NAMED ENTITY EXTRACTION FROM SPONTANEOUS SPEECH IN HOW MAY I HELP YOU?^{SM,TM}

F. Béchet[*]*, A. Gorin, J. Wright, D. Hakkani Tur*

AT&T Laboratories-Research
180 Park Avenue, Florham Park, New Jersey 07932, USA
{bechet,algor,jwright,dtur}@research.att.com

ABSTRACT

The understanding module of a spoken dialogue system must extract, from the speech recognizer output, the kind of request expressed by the caller (the *call type*) and its parameters (numerical expressions, time expressions or proper-name). The definition of such parameters (called *Named Entities*, NE) is linked to the dialogue application. Detecting and extracting such *contextual* NEs for the *How May I Help You?*^{SM,TM} application is the subject of this study. By detecting NEs with a statistical tagger on 1-best hypotheses and by extracting their values with local models on word-lattices, we show very significant improvements compared to the traditional approach which uses regular expressions on the 1-best hypothesis only.

1. INTRODUCTION

A spoken dialogue system can be seen as an interface between a user and a database. The role of the Dialogue Manager (DM) module is to determine, firstly what kind of query the database is going to be asked and secondly with which parameters. In the *How May I Help You?* (HMIHY) application [1], currently deployed for the AT&T customer care service, if a user wants his account balance, the query will be *accessing the account balance field of the database* with the customer identification number as the parameter.

Such database queries are named *call type* and their parameters are the information items, independent from the call type, which are contained in the user's request. They are often called *Named Entities* (NEs) and they refer to three different kind of information: proper-name identifiers, time identifiers and numerical expressions.

In the framework of spoken dialogue system, the definition of a NE is associated to its meaning for the application targeted. For example, in the HMIHY customer care application, most of the relevant time or monetary expressions are those related to an item on a customer's bill (the date of the bill, a date or an amount of a call for example). In this paper, we will refer to *contextual named entities* for expressing this restricted definition of NEs.

Detecting and extracting such contextual NEs tags and values from spontaneous speech in the HMIHY application is the subject of this study. After presenting related works and state-of-the-art techniques, we will describe our statistical NE tagger and the extraction models developed for processing the HMIHY NE tag set. Finally, we will give some results both for the detection and extraction tasks on a test corpus extracted from real dialogues between customers and the HMIHY service.

[*]also: LIA, University of Avignon, BP 1228 84911 Avignon Cedex 09, FRANCE email : *frederic.bechet@lia.univ-avignon.fr*

2. RELATED WORK

One task of the HUB-4 DARPA program was dedicated to NE extraction from Broadcast News speech data. It is therefore difficult to compare the results of this program and those which can be obtained on a spoken dialogue corpus: the kind of speech (spontaneous vs. broadcast news) and the NEs targeted are very different. Nevertheless, it is interesting to point out that most of the systems used for this evaluation follow a statistical tagging approach, which has proven to be much more robust to recognition errors and lack of text formatting than rule based methods [2].

Another work which can be related to our study is the spoken digit recognition task. [3] presents the performances of state-of-the-art digit recognizers on various speaking conditions. The digit error rate varies from less than 1% for read digit strings to more than 5% for conversational speech over the telephone. When digits are related to a NE, like a phone number or a time expression, the correct measure to consider is the *understanding error rate* rather than the digit error rate. This measure corresponds to the digit values of the NE considered regardless from the way they have been expressed. For example, the digit value corresponding to the NE phone number: $three\ six\ four\ forty\ two\ forty\ eight\ area\ code\ two\ oh\ one$ is: 201 364 4248. [3] reports an understanding error rate of 24% (with a 5.5% digit error rate) for phone number values expressed in a dialogue context, as an answer to the prompt: *Please give me your phone number*. The NEs, here, are represented by hand-written rules which translate the ASR output into digit strings.

Finally, the closest application our work can be related to is the contextual NE extraction task on the Voicemail database, which contains several hours of conversational telephone speech. [4] reports experiments on phone number extraction by means of several corpus-based and rule-based methods with an understanding error rate of 46% and a global WER of 35%. It is important to point out that this result can't be compared to the one given previously for digit string recognition, as the phone numbers in this case are embedded in the sentences and aren't the result of a direct query. The same difference of performance will be noticed on the HMIHY data according to the kind of prompt considered.

3. DETECTING NAMED ENTITIES

The NE detection module in itself, even with no value extraction, is an important module of a spoken dialogue system. For example, the tags detected can help the call-type classification process [5].

NEs are usually represented by either hand written rules or statistical models [2][4][6]. Even if statistical models seem to be much more robust to ASR errors, in both cases the models are derived from transcribed speech and the ASR errors are not taken into account explicitly by the models. This strategy certainly emphasizes the precision of the detection, but a great loss in recall can occur by not modeling the ASR behavior. For example, a word can be considered as a very salient information for detecting a particular NE tag. But if this word is, for any reason, very often badly recognized by the ASR system, its salience won't be useful on the ASR output.

For this reason, and thanks to the amount of data available, we have decided to explicitly model the ASR behavior in our contextual NE tagger. After presenting which context is considered salient for identifying NEs, we will describe the training process of our NE tagger, on the ASR output, with constraints derived from the transcribed speech.

3.1. Selecting NE context

As it has been previously shown, not all the NEs are considered relevant for the DM. Therefore, our NE tagger must model not only the NE itself (e.g. *20 dollars and 40 cents*) but its whole context of occurrence (e.g. *... this 20 dollars and 40 cents call ...*) in order to disambiguate relevant NEs from others.

An automatic selection method, for these relevant NE contexts, has been developed. The input of this process is a corpus of transcribed speech dialogues with, for each sentence, the list of NE tags included in each of them according to the manual labelers. This method uses both statistic and syntactic criteria in the following way:

1. a statistical text classifier is first trained on the corpus in order to classify each sentence according to the list of NE tags contained in it;
2. for each sentence, all the words or group of words selected by the classifier for characterizing the sentence according to the NE tags are marked;
3. then, a Part-Of-Speech tagging and a syntactic bracketing are performed on the corpus;
4. finally, the relevant context of a NE tag in a sentence is the concatenation of all the syntactic phrases containing a word marked by the classifier.

After processing the whole training corpus, two corpora are attached to each tag: the corpus C_{NE} containing only NE contexts (e.g. `<PH>my phone area code d3 number d7</PH>`) and the corpus C_{BK} which contains the background text without the NE contexts (e.g. `I'm calling about <PH> </PH>`). In both cases, non-terminal symbols are used for representing numerical values and proper names.

3.2. NE tagger

3.2.1. Probabilistic model

For the NE detection process, we use a probabilistic tagging approach similar to the one presented in [7]. This is a Hidden Markov Model (HMM) where each word of a sentence is emitted by a state in the model. The hidden state sequence $S = (s_1 \dots s_N)$ corresponds to the following situations: beginning a NE, being inside a NE, ending a NE, being in the background text.

Finding the most probable sequence of state which have produced the known word sequence $W = (w_1 \dots w_N)$ is equivalent to maximizing the probability: $P(S|W)$. By using Bayes'rule and the assumption that the state at time t is only dependent on the state and observation at time $t-1$, we obtain equation 1.

$$P(S|W) \approx arg\max_{S} \prod_{t=1}^{N} P(w_t|w_{t-1}, s_t) P(s_t|s_{t-1}, w_{t-1}) \quad (1)$$

The first term, $P(w_t|w_{t-1}, s_t)$, is implemented as a state-dependent bigram model. For example, if s_t is the state *inside a* `PHONE`, this first term corresponds to the bigram probability $P_{phone}(w_t|w_{t-1})$ estimated on the corpus C_{NE} introduced in the previous section. Similarly, the bigram probability for the background text, $P_{bk}(w_t|w_{t-1})$, is estimated on the corpus C_{BK}.

The second term is the state transition probability of going from the state $t-1$ to the state t. These probabilities are estimated on the training corpus, once the NE context selection process has been done.

3.2.2. Modeling the ASR system behavior

Taking into account the recognition errors explicitly in the models which represent the NEs is one of the main sources of originality of this work. In our method, the whole training corpus is processed by the ASR system in order to learn automatically the confusions and the mistakes which are likely to occur in the deployed system. This ASR output corpus is then aligned, at the word level, with the transcription corpus. During the training of the state-dependent bigram model presented previously, the corpora C_{NE} and C_{BK} are replaced by their corresponding sections in the ASR output corpus.

Such a method balances the inconvenience of learning directly a model on a very noisy channel by structuring the noisy data according to constraints obtained on the clean channel. This leads to an increase in performance like it will be shown in table 1.

3.3. Tagging Named Entities

The NE tagging process consists in maximizing the probability expressed by equation 1 by means of a search algorithm. In order to be able to tune the precision and the recall of our model for the deployed system, each NE detected by our tagger is scored by a text classifier.

The text classifier is trained as follow:

1. the ASR output of the training corpus is processed by the NE tagger with no rejection;
2. on one side, all the NEs which are correctly tagged according to the manual labels are kept;
3. on the other side, all the false positive detections are labeled with the tag `OTHER`;
4. then the text classifier is trained in order to separate the NE tags from the `OTHER` tags.

During the tagging process, the scores given by the text classifier are used as confidence scores to accept or reject a NE tag according to a given threshold.

4. EXTRACTING NE VALUES FROM ASR OUTPUT

ASR systems can generate, as output, a word-lattice as well as the highest probability hypothesis called the 1-best hypothesis. In the HMIHY customer care application, the Word-Error-Rate (WER) of the 1-best hypothesis is around 30%. However, by performing an alignment between the transcribed data and the word lattices produced by the ASR system, the WER of the aligned corpus (called the *oracle WER*) dropped to around 10%.

Based on these results, we developed a 2-step approach for extracting NE values: firstly, because we can't predict exactly what the user is going to say after a given prompt, we detect the NEs on the 1-best hypothesis produced by the ASR system; secondly, once we have detected areas in the speech input which are likely to contain NEs with a high confidence score, we extract the NE values from the word lattice with a local model (specific to each NE tag) but *only* on the areas selected by the NE tagger.

These local models are regular grammars coded as Finite-State-Machines (FSM) which are automatically obtained, from the training corpus, as follows:

1. in a first step, the transcribed training corpus is processed by the NE tagger in order to extract NE contexts on the clean text;
2. only the NE contexts correctly tagged according to the manual labels are kept;
3. then all the digits, natural numbers and proper-names are replaced by corresponding non-terminal symbols ;
4. finally all the patterns representing a given tag are merged in order to obtain one FSM for each tag, coding the regular grammar of the patterns found in the corpus.

The extraction process consists in a composition operation between the FSM associated to a NE tag and an area of the word-lattice where such a tag have been detected by the NE tagger. Because having a logical temporal alignment between words makes the transition between 1-best hypothesis and word-lattice easier, and because posterior probabilities are powerful confidence measures for scoring words, the word-lattices produced by the ASR system are first transformed into a chain-like structure as described in [8] (also called *sausages*).

Once the FSM is composed with the portion of the chain corresponding to the NE area detected, a search process looks for the best path according only to the confidence scores attached to each word in the chain. Indeed, the FSM is not weighed as all the patterns extracted from the training data are considered valid. Therefore, it's only the posterior probability of each sequence of word following the patterns which is taken into account. According to the kind of NE tag extracted, a simple filtering process of the best path in the FSM is performed in order to produce a value.

5. EXPERIMENTS

5.1. Experimental setup

The experiments reported in this section have been made with a 70K sentence training corpus extracted from real dialogues of the deployed application and a 20K sentence test corpus from the same period of time. The total WER is 27.4%. All the NE detection and extraction models have been trained on the same corpus and the text classifier used for training the NE tagger and scoring the NE contexts is a decision-tree classifier based on the Semantic-Classification-Trees introduced for the ATIS task by [9] and used for semantic disambiguation in [10].

5.2. Detection results

The detection results are given for the following contextual NEs: phone number and money expression referring to a cost on the bill. The phone number detection task is split in two situations:

- `phone(1)` corresponds to answers to the system prompt *Please give me your home phone number*;
- `phone(2)` corresponds to answers to all the other prompts.

The baseline results for the detection task are those obtained by using regular expressions for representing the NEs. These regular expressions detect phone numbers and monetary values expressed in various standard ways. Table 1 presents the precision, recall and F-measure results obtained on the ASR output with such regular expressions and those obtained by mean of our NE tagger with no rejection. The column *F'* corresponds to the results obtained when the tagger is trained only on dialogue transcriptions and not on the ASR output like presented in section 3.2.2.

ASR output	**Regular expression**			**NE tagger**			
Detection	P	R	F	P	R	F	*F'*
phone(1)	97.6	90.7	**94**	97	95.9	**96.2**	*92.6*
phone(2)	71.6	80.7	**75.9**	76.1	88.9	**83.7**	*79.6*
money	52.9	71.7	**60.9**	58.1	76.2	**70**	*67.4*

Table 1. Precision, Recall and F-measure results for the NE detection task

One can see that a very significant improvement in the F-measure is obtained by means of the NE tagger, specially thanks to the recall measure. Using the ASR output for training the models leads also to an improvement by increasing the robustness to recognition errors.

Another main advantage of the NE tagger approach is the possibility to tune the acceptance or the rejection of a NE according to its confidence score given by the text classifier. Figure 1 shows the ROC curves for the detection of the phone number NE tags. The results are compared with those obtained with a standard regular expression approach. Each point of the regular expression curves corresponds to a different value in the number of digits that a phone number must contain in order to be detected.

For example, with an operating point of 20% false rejection, the precision for the tag `phone(2)` jumps to 85% while for the same false rejection rate, the regular expression approach only obtains a precision below 75%.

5.3. Extraction results

The NE value extraction process has been evaluated on the phone number tag only as the reference values are not yet available for the other tags in the HMIHY? transcribed data. Our baseline is the extraction process made on the 1-best hypothesis with a regular expression. Table 2 shows the results obtained by applying local models to word-lattices on the areas detected by the NE tagger.

These results presents the understanding accuracy, which indicates a perfect match between the whole digit string representing the phone value in the transcribed corpus and the one extracted

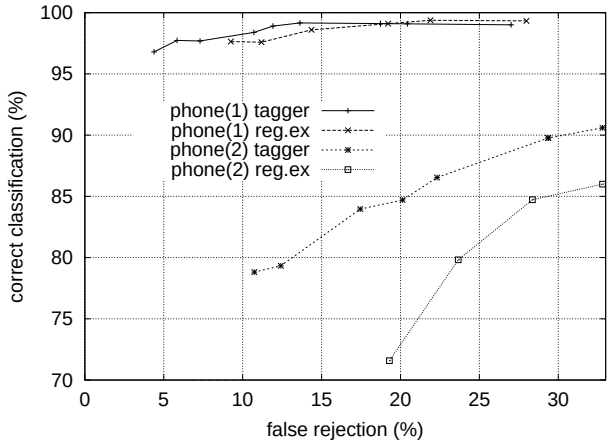

Fig. 1. ROC detection curves for phone number tags

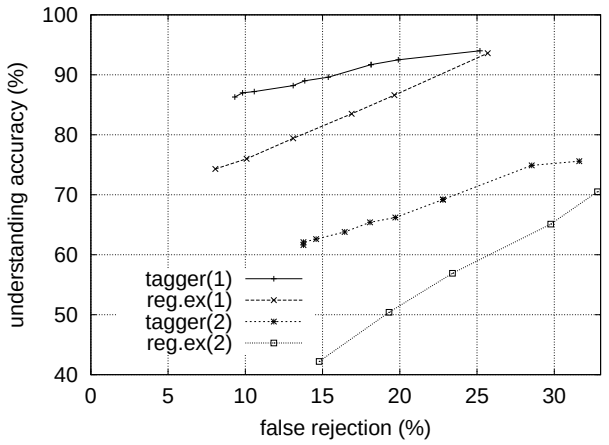

Fig. 2. Understanding accuracy according to false rejection rate

by our models. Table 2 reports the amount of correct phone number values compared to the total amount of phone numbers in the transcription corpus.

Figure 2 shows the understanding accuracy as a function of the false rejection rate. To obtain these curves we compare the amount of correct phone number values to the amount of phone number tags detected by either the tagger or the regular expression approach.

tag	Regular expression	NE tagger
phone(1)	70%	**79.3%**
phone(2)	52.6%	**61.2%**

Table 2. Understanding accuracy on the ASR output

By choosing an operating point of 15% false rejection for the tag `phone(1)` and 25% for the tag `phone(2)` we obtain an increase of almost 10% absolute in the understanding accuracy between the regular expression approach and the NE tagger.

6. CONCLUSION

The results obtained by our NE tagger, both for detection and extraction, show a very significant improvement compared to those obtained by a standard regular expression approach. Even for the tag `phone(1)`, where the detection task is easy (if *any* sequences of digits is recognized, then it's a phone number), we obtain an absolute 5% improvement for the detection recall and an absolute 9% improvement for the understanding accuracy. This validates our 2-step approach which uses 1-best hypotheses for detecting NEs and word lattices for extracting values.

7. REFERENCES

[1] Gorin A., Riccardi G., and Wright J., "How May I Help You ?," in *Speech Communication*, 1997, vol. 23, pp. 113–127.

[2] D. M. Bikel, R. Schwartz, and R. Weischedel, "An algorithm that learns what's in a name," *Machine Learning : Special Issue on Natural Language learning*, vol. 34, no. 1-3, pp. 211–231, 1999.

[3] M. Rahim, G. Riccardi, L. Saul, J. Wright, B. Buntschuh, and A. Gorin, "Robust numeric recognition in spoken language dialogue," in *Speech Communication*, 2001, vol. 34, pp. 195–212.

[4] Huang J., Zweig G., and Padmanabhan M., "Information extraction from voicemail," in *Proceedings of the 39th Annual Meeting Association for Computational Linguistics, Toulouse, France*, 2001, pp. 290–297.

[5] Wright J. H., Gorin A. L., and Riccardi G., "Automatic acquisition of salient grammar fragments for call-type classification," in *Proceedings of EUROSPEECH'97, Rhodes*, 1997.

[6] Kim J.H. and Woodland P.C., "A rule-based named entity recognition system for speech input," in *Proceedings of ICSLP'2000, Beijing, China*, 2000.

[7] David D. Palmer, Mari Ostendorf, and John D. Burger, "Robust information extraction from spoken language data," in *Proceedings of EUROSPEECH'99, Budapest*, 1999.

[8] Mangu L., Brill E., and Stolcke A., "Finding consensus among words: lattice-based word error minimization," in *Proceedings of EUROSPEECH'99, Budapest*, 1999, pp. 495–498.

[9] Kuhn R. and De Mori R., "The application of semantic classification trees to natural language understanding," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 449-460, 1995.

[10] F. Bechet, A. Nasr, and F. Genet, "Tagging unknown proper names using decision trees," in *38th Annual Meeting of the Association for Computational Linguistics, Hong-Kong, China*, 2000, pp. 77–84.