilize your patients as soon as possible and move you out of this area so we can move you to a more secure location .
doctor		
	36.58	my PATIENTS are stable right NOW .
	40.489	and ,
	41.395	i i don't understand why you're coming in here ,
	44.926	to tell me to move patients out of here ,
	47.583	from a clinic that's been here for almost a YEAR .
	50.311	and now i have to move my patients ?

Figure 1: A sample roleplay dialogue in SASO-ST

4 Chat-Bot methods

The methods described in the this section view dialogue as a sequence of tokens. They employ simple Information Retrieval techniques to create chat-bots that are trained in an unsupervised manner. Since there is no annotation effort other than building the dialogue corpus from roleplays and WoZ, these methods allow for rapid prototype development.

We will have to move the hospital .	
S.mood	declarative
S.sem.task	move-clinic
S.sem.speechact.type	statement
S.sem.type	event
S.sem.modal.deontic	must
S.sem.agent	we
S.sem.event	move
S.sem.theme	hospital
S.sem.time	future

Figure 2: An example of semantic annotation

In building these prototypes we have chosen to fix the input modality to typed text and the interface is in the form of a chat session. The turns strictly alternate between the doctor (system) and the captain (user). The screenshot of the interface is as seen in the fig 3.

The general idea is to retrieve one of the doctor utterances from the corpus and present it to the user as the system response. We implemented 4 types of chat-bots. They capture different aspects of local and global coherence of the dialogue.

4.1 random bot

This type of bot provides a zero baseline and does not capture global or local coherence. A set of utterances with doctor as the speaker is compiled from the corpus. The bot just replies to any utterance of the captain with a randomly selected utterance from this list. There are around 435 doctor utterances to randomly choose from.

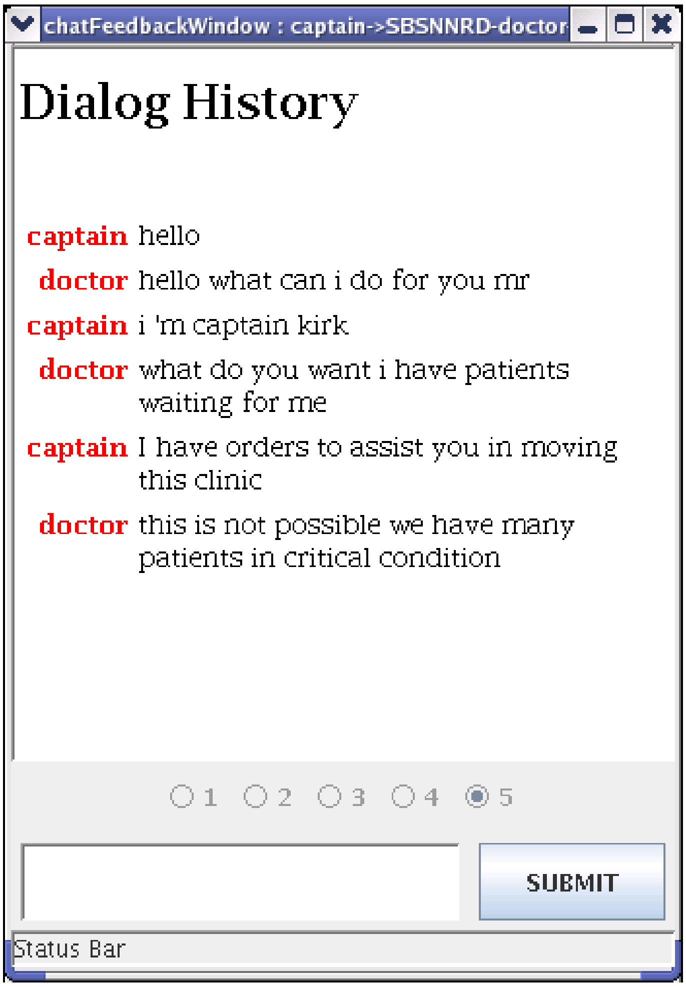
4.2 nearest context

This type of bot captures local coherence. In this type rather than choosing the reply randomly from all available doctor utterances we decide to choose the one which has the most similar context as compared to the context of the current ongoing dialogue. The context is defined as last *n* turns. Here we have chosen n=2. To find the similarity between the contexts we represent the context using vector space model as in information retrieval [Manning and Schutze, 1999]. Fig 5 shows an example of the feature vector used to represent the context of the dialogue. In this vector the unigrams from utterances form the features. These unigrams are augmented with the speaker and the distance in time in units of turns. The latest turn is at a distance of 0, the previous at 1 and so on. The weights for the features are derived from tf-idf scores. For the systems to be more reactive to the latest input, we weigh these tf-idf scores depending on how far back in the history the utterance is.

Let W_i^j be the weight assigned for unigram w_i which appears j turns ago. Then W_i^j is given by,

$$W_i^j = TF(w_i) \times IDF(w_i) \times H(j) \qquad (1)$$

$$TF(w_i) = 1 + \log(\#w_i) \qquad (2a)$$



chatFeedbackWindow : captain->SBSNNRD-doctor

Dialog History

captain hello

doctor hello what can i do for you mr

captain i 'm captain kirk

doctor what do you want i have patients waiting for me

captain I have orders to assist you in moving this clinic

doctor this is not possible we have many patients in critical condition

○ 1 ○ 2 ○ 3 ○ 4 ● 5

Figure 3: A screenshot of user interface

where $\#w_i$ is the number of times w_i appears in the utterance

$$IDF(w_i) = \log\left(\frac{N}{df_i}\right) \qquad (2b)$$

where N is the total number of utterances
and df_i is the number of utterances containing w_i

$$H(j) = \exp\frac{-j^2}{2} \qquad (2c)$$

This is a type of memory based or Instance based learning. The training phase only involves identifying all the contexts associated with utterances and storing the vector space representations for them in memory. When it's time to predict the next utterance for the doctor the job is to find a context c_k which is most similar to the context of the current dialogue c. The utterance u_k associated with context c_k will be the reply. Here k is given by,

$$argmin_{i=1..n}\left(||\bar{c}_i - \bar{c}||\right) \qquad (3)$$

where the feature vectors $\bar{c}_i$ and $\bar{c}$ are L_2 normalized.

captain	hello
doctor	so how are you going to do that
captain	hello doctor i am captain kirk
doctor	uh i have i have patients from both sides from ah there have been injured from american forces as well as other locals

Figure 4: example interaction for random bot

⋮

doctor	what do you want i have patients waiting for me
captain	I have orders to assist you in moving this clinic

<captain>_0_i	1.08705147
<captain>_0_have	1.66113929
<captain>_0_orders	6.24610677
<captain>_0_to	1.03117101
<captain>_0_assist	4.16666522
<captain>_0_you	0.56935296
<captain>_0_in	2.18566375
<captain>_0_moving	2.98801023
<captain>_0_this	1.7327793
<captain>_0_clinic	2.41746537
<doctor>_1_what	1.21161264
<doctor>_1_do	1.18616006
<doctor>_1_you	0.34533003
<doctor>_1_want	1.86086817
<doctor>_1_i	0.65933004
<doctor>_1_have	1.00753191
<doctor>_1_patients	1.02003814
<doctor>_1_waiting	2.45577118
<doctor>_1_for	1.32567208
<doctor>_1_me	1.66771622

Figure 5: feature vector representing the context of previous *n* turns. *(here n=2)* (same example as in fig 3)

Apart from this we try to avoid repetition in the dialogue. If the predicted best response u_k is same as that predicted in previous prediction the next best reply is chosen.

4.3 segmented - nearest context

This type of bot tries to capture both global and local coherence. The chat bot described earlier suffers from short-sightedness. It is designed to be coherent locally, but in the broader sense of the dialogue phenomenon there are certain responses that don't make much sense. These are a direct result of the fact that context cannot be faithfully represented using just n previous turns.

We observed most of the failures are because of the violations of presuppositions. Presupposition is the assumption that must be held by both parties for an utterance to make sense. e.g. Doctor saying " I don't want to move the clinic " makes sense only if the the dialogue participants both believe that doctor may be required to move the clinic. And in this case it comes down to whether moving the clinic was previously mentioned or not.

captain	hello doctor
doctor	hello
captain	i am captain kirk nice to meet you
doctor	i'm uh doctor perez uh i'm very busy right now i i only have uh just a few minutes to uh
captain	i want to talk to you about an important issue
doctor	uh captain i i appreciate you have given us a lot of very good information to uh which roads are safe and where the landmines are and i need you+ uh i i cannot move this location though because of all these patients they're they're too critical right now i'm working on a on a on a young girl with amoebic dysentery and and she she requires my attention at all times i there is no way i i+ these people are in no shape to to+

Figure 6: example interaction for nearest context bot

Hence we split the dialogue in segments that try to keep track of whether several key concepts were introduced in the dialogue or not. These key concepts are picked by an expert who understands the domain well. Fig 7 illustrates the compiled list for SASO-ST dialogues. It identifies concepts by listing down the words with similar sense.

[move]	move relocate
[supplies]	antibiotics supplies medicines plasma
[transportation]	vehicle transportation trucks

Figure 7: list of key concepts along with the representative unigrams compiled for SASO-ST dialogues

With this information the system can determine the segment signature for every context. e.g. the segment signature for the context in fig 5 will be,

```
[move 1, supplies -1, transportation -1]
```

The decision procedure for picking up the response based on given context is same as earlier but with one additional constraint. The most similar context c_i must have the same segment signature as the current context c.

Notice that it is possible to misrecognize the current segment of the dialogue. This may happen because a certain similar word was used for signaling the concept but this word was not in the list. e.g. using " shifting the clinic " instead of

captain	hello doctor
doctor	hello
captain	i am captain kirk nice to meet you
doctor	i'm uh doctor perez uh i'm very busy right now i i only have uh just a few minutes to uh
captain	i want to talk to you about an important issue
doctor	ok

Figure 8: example interaction for segmented - nearest context bot

captain	hello
doctor	yes
captain	i am captain kirk nice to meet you
doctor	you are the threat i need protection from you
captain	no we are here to protect you and your clinic
doctor	are you injured

Figure 9: example interaction for segmented - random bot

" moving the clinic ". Similarly " moving the chair " instead of " moving the clinic ". This clearly shows that single unigram spotting is not adequate for detecting dialogue segment transitions.

4.4 segmented - random

This type of bot tries to capture only the global coherence. The last mentioned chat-bot system tries to achieve local appropriateness by finding the nearest similar context and also the global appropriateness by keeping track of the segments. To understand which of the two factors makes more significant impact we implemented the fourth type of bot. It keeps track of the segment signature of the context but picks up one of the utterance randomly with that signature.

5 Evaluation

To evaluate the merits of these methods we asked volunteers to conduct a conversation with the simulated doctor. These volunteers had two roles - as a participant in negotiation conversation and also as a judge of the responses from the doctor. The interface shown in fig 3 allows the volunteers to judge the doctor's response on a scale of 1 to 5 for appropriateness. Here 1 stands for a totally non-sensical reply and 5 is the perfectly appropriate response. This is a subjective metric and we believe that the conversation participant is in the best position to judge the appropriateness of the response.

Each bot type was used in 5 conversations. Each volunteer had conversations with all types of bots. The presenting order of the bots was balanced.

The average ratings for various types of chat-bots is summarized here. nearest context, segmented - nearesest context and segmented - random are all significantly better (t-test, $p < 0.05$) over the random baseline. segmented - nearest context is significantly better (t-test, $p < 0.05$) than segmented - random or nearest context approaches.

	Without Segments		With Segments	
Without Context	avg	2.6764	avg	3.0430
	stddev	1.2758	stddev	1.2930
	size	136	size	93
With Context	avg	3.0625	avg	3.4722
	stddev	1.5438	stddev	1.3703
	size	112	size	108

Figure 10: Results for various types of chat-bots

6 Discussion and Future work

In the preliminary evaluation and the subjective feedback from the users it appears that type-3 system performs suprisingly well. The reason behind this success stems from the fact that these converstaions are restricted because they have to follow the story line. Also since the task structure is shallow, just identifying the correct dialogue segment helps quite a bit.

The way these methods differ from information-state based dialogue systems is that there is no need for annotation. Theoretically information state based systems can achieve arbitrary levels of perfection given enough rules are authored for it. But rule authoring still remains a practical limiting factor for such systems. We do miss the deep understanding that an information-based system gives, such as principled connection to emotions and plan reasoning.

When compared to general purpose chat-bot systems like Eliza or Alice, our domain of interaction is well defined. e.g. chat-bot systems have to be ready to talk about a variety of topics, including favorite movies, sports etc. while our system just has to know about the negotiation scenario between the captain and the doctor.

Some recent task-oriented dialogue systems use reward structure for optimizing the dialogue policies. These systems generally have a clear defination of success and failure which helps in designing reward structure. They typically learn the optimal policies using MDP [Levin and Pieraccini, 1997], POMDP [Roy *et al.*, 2000] framework. But in case of games or simulation training dialogue systems don't have easy access to the notion of success and failure. It is very hard to design this reward system. The best that can be done is to imitate human behavior, which is exactly what our system is designed to do.

There are a number of avenues of future work that we would like to employ. First, we can compare our dialogue agents to other dialogues, e.g., some of the original role-play dialogues and dialogues with the SASO-ST system. We would need to alter our evaluation mechanism though, since our evaluations are done by the dialogue participants themselves in real time rather than by a third party.

We will also connect our dialogue manager to the virtual human's body and speech recognizer and synthesizer, to see whether embodiment has an effect on acceptability. We will also use the same techniques to create systems for other dialogue agents.

Also, we will try to automatically identify the key concepts used to segment the dialogue, as well as looking at what in-

Speaker	Utterance	Rating
doctor	uh i'm sorry what was your name	5
captain	i'm captain kirk	
doctor	captain it's nice to meet you i don't have much time so i would appreciate it if you could make uh make this quick	4
captain	ok doctor . I want to talk about the security of the area this hospital is located in.	
doctor	well they i don't know that there is very good uh anwhere this this one seems to be full of patients who i need to be treating but the problem is the roads are not always clear and it's necessary uh you know that we are able to get supplies and and we're not	2
captain	yes doctor, it is hard to get supplies here, because of the danger. We do have access to supplies, but it will be hard to bring them here.	

Figure 11: Illustration of the problem due to the granularity of the utterance. This dialogue is generated by using segmented-nearest context bot and shows the evaluation by the participant. Last utterance from doctor gets a low rating.

formation would both improve dialogue quality and be able to be extracted automatically or authored with little effort. We will also investigate how these methods can be applied to tasks which have a more deeper structure.

Our system works by selecting the appropriate utterance from the ones it has seen. Using human generated utterances has the advantage of being more natural and fluent. But the main assumption that a dialogue can be carried out by retreiving an utterance from the training data rather than constructing it from a high level abstract representation can be a considerable limitation. This is felt strongly when the system comes across completely unseen contexts. Typically we found that type-3 system gets stuck in a loop where the dialogue does not move to the next phase.

Also the granularity of the utterances is an issue. Fig 11 shows an example where the utterance refers to security issues by mentioning the problems of the blocked roads but also talks about other things not mentioned in the preceeding context. This makes the utterance less coherent. We are looking into stochastic models for discourse coherence [Barzilay and Lapata, 2005; Soricut and Marcu, 2006] which can help recognize which utterances are best suited given the context.

References

[Allen, 1995] James F. Allen. The trains project. *Journal of Experimental and Theoretical AI*, 1995.

[Barzilay and Lapata, 2005] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. In *Proc. ACL-05*, 2005.

[Chu-Carroll and Carpenter, 1999] Jennifer Chu-Carroll and Bob Carpenter. Vector-based natural language call routing. *Journal of Computational Linguistics*, 25(30):361–388, 1999.

[Leuski *et al.*, 2006] Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, 2006.

[Levin and Pieraccini, 1997] Esther Levin and Roberto Pieraccini. A stochastic model of computer-human interaction for learning dialogue strategies. In *Proc. Eurospeech '97*, pages 1883–1886, Rhodes, Greece, 1997.

[Manning and Schutze, 1999] Chris Manning and Hinrich Schutze. *Foundations of Statical Natural Language Processing*, chapter 15. MIT Press. Cambridge, MA, 1999.

[Roy *et al.*, 2000] N. Roy, J. Pineau, and S. Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL2000)*, Hong Kong, 2000., 2000.

[Seneff *et al.*, 1991] E. Seneff, L. Hirschman, and V.W. Zue. Interactive problem solving and dialogue in the atis domain. pages 354–359, February 1991.

[Soricut and Marcu, 2006] Radu Soricut and Daniel Marcu. Discourse generation using utility-trained coherence models. In *Proc. ACL-06*, 2006.

[Traum and Larsson, 2003] David Traum and Staffan Larsson. The information state approach to dialogue management. In Jan van Kuppevelt and Ronnie Smith, editors, *Current and New Directions in Discourse and Dialogue*. Kluwer, 2003.

[Traum *et al.*, 2005] David Traum, William Swartout, Jonathan Gratch, and Stacy Marsella. Virtual humans for non-team interaction training. AAMAS-05 Workshop on Creating Bonds with Humanoids, July 2005.

[Wallace, 2003] Richard Wallace. *Be Your Own Botmaster, 2nd Edition*. ALICE A. I. Foundation, 2003.

[Weizenbaum, 1966] Joseph Weizenbaum. Eliza–a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, January 1966.