OOV Sensitive Named-Entity Recognition in Speech

Carolina Parada, Mark Dredze, and Frederick Jelinek

Center for Language and Speech Processing and
Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore MD
carolinap@jhu.edu, mdredze@cs.jhu.edu

Abstract

Named Entity Recognition (NER), an information extraction task, is typically applied to spoken documents by cascading a large vocabulary continuous speech recognizer (LVCSR) and a named entity tagger. Recognizing named entities in automatically decoded speech is difficult since LVCSR errors can confuse the tagger. This is especially true of out-of-vocabulary (OOV) words, which are often named entities and always produce transcription errors. In this work, we improve speech NER by including features indicative of OOVs based on a OOV detector, allowing for the identification of regions of speech containing named entities, even if they are incorrectly transcribed. We construct a new speech NER data set and demonstrate significant improvements for this task.

Index Terms: Named Entity Recognition, OOV Detection

1. Introduction

Named entity recognition (NER) in text, a key step in information extraction, is typically treated as a sequence labeling task in which entities are labeled as people, locations and organizations [1]. Evaluations have focused on newswire text and manually transcribed broadcast news. However, NER in automatic speech recognition (ASR) produced transcripts is a challenge due to recognition errors and the lack of common named entity markers (punctuation, capitalization, numerals, etc.) Understandably, performance lags behind that of text applications. Attempts to improve speech NER have included transcript normalization [2], incorporating speech recognition confidence features [3, 4], or tagging LVCSR word lattices [5]. A difficult unaddressed problem comes from out-of-vocabulary (OOV) terms: words that are missing from the LVCSR vocabulary. Since many OOVs are proper names (66% of the OOVs in our corpus are named entities,) OOV recognition errors are particularly damaging for NER.

In this work, we improve speech NER by allowing the tagger to identify incorrectly decoded sections of speech where a named entity was spoken. Finding such audio regions allows for targeted manual transcription, or automated OOV recovery efforts. To recognize OOV NEs, we augment the features in an NER system to include indications of possible OOVs in the transcript using an OOV detection system [6]. These features yield significant improvements for OOV NEs in particular, as well as NEs in general.

To evaluate our approach, we introduce a new broadcast news speech data set annotated for named entities using Amazon Mechanical Turk. We describe the methods used to create this data set and its properties. Additionally, we provide these collected annotations to encourage research in this area.

2. Named Entity Recognition for OOVs

No matter the vocabulary size, LVCSR systems will encounter and mis-recognize OOVs, especially in new domains or genres. These often include named entities; in our English broadcast news data set 66% of the OOVs are named entities, accounting for 21% of all named entities. This problem is often ignored in NER in speech [7, 4]; and some cope with OOV entities by adapting the vocabulary and the language model to the specific time interval of the test set [8].

To recognize OOV NEs, we augment a standard NE tagger to include features indicative of OOV terms. The tagger should ignore the decoded words for OOV regions and rely on context to identify the named entity. For example, if the tagger sees the string "FORMER PRESIDENT MOST OF IT SAID" it would likely find no named entity. However, "MOST OF IT" is an obvious transcription error (for "MILOSEVIC") and if the tagger knew "MOST OF IT" was OOV, it could focus on context ("Former President X said") and identify the audio corresponding to "X" as a named entity.

Our work is similar to that of Huang [3] and Sudoh et al.[4] Huang uses a confidence based approach to identify transcript errors and ignores the decoded word sequence in the error region, using the context to query relevant documents for OOV recovery. He uses features from the recovered word and its context as input for a standard NER system. In this work, we are concerned with identification and not recovery. However, identified named entities in incorrectly transcribed audio could be targeted for recovery using an OOV recovery system [9]. Both Huang and Sudoh et al. rely on the word posterior probability as a confidence metric. Sudoh et al. combine this metric with the decoded word sequence and contextual POS tag information using SVMs to detect unreliable regions. We consider a similar approach (errordet) as a baseline.

Our approach uses the output of an OOV detector as indicative of NE regions. In the next section, we introduce our NE tagger and describe how we incorporate OOV information.

2.1. Named Entity Tagger and OOV Detector

We use a conditional random field (CRF) based named entity tagger [10], with a first order Markov model, BIO encoding (B-ORG, I-ORG, etc.), and a standard set of orthographic features [11] (Baseline). We used the default parameters in Mallet [1] and a Gaussian prior of $\sigma^2 = 10$ (results were generally insensitive to σ.) The number of training iterations was selected using development data. On CONLL 2003 English data [1], the tagger achieves an overall development F1 of 88.34 and test F1 of 81.41, which is close to state of the art on this task.

[1] http://mallet.cs.umass.edu

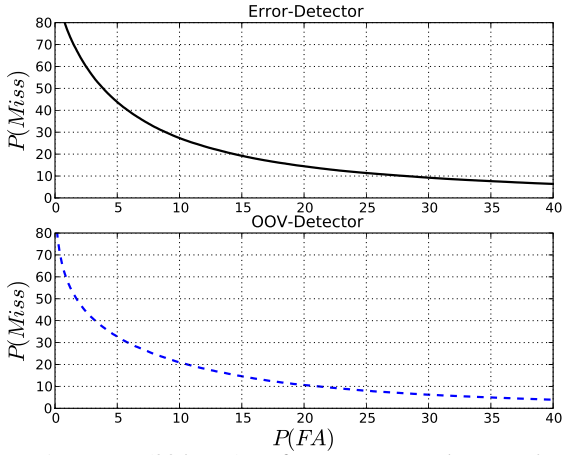

Figure 1: Test set (90 hours) performance: Error detector [4] on errors (top) and OOV detector [6] on OOVs (bottom.)

We use an OOV detector to incorporate OOV information into the tagger. Parada et al. [6] introduced a state of the art confidence based OOV detector using features from a hybrid LVCSR system, combining words and sub-word units (variable length phone sequences). This was combined with confidence estimates from the confusion network (word entropy in each confusion network region) as well as contextual features, such as adjacent words, language model scores, etc. These features are used in a CRF to tag confusion network bins as OOV or IV. We obtained the same training splits as Parada et al. to match their OOV detector results (Figure 1.)

Several of the baseline NER features use part of speech tags. Since most POS taggers are trained for text, we obtained tags for speech by using a syntactic language model [12, 13], which estimates the joint probability of the word and its syntactic tag based on the preceding words and tags (a trigram context.) This tagger can produce tags for suspected OOVs.

We trained the language model on 130 million words from Hub4 CSR 1996 [14]. Tags, extracted from parse trees from a modified Berkeley parser [15], incorporated the word's POS, the label of its immediate parent, and the relative position of the word among its siblings. [2] Since we are considering OOV detection, the language model was restricted to the LVCSR system's vocabulary.

2.2. Confidence Baseline

A common approach to improve NER performance on speech is to incorporate a confidence estimate for predicting decoding errors (such as those caused by OOVs) [3, 4] . Therefore, we compare our approach with both a baseline NER feature set described in Section 2.1, and with an additional confidence estimate baseline (Figure 1.)

The confidence baseline uses the features of Sudoh et al. [4] to create a CRF error predictor: the decoded word, POS tag, and posterior probability, as well as these features from a ± 2 word window. This system shows superior error detection performance to only using the word posterior probability. The training data was obtained from a standard word-based LVCSR system whose errors are known by aligning with the reference transcription. The probability of error, provided by the CRF error predictor, was quantized into 10 bins generating binary features (errordet).

[2] The *parent* tagset of Filimonov and Harper [13].

2.3. NER with OOVs

We incorporate OOV information into the NER tagger by generating features based on the OOV detector. Our goal was to inform the NER tagger when we suspected that a word may be an OOV, which could make it more likely to be a named entity. We also sought to remove unreliable features, i.e. incorrectly decoded words. We developed three feature sets:

- **oovdet**: The probability of a word being OOV according to the OOV detector (Section 2.1).[3]
- **context**: The oovdet confidence feature from a ± 2 word window around the current word.
- **replace**: Replace the decoded word with the token `OOV` if the detector has a confidence threshold above 0.9 (tuned on development data.) This explicitly removes the confusion of an incorrectly decoded word, and the system must rely on the context to tag the words, as well as a prior that OOVs may be NEs.

3. Experiment Setup

To focus attention on the OOV problem, we used the data set constructed by Can et al. [16], originally designed to evaluate Spoken Term Detection (STD) of OOVs (`OOVCORP`.) The corpus contains 100 hours of transcribed English broadcast news speech emphasizing OOVs. There are 1290 unique OOVs in the corpus, which were selected with a minimum of 5 acoustic instances per word. Example OOVs include: `PUTIN`, `QAEDA`, `HOLLOWAY`, `COROLLARIES`, `HYPERLINKED`, `NATALIE`. Short OOVs (less than 4 phones) inappropriate for STD were explicitly excluded. This resulted in roughly 24K (2%) OOV tokens.

We used the IBM Speech Recognition Toolkit [17][4] to obtain a transcript of the audio. Acoustic models were trained on 300 hours of HUB4 data [18] and utterances containing OOV words as marked in `OOVCORP` were excluded. The language model was trained on 400M words from various text sources with a 83K word vocabulary. The LVCSR system's WER on the standard RT04 BN test set was 19.4%.

Excluded utterances amount to 100 hours. Five hours were used for training the OOV and error detectors, and 48 hours were annotated for named entity training and evaluation. From this set, 25 hours were used for NE tagger training, 5 hours for development, and the remaining 18 hours for testing. Both train and test sets have a 2% OOV rate.

Since our features are based on sub-word units, we used a hybrid LVCSR system. Sub-word units are variable length phone sequences derived using Rastrow et al. [19]. The vocabulary of a hybrid LVCSR system contains a word and a sub-word lexicon; sub-words are used to represent OOVs in the language model text. Language model training text is obtained by replacing low frequency words (assumed OOVs) by their fragment representation. Pronunciations for OOVs are obtained using grapheme to phoneme models (see Rastrow et al.) Our hybrid system's lexicon has 83K words, 20K sub-words. The 1290 excluded words are OOVs to both the word and hybrid systems. Features were extracted from confusion networks from the word and hybrid LVCSR systems.

[3] Real valued features were quantized into 10 bins (binary features.)

[4] The IBM system used speaker adaptive training based on maximum likelihood with no discriminative training.

OOV	System	Overall			IV entities			OOV entities			Unobserved OOVs		
		F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec
-	Baseline	58.5	64.7	53.3	59.9	64.6	55.9	51.0	63.4	42.7	29.9	41.5	23.4
	errordet	58.6	66.5	52.4	60.2	66.9	54.7	51.3	63.8	42.9	36.8	50.5	28.9
Auto	oovdet	60.1	66.4	54.9	61.7	66.6	57.5	52.9	64.4	44.9	37.5	49.5	30.1
	oovdet+context	59.8	66.1	54.6	61.0	65.8	56.8	54.2	66.6	45.7	33.7	45.8	26.7
	oovdet+replace	59.8	67.9	53.5	61.4	68.3	55.7	52.4	65.1	43.9	35.6	48.0	28.3
	oovdet+context+replace	**60.7**	68.2	54.7	**62.2**	68.4	57.1	**53.6**	66.2	45.1	**37.7**	50.5	30.1
Oracle	oov	61.7	68.2	56.4	61.7	70.2	55.0	61.2	62.5	60.0	47.2	47.9	46.5
	oov+oovdet+context+replace	**62.3**	68.0	57.5	**62.1**	68.6	56.8	**61.7**	65.9	58.0	**52.0**	54.1	50.0

Table 1: Performance for all named entities (Overall), for entities containing all words in the LVCSR vocabulary (In-Vocabulary or IV entities), for entities containing at least one OOV word (OOV entities), and for OOV entities which are not present in training for NER system (Unobserved OOVs).

3.1. Named Entity Annotations

The 48 hours set aside for named entity training and evaluation did not contain named entity annotations. We annotated this data using Amazon Mechanical Turk (MTurk), which provides a platform where human intelligence tasks (HITs) can be given to users (turkers) for annotation. HITs tend to be simple tasks that are easy for humans to accomplish but remain challenging for computers, such as labeling images or translating sentences. Turkers are paid a micro-payment for each HIT, typically a few cents. Studies have shown that high quality annotations are possible by using multiple turkers, simple HITs, and embedded gold standard examples to evaluate quality [20].

We used MTurk to obtain named entity labels for the manual speech transcripts from our corpus. Our annotation guidelines were based on the CONLL shared task [1]. We used an interface similar to that presented in Finin et al. [21], modified to allow users to indicate the presence of two adjacent named entities of the same type, such as "Thanks Jim Sarah reporting live from Boston". Here Jim and Sarah are two different people with a pause in the speech that does not appear in the transcript. Such occurrences are exceedingly rare in text, but common in speech where pauses or speaker turn taking creates sequential named entities. Each HIT contained 5 speech utterances, 1 of which was an utterance chosen from 250 utterances for which we obtained expert NE annotation (provided by the authors) included for quality control. Each HIT was completed by two different turkers at a rate of \$0.10 each, yielding a rate of \$0.02 an utterance. Utterances completed by unreliable turkers (poor scores on the included gold examples) were resubmitted to obtain additional annotations. Total cost was \$530, including repeated annotations and productivity bonuses. Details on annotation instructions are in the Appendix.

For each turker, we evaluated his or her average F1 score on gold utterances and removed annotations by turkers with an F1 score below 0.5. We then compared the two annotations for each utterance and selected labels agreed to by both turkers. In cases of dispute, we select the label assigned by the turker with highest F1. This yielded a total of 9971 annotated utterances (510K tokens). The inter-annotator agreement between turkers computed using Cohen's κ [22] was .72.[5]

The average F1 score of the final turker annotations on the 257 utterances with gold annotations was 87%. The 9971 utterances contain 34,293 named entities, 14,967 people, 10,680 locations and 8,646 organizations. In this data, 21% of the named entities are OOVs (38.32% of PER, 6% of LOC, and 14% of ORG) and 66% of the OOVs are named entities.

[5]Kappa coefficient can be interpreted as: $0-.2$ slight, $.2-.4$ fair, $.4-.6$ moderate, $.6-.8$ substantial and higher is almost perfect [23].

4. Results

We evaluated our named entity tagger with the various OOV feature sets and two baselines: standard NER features (baseline), and the addition of error detection features (errordet). We report F1 results overall, for entities containing all words in the vocabulary (In-Vocabulary or IV entities), and for entities containing at least one OOV word (OOV entities) (Table 1). While the error detector (errordet) yields a small improvement (0.15), the equivalent OOV features (oovdet) yield a much larger improvement (1.67). Using all of our OOV features (oovdet+context+replace) achieves a 2.25 improvement over the baseline.[6] Example improvements included the utterance "opposition claims VOJISLAV KOSTUNICA should be declared winner", which is decoded as: "opposition claims BORISLAV CUSTOM ME JUST should be declared winner". The named entity "VOJISLAV KOSTUNICA" is misrecognized because both words are OOVs, but the improved system correctly labeled it as a person.

Note that in our data set partitioning, the tagger may learn the context of an OOV in the NER training set, which matches the context for that same OOV in test, allowing the system to correctly label it as an entity. However, in a real application, with a constantly changing vocabulary, OOVs seen during the tagger training are likely to be different from those in the test set. To evaluate this scenario, we report performance of "Unobserved OOV" entities: named-entities containing words which are not in the recognizer's vocabulary, and are unobserved in the training set for the OOV detector or NER training and development set. These words only appear in the test set, never in the training or development set. As expected, the performance for these entities is lower than the overall OOV performance, however the proposed system (oodet+context+replace) achieves a 7.8 absolute improvement over the baseline (29.9 to 37.7). Eighteen percent of all OOV entities in the test-set were Unobserved.

Additionally, we achieve improvements in performance for IV entities, where there is a large increase in precision (64.58 to 68.42). We attribute this gain to the fact that now the learner is not forced to mark common word strings like "most of it" (for MILOSEVIC) or "custom me just" (for KOSTUNICA) as named entities unless the OOV features indicate otherwise. Furthermore, these incorrectly decoded words are replaced by `OOV`, removing misleading features.

We also sought to determine additional gains we might achieve by improved OOV detection. We replaced the OOV detector by an oracle predictor, manually tagging OOV regions

[6]Statistically significant at $p = 0.001$ using the paired permutation test.

by finding time segments in the manual transcripts containing words which are not in the LVCSR system vocabulary. True OOV regions were marked with a new feature (oov.) The best performance was obtained when adding this oracle OOV feature, replacing the decoded word by "OOV" if this feature fires, and including the context feature described in Section 2.3 (oov+oovdet+context+replace). This improves an additional 8.08 points on OOVs over the best OOV predictor results, and 1.61 improvement overall. This demonstrates that our automatic results achieve an almost 60% error reduction towards oracle OOV detection. Additionally, the oracle results achieve similar performance for OOV and IV, indicating that remaining NER errors may not be attributed to OOVs and that given correct OOV predictions, our NE tagger effectively addresses the OOV NER problem.

5. Conclusion

We have presented a novel approach for OOV sensitive named entity recognition in automatically transcribed speech, targeting NEs containing words which are not present in the LVCSR system's vocabulary. We augmented the features used by a CRF NER tagger to indicate possible OOVs in the transcript. Our system obtains a statistically significant improvement in overall performance using automatic OOV detection and our automatic results achieve an almost 60% error reduction over the baseline compared to oracle results. Additionally, we show that oracle OOV features close the gap between IV and OOV NER performance. Finally, we introduced a new broadcast news speech data set annotated for named entities using Amazon Mechanical Turk (available online at time of publication.)

6. Acknowledgements

We gratefully acknowledge the IBM Speech Group for providing the recognition lattices used in these experiments, and Denis Filimonov for the POS tags used in the baseline tagger.

7. References

[1] Erik Tjong Kim Sang and Fien De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *CONLL*, 2003.

[2] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *ICASSP*, 2009, pp. 4741–4744.

[3] Fei Huang, *Multilingual Named Entity Extraction and Translation from Text and Speech.*, Ph.D. thesis, Carnegie Mellon University, 2005.

[4] Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki, "Incorporating speech recognition confidence into discriminative named entity recognition of speech data," in *ACL*, 2006.

[5] James Horlock and Simon King, "Named entity extraction from word lattices," in *Eurospeech*, 2003.

[6] Carolina Parada, Mark Dredze, Denis Filimonov, and Fred Jelinek, "Contextual information improves oov detection in speech," in *NAACL*, 2010.

[7] David Miller, Michael Kleber, Chia lin Kao, and Owen Kimball, "Rapid and accurate spoken term detection," in *INTERSPEECH*, 2007.

[8] Benoît Favre et. al, "Robust named entity extraction from large spoken archives," in *EMNLP*, 2005.

[9] Carolina Parada, Abhinav Sethy, Mark Dredze, and Fred Jelinek, "A spoken term detection framework for recovering out-of-vocabulary words using the web," in *International Speech Communication Association (INTERSPEECH)*, 2010.

[10] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *CONLL*, 2003.

[11] Ryan McDonald and Fernando Pereira, "Identifying gene and protein mentions in text using conditional random fields," *BMC Bioinformatics*, vol. 6(Suppl 1), no. S6, 2005.

[12] Denis Filimonov and Mary Harper, "Measuring tagging performance of a joint language model," in *INTERSPEECH*, 2009.

[13] Denis Filimonov and Mary Harper, "A joint language model with fine-grain syntactic tags," in *EMNLP*, 2009.

[14] John Garofolo, Jonathan Fiscus, William Fisher, and David Pallett, *CSR-IV HUB4*, Linguistic Data Consortium, 1996.

[15] Zhongqiang Huang and Mary Harper, "Self-Training PCFG grammars with latent annotations across languages," in *EMNLP*, 2009.

[16] Dogan Can et. al, "Effect of pronounciations on OOV queries in spoken term detection," *ICASSP*, 2009.

[17] H. Soltau et al., "The IBM 2004 conversational telephony system for rich transcription," in *ICASSP*, 2005.

[18] Jonathan Fiscus, John Garofolo, Mark Przybocki, William Fisher, and David Pallett, *1997 English Broadcast News Speech (HUB4)*, Linguistic Data Consortium, 1998.

[19] Ariya Rastrow, Abhinav Sethy, and Bhuvana Ramabhadran, "A new method for OOV detection using hybrid word/fragment system," *ICASSP*, 2009.

[20] Chris Callison-Burch and Mark Dredze, "Creating speech and language data with amazon's mechanical turk," in *Workshop on Creating Speech and Language Data With Mechanical Turk at NAACL-HLT*, 2010.

[21] Tim Finin et al., "Annotating named entites in twitter data with crowdsourcing," in *Workshop on Creating Speech and Language Data With Mechanical Turk at NAACL-HLT*, 2010.

[22] Joseph Fleiss and Jacob Cohen, "The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability.," *Educational and Psychological Measurement*, vol. 33, pp. 613–619, 1973.

[23] J. Richard Landis and Gary G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, pp. 159–174, 1977.

A. Annotation Guidelines

The following directions were provided to Turkers.
An entity is a object in the world like a place or person. A named entity is a phrase that uniquely refers to an object by its proper name (Hillary Clinton), acronym (IBM) or abbreviation (Minn.). Here are some more examples of named entities for each of the types we are interested in. Note that there is no capitalization in the sentences.

- Person: first, middle, and last names of people, animals and fictional characters.

  Person examples: barack obama; palins; john; terry lewis;
- Organization: companies, brands, political movements, government bodies (councils, courts, ministries), musical companies, public organizations (schools, universities, charities), other collections of people (sport clubs, associations, etc.)

  Organization examples: c.n.n., the white house, congress., valujet; the washington post; oxford university (when considered in context as an organization);
- Place: roads, regions (towns, cities, countries, etc.), natural locations (mountains, valleys, national parks, etc.), public places (squares, museums, airports, stations, hospitals, parks, universities, etc.), commercial places (pubs, restaurants, hotels, etc.), assorted buildings (houses, halls, rooms, castles), abstract places ("the free world")

  Place examples: baltimore, md; washington; dade county, florida; mt. everest; hoover dam; u.s.; oxford university (when considered in context as a location);