IN PROCEEDINGS OF EUROSPEECH'97, RHODES, GREECE, SEPTEMBER 1997

STATISTICAL ANALYSIS OF DIALOGUE STRUCTURE

Ye-Yi Wang and Alex Waibel

Language Technology Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
Email: {yyw,waibel}@cs.cmu.edu

ABSTRACT

We introduce a statistical model for dialogues. We describe a dynamic programming algorithm that can be used to bracket a dialogue into segments and label each segment with its speech act. We evaluate the performance of the model. We also use this model for language modelling and get perplexity reduction.

1 INTRODUCTION

Dialogue structure provides important information for spoken language understanding. This structure comprises the current topic, discourse state, and speech act, etc. Many researchers used topic information to reduce the perplexity of a task [1, 2]. In our experiments, we also found that dialogue structure information also helps to reduce ambiguities and improve spoken language translation performance.

While knowledge-based approaches are used widely and successfully in dialogue structure analysis[3, 4], they require intensive human effort in defining linguistic structures and developing grammars to detect the structures. We would like to build a model that is able to learn from examples to analyze dialogue structures.

[5] used a statistical approach for dialogue analysis. They modeled dialogue structure with a 6-state Markov chain. Each state represents a speech act, and it emits words to form sentences of that speech act. The transition and emission probability can be obtained from labeled data with maximum likelihood estimation:

$$\begin{aligned} a_{ij} &= P(q_t = S_j \mid q_{t-1} = S_i) \\ &= \frac{\text{number of transitions from } S_i \text{ to } S_j}{\text{number of transitions from } S_i} \\ b_i(k) &= P(v_k \text{ at } t \mid q_t = S_i) \\ &= \frac{\text{number of times observing } v_k \text{ in } S_i}{\text{number of times in } S_i} \end{aligned}$$

A deficiency of this model is that it treats words as unrelated items randomly emitted from a state. It does not take into account a much stronger constraint that words must form a legitimate sentence of a speech act. Because of this, the model is inclined to shift among states too often to maximize the probability of individual words. In the following section, we present a model that uses ngram language models to constrain word sequences that can emit from a dialogue state.

2 DIALOGUE MODEL

2.1 Dialogue and Bracketing Scheme

In this paper, a *dialogue* is a sequence of *utterances*. An utterance is a dialogue participant's turn-taking. It may be consist of several *segments*. A *speech act* is associated with each segment that reveals the functionality of the segment. A segment can be a sentence, a clause or a phrase with its own speech act.

In an unmarked dialogue there is no segment boundary or speech act information labeled. Only utterance boundaries are available. An example of unmarked dialogue is shown below:

```
A: hello Dr. Noah
B: hi Tor let's set up a meeting for a
   couple hours in the next two weeks
   when's good for you
A: let's see how about Friday the second
   in the morning
B: I'm busy that morning
......
```

A bracketing scheme B breaks long utterances in a dialogue into sequences of segments, and labels each segment with its speech act. Below is an example of a bracketing scheme for the previous dialogue:

```
[nicety] (Hello Dr. Noah)
[nicety] (Hi Tor)
```

```
[suggest-meeting] (let's set up a meeting
                   for a couple hours)
[temporal] (in the next two weeks)
[your-availability] (when's good for you)
[interject] (let's see)
[suggest-time] (how about Friday the second
                in the morning)
[my-unavailability] (I'm busy that morning)
......
```

This is not the only bracketing scheme for the dialogue. For example, the second utterance can also be bracketed as following:

```
[nicety] (Hi Tor)
[suggest-meeting] (let's set up a meeting
                   for a couple hours)
[your-availability] (in the next two weeks
                     when's good for you)
```

In a bracketing scheme B for a dialogue D, B_i denotes the i-th segment of B, A_i represents the speech act label for B_i. $|B|$ is the number of segments in B, and $|B_i|$ or $|D|$ is the number of words in B_i or D respectively. B_i's form a partition of B: there is no overlapping between B_i and B_{i+1} for $i = 1, \cdots, |B|$, and $\sum_{i=1}^{|B|} |B_i| = |D|$.

2.2 Bracketing Model

The generation of a dialogue D can be depicted by the following process:

1. At time 0, the dialogue is at a null segment with the speech at <d>, the start of the dialogue. For $i = 1, 2, \cdots$, do the following:

2. At time i, generate the speech act A_i according to a distribution $P(A_i \mid A_0^{i-1}, B_0^{i-1})$. If $A_i =$ </d>, which is the dialogue end speech act, the generation is complete. Otherwise take the next step.

3. Generate a segment of words B_i with the speech act A_i according to a distribution $P(B_i \mid A_0^i, B_0^{i-1})$.

We can make the following independence assumptions in this model:

1. $P(A_i \mid A_0^{i-1}, B_0^{i-1}) = P(A_i \mid A_{i-1})$.

2. $P(B_i \mid A_0^i, B_0^{i-1}) = P(B_i \mid A_i)$.

And $P(B_i \mid A_i)$ can be modeled with a speech act dependent ngram model:

$$
\begin{aligned}
& P(B_i \mid A_i) \\
&= \sigma(B_i) \prod_{j=1}^{|B_i|+1} P(b_{ij} \mid b_{i1}^{j-1}, A_i) \\
&= \sigma(B_i) \prod_{j=1}^{|B_i|+1} P(b_{ij} \mid b_{i(j-N+1)}^{j-1}, A_i) \qquad (1)
\end{aligned}
$$

here $b_{i0} =$ <s> and $b_{i(|B_i|+1)} =$ </s>. The function $\sigma(B_i) = 1$ when there is no utterance boundary between b_{ij} and $b_{i(j+1)}$ for $j = 1, \cdots, |B_i| - 1$. It guarantees that the bracketing scheme B respects the natural utterance boundaries — no segment can stride over two utterances.

Hence we can get the probability of a dialogue D:

$$
\begin{aligned}
P(D, B) &= \prod_{i=1}^{|B|} P(A_i \mid A_{i-1}) P(B_i \mid A_i) \\
&\quad \times P(\texttt{</d>} \mid A_{|B|}) \qquad (2) \\
P(D) &= \sum_B P(D, B). \qquad (3)
\end{aligned}
$$

here $A_0 =$ <d>.

2.3 Parameter Estimation

In the bracketing model, there are two types of parameters: a speech act transition distribution $P(A_i \mid A_{i-1})$ depicts the likelihood of switching from one speech act to another; and for each speech act A, the speech act dependent ngram distribution $P_A(w_i \mid w_{i-N+1}^{n-1}) = P(w_i \mid w_{i-N+1}, \ldots, w_{n-1}, A)$ models the likelihood of generating a sentence under the speech act A. Labeled data can be used to estimate these parameters. Data are labeled like the above example for bracketing schemes, so the parameters in the bracketing model can be estimated with maximum likelihood estimation: the speech act sequence in the labeled data can be used to estimate the speech act transition parameters, and the texts for each speech act can be used to train the ngram for that speech act.

3 ALGORITHMS FOR THE BRACKETING MODEL

Given a bracketing model, we are facing two problems. The first is how to effectively compute $P(D)$; the second is how to find the most probable bracketing scheme, i.e., the most probable structure of a dialogue. Both problems can be solved with dynamic programming.

We define $Q_{ij}(A)$ to be the probability that $d_i d_{i+1} \ldots d_j$ is a complete sentence generated for the speech act A, regardless of the context around it. $Q_{ij}(A)$ can be computed as

$$
\begin{aligned}
Q_{ii}(A) &= P_A(d_i \mid \texttt{<s>}) P_A(\texttt{</s>} \mid d_i) \\
Q_{ij}(A) &= \frac{Q_{i(j-1)}(A) P_A(d_j \mid d_{j-1}) P_A(\texttt{</s>} \mid d_j)}{P_A(\texttt{</s>} \mid d_{j-1})} \\
&\quad \times \sigma(d_{j-1} d_j)
\end{aligned}
$$

To solve the first problem, we define $\alpha(k, A)$ to be the probability that $d_k \cdots d_n$ starts with a segment of speech act A (here $n = |D|$) in all possible bracketing schemes. Then

$$
P(D) = \sum_A P(A \mid \texttt{<d>}) \alpha(1, A). \qquad (4)
$$

$\alpha(k, A)$ has the following recursive relation that allows us to use dynamic programming:

$$
\begin{aligned}
\alpha(n, A) &= Q_{nn}(A) P(\texttt{</d>} \mid A) \\
\alpha(k, A) &= \sum_{k \le j < n, C} \alpha(j+1, C) Q_{kj}(A) P(C \mid A) \} \\
&\quad + Q_{kn}(A) P(\texttt{</d>} \mid A) \qquad (5)
\end{aligned}
$$

In the second problem, we would like to bracket a dialogue $D = d_1 d_2 ... d_n$ into a sequence of segments labeled with their speech acts. The bracketing scheme should be the most probable one:

$$\hat{B} = \arg\max_{B} P(D, B) \qquad (6)$$

where $P(D, B)$ is defined in (2). We call the scheme $\hat{B}$ the Viterbi bracketing. It can be found with dynamic programming.

Let $\gamma(k, A)$ be the maximum probability that $d_k \cdots d_n$ starts with a segment of speech act A in a bracketing scheme. Then

$$
\begin{aligned}
\gamma(n, A) &= Q_{nn}(A) P(\texttt{</d>} \mid A) \\
\gamma(k, A) &= \max\{ Q_{kn}(A) P(\texttt{</d>} \mid A), \\
&\quad \max_{k \le j < n, C} \gamma(j+1, C) Q_{kj}(A) P(C \mid A) \} \\
\xi(k, A) &= (n, A) \\
&\quad \text{when } \gamma(k, A) = Q_{kn}(A) P(\texttt{</d>} \mid A) \\
\xi(k, A) &= \arg\max_{(k < j \le n, C)} \gamma(j+1, C) Q_{kj}(A) P(C \mid A) \\
&\quad \text{otherwise} \qquad (7)
\end{aligned}
$$

here $\xi(k, A) = (j, C)$ is used to remember the end position of the first bracket and the speech act of the following bracket in the bracketing scheme for $d_k \cdots d_n$ that starts with the speech act A and maximizes $\gamma(k, A)$. By choosing $\hat{A} = \arg\max_A P(A \mid \texttt{<d>}) * \gamma(0, A)$ as the speech act of the first bracket and backtracking with ξ (starting from $\xi(0, \hat{A})$), the optimal bracketing and labeling of the whole dialogue can be recovered.

The time complexity of the dynamic programs algorithms is $O(n^2 * l^2)$, where l is the number of different speech act labels, and n is the length of a dialogue.

4 PERFORMANCE

We applied the bracketing model to the JANUS [6] scheduling data with 19 different speech acts, exclusive of <d> and </d>.

We hand bracketed 96 dialogues with 1400 utterances or around 40,000 words. The data were segmented into around 6900 segments and each is labeled with a speech act. We trained a speech act bigram model and 19 speech act dependent word bigram models. We smoothed the speech act dependent bigram models with a speech act independent model using deleted interpolation[7]. The speech act independent bigram model was trained with a corpus of around 420,000 words in the same domain.

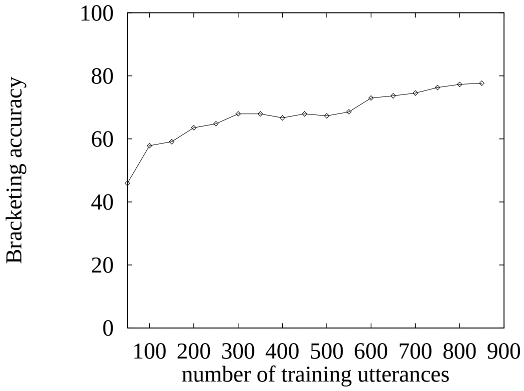

Figure 1: Bracketing performance versus amount of training data: The amount of training data were measured as the number of utterances.

Missing segment	22
Added segment	33
Segment with missing words	9
Segment with added words	4

Table 1: Number of different kinds of errors

We bracketed a test set of 8 dialogues (117 utterances) with the Viterbi dynamic programming algorithm. Figure 1 shows the performance on those test data. The accuracy was measured as the percentage of the target speech act segment discovered by the algorithm. We classified errors into four different types, namely *missing segments*, *added segments*, *segments with missing words*, and *segments with added words*. Table 1 gives the statistics of different types of errors for the total 68 errors with our testing data.

This error classification does not form a partition of the error space. Different types of errors may have the same cause. For example, adding some words to one segment made these words missing in another segment, or make a whole segment composed of these words missing. Many of the *missing segment* and *added segment* errors were caused by mislabelling of correct segmentations.

We also measured the performance in the same way as [5], namely for each word in a dialogue we found its *state* — the speech act of the segment that the word is in, and compared it with its state in a reference bracketing. In this case, mis-segmentation by adding or deleting a few words around segment boundaries would not hurt the performance measure very much. The correct word to state classification rate was 89%. This is much better than the 74.1% accuracy in [5]. It is even more significant if we consider that the task in [5] was an easier one with only five speech acts for a similar JANUS data set.

We also calculated the perplexity of a 20 dialogues test set with around 2,400 words. We did this with a speech act independent bigram model, the bracketing

Model	Perplexity
SA Independent Bigram	42.2
Bracketing Model	39.6
Viterbi Bracketing Model	43.7

Table 2: Test Data Perplexities of Different Models.

model that computes dialogue probability with (4), and a Viterbi bracketing model that computes the probability of a dialogue with its Viterbi bracketing scheme only. Table 2 shows the perplexities.

5 DISCUSSION

The bracketing algorithm requires hand made training samples. Although the simplicity of the dialogue structure and a special Emacs editing mode make hand-labelling much easier, it is still a tedious and time consuming task. However, data labeling can proceed with bootstrapping. We can start with labeling a few hundred utterances, train a system with these initial data, use the trained system to bracket more data, fix the errors in the newly bracketed data, and use them for further training. Figure 1 shows that the system has already achieved adequate performance with the first few hundred samples. If we start with 300 sentences, we only have to fix 30% of errors in the new data bracketed by the trained system.

6 CONCLUSION

We introduced a statistical dialogue model. Dynamic programming algorithms were used to compute the likelihood of a dialogue and to find its Viterbi structure. By constraining a segment of words with a speech act dependent language model, the model achieved better performance than the previous work. Used for language modelling, the bracketing model resulted in perplexity reduction.

7 ACKNOWLEDGEMENTS

We would like to thank John Lafferty for enlightening discussions and suggestions on this work. This research was partly supported by the Advanced Telecommunications Research (ATR) and the Verbmobil Project. The views and conclusions in this document are those of the authors.

8 REFERENCES

[1] S. Young. Dialog Structure and Plan Recognition in Spontaneous Spoken Dialog. In *EUROSPEECH 1993*, volume 2, pages 1169–1172, 1993.

[2] R. Kneser and V. Steinbiss. On the Dynamic Adaptation of Stochastic Language Models. In *ICASSP '93*, volume 2, pages 586–589. IEEE, 1993.

[3] B. J. Grosz and C. J. Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12 (3), 1986.

[4] Diane J. Litman and James F. Allen. Discourse processing and commonsense plans. In *Intentions in Communications*, 1990.

[5] M. Woszczyna and Alex Waibel. Inferring Linguistic Structure in Spoken Language. In *ICSLP 1994*, 1994.

[6] B. Suhm, P.Geutner, T. Kemp, A. Lavie, L. Mayfield, A. McNair, I. Rogina, T. Schultz, T. Sloboda, W. Ward, M. Woszczyna, and A. Waibel. JANUS: Towards multilingual spoken language translation. In *Proceedings of the ARPA Speech Spoken Language Technology Workshop, Austin, TX, 1995*, 1995.

[7] F. Jelinek and E. L. Mercer. Interpolated Estimation of Markov Source Parameters from Sparse Data. In D. Gelsema and L. Kanal, editors, *Pattern Recognition in Practice*. North-Holland, 1980.