Combining word-based features, statistical language models, and parsing for named entity recognition

Joseph Polifroni1, Stephanie Seneff2

1 Nokia Research Center
4 Cambridge Center, Cambridge, MA 02142 USA
2 MIT CSAIL
32 Vassar Street, Cambridge, MA 02139 USA

joseph.polifroni@nokia.com, seneff@csail.mit.edu

Abstract

As users become more accustomed to using their mobile devices to organize and schedule their lives, there is more of a demand for applications that can make that process easier. Automatic speech recognition technology has already been developed to enable essentially unlimited vocabulary in a mobile setting. Understanding the words that are spoken is the next challenge. In this paper, we describe efforts to develop a dataset and classifier to recognize named entities in speech. Using sets of both real and simulated data, in conjunction with a very large set of real named entities, we created a challenging corpus of training and test data. We developed a multi-stage framework to parse these utterances and simultaneously tag names and locations. Our combined system achieved an f-measure of 0.87 on extracted proper nouns, with a 95% accuracy on distinguishing names from locations.

1. Introduction

Large-vocabulary automatic speech recognition (ASR) is now accurate and fast enough for use in mass market applications. Speech as an input device for text messages, email, web searches, and even Facebook status updates has become relatively common for users of high-end mobile devices. These applications typically involve little or no understanding beyond the recognition of keywords and some other semantic types (e.g., contact names) within limited syntactic contexts. The logical next step is to incorporate some level of natural language understanding in order to allow these applications to perform more complicated interactions.

In one of the most successful paradigms for integrating speech into mobile devices, the user is encouraged to speak in an unconstrained way, and the task of the Human Language Technology is primarily to transcribe the speech. With the goal of maintaining the unconstrained nature of the interaction while adding a higher level of understanding, we describe here a system that preprocesses utterances for named entity extraction and achieves high parse coverage as a consequence.

The two-stage classifier makes a three-way distinction among name, location, and unclassed word, and outputs an FST encoding all possibilities for each word. The FST is then processed through a statistical language model to yield an N-best list of sentence candidates, which are then rescored by a syntactic parser to yield the final selection. For training data, we used a combination of real user data provided by a Nokia partner, data collected from users in a laboratory setting, simulated data from a calendar application, points of interest from Navteq, and census data.

The data profile is similar to that of data used for text-based named entity recognition, although many of the cues used by text-based systems, such as punctuation and capitalization, are unavailable with speech. In simulating the results of the ASR component, we have thus far avoided the compounding effect of errors from the ASR engine. We have intentionally designed the various components, however, with robustness in mind, and make use of only local context for classification. At further levels of processing, we rely on highly abstracted representations of words to ameliorate sparse data effects.

The remainder of the paper begins by describing some previous work in this field. Section 3 defines the specific problem we are addressing in more detail, and Section 4 describes the data simulation process. Section 5 provides a detailed description of our methodology, including the two-stage classifier, statistical language model, and parser. Section 6 presents results on named entity extraction and parse coverage. We conclude with a look to future work.

2. Previous work

Named entity recognition technology has been applied to large speech corpora, most notably Broadcast News, associated with the DARPA HUB-4 task [2, 3, 4], or with similar corpora in Chinese [5] or French [6]. Bechet et al. [7] extract named entities from spontaneous speech within the HMIHY corpus from AT&T. They concentrate on the extraction of phone numbers from utterances spoken to a customer care application. Huang et al. [8] and Jansche and Abney [9] perform named entity extraction on two separate speech corpora of a similar nature, i.e., voicemail transcripts. In both cases, they were looking for “caller phrases,” phrases within the voicemail, typically near the beginning of the message, where the caller identifies him-/herself. In addition, caller phone numbers were extracted. The HMIHY and voicemail corpora, with their spontaneous nature, are closer to the type of speech we are trying to emulate.

More recent work at AT&T, focusing on voice search, has also sought to extract locations from spoken input, along with query search terms [10]. This work attempts to parse spoken
3. Defining the problem

We began our investigations with the set of data from an industrial partner, comprised of short notes-to-self dictated on Nokia mobile phones, by real users, using an ASR system. We call this set the NoteToSelf data. One of the first things we noticed about these data were the large number of utterances (approximately 17%) that contained reminders to “go to” particular locations, “pick up” or “buy” things, also at particular locations, or “call” or “email” particular people. In addition, a further 11% contained references to calendar entries (e.g., “meeting” or “appointment”) or specific time expressions. Using a combination of automated and hand-crafted means, we derived a subset of these data that contained names, locations, and time expressions, examples of which are shown in Table 1.

From these data, it was clear that a rule-based system for parsing, or even extracting keywords, would not be completely successful. The sorts of things people were leaving as reminders included a range of activities, errands, and daily occupations that would be difficult to anticipate, even with a user model that incorporated contact lists and geo-location data. At the same time, it was also clear that these utterances contained information that could be usefully processed to, for example, fill in a calendar or provide a link to a webpage or map.

Nokia itself had collected a large corpus of SMS messages, dictated by real users albeit in a laboratory setting. These messages had been transcribed and lightly annotated (i.e., for numbers). This corpus, which we will refer to as SMSData, gave us further insight into both variability and consistency surrounding named entities in this type of data. For example, users said both toki and toki seafood buffet to refer to a particular restaurant, yet each instance was preceded by either “to” or “at”. Likewise the fast-food chain in Southern California was called in and out, in and out burger and in and out burgers yet consistently had “from”, “at”, or “to” in front of it.

We also had a set of calendar utterances spoken to a dialogue system developed jointly by Nokia and MIT to interface to an online diary. This system allowed users to add, change, and move calendar entries using voice. We were able to mine the approximately 1300 utterances collected for this domain to provide prototypes for the sorts of calendar-type interactions seen in the NoteToSelf data.

4. Creating the data sets

4.1. Utterance patterns

Altogether, using NoteToSelf, calendar, and SMSData, we created approximately 450 unique templates, which we randomly assigned to training (80%) and test (20%) sets. Table 2 shows example templates. Although there is an overlap in the filler words used in each template, the sentence patterns are different between training and test sets and the named entities used to instantiate the variables are different (see below).

4.2. Named entities

Through Navteq, a subsidiary of Nokia, we had access to over 8 million points of interest within the United States.1 Within the Navteq points of interest data themselves there was variability in how entities were represented. Some restaurant names, for example, encoded a cuisine type (e.g., “peach blossom chinese restaurant”) while many did not. The designator word for restaurant was variable (e.g., “restaurant”, “cafe”) and, in many cases, absent from the name (“el amigo”) in our data.

From the same source as the NoteToSelf data, we also had a set of contact names comprising approximately 170K unique entities. The users and contact lists were completely anonymous, but we took the added precaution of extracting given and surnames separately and randomizing these.

Each set of variables (i.e., grocery, restaurant, etc.) was filtered to contain only unique entries. Unique entries for each category were divided into training (80%) and test (20%) sets. There was no overlap in the named entities for any category between training and testing.

Aliases for variables were created algorithmically for 50% of each set by removing common words/terms such as “cafe,” “coffee shop,” or “restaurant” (+ cuisine type) from the full variable names. The resulting set included variants such as “carl’s” and “carl’s coffee shop”, as well as “ho sai kai chinese restaurant”, “ho sai kai restaurant”, and “ho sai kai”. For a more complete analysis of the data we created, and a comparison with other text corpora for named entity recognition, see [11].

A big advantage of data simulation is that the simulated utterances are automatically annotated. Each word in each instantiated variable is marked as being either a name or a location.

5. System Configuration

This section describes the three components of our system in more detail: classification, language modeling, and parsing.

1Navteq has equivalent data worldwide, which we hope to use when expanding the work to other countries/languages.

2Proper names are not capitalized in this paper, reflecting ASR output of such words.
Table 3: Perplexity and entropy as measured on two separate letter bigram language models, one constructed using words and one constructed using names. The measures are shown on data drawn from common words and names.

<table>
<thead>
<tr>
<th></th>
<th>Word Bigram</th>
<th>Name Bigram</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Perplexity</td>
<td>Entropy</td>
</tr>
<tr>
<td>Names</td>
<td>17.8</td>
<td>4.15 bits</td>
</tr>
<tr>
<td>Words</td>
<td>13.74</td>
<td>3.78 bits</td>
</tr>
<tr>
<td></td>
<td>14.72</td>
<td>2.88 bits</td>
</tr>
<tr>
<td></td>
<td>15.70</td>
<td>3.97 bits</td>
</tr>
</tbody>
</table>

5.1. Classification

The classification process used a maximum entropy-based classifier. Output were three classes for each word, name, location, or unclassed word, with associated probabilities. This output was encoded as an FST, with arcs corresponding to each hypothesis type. For classification, we distinguished between context-independent features, which involve only the word itself, and context-sensitive features, which utilized neighboring contexts.

5.1.1. Context-independent features

One powerful low-level feature utilizes the statistical properties of letter sequences within common words vs. proper nouns. We trained letter bigram and trigram models on approximately 80K common words in English, and a parallel letter bigram model on approximately 100K common given and surnames taken from the US Census Bureau. We used entropy as a context-independent feature, measured for each word against trigrams computed from words and from names.

5.1.2. Context-dependent features

In a real world setting, our classifier would be able to take advantage of the contact list for an individual user. Any word that appears within an utterance that is also in a user’s contact list is more likely to be a name (although, obviously not guaranteed to be so, e.g., “bill” in “remember to pay the phone bill”). To simulate the use of contact lists data, we assembled given names and surnames into “full names” and created separate sets for training and testing. When any word in these sets was needed to instantiate a variable in an utterance (e.g., “givenname”, “surname”, or “fullname”), we chose from entirely separate sets of exemplars for training and test utterances. We investigated the relative contribution of this knowledge to overall performance by conducting two separate experiments, simulating either 50% or 70% match statistics with contact list entries.

In addition to the features mentioned above, we used the presence of each word in dictionaries of common words and common locations, as well as the word’s unigram score calculated on the Google 1 T 5-gram corpus.

5.2. Language Modeling

We felt confident that language modeling should be able to improve performance over the original classifier output, particularly if we could capture the statistics on multi-word sequences representing locations. On the other hand, we wanted to carefully avoid sparse data problems, so that we could take advantage of sentence-structure statistics learned from a small set of sentence patterns and hope that they would apply to a different set of patterns (those in the test templates). We therefore mapped most of the words to the three classes, “word,” “loc,” and “name,” retaining as explicit words only the very common glue words such as auxiliaries and prepositions. We allowed only 100 words to survive as individual words in the language model. An additional 2000 words were retained but incorporated into a single “common word” class in the language model (these were established from a set of common English words found on the web). Words belonging to obvious classes like month name and weekday were retained and grouped into their corresponding class for the class n-gram statistics. All of the remaining words were simply represented as “word.” To better capture statistics on location sequences, we grouped all locations into a single compound word encoding the number of location words in the unit (e.g., “loc_loc_loc_loc” for “Basketball Hall of Fame.”) Likewise, we used “name_name” to represent a first and last name sequence.

A class trigram language model was trained on the appropriately converted training sentences, and then applied on the test set by rescoring 10-best outputs from the classifier.
5.3. Parsing

Our main goal is to be able to parse these utterances and extract relevant [key: value] information to encode the intended meaning and enable the system to interpret and act on them in some way. We determined that, without the tags, the utterances are much more difficult to parse, requiring more time and often failing in part due to pruning and search constraints. The tags provide a major simplification of the parse task, since complex noun sequences can be grouped under the tagged-entity category without requiring any further analysis.

For parsing, we utilized the TINA framework [12] which is based on context free rules augmented with long distance constraints to handle movement and agreement, and includes a probability model that is typically automatically trained on a designated corpus. The grammar is lexicalized, and covers most patterns found in the sentence structure of English, especially for the spoken language. For these experiments, we trained a general-purpose syntax-based grammar on the utterances from our training set, and then used the parser to rerank N-best lists produced by the trigram language model. Our hope was that the parser would be able to improve upon the statistical language model by being able to better capture long-distance constraints of syntactic structure.

6. Results

We created two sets of utterances using our template-based methods: a 20,000 utterance training set generated from 80% of the template patterns, and a 4000 utterance test set, generated from the remaining 20% of the templates and with instantiations of names and locations from an orthogonal source from that used for the training data. These sets were augmented with 240 and 60 utterances, respectively, drawn from NoteToSelf, containing no names or locations.

We report results in terms of a two-stage process: (1) how well could it identify named entities as such, and (2) how well could it tag them as being a "name" or a "location." To be considered correctly detected, a name or location has to be found in its entirety, and without any spurious additional words.

One question we were interested in addressing is the degree to which explicit knowledge of the contact list might help in name recognition. The top part of Table 4 shows the results obtained when only 50% of the names were in the contact list; the bottom, the results when this number increases to 70%. One surprising outcome is that there is little difference in performance between the two sets. The 70% condition gives a slight edge on correctly tagging the entity as a name or location, but actually comes out a bit worse on f-measure, when only the classifier is used.

The trigram language model and the parser were both able to substantially improve performance, for both experiments. Both recall and precision improved, as well as the tag label. The only exception is the tag label for the trigram language model in the 70% case, which dropped slightly from 94.6% to 94.5%. The best result for detection f-measure was 88.7%, obtained after parse selection for the 70% contacts presence condition.

Interestingly, the parser was able to produce a parse analysis for 97.7% of the utterances, a substantial improvement over its parsing coverage on utterances without the tags. Furthermore, only a tiny fraction of the utterances (0.7%) received a "robust"

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f measure</th>
<th>% correct tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>classifier</td>
<td>0.777</td>
<td>0.821</td>
<td>0.798</td>
<td>93.3</td>
</tr>
<tr>
<td>trigram LM</td>
<td>0.798</td>
<td>0.854</td>
<td>0.825</td>
<td>93.8</td>
</tr>
<tr>
<td>parser</td>
<td>0.854</td>
<td>0.883</td>
<td>0.868</td>
<td>94.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>classifier</td>
<td>0.767</td>
<td>0.821</td>
<td>0.793</td>
<td>94.6</td>
</tr>
<tr>
<td>trigram LM</td>
<td>0.800</td>
<td>0.860</td>
<td>0.829</td>
<td>94.5</td>
</tr>
<tr>
<td>parser</td>
<td>0.854</td>
<td>0.887</td>
<td>0.870</td>
<td>94.9</td>
</tr>
</tbody>
</table>

Table 4: Results of named entity recognition for the case when 50% of the names were assumed present in the contacts list (top table) and when 70% of the names were assumed present (bottom table). A name or location is considered correct only if the entire name is represented perfectly, with no additional words.

We report analysis as opposed to a “full parse” analysis. These results are very encouraging, and lead us to believe that “understanding” in addition to “recognition” is well within reach.

7. Acknowledgements

We thank Nokia for the MIT-Nokia collaboration, as well as Navteq for the use of their extremely valuable database.

8. References