Extracting Caller Information from Voicemail

Geoffrey Zweig, Jing Huang, Mukund Padmanabhan

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
USA
gzweig, jhuang, mukund@watson.ibm.com

Abstract

In this paper we address the problem of extracting the identities and phone numbers of the callers in voicemail messages. Previous work in information extraction from speech includes spoken document retrieval and named entity detection. This task differs from the named entity task in that the information we are interested in is a subset of the named entities in the message, and consequently, the need to pick the correct subset makes the problem more difficult. Also, the caller's identity may include information that is not typically associated with a named entity. In this work, we present two information extraction methods, one based on hand-crafted rules, and one based on a maximum entropy model. We find that both systems give good performance when applied to manually-derived transcriptions, and that the maximum entropy system can reliably identify the time intervals containing phone numbers, even in the presence of significant decoding errors.

1. Introduction

In recent years, the task of automatically extracting information from data has grown in importance, as a result of an increase in the number of publicly available archives and a realization of the commercial value of the available data. One aspect of information extraction (IE) is the retrieval of documents. Another aspect is that of identifying words from a stream of text that belong in pre-defined categories, for instance, "named entities" such as proper names, organizations, or numerics. Though most of the earlier IE work was done in the context of text sources, recently a great deal of work has also focused on extracting information from speech sources. Examples of this are the Spoken Document Retrieval (SDR) [1] and named entity (NE) extraction [2, 3, 4] tasks. The SDR task focused on Broadcast News and the NE task focused on both Broadcast News and telephone conversations.

In this paper, we focus on a source of conversational speech data, voicemail, that is found in relatively large volumes in the real-world, and that could benefit greatly from the use of IE techniques. The goal here is to query one's personal voicemail for items of information, without having to listen to the entire message. For instance, "who called today?", or "what is X's phone number?". Because of the importance of these key pieces of information, in this paper, we focus precisely on extracting the identity and the phone number of the caller. We measure the success of our systems both in terms of a word-based measure, and in terms of the time-overlap between the identified and correct portions of speech. This latter measure is relevant to a speech-recognition based system in which the identified segments are replayed to the user. Other attempts at summarizing voicemail have been made in the past [5], however the goal there was to compress a voicemail message by summarizing it, and not to extract the answers to specific questions.

An interesting aspect of this research is that because a transcription of the voicemail is not available, speech recognition algorithms have to be used to convert the speech to text and the subsequent IE algorithms must operate on the transcription. One of the complications that we have to deal with is the fact that the state-of-the-art accuracy of speech recognition algorithms on this type of data is only in the neighborhood of 60-70% [6].

The task that is most similar to our work is named entity extraction from speech data [2]. Although the goal of the named entity task is similar - to identify the names of persons, locations, organizations, and temporal and numeric expressions - our task is different, and in some ways more difficult. There are two main reasons for this: first, caller and number information constitute a small fraction of all named entities. Not all person-names belong to callers, and not all digit strings specify phone-numbers. In this sense, the algorithms we use must be more precise than those for named entity detection. Second, the caller's identity may include information that is not typically found in a named entity, for example, "Joe on the third floor", rather than simply "Joe". We discuss our definitions of "caller" and "number" in Section 2.

To extract caller information from transcribed speech text, we implemented two different systems. The first is a simple rule-based system that uses trigger phrases to identify the information-bearing words. The second is a maximum entropy model that tags the words in the transcription as belonging to one of the categories, "caller's identity", "phone number" or "other". We evaluate these systems on manual voicemail transcriptions as well as the output of a speech recognizer.

The rest of the paper is organized as follows: Section 2 describes the database we are using; Section 3 contains a description of the baseline system; Section 4 describes the maximum entropy model and the associated features; Section 5 contains our experimental results and Section 6 concludes our discussions.

2. The Database

Our work focuses on a database of voicemail messages gathered at IBM, and made publicly available through the LDC. This database and related speech recognition work is described fully in [6]. We worked with approximately $5,500$ messages, which we divided into $4,200$ messages for training, 300 for development, and 1000 for evaluation. The messages were manually transcribed with about 3% errors, and then a human tagger identified the portions of each message that specified the caller and any return numbers that were left. Caller and number information was determined as follows. The caller was defined to be the consecutive sequence of words that best answered the question "who called?". The definition of a number we used is a sequence of consecutive words that enables a return call to be placed. Thus, for example, a caller might be "Angela *from P.C. Labs*," or "Peggy Cole *Reed Balla's secretary*". Similarly, a number may not be a digit string, for example: "*tieline* eight oh five six," or "*pager* one three five". No more than one caller was identified for a single message, though there could be multiple numbers. The training of the maximum entropy model and statistical transducer are done on these annotated scripts.

3. A Baseline Rule-Based System

In voicemail messages, people often identify themselves and give their phone numbers in highly stereotyped ways. So for example, someone might say, "Hi Joe it's Harry..." or "Give me a call back at extension one one eight four." Our baseline system takes advantage of this fact by enumerating a set of transduction rules - in the form of a *flex* program - that transduce out the key information in a call.

The baseline system is built around the notion of "trigger phrases". These phases are the patterns that are used by the flex program to recognize caller's identity and phone numbers. Examples of trigger phrases are "Hi this is", and "Give me a call back at". When the flex program encounters a trigger phrase, it enters a special name (or number) matching state, and prints the words that are matched in this state.

In addition to trigger phrases, "trigger suffixes" proved to be useful for identifying phone numbers. For example, the phrase "thanks bye" frequently occurs immediately after the caller's phone number. In general, a random sequence of digits cannot be labeled as a phone number; but, a sequence of digits followed by "thanks bye" is almost certainly the caller's phone number. So when the flex program matches a sequence of digits, it stores it; then it tries to match a trigger suffix. If this is successful, the digit string is recognized a phone number string. Otherwise the digit string is ignored.

Our baseline system has about 200 rules. Its creation was aided by an automatically generated list of short, commonly occurring phrases that were then manually scanned and added to the flex program. It is the simpler of the systems presented, and achieves a good performance level, but suffers from the fact that a skilled person is required to identify the rules.

4. Maximum Entropy Model

Maximum entropy modeling is a powerful framework for constructing statistical models from data, and has achieved state-of-the-art performance in a variety of difficult classification tasks such as part-of-speech tagging [7], prepositional phrase attachment [8] and named entity tagging [9]. In the following, we briefly describe the application of these models to extracting information from voicemail messages.

The problem of extracting the caller's identity and phone number can be thought of as a tagging problem, where the tags are "caller's identity," "caller's phone number" and "other." The objective is to tag each word in a message with one of these labels. Further, in order to segment repeated patterns, for each tag t there are two sub-tags: $begin\_t$ and t. For example, "*hi jim this is patricia at bank united ... call me back at two nine two three*" would be tagged as "*other other other other begin_caller caller caller caller ... other other other other begin_number number number number*".

	Features		
$\forall w_i$	$w_i = X$	&	$t_i = T$
	$t_{i-1} = X$	&	$t_i = T$
	$t_{i-2}t_{i-1} = XY$	&	$t_i = T$
	$w_{i-1} = X$	&	$t_i = T$
	$w_{i-2} = X$	&	$t_i = T$
	$w_{i+1} = X$	&	$t_i = T$
	$w_{i+2} = X$	&	$t_i = T$

Table 1: Unigram features of the current history h_i.

The information that can be used to predict the tag of a word is the context of its surrounding words and their associated tags. Let $\mathcal{H}$ denote the set of possible word and tag contexts, called "*histories*", and $\mathcal{T}$ denote the set of tags. The maxent model is then defined over $\mathcal{H} \times \mathcal{T}$,and predicts the conditional probability $p(t|h)$ for a tag t given the history h. The computation of this probability depends on a set of binary-valued "features" $f_i(h, t)$.

Given some training data and a set of features, the maximum entropy estimation procedure computes a weight parameter α_i for every feature f_i and parameterizes $p(t|h)$ as follows:

$$p(t|h) \quad = \quad \frac{\prod_i \alpha_i^{f_i(h,t)}}{Z}$$

where Z is a normalization constant.

The role of the features is to enumerate co-occurrences of histories and tags, and find histories that are strong predictors of specific tags. (for example, the tag "begin_caller" is very often preceded by the word sequence "this is"). If a feature is a very strong predictor of a particular tag, then the corresponding α_i would be high. It is also possible that a particular feature may be a strong predictor of the absence of a particular tag, in which case the associated α_i would be near zero.

Training a maximum entropy model involves the selection of the features and the subsequent estimation of weight parameters α_i. The testing procedure involves a search to enumerate the candidate tag sequences for a message and choosing the one with highest probability. We use the "beam search" technique of [7] to search the space of all hypotheses.

4.1. Features

Designing effective features is crucial to the maxent model, and in the following sections, we describe the various features that we experimented with. In all cases, we first preprocessed the text in the following ways: (1) we mapped rare words (with counts less than 5) into the generic word "UNKNOWN"; and (2) mapped words in a name dictionary to the symbol "NAME." The first step is a way to handle out-of-vocabulary words in test data; the second step takes advantage of words that are known to be names. This mapping makes the model focus on learning features which help to predict the location of the caller identity and leaves the actual specific names for later extraction. We describe the features in detail next.

4.1.1. Unigram lexical features

Unigram features consider the combination of at most one word and a specific tag. However, the word need not be the immediately preceding word. To compute these features, we used the

	Features		
$\forall w_i$	$w_i = X$	&	$t_i = T$
	$t_{i-1} = X$	&	$t_i = T$
	$t_{i-2}t_{i-1} = XY$	&	$t_i = T$
	$w_{i-2}w_{i-1} = XY$	&	$t_i = T$
	$w_{i-1}w_i = XY$	&	$t_i = T$
	$w_i w_{i+1} = XY$	&	$t_i = T$
	$w_{i+1}w_{i+2} = XY$	&	$t_i = T$

Table 2: Bigram features of the current history h_i. Two unigram feature templates are also included to improve performance.

neighboring two words, and the tags associated with the previous two words to define the history h_i as

$$h_i = w_i, w_{i+1}, w_{i+2}, w_{i-1}, w_{i-2}, t_{i-1}, t_{i-2}$$

The features are generated by scanning each pair (h_i, t_i) in the training data with feature template in Table 1.

4.1.2. *Bigram lexical features*

The trigger phrases used in the rule-based approach generally comprise of several words, and turn out to be good predictors of the tags. In order to incorporate this information in the maximum entropy framework, we decided to use ngrams that occur in the surrounding word context to generate features. Due to data sparsity and computational cost, we restricted ourselves to using only bigrams. The bigram feature template is shown in Table 2.

4.1.3. *Dictionary features*

Dictionary features capture a-priori knowledge of word classes and phrases. To do this, we tag each word in the training and test data as being either a number, part of a pre-caller phrase, part of a post-number phrase, or other. This is done with two dictionaries: one containing numbers, and the other for phrases. The stream of dictionary codes c_i is then added to the words w_i and tags t_i to form an enhanced history on which features are defined.

4.1.4. *Learning from errors*

To learn from errors, we first use the current maximum entropy model to decode the training data, and then enhance the history h_i by adding the decoded tags. We then generate new features and train a new model which corrects some errors made by the old model.

4.2. Feature selection

The universe of possible features is extremely large, and in practice must be reduced. The simplest way of doing this is to impose a feature-count cutoff, and, for example, ignore features whose counts are less than 10. This method results in about $10,000$ features. Even smaller models can be obtained with an incremental feature selection scheme where we start with a uniform distribution $p(t|h)$, and no features, and at every iteration add a new batch of features to the existing set. The procedure stops when the gain in likelihood on a cross-validation set becomes small.

	P/C	R/C	F/C	P/N	R/N	F/N
baseline	72.9	67.8	70.3	81.1	83.3	82.2
ME1-U	87.9	75.3	81.1	90.2	77.8	83.5
ME1-B	88.8	79.8	84.1	88.1	78.1	82.8
ME2-U-f1	87.9	75.8	81.4	89.7	82.3	85.8
ME2-U-f1-L	88.2	76.1	81.7	90.0	85.9	87.9
ME2-U-f12	87.3	77.6	82.2	89.5	82.7	86.0
ME2-B-f12	88.3	80.0	83.9	89.3	82.7	85.9
ME2-U-f12-I	86.9	77.5	81.9	88.8	81.2	84.8
ME2-B-f12-I	87.0	78.9	82.8	90.3	82.4	86.2

Table 3: Precision and recall rates for different systems on manual voicemail transcriptions.

5. Experimental Results

To evaluate the performance of the different systems, we use the conventional *precision*, *recall* and *F-measures*. We compute these both based on word agreement and time-interval overlap. The word-based metric measures the agreement between the automatically extracted portions of text and the desired portions, while the time-based metric measures the degree of temporal overlap between the actual speech data underlying the extracted and desired words. Significantly, in the word-based metric, we insist on **exact** matches for an answer to be counted as *correct*. The reason for this is that any error is liable to render the information useless, or detrimental. For example, an incorrect phone number can result in unwanted phone charges, and unpleasant conversations. This is different from typical named entity evaluation, where partial matches are given partial credit. Therefore, it should be understood that the precision and recall rates computed with this strict criterion *cannot* be compared to those from named entity detection tasks.

A summary of our results with the word-based metric is presented in Tables 3 and 4. Table 3 presents precision and recall rates when manual word transcriptions are used; Table 4 presents these numbers when speech recognition transcripts are used. On the heading line, P refers to precision, R to recall, F to F-measure, C to caller-identity, and N to phone number. Thus P/C denotes "precision on caller identity".

In these tables, the maximum entropy model is referred to as ME. We present results for the following variants. The number of features used in each is also indicated.

- ME1-U: unigram lexical features only (9704);
- ME1-B: bigram lexical features only (25828);
- ME2-U-f1: unigram features with a number dictionary (9722);
- ME2-U-f1-L: previous with error correction (9815);
- ME2-U-f12: ME2-U-f1 with the trigger-phrase dictionary features (9747);
- ME2-U-f12-I: previous with incremental feature selection (910);
- ME2-B-f12: bigram features and both dictionaries (25871);
- ME2-B-f12-I: previous with incremental feature selection (2125);

These results indicate that it is possible to achieve F-measure performance between 80 and 88% with a wide variety of feature schemes, operating on the reference scripts. Furthermore, the use of dictionary features systematically improves the recall of names and numbers by 3 to 5% relative. Similarly, the use of bigram features improves recall by 2 to 6 %, though with significantly more features than unigrams.

	P/C	R/C	F/C	P/N	R/N	F/N
baseline	21.7	17.2	19.2	52.3	53.8	53.0
ME2-U-f1	23.5	15.6	18.8	56.0	51.5	53.7

Table 4: Precision and recall rates for different systems on decoded voicemail messages.

	P/C	R/C	F/C	P/N	R/N	F/N
baseline	66.3	66.0	66.1	70.8	71.9	71.3
ME2-U-f1	82.8	72.1	77.1	84.4	81.3	82.8

Table 5: Precision and recall rates for different systems on replaced decoded voicemail messages.

When the incremental feature selection is used, the number of features is reduced by a factor of 10 with little performance loss. This indicates that the main power of the maxent model comes from just a small portion of the features.

We also note that the maximum entropy approach beats the baseline in terms of precision, and also on the recall of the caller's identity. We believe this is because the baseline has an imperfect set of rules for determining the end of a "caller identity" description. On the other hand, the baseline system has higher recall for phone numbers.

A comparison of Table 3 and Table 4 indicates that there is a significant difference in performance between manual and decoded transcriptions. As expected, the precision and recall numbers are worse in the presence of transcription errors. The degradation due to transcription errors could be caused by either: (i) words in the context surrounding the names and numbers being corrupted; or (ii) the information itself being corrupted. To investigate this, we replaced the regions of decoded text that correspond to the correct caller identity and phone number with the correct manual transcription, and redid the test.

The results are shown in Table 5. Compared to the results on the manual transcription, the precision and recall numbers for the maximum-entropy tagger are just slightly worse. This indicates that the **corruption of the information content** due to transcription errors is much more important than the corruption of the surrounding context.

If measured by the string error rate, none of our systems can be used to extract exact caller and phone number information directly from decoded voicemail. If, however, the extracted portions of speech have a significant time overlap with the information-bearing regions, then it is still possible to convey useful information by playing short segments of a message to a user. To investigate the feasibility of this approach, we computed precision and recall based on temporal overlap. To do this, we identified the time intervals of information extracted from the decoded scripts, and computed the amount of overlap with the correct information-bearing intervals. Denoting the overlap by V, the total amount (length) of information extracted from the decoded script by D, and the total amount present in the reference script by R, the performance is measured by Precision = V/D and Recall = V/R. Table 6 shows the results: an 80% F-measure for phone numbers and 50% F-measure for callers, indicating that useful phone-number information can be extracted.

	P/C	R/C	F/C	P/N	R/N	F/N
baseline	77.0	36.0	49.1	84.8	76.2	80.3
ME2-U-f1	73.2	40.5	52.2	84.6	78.6	81.5

Table 6: Precision and recall of time-overlap for different systems on decoded voicemail messages.

6. Conclusion

In this paper, we study how to extract caller information from voicemail messages. This information is useful for voicemail indexing and retrieval. In contrast to traditional named entity tasks, we are interested in identifying just a selected subset of the named entities that occur. We implemented a rule-based baseline and a maximum entropy system, and tested them on both manual transcriptions and transcriptions generated by a speech recognition system. Both the baseline and the maximum entropy model performed well on manually transcribed messages, but degraded significantly in the presence of speech recognition errors. Our results show that the degradation is due to recognition errors in the information bearing text - and not its in its surroundings, and that the time intervals containing phone numbers can be extracted even in the presence of a 35% word-error-rate in recognition.

7. References

[1] NIST. 1999. *Proc. of the Eighth Text REtrieval Conference (TREC-8).*

[2] DARPA. 1999. *Proc. of the DARPA Broadcast News Workshop*.

[3] David Miller, Sean Boisen, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. 2000. Named entity extraction from noisy input: Speech and ocr. ANLP-NAACL 2000, pp. 316–324.

[4] Ji-Hwan Kim and P.C. Woodland. 2000. A rule-based named entity recognition system for speech input. ICSLP-2000, Beijing, China.

[5] Konstantinos Koumpis and Steve Renals. 2000. Transcription and summarization of voicemail speech. ICSLP-2000, Beijing, China.

[6] J. Huang, B. Kingsbury, L. Mangu, M. Padmanabhan, G. Saon, and G. Zweig. 2000. Recent improvements in speech recognition performance on large vocabulary conversational speech (voicemail and switchboard). ICSLP-2000, Beijing, China.

[7] Adwait Ratnaparkhi. 1996. A Maximum Entropy Part of Speech Tagger. In Eric Brill and Kenneth Church, eds, *Conference on Empirical Methods in Natural Language Processing*, University of Pennsylvania, May 17–18.

[8] Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proc. of the Human Language Technology Workshop*, pp. 250–255, Plainsboro, N.J. ARPA.

[9] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. 1998. Nyu: Description of the mene named entity system as used in MUC-7. In *Seventh Message Understanding Conference(MUC-7)*. ARPA.