Named Entity Detection in ASR Hypothesis

**Keywords:** named entity detection, speech recognition, post-processing

Abstract[[1]](#footnote-1)

In this paper, we discussed the problem of named entity extraction from ASR hypothesis. Speech recognition in the realm of broadcast news severely suffers with out-of-vocabulary problem. Majority of the OOV tokens fall under named entity category. We approached this problem …

Introduction

Named Entity Recognition is an important task in Information Retrieval and Extraction tasks. Broadcast news

* 1. Introduction

In speech recognition systems, vocabulary is usually domain specific and predetermined. An ASR system has a limited vocabulary just enough to recognize popularly uttered words. An IR system can index limited number of words contained in the indexed documents.

Hethrington found that in speech recognition an OOV token contributes on average 1.5 errors because longer OOVs (names and places), the formatting instructions below will already be enforced. Primary sources of OOV person names:

* “New” names of global importance
* Newsworthy leaders, terrorists, criminals etc.
* News Reporters, readily available from news agency.
* Spelling and morphological variants
* Sports Figures
* Villagers and human interest personalities
  1. Statistics

The significant problem in vocabulary construction is identifying and ranking the most important vocabulary items. As we know that new words are constantly introduced into usage Submission to ICML 2009 will be entirely electronic, via a web site (not email). The URL The right strategy to decide the vocabulary size depends on the domain of operation. In this paper we focus on broadcast news domain and the kind of speech is different from spontanenous speech. This domain si different from switch-board corpus.

**Problem of OOVs:** Name phrases contain significant information about the sub-text (either phrase or sentence) and they occur quite frequently in several sources. Names are most frequent source of ASR errors in the broad cast news data. The table following stands as an evidence to this claim:

<TABLE here>

**Nature of errors:** Proper names are an important focus of error correction work because there is a strong correlation between names and OOV words. There are in vocabulary word errors caused because of mispronounciation. OOV errors are caused because of lack of vocabulary. Within OOVs we could observe a minor portion of errors are caused due to plural and possessive-marked words. For example: King(IV), King’s (OOV), Kings (OOV). There are other types of errors which do not occur in name phrases but given the magnitude of 57% of OOVs (cite palmer p34 and the palmer’s dissertation)

<example from our presentation>

After looking at this example we could figure out it is extremely difficult to anticipate new names of global importance. There are always new names appear in the news and we will never be able to model them in the ASR vocabulary. Therefore, we lose a lot of information when names of newsworthy people are OOVs. A great deal of OOV errors appear in document retrieval tasks where query terms could be morphological variants of indexed keywords. Also, there are several instances where one could find multiple transliterations of foreign names eg.,

Qadafi, Kadafi, Gadafi.

* 1. Related Work

There has been great deal of work applying text-based techniques to not-so-well behaved text like hand-written, OCR and spoken language data.

Palmer and Ostendorf’s work focuses on graphical representation of errors amidst running text. They propose a joint NE and error detection HMM-based model.

Confusion networks are employed (Evermann and Woodland 2000) to tackle this problem, where a richer representation of errors improve the accuracy on OOV detection.

Recent work on building candidate name list has been proposed (Schwarm and Ostendorf 2002) which is conditioned over text data to acquire an adapted language model. This approach could improve the second pass recognition while using adapted language model and dictionary.

Most of the earlier approaches focused on improving ASR level

In this work, we confine ourselves to solve the problem of named entity region detection with the help of various levels of information provided by the surrounding context.

* 1. Reacting to Reviews

We will continue the ICML tradition in which the authors are given the option of providing a short reaction to the initial reviews. These reactions will be taken into account in the discussion among the reviewers and PC-members.

* 1. Submitting Final Camera-Ready Copy

Final versions of papers accepted for publication should follow the same format and naming convention as initial submissions, except of course that the normal author information (names and affiliations) should be given. See Section 2.3.1 for details of how to format this.

The footnote, “Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.”' must be modified to “Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009. Copyright 2009 by the author(s)/owner(s).” For those using the LaTeX style file, simply change \usepackage{icml2009} to \usepackage[accepted]{icml2009}. Authors using Word must edit the footnote on the first page of the document themselves.

Camera-ready copies should have the title of the paper as running head on each page except the first one. The running title consists of a single line centered above a horizontal rule which is 1 point thick. The running head should be centered, bold and in 9 point type. The rule should be 10 points above the main text. For those using the LaTeX style file, the original title is automatically set as running head using the fancyhdr package which can be obtained at the ICML 2009 web site. In case that the original title exceeds the size restrictions, a shorter form can be supplied by using

\icmltitlerunning**{…}**

just before \begin{document}. Authors using Word must edit the header of the document themselves.

Approach

In tackling the problem, we focused most of our initial efforts on the feature engineering process.

* 1. Features

Due to the mixed nature of the problem, we explored a variety of text-based, as well as speech-based features, utilizing both traditional features of NER, as well as novel features inspired by the speech community.

* + 1. Bag of words

For our baseline model, we use the simple bag-of-word feature. This feature simply includes the surface form of all the words inside a window of fixed size around each word. This is a simple feature, which has proved to work well in Named Entity Recognition with clean text (CITATION NEEDED).

We did not perform any regularization on the text, e.g. stemming, lemmatization…, because we believe it would not help in a domain with noisy input such as ours. These techniques perhaps would help reduce the error rate of an ASR in cases where there can be confusion between possessive endings and plural forms. However, in NER on speech hypothesis, it is not clear that it would help.

* + 1. Phonetic features

In the speech domain, it is likely that an incorrect hypothesis word sounds similar to the original word. This intuition guided us to a family of features based on the phonetic structure of words. For each word in the speech hypothesis, we used an off-the-shelf text-to-phone software (Fisher, 1999) to find the phones that comprise it. From this sequence of phones, we extracted various features, such as phone unigrams, phone bigrams. To further de-emphasize the influence of errors caused by similar sounding phones, we devised various groupings of phones into phone classes, and then used these class names in place of the exact phone names. We used phone class sequence, “bag of classes”, as well as phone class pattern as our features, where the phone class pattern feature is computed as the regular expression representing the sequence of phones. Table 1 shows one way of grouping phones into phone classes.

*Table 1*. Example grouping of phones into phone classes

|  |  |
| --- | --- |
| Class | Member phones |
| 1 | b, p |
| 2 | d, t, dx |
| 3 | g, k, q |
| 4 | jh, ch |
| 5 | s, sh, z, zh |
| 6 | dh, th |
| 7 | f, v |
| 8 | l, r, e, y, w |
| 9 | m, n, nx, ng, em, en, eng |
| 10 | hh, hv |
| 11 | iy, ih, eh, ae, ay, aw |
| 12 | aa, er, ah, ax, ao |
| 13 | uw, uh, ow, oy |
| 14 | axr, ax-h |

* + 1. Base phrase chunk labels

Names always appear in noun phrases. Therefore, to detect names, we attempted to find noun phrases in the data to use as features. Quick experiments with the Stanford parser (Klein, Manning, 2003) showed that it was quite robust to noise in the data. For example, given “in the frustrate”, it tagged as a noun the word “frustrate”, which according to WordNet (Miller, 2009) never acts as a noun. While the Stanford parser is a regular syntactic parser and gives syntax labels for each word, we used it as a shallow parser, stopping at the base noun phrase level. The feature is a binary one, which distinguishes whether a word appears in a noun phrase or not.

* + 1. Other features

Other suitable features to use on a speech output corpus include acoustic confidence and language model confidence. As elaborated in Section 3.2.1, however, half of our experiments are with provided speech hypothesis, which did not come with confidence information, making it impossible for us to include these features. We therefore chose to exclude them to provide a more straightforward comparison between our experiments.

We should also mention some common features that we chose not to use. Part-of-speech tags, character n-grams and word shape are such features. With our noisy data, we decided to exclude these features to mitigate potential damage caused by not having the correct text from which to extract them.

* 1. Learning models

We used Conditional Random Fields (Lafferty, McCallumn, Perreira, 2001), which is the most popular model in sequential labeling problems, as the learning model with which to experiment the features. We also experimented with Conditional Markov Models using probabilistic SVM (Taskar, Guestrin, Koller, 2003), MEMM (McCallum, Freitag, 2000), as well as Voted Perceptron HMM (Collins, 2002). These methods, however, did not perform better or notably faster than CRFs, so we did not focus our effort on experimenting with multiple learners.

Evaluation

In this section, we discuss the dataset we used, our experiment setups and finally our experiment results.

* 1. Data

We used Broadcast News data from the Linguistic Data Consortium’s TDT4 2004 Corpus (Mitchell et al., 2005). The corpus came with the reference text, the speech data, as well as the ASR hypothesis obtained using a commercial decoder. The dataset contained 312 hours of speech, with a total of 2,444,334 tokens. A small portion of this dataset was annotated by LDC through the 2004 Automatic Content Extraction (ACE) project (Mitchell et al., 2004). The annotations contained 7 types of Name labels, provided as offset markups into the reference files. In total, it was an equivalence of 8 hours of speech, containing 33,479 tokens, with 3164 named entities. Table 1 summarizes the data. Table 2 gives a breakdown of the entire dataset (labeled and unlabeled) by the news source.

* 1. Data preparation

To prepare for the experiments, we first had to label the speech hypothesis using the given annotations in the reference text.

* + 1. Speech hypothesis

We had two sources of speech hypothesis data to work with, one readily available from the LDC corpus, and one that we produced ourselves using the original speech data. The ASR system that was used in producing the LDC speech hypothesis was the Dragon System by Nuance Communications, which used a vocabulary of around 60k word vocabulary. This system produced relatively good recognition output, due to its large vocabulary size.

We also had access to the original speech data, on which we were able to run our own ASR system. The intention was to use a deliberately less powerful ASR, with a smaller vocabulary size, to imitate real world situation in which the system knows only the most common names. In such a setting, errors in name recognition would pose a much great challenge to speech recognition. However, we did not have speech data for one of the news sources (VOA). In addition, some of the data was corrupt, making it impossible for our ASR to decode. This resulted in only 2.5 hours of speech data that had annotations.

*Table 1*. Data statistics – labeled vs. unlabeled

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Hours | Tokens | News Stories | Named Entities |
| Labeled | 8 | 33,479 | 147 | 3,164 |
| Unlabeled | 304 | 2,410,885 | 13,280 | ? |

*Table 2*. Breakdown by news source – all data

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hours | Tokens | News Stories |
| ABC | 38.5 | 277,957 | 1,692 |
| CNN | 64.5 | 430,371 | 4,698 |
| NBC | 35 | 237,549 | 1,234 |
| PRI | 62 | 558,867 | 1,965 |
| VOA | 69 | 616,043 | 2,694 |
| MNB | 43 | 290,068 | 997 |

* + 1. Reference to Hypothesis Alignment

Because the annotations we had were done for reference text, and the input to our NER system is speech hypothesis, we had to perform alignment between the reference text and the speech hypothesis in order to get labels for our input data. We used SCLite[[2]](#footnote-2) for this task. This process introduces further noise to our data – we no longer have a “gold standard” human annotation for it.

* 1. Experiment setup

We used various combinations of the feature set described in Section 2 to test our method. We only used the labeled data, and ran a 10-fold cross validation experiment for each model – feature set combo. We used the MinorThird package (Cohen, 2004) to extract the features, as well as train and test our learner. The next subsection shows our experiment results.

* 1. Figures

You may want to include figures in the paper to help readers visualize your approach and your results. Such artwork should be centered, legible, and separated from the text. Lines should be dark and at least 0.5 points thick for purposes of reproduction, and text should not appear on a gray background.

Label all distinct components of each figure. If the figure takes the form of a graph, then give a name for each axis and include a legend that briefly describes each curve. Do not include a title inside the figure; instead, the caption should serve this function.

Number figures sequentially, placing the figure number and caption *after* the graphics, with at least 0.1 inches of space before the caption and 0.1 inches after it, as in Figure 1. The figure caption should be set in 9 point type and centered unless it runs two or more lines, in which case it should be flush left. You may float figures to the top or bottom of a column, and you may set wide figures across both columns (use the environment figure\* in LaTeX), but always place two-column figures at the top or bottom of the page.

* 1. Algorithms

If you are using LaTeX, please use the “algorithm” and “algorithmic” environments to format pseudocode. These require the corresponding stylefiles, algorithm.sty and algorithmic.sty, which are supplied with this package. Algorithm 1 shows an example.

**Algorithm 1** Bubble Sort

**Input**: data *x* , size *m*

**repeat**

Initialize *noChange = true*.

**for**  *i = 1* to m-1

**if** *xi*  > xi+1

Swap *xi*  and xi+1

*noChange = false*

**end if**

**end for**

**until**  *noChange* is *true*

* 1. Tables

You may also want to include tables that summarize material. Like figures, these should be centered, legible, and numbered consecutively. However, place the title *above* the table with at least 0.1 inches of space before the title and the same after it, as in Table1. The table title should be set in 9 point type and centered unless it runs two or more lines, in which case it should be flush left.

Tables contain textual material that can be typeset, as contrasted with figures, which contain graphical material that must be drawn. Specify the contents of each row and column in the table's topmost row. Again, you may float tables to a column's top or bottom, and set wide tables across both columns, but place two-column tables at the top or bottom of the page.

*Table 1*. Classification accuracies for naive Bayes and flexible Bayes on various data sets.

|  |  |  |  |
| --- | --- | --- | --- |
| Data Set | Naïve | Flexible | Better? |
| Breast | 95.9 ± 0.2 | 96.7 ± 0.2 | ✓ |
| Cleveland | 83.3 ± 0.6 | 80.0 ± 0.6 | x |
| Credit | 74.8 ± 0.5 | 78.3 ± 0.6 |  |
| Glass2 | 61.9 ± 1.4 | 83.8 ± 0.7 | ✓ |
| Horse | 73.3 ± 0.9 | 69.7 ± 1.0 | x |
| Meta | 67.1 ± 0.6 | 76.5 ± 0.5 | ✓ |
| Pima | 75.1 ± 0.6 | 73.9 ± 0.5 |  |
| Vehicle | 44.9 ± 0.6 | 61.5 ± 0.4 | ✓ |

* 1. Future Work

Apply semi-supervised learning approach to leverage on the vast amount of unlabeled data

1. Explore the speech output more closely to spot patterns that might have helped bag of word features, e.g. some common words that are consistently hypothesized in place of a name

Sometimes, speech output is consistent enough while it repeats mistakes on certain phrases. We would like to explore such recurrent patterns to

1. Include acoustic confidence feature

Acknowledgments

**Do not** include acknowledgements in the initial version of the paper submitted for blind review.

If a paper is accepted, the final camera-ready version can (and probably should) include acknowledgements. In this case, please place such acknowledgements in an unnumbered section at the end of the paper. Typically, this will include thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

Alexis Mitchell et al. (2004). TIDES Extraction (ACE) 2003 Multilingual Training Data. *Linguistic Data Consortium, Philadelphia.*

Alexis Mitchell et al. (2005). ACE 2004 Multilingual Training Corpus. *Linguistic Data Consortium, Philadelphia.*

Cohen, W. (2004). *MinorThird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data*, [http://minorthird.sourceforge.net](http://minorthird.sourceforge.net/).

Collins, M. (2002). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. *Empirical Methods in Natural Language Processing.*

Fisher, W. (1999). A Statistical Text-to-Phone Function Using Ngrams and Rules.  *IEEE International Conference on Acoustics, Speech, and Signal Processing 1999, pp. II-649-652.*

Klein, D., Manning, C. (2003). Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, Cambridge, MA: MIT Press, pp. 3-10.

Klein, D., Manning, C. (2003). Accurate Unlexicalized Parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423-430.

Lafferty, J., McCallum, A., & Perreira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the International Conference on Machine Learning*.

McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proceedings of the International Conference on Machine Learning*.

Miller, G. (2009). "WordNet - About Us." *WordNet. Princeton University*. "[http://wordnet.princeton.edu"](http://wordnet.princeton.edu\%22)

Taskar, B., Guestrin, C., & Koller, D.(2003). Max- margin markov networks*. Advances in Neural Informa- tion Processing System.*

1. Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute. [↑](#footnote-ref-1)
2. http://www.itl.nist.gov/iad/mig/tools/ [↑](#footnote-ref-2)