

# Bootstrapping Named Entity Extraction for the Creation of Mobile Services

Joseph Polifroni, Imre Kiss, Mark Adler

Nokia Research Center Hollywood/Cambridge  
4 Cambridge Center  
Cambridge, MA 02142 USA  
joseph.polifroni@nokia.com, imre.1.kiss@nokia.com, mark.adler@nokia.com

## 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 use these data to develop a classifier to identify *names* and *locations* on a word-by-word basis. In this paper, we describe the process of creating the data and determining a set of features to use for named entity recognition. We report on our classification performance on these data, as well as point to future work in improving all aspects of the system.

## 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. In order to maintain the unconstrained nature of the interaction while adding a higher level of understanding, we describe here our initial experiments in developing a classifier to extract names and locations, along with a regular expression matcher.

We examine here a method for rapid development and subsequent refining of such a classifier for mobile devices using named entity extraction technology. To train the classifier, we used a combination of real user data provided by a Nokia partner, data collected from real users in a laboratory setting, simulated data, points of interest from Navteq, and census data. The classifier made a three-way distinction, i.e., whether the word was a name, a location, or an unclassified word, and output a probability for each.

The purpose of this paper is to discuss how we used the data at hand, in combination with simulated data, to train a classifier for use on real user data. We begin by describing some previous work in this field. Section 3 defines the specific problem we are addressing in more detail. In Sections 4 and 5 we describe how we create the simulated data we used for our experiment, using a variety of real-world resources. Section 6 analyzes the resulting data and compares and contrasts it to other corpora used in this field. In

Section 7, we discuss the classifier we developed to perform the named entity recognition. Section 8 describes the initial results and, in Sections 9, we discuss results and enhancements we are currently working on to improve the system.

## 2. Previous work

Named entity recognition is a well-established research area, with much of the initial research focussed on named entity recognition in text. Recognized entities typically fall into broad categories such as *person*, *location*, or *organization* as defined for the MUC-7 Conference (Chinchor, September 1997). In addition some systems recognize number sequences such as phone numbers or dates (Béchet et al., 2004; Favre et al., 2005). In this work, we focus on identifying *names* and *locations*. (*Dates* and *times* are also recognized and canonicalized by our system but that is outside the scope of this paper.)

In applying named entity recognition technology to large text corpora, effort can be placed in either the recognition of the entities (Canada et al., 2006) or the extraction of large lists of similar named entities (Etzioni et al., 2005). In both cases, the types of named entities is expanded as new categories are automatically acquired. This work critically depends on the information that can be found in web-based corpora, such as recurring patterns that indicate particular types of nouns/noun phrases (Mikheev et al., 1999) or the identification of probable proper noun phrases using parsing techniques (Collins and Singer, 1999).

Because we seek to identify named entities in utterances that are either typed or spoken to a mobile device, we cannot count on things like punctuation or capitalization. Furthermore, we cannot count on grammaticality, even in the absence of recognition errors. Although context is important (e.g., the presence of a name in a list of contacts), the input is informal and so the presence of certain markers for *person*, e.g., a title such as *Mr.* (Bikel et al., 1999; Kubala et al., 1998) cannot be relied on. In this respect, the work is more similar to that done on information extraction from voicemail (Huang et al., 2001; Jansche and Abney, 2002). Recent work at AT&T, focussing on voice search, has also sought to extract locations from spoken input, along with

User1	meet <b>michelle</b> at two o'clock
User2	get <b>jan</b> a ticket for <b>mark e.</b> at <i>cafe veritas</i>
User3	stop by to see <b>miranda</b> at <i>william beaumont</i>
User4	take <b>jay</b> to <i>basketball hall of fame</i> on june fourteenth

Table 1: Example utterances with **names** shown in boldface and *locations* shown in italics.

SMS1	will you go to wal-mart and get a new television remote
SMS2	i'm meeting up with jeany paul and shane tonight for dinner at the olive garden
SMS3	i am going to fly into ontario airport around six pm
SMS4	let's meet for dinner at that korean barbecue place on western

Table 2: Example SMS messaged dictated by users in a laboratory setting.

query search terms (Feng et al., 2009). This work attempts to parse spoken language input to impose a level of natural language understanding. The work we describe here, while not specifically focussed on parsing at present, can also be used to identify certain entities used by parsers, to help make parsing more reliable.

### 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. In examining these data, we noticed that, in a relatively large subset of the data, users referred to proper names, locations, and time expressions. Using a combination of automated and hand-crafted means, we derived a subset of these data that contained these three entity types, 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 contacts lists and geo-location data. Not all names mentioned in the data could be reliably assumed to be in a user's contacts list (e.g., **mark e.** in utterance *User2* in Table 1). Some names were, in fact, locations (e.g., *william beaumont* in utterance *User3*). Furthermore, the range of locations referred to could be well outside a geographic area identified for a user (e.g., utterance *User4*). 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.

In addition to the variability in the structure of how notes were expressed, we also noticed, unsurprisingly, indications of variability in how users referred to locations. The location in utterance *User3*, for example can be found in Web resources, as "william beaumont hospital", "beaumont medical center", "william beaumont army medical center", and "beaumont hospital," none of which match the actual string spoken by the user. This problem is common in natural language interfaces to location-based services, i.e., canonical representations found in gazetteers, etc. frequently do not match the ways users refer to places they are familiar with.

We experimented with various ways of discovering word units automatically, using Web-based resources, but our preliminary efforts produced noisy results. Furthermore, as seen in utterance *User3*, the variations users spoke could not be reliably discovered from how the named entities were represented on the Web. We, therefore, made the decision that we would not attempt to create word units from multi-word phrases, for either names or locations. This meant that the sequence of words denoting *location* in utterances labelled "User3" and "User4" in Table 1 would each be labelled as a *location*.

Although word units would presumably make the work of both the ASR engine and the named entity classifier easier, we want to allow users to say multi-word units in as natural a way as possible. To do this, we must be able to identify *any* sequence of words as a *name* or a *location*. This meant that, for example, in the name of the museum point of interest *basketball hall of fame*, each of the words were labelled as "location" and would have to be identified as such by our classifier.

Our experience with the *NoteToSelf* corpus helped us define the problem: a trainable, statistical method to extract information from spoken utterances, to enable a more semantically rich interaction. Specifically, we wish to extract named entities (subclassified in this work as either *person names* or *locations*) and time expressions (subclassified into dates and times). In addition to the initial set of note-to-self data described above, we had a variety of data resources at hand to help us address this problem. These will be described in the next section.

## 4. Creating the data sets

### 4.0.1. Utterance patterns

In addition to seeing the usefulness of such technology on a mobile platform, we were also encouraged to undertake this work because of a unique combination of knowledge/data sources we had at our disposal. As mentioned in Section 3, we had access to the *NoteToSelf* corpus, transcriptions of notes dictated by real users to an ASR application running on Nokia phones.

In addition to these data, 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). We will refer to this corpus as *SMS*. Table 2 shows an example of some of these utterances.

Template1	meeting with <b>:givenname</b> after work at <i>:coffeeshop</i>
Template2	go to <i>:grocery</i> to pick up [:groceryitem] [:groceryitem] and [:groceryitem]
Template3	let's go for a drink at <i>:restaurant</i> [:relday]
Template4	get to <i>:store</i> in <i>:city</i> before [:clocktime] [:day]

Table 3: Example templates for generating training and test data. Variables are preceded by colons. Variables classed as **names** are shown in **boldface**, *locations* in italics. Other variables such as time expressions are shown within square brackets.

The *SMS* corpus differed statistically from the *NoteToSelf* corpus. The average length of the utterances in the *NoteToSelf* corpus was relatively short, 4.9 words, while the average length of the *SMS* corpus was 9.3. This difference could reflect the fact that the users in the former were speaking to a real system, where the likelihood of error may have affected their willingness to speak for long periods of time. However, these data showed the same phenomena on the local level that we noticed with the *NoteToSelf* data, i.e., that certain contextual cues (e.g., prepositions) indicated names and locations.

#### 4.0.2. Named entities

Through Navteq, a subsidiary of Nokia, we had access to over 8 million points of interest within the United States.<sup>1</sup> 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”<sup>2</sup>) while many did not. The designator word for restaurant was variable, as well, (e.g., “restaurant”, “cafe”, “bar”) and, in many cases, absent from the name (“el amigo”) in our data.

From the same source from which we obtained 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.

## 5. Simulating data

The combination of the resources for utterance and named entity data meant that we had a reasonably large set of real utterances that we could use to create templates, and a very large set of named entities that we could use to instantiate variables within these templates.

Altogether, using *NoteToSelf* and *SMS* data, we created approximately 450 unique templates, which we randomly assigned to training (80%) and test (20%) sets. Table 3 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). 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%)

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

<sup>2</sup>Proper names are not capitalized in this paper, reflecting ASR output of such words.

Restaurants	Grocery Stores
mandarin wong	park's
mama lena's restaurant&pizza	george's international grocery
crabby jack's	dot's

Table 4: Example variables taken from 8 million Navteq points of interest. Approximately 50% of the data contain name “aliases,” i.e., versions of the name from which designators such as “restaurant” or “market” had been removed.

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”. Table 4 shows an example of variables for restaurant and grocery store names.

One advantage of simulating data, at least in the initial stages of system development, is that we were able to derive annotated training data for the classifier at the same time that we created our utterances. As each variable is instantiated, it is marked, word-by-word, as being either *name*, *location*, or *unclassified word*.

## 6. Data analysis

Palmer and Day(1997) perform a statistical analysis of corpora comprised of newswire articles used for the MUC-6, MET, and TIPSTER evaluations. Although our task and corpus are both quite different, we thought a comparison between the two corpora would be illustrative. Palmer and Day looked at the *type-token distinction*, as a way of understanding the frequency of occurrence of words in the corpora. They report on the token/type ratio, a ratio of the number of words in the text to the number of unique words. For the English corpora, they found a ratio of 4.3. In our corpus, we found a much higher ratio, of 19.3. This is closer to the value we found in the *NoteToSelf* corpus, i.e., 13.9. We attribute this to the fact that spoken corpora tend to have fewer unique words in general. In fact, in all three of the corpora we used, *NoteToSelf*, *SMS*, and the simulated data we created, we found that the top-1000 most frequently occurring words accounted for between 87 and 91% of the total words.

The measure Palmer and Day call *vocabulary transfer rate*, the percentage of words that occur in both the training and test corpus, highlights another difference between their text data and our simulated data. Palmer and Day examined this

	Palmer&Day	SMS data
All NEs	21.2%	8.6%
Locations	42.7%	11.4%
Person names	13.2%	2.0%

Table 5: Transfer rates for named entity types in the data analyzed in Palmer & Day (1997) compared with the data in the our simulated data.

metric for named entities in their corpus. They found an overall transfer rate of 21.2% for the English data in their corpora, for all named entity types<sup>3</sup>, compared with 8.6% in our data. Furthermore, the transfer rates for *location* named entities was 42.7% in the Palmer and Day corpora, compared with 11.4% in our data. For *person* names, the Palmer and Day corpora had a transfer rate of 13.3% compared to only 2.0% in our data. Table 5 summarizes these statistics. This also shows the effects of our separation of names and locations into exclusive training and test sets, meaning that there was very little overlap at the word level for these types.

These numbers reflect the nature of the corpora we set out to create. Our primary interest was in testing the ability of our classifier to identify sequences of words that were acting as *names* or *locations*. To that end, we wanted as rich a set as possible of named entity types comprising those sets.

## 7. Classification

### 7.1. Determining features

as either *name*, *location*, or *unclassified word*. One of the first features we looked at was part-of-speech (POS). Previous studies have shown the effectiveness of using POS information for identifying named entities (Collins and Singer, 1999), although these studies were done on text resources. We performed experiment on the utterance templates while in the initial stages of our investigation to determine if POS tagging would work for our data set. We generated a set of parallel utterances labelled for part-of-speech in a way similar to the method we used to automatically annotate named entity types and we used these data to train an off-the-shelf POS tagger. Templates were annotated using a total of 18 part of speech tags, including ones for singular and plural nouns, three different types of verbs (base form, third person singular, other), adjective, and adverb. The tag for proper noun was automatically attached when location and name variables were instantiated.

The performance of the POS tagger on our data was extremely unreliable. Performance improved when we added more utterances from the test set into our training, but we still felt that convergence was far off. Although it is a simple process to add new POS templates to the data when new patterns are found, we also wanted to avoid this stage if possible, given that it is hand-crafted and requires some expertise.

In a real world setting, a feature that would be available to our classifier is associated with the contacts list for an

<sup>3</sup>The named entity types in the Palmer and Day corpora included *person*, *location*, and *organization*.

	Word Bigram		Name Bigram	
	Perplexity	Entropy	Perplexity	Entropy
Names	17.8	4.15 bits	14.72	2.88 bits
Words	13.74	3.78 bits	15.70	3.97 bits

Table 6: 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.

individual user. Any word that appears within an utterance that is also in a user’s contacts list is likely to be name. However, a simple approach of simply flagging any word in the contacts list as a name could result in false positives. A contact named “bill” is not being referred to in an utterance such as “remember to pay the phone bill.” One advantage of the statistical approach we are taking is that the presence of the word in the contacts list alone is not the sole determiner of a word’s identity. Other contextual cues, as captured in the features listed in Section 7.2, provide further evidence of a word’s identity.

To simulate the use of contacts lists data, we first extracted single-word entries in the “given” and “surname” fields from the contacts data we had. We then randomized these two fields and reassembled entries from each into simulated full names, giving us three sets of contacts data, separated into training and test components. We randomly assigned entries in each of these sets to simulated user ids created for training and test utterances. When any of these three sets were needed to instantiate variables in the utterances (e.g., “:givenname”, “:surname”, and “:fullname”), we chose from utterances in the simulated user’s contacts list a given proportion of the time. This meant that, for a given proportion of the proper names in our dataset, we could use the feature *:inContactsList* set to “true.” For the results reported here, that proportion is 50%, i.e., 50% of the names used in the utterances were marked as members of a user’s contacts list.

Another feature that we examined in our preliminary investigations were properties associated with words themselves. Although we had assumed that contextual information would be important to our classifier (e.g., identity of preceding preposition), we knew that it alone would not be sufficient. Our experiments had shown statistical differences in the contextual patterns of letters within words vs. those found in names. These experiments involved training a letter bigram model on approximately 80K common words in English (Beale, 2003), and a parallel letter bigram model on approximately 100K common given and surnames taken from the US Census Bureau (Bureau, 1990). To test our hypothesis that the letters in these words patterned differently, we measured perplexity and entropy for these two models two sets of words drawn from other data sources: (1) a set of the 200 most common words found in a the *NoteToSelf* and *SMS* corpora (names excluded by hand) and a set of 200 names drawn at random from the contacts list we had from our industrial partner. Table 6 shows the perplexity and entropy for these two datasets as measured on our two bigram models.

As can be seen in Table 6, values for both perplexity and

entropy are lower for each measured dataset on the bigram model constructed from equivalent data. Perplexity and entropy are lower for names when measured on the name letter bigram, with a similar pattern for words measured on the word bigram. Given these results, we felt we had a set of features to begin exploring named entity classification.

## 7.2. Building the classifier

We used a maximum entropy classifier whose output was a probability estimate, for each word, of each of the three classes. The classifier works on a word-by-word basis, with a post-processing step to find the sequence of words identified as units.

Because we suspected that immediate context is a powerful predictor for named entity classification, we ran the classifier using a 2-pass strategy. The first pass used the features described above. In the second pass, we added a feature representing the identity of words immediately to the left and right, as classified in the first pass. Thus, this feature had one of three values, *name*, *location*, and *unclassified word*.

The set of features used included:

- position within utterance (i.e., an integer measure of the distance of each word from the beginning and from the end of the utterance);
- entropy of word with respect to the *name bigram*;
- entropy of word with respect to the *word bigram*;
- delta entropy between scores from *name* and *word bigram*;
- presence of the word in gazettiers comprised of common names, cities, and a dictionary of common words;
- the same gazetteer feature computed on the 2 preceding words;
- the identity of the preceding word in a set comprised of (“at”, “for”, “from”, “in”, “on”, “to”, “with”);
- the classified identity of the preceding and following word in the first pass classification;

## 8. Results

Table 7 shows the performance of the classifier when trained on utterances generated using 80% of the templates in our complete set, and tested on set of utterances generated from half the remaining templates (comprising 10% of the total number of templates), randomly chosen. The results show the effects of both single and dual-pass strategies. Performance improves significantly with the second pass, on both *names* and *locations*. Performance for *location* entities improves most dramatically. Although the average length of a *location* entity is only 1.88 words, there are many more *location* entities that are of length 2 or greater (e.g., *bird’s nest coffee house*). Contextual information is most critical for these entities. *Name* entities, by contrast, are all of either length 1 or 2 (i.e., a given name, surname, or fullname).

To test the effects of adding a small set of additional templates, we trained another model using the original training

set (results shown in Table 7), augmented with utterances generated from the half of the remaining templates that were not used in the test set, randomly chosen (again, comprising 10% of the total). The results of this run are shown in Table 8. Overall performance improves on all measures for the 2-pass classification strategy. In the single-pass system, performance drops most markedly for names. An examination of the data reveals that the 90% training set contained a previously unseen overlap in local context for the two types of named entities (i.e., the same preposition preceding both). The effect this had on performance was removed in the second pass of the system. These trade-offs are sure to emerge as more contexts are considered, leading us to believe that incorporation of more global features from a language model might be useful. We intend to examine this is further work (see below).

## 9. Discussion/Future work

We feel our classification results are encouraging. Starting with real data, we have created a set of training and test utterances that we feel are useful for developing strategies for named entity extraction in speech. We feel the use of utterance templates is justified in this domain, when these templates are based on real user utterances. Given that our goal is to extract named entities from these utterances, we feel it is a strength of our approach that we are able to create a corpus rich in these phrase types. Furthermore, our technology is designed to use properties of words themselves, as well as local context. In this regard, we are hopeful that we can reach convergence using a relatively small set of templates.

Although we do not anticipate that we can produce a full syntactic analysis of the utterances spoken by users to a mobile device, we do think that local syntactic context is important. Some of the features used in classification were designed to capture this context. In addition, we have begun experiments to more fully exploit this information in a language model. We have created a simplified language model using just the classes output from the classifier (i.e., *name*, *location*, and *word*), along with lexicalized units derived from a list of common English words. We create Finite State Transducers (FSTs) from this language model and compose it with FSTs built from classifier output. Initial experiments show this to be a very promising way of improving unit error rate.

While not currently focussing on it, we think this technology will be useful even in cases where a full syntactic parse is available. Urban navigation systems, for example, when ported to new cities, encounter the problem of processing new sets of locations. A system that could delimit such entities could be used, in conjunction with a grammar, to understand a new domain.

In mining the data we had for utterance templates, we took a more-is-better approach. In doing so, we found many utterances that seem ideally suited to the degree of automatic processing we propose, e.g., “call trotter’s and make reservations for 8 pm.” Identifying the time expression would enable a system to set up a reminder, and identifying simply the location name (i.e., “trotter’s”) would allow a device to find and display a phone number. Without understanding the user’s intent, the system could provide helpful information. However, we also found (and used) many utterances

	Names			Locations		
	Precision	Recall	F-measure	Precision	Recall	F-measure
1-pass	86.6	79.0	72.4	85.5	73.7	79.2
2-pass	87.0	74.3	80.1	91.6	81.1	86.0

Table 7: Results of the named entity classification on an individual word basis. 80% of the utterance templates were used in training and 10% in testing. Named entities were different between training and test sets.

	Names			Locations		
	Precision	Recall	F-measure	Precision	Recall	F-measure
1-pass	83.0	75.9	79.3	85.6	72.9	78.7
2-pass	88.5	86.0	87.2	93.0	84.2	88.4

Table 8: Results of the named entity classification on an individual word basis. 90% of the utterance templates were used in training and 10% in testing. Named entities were different between training and test sets.

such as “we are all going to pizza hut after biology class,” where it is unclear if any further processing is necessary or relevant. We anticipate another classification stage might be needed to target utterances for which action is desired. This could improve system performance by identifying a smaller set of relevant utterances to work on.

## 10. Conclusions

We began with the desire to add semantic complexity to speech applications while not burdening the user with the need to learn a great deal about system capabilities. Users do learn, of course, and system developers rely on that learning, but what this means to us is that applications should be easy to bootstrap and then extensible by means of lightweight training. We have described here our initial efforts to achieve those goals.

The two most important resources we used here were simulated data and a large knowledge source of points of interest data. We will next examine how well the algorithms work when given data from different domains and languages. The underlying algorithms are portable. We anticipate language specific issues having to do with non-alphabetic languages but believe that these can be addressed with a richer set of features in classification.

## 11. Acknowledgements

We wish to thank Navteq for the use of their extremely valuable database of points of interest.

## 12. References

- Alan Beale. 2003. English vocabulary word list. <http://wordlist.sourceforge.net/12dicts-readme.html>.
- Frédéric Béchet, Allen L. Gorin, Jeremy H. Wright, and Dilek Hakkani-Tür. 2004. Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How may i help you? *Speech Communication*, 42(2):207–225.
- Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what’s in a name.
- U.S. Census Bureau. 1990. <http://www.census.gov/genealogy/www/data/1990surnames>.
- Council Canada, David Nadeau, Peter D. Turney, and Stan Matwin. 2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity.
- Nancy Chinchor. September, 1997. Muc-7 named entity task definition dry run version 3.5.
- Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification.
- Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134.
- Benoît Favre, Frédéric Béchet, and Pascal Nocéra. 2005. Robust named entity extraction from large spoken archives. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 491–498, Morristown, NJ, USA. Association for Computational Linguistics.
- Junlan Feng, Srinivas Bangalore, and Mazin Gilbert. 2009. Role of natural language understanding in voice local search. In *Proceedings Interspeech, 2009*.
- Jing Huang, Geoffrey Zweig, and Mukund Padmanabhan. 2001. Information extraction from voicemail. In *In Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 290–297.
- Martin Jansche and Steven P. Abney. 2002. Information extraction from voicemail transcripts. In *In Proc. Conference on Empirical Methods in NLP*.
- Francis Kubala, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. 1998. Named entity extraction from speech. In *in Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 287–292.
- Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *In Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8.