Open-Domain Name Error Detection using a Multi-Task RNN

Hao Cheng **Hao Fang** **Mari Ostendorf**
Department of Electrical Engineering
University of Washington
{chenghao,hfang,ostendor}@uw.edu

Abstract

Out-of-vocabulary name errors in speech recognition create significant problems for downstream language processing, but the fact that they are rare poses challenges for automatic detection, particularly in an open-domain scenario. To address this problem, a multi-task recurrent neural network language model for sentence-level name detection is proposed for use in combination with out-of-vocabulary word detection. The sentence-level model is also effective for leveraging external text data. Experiments show a 26% improvement in name-error detection F-score over a system using n-gram lexical features.

1 Introduction

Most spoken language processing or dialogue systems are based on a finite vocabulary, so occasionally a word used will be out of the vocabulary (OOV), in which case the automatic speech recognition (ASR) system chooses the best matching in-vocabulary sequence of words to cover that region (where acoustic match dominates the decision). The most difficult OOV words to cover are names, since they are less likely to be covered by morpheme-like subword fragments and they often result in anomalous recognition output, e.g.

REF: what can we get at **Litanfeeth**
HYP: what can we get *it leaks on feet*

While these errors are rare, they create major problems for language processing, since names tend to be important for many applications. Thus, it is of interest to automatically detect such error regions for additional analysis or human correction.

Named entity recognition (NER) systems have been applied to speech output (Palmer and Ostendorf, 2005; Sudoh et al., 2006), taking advantage of local contextual cues to names (e.g. titles for person names), but as illustrated above, neighboring words are often affected, which obscures lexical cues to name regions. Parada *et al.* (2011) reduce this problem somewhat by applying an NER tagger to a word confusion network (WCN) based on a hybrid word/fragment ASR system.

In addition to the problem of noisy context, automatic name error detection is challenging because name errors are rare for a good recognizer. To learn the cues to name errors, it is necessary to train from the output of the target recognizer, so machine learning is faced with infrequent positive examples for which training data is very sparse. In addition, in an open domain system, automatically-learned lexical context features from one domain may be useless in another.

In this paper, we address these general problems – detecting rare events in an open-domain task – specifically for name error detection. Prior work addressed the problem of skewed priors by artificially increasing the error rate by holding names out of the vocabulary (Chen et al., 2013) or by factoring the problem into sentence-level name detection and OOV word detection (He et al., 2014) (since OOV errors in general are more frequent than name errors). Sentence-level features are also shown to be more robust than local context in direct name error prediction (Marin, 2015). While these techniques provide some benefit, the use of discrete lexical context cues is sensitive to the limited amount of training data available.

Our work leverages the factored approach, but improve performance by using a continuous-space sentence representation for predicting presence of a name. Specifically, we modify a recurrent neural network (RNN) language model (LM) to predict both the word sequence and a sentence-level name indicator. Combining the LM objective with name prediction provides a regularization effect in training that leads to improved sentence-level

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 737–746,
Lisbon, Portugal, 17-21 September 2015.

name prediction. The continuous-space model is also effective for leveraging external text resources to improve generalization in the open-domain scenario.

The overall framework for speech recognition and baseline name error detection system is outlined in Section 2, and the multi-task (MT) RNN approach for sentence-level name prediction is introduced in Section 3. Experimental results for both sentence-level name detection and name error detection are presented in Section 4, demonstrating the effectiveness of the approach on test data that varies in its match to the training data. As discussed in Section 5, the sentence-level model is motivated by similar models for other applications. The paper summarizes key findings and discusses potential areas for further improvement in Section 6.

2 System Overview and Tasks

The name error detection task explored in this work is a component in a bidirectional speech-to-speech translation system for English to/from Iraqi Arabic with human-computer dialogue interaction for error resolution (Ayan et al., 2013), developed during the DARPA BOLT project. The training data consists of a range of topics associated with activities of military personnel, including traffic control, military training, civil affairs, medical checkups, and so on. However, the system is expected to handle open-domain tasks, and thus the evaluation data covers a broader range of topics, including humanitarian aid and disaster relief, as well as more general topics such as sports, family and weather. The dialogues often contain mentions of names and places, many of which are OOV words to the ASR system. As illustrated in the previous section, ASR hypotheses necessarily have errors in OOV regions, but because specific names are infrequent, even in-vocabulary names can have these types of error patterns. Therefore, developing a robust name error detector for ASR hypotheses is an important component of the system to resolve errors and ambiguity.

Detecting OOV errors requires combining evidence of recognizer uncertainty and anomalous word sequences in a local region. For name errors, lexical cues to names are also useful, e.g. a person's title, location prepositions, or keywords such as "name". The baseline system for this work uses structural features extracted from a confusion network of ASR hypotheses plus ASR word confidence to represent recognizer uncertainty, and word n-gram context to the left and right of the target confusion network slot. These features are combined in a maximum entropy (ME) classifier trained to predict name errors directly. This is the same as the baseline used in (He et al., 2014; Marin, 2015; Marin et al., 2015), but with a different ASR system.

Training a classifier to predict whether a sentence has a name is easier than direct name error prediction, because the positive class is less rare, and it does not require recognizer output so more data can be used (e.g. speech transcripts without recognizer output, or written text). In addition, since the words abutting the name are less reliable in a recognition hypothesis, the information lost by working at the sentence level is minimal. The idea of using sentence-level name prediction is proposed in (He et al., 2014), but in that work the sentence name posterior is a feature in the ME model (optionally with word cues learned by the sentence-level predictor). In our work, the problem is factored to use the acoustic confusibility and local word class features for OOV error prediction, which is combined with the sentence-level name posterior for name error prediction. In other words, only two features (posteriors) are used in training with the sparse name error prediction data. An ME classifier is then used to combine the two features to predict the word-level name error. The word-level OOV detector is another ME binary classifier; the full set of features used for the word-level OOV detector can be found in (Marin, 2015).

The main innovation in this work is that we propose to use a multi-task RNN model for the sentence-level name prediction, where the training objective takes into account both the language modeling task and the sentence-level name prediction task, as described in the next section.

3 Multi-task Recurrent Neural Network

The RNN is a powerful sequential model and has proven to be useful in many natural language processing tasks, including language modeling (Mikolov et al., 2010) and word similarity (Mikolov et al., 2013c). It also achieves good results for a variety of text classification problems when combined with a convolutional neural network (CNN) (Lai et al., 2015). In this paper, we

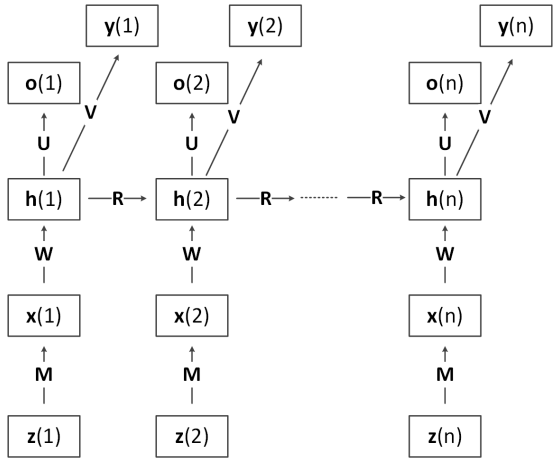


Figure 1: The structure of the proposed MT RNN model, which predicts both the next word $o(t)$ and whether the sentence contains a name $y(t)$ at each time step.

propose an MT RNN for the sentence-level name prediction task, which augments the word prediction in the RNN language model with an additional output layer for sentence-level name prediction. Formally, the MT RNN is defined over $t = 1, \ldots, n$ for a sentence of length n as:

$$\begin{aligned} \mathbf{x}(t) &= M\mathbf{z}(t), \\ \mathbf{h}(t) &= f\left(W\mathbf{x}(t) + R\mathbf{h}(t-1)\right), \\ \mathbf{o}(t) &= s\left(U\mathbf{h}(t) + \mathbf{b}_1\right), \\ \mathbf{y}(t) &= s\left(V\mathbf{h}(t) + \mathbf{b}_2\right). \end{aligned}$$

where $\mathbf{z}(t) \in \mathbb{R}^V$ is the 1-of-V encoding of the t-th word in the sentence; $\mathbf{x}(t) \in \mathbb{R}^d$ is the d-dimensional continuous word embedding corresponding to the t-th word; $\mathbf{h}(t) \in \mathbb{R}^k$ is a k-dimensional embedding that summarizes the word sequence through time t; and $\mathbf{o}(t)$ and $\mathbf{y}(t)$ are respectively the output layers for the language modeling task and the sentence-level prediction task. The parameters of the model (learned in multi-task training) include: $M \in \mathbb{R}^{d \times V}$, which is usually referred to as the word embedding matrix; projection matrices $U \in \mathbb{R}^{V \times d}$ and $V \in \mathbb{R}^{2 \times d}$; and bias terms $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^d$. f and s are respectively the sigmoid and softmax functions. Note that the word sequence associated with a sentence includes start and end symbols.

The structure of the proposed MT RNN is shown in Fig. 1. At each time step, the hidden vector is used to predict both the next word and a sentence-level indicator of the presence of a name, providing a probability distribution for both variables. Thus, the hidden vector h_t provides a continuous representation of the word history that emphasizes words that are important for predicting the presence of a name. The vector at time n can be thought of as a sentence embedding. The sentence-level output y_t differs from the word-dependent label predictor typically used in named entity detection (or part-of-speech tagging) in that it is providing a sequentially updated prediction of a sentence-level variable, rather than a word-level indicator that specifies the location of a named entity in the sentence. The final prediction y_n is used as a feature in the name error detection system. The sentence-level output y_t does not always converge to y_n gradually nor is it always monotonic, since the prediction can change abruptly (either positively or negatively) as new words are processed. The sentence-level variable provides a mechanism for capturing long distance context, which is particularly useful for speech applications, where both the name of interest and the words in its immediate context may be in error.

The training objective is the combination of the log-likelihood of the word sequence and that of the sentence-level name prediction:

$$\sum_{t=1}^{n} \left[(1-\lambda)\log P(w(t)|h(t)) + \lambda \log P(y(t)|h(t))\right], \quad (1)$$

where $h(t) = [w(1), \ldots, w(t-1)]$ and λ is the weight on the log-likelihood of the sentence-level name labels.

Another way to train the model is to predict the sentence-level name label *only* at the end of the sentence, rather than at every time step. Preliminary experiment results show that this model has inferior performance. We argue that training with only the sentence-final name label output can result in unbalanced updates, i.e., information from the language modeling task is used more often than that from the sentence-level name prediction task, implying that balancing the use of information sources is an important design choice for multi-task models.

The training objective is optimized using stochastic gradient descent (SGD). We also experiment with AdaGrad (Duchi et al., 2011), which has shown to be more stable and converge faster for non-convex SGD optimization. Since language model training requires a normalization operation each time over the whole vocabulary (~60K) which is computationally intensive, we further speed up training by using noise contrastive esti-

mation (NCE) (Mnih and Teh, 2012). For all models using NCE, we fix the number of negative samples to 50.

All the weights are randomly initialized in the range of $[-0.1, 0.1]$. The hidden layer size k is selected from {50, 100, 200} and the task mixing weight λ is select in {0.2, 0.4, 0.6, 0.8}, based on development set performance. We set the initial learning rate to 1 and 0.1 for SGD with and without Adagrad respectively. In our experiments, we observe that models trained with AdaGrad achieve better performance, so we only report the models with AdaGrad in this paper. At each training epoch, we validate the objective on the development set. The learning rate is reduced after the first time the development set loglikelihood decreases, and the whole training procedure terminates when the development set loglikelihood decreases for the second time.

Domain	Topics
TRANSTAC	Traffic Control, Facilities Inspection, Civil Affairs, Medical, Combined Training, Combined Operations
HADR	Humanitarian Aid, Disaster Relief
General	Family, Gardening, Sports, Pets, Books, Weather, Language, Phone

Table 1: Topics in different domains.

Target Class	BOLT-Train	Dev1	Dev2	Test1	Test2
name sentences	7.6	7.5	8.2	8.1	14.0
name errors	0.8	1.1	0.9	0.7	0.8
OOVs	1.7	1.8	1.8	1.8	1.7

Table 2: Data splits and statistics, including the percentage of sentences containing names (name sentences), the percentage of hypothesized words that are name errors (name errors), and the percentage of words that are OOVs (OOVs).

4 Experiments

4.1 Data

There are two types of datasets used in this paper: the BOLT dataset and a collection of Reddit discussions. The first dataset was collected during the DARPA TRANSTAC and BOLT projects. The ASR hypotheses of totally 7088 spoken sentences makes up of the training dataset (BOLT-Train) for both sentence-level name prediction and word-level name error detection. There are two development sets: Dev1 and Dev2. Dev1 is used for parameter tuning for all models, and Dev2 is used for training the ME-based word-level name error detector using the word-level OOV posterior and sentence-level name posterior. For RNN models, we tune the hidden layer size and the multi-task weight λ; for the ME-based word-level name error detector, we tune the regularization parameter. Two test sets are used to evaluate sentence-level name prediction (Test1) and word-level name error detection (Test2), based on the BOLT phase 2 and 3 evaluations, respectively.

As shown in Table 1, the BOLT topics are categorized into three domains: TRANSTAC, HADR and General. The BOLT-Train, Dev1 and Dev2 sets contain only speech from the TRANSTAC domain, whereas the Test1 and Test2 sets contain all three domains. Detailed data split statistics and domain information are summarized in Table 2. Note that there are very few positive samples of name errors (roughly 1%), whereas for the sentence-level name prediction task and the word-level OOV prediction task, the data skewness is somewhat less severe (roughly 8% sentences with names in most of our data sets).

In order to address the issue of domain mismatch between the training data and the broad-domain subsets of the test data, we collect text from *Reddit.com* which is a very active discussion forum where users can discuss all kinds of topics. Reddit has thousands of user-created and user-moderated *subreddits*, each emphasizing a different topic. For example, there is a general ASKREDDIT subreddit for people to ask any questions, as well as subreddits targeted for specific interests, like ASKMEN, ASKWOMEN, ASKSCIENCE, etc. Although the Reddit discussions are different from BOLT data, they have a conversational nature and names are often observed within the discussions. Therefore, we hypothesize that they can help improve sentence-level name prediction in general. We collect data from 14 subreddits that cover different kinds of topics (such as politics, news, book suggestions) and vary in community size. The Stanford Name Entity Recognizer (Finkel et al., 2005) is used to detect names in each sentence, and a sentence-level name label is assigned if there are any names present.[1] The data are tokenized using Stanford

[1]The Stanford Name Entity Recognizer achieves 82.3% F-score on the references of Dev1. Thus, it is expected to

Model	TRANSTAC	HADR	General	All
BOW + ME	52.9	32.5	11.1	40.8
RNN + ME	11.2	21.7	30.5	15.0
SG + ME	14.7	12.6	16.3	14.6
ST RNN	40.3	37.5	**37.8**	39.3
MT RNN	**59.8**	**52.0**	23.8	**51.1**

Table 3: F-scores on Test1 for sentence-level name prediction for models trained on BOLT data.

CoreNLP tools (Manning et al., 2014) and are lowered cased after running the Stanford Name Entity Recognizer. In total, we obtain 135K sentences containing at least one name and 360K sentences without names.

4.2 Sentence-level Name Prediction

To evaluate the effectiveness of the proposed MT RNN model, we first apply it to the sentence-level name prediction task on ASR hypotheses. For this task, each sample corresponds to a hypothesized sentence generated by the ASR system and a ground-truth label indicating whether there are names in that sentence. We compare the MT RNN with four contrasting models for predicing whether a sentence includes a name.

- **BOW + ME**. An ME classifier using a bag-of-words (BOW) sentence representation.
- **SG + ME**. An ME classifier is used with the sentence embedding as features, where the embedding uses the skip-gram (SG) model (Mikolov et al., 2013a) to get word-level embeddings (with window size 10) and sentence embeddings are composed by averaging embeddings of all words in the sentence.
- **RNN + ME**. A simple RNN LM is trained (i.e., λ in (1) is set to 0), and the hidden layer for the last word in a sentence provides a sentence-level embedding that is used in an ME classifier trained to predict the sentence-level name label.
- **ST RNN**. A single-task (ST) RNN model is trained to directly predict the sentence-level name for each word (i.e., λ in (1) is set to 1).

All models are trained on either BOLT-Train or BOLT-Train + Reddit, and tuned on Dev1 including the dimension of embeddings, ℓ_2 regularization parameters for the ME classifiers, and so on.

The domain-specific F-scores on Test1 are summarized in Table 3 for training only with the BOLT

give useful labels for the Reddit data.

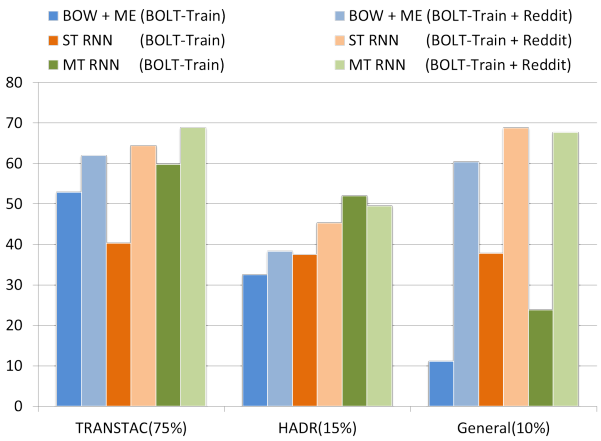


Figure 2: F-scores on Test1 for sentence-level name prediction for models trained with and without Reddit data. The number in the parenthesis indicates the portion of the domain in the data.

data. The proposed MT RNN achieves the best results on the TRANSTAC and HADR domains, and it has significant overall performance improvement over all baseline models. Not surprisingly, the two approaches that used unsupervised learning to obtain a sentence-level embedding (SG + ME, RNN + ME) have the worst performance on the TRANSTAC and HADR domains, with both having very low precision (6-18%), with best precision on the general domain. The BOW + ME and the ST RNN have similar performance in terms of overall F-score, but the BOW + ME model is much better on the TRANSTAC domain, whereas the ST RNN achieves the best results on the General domain. The main failing of the BOW + ME model is in recall on the General domain, though it also has relatively low recall on the HADR domain. Note that the General domain only accounts for 10% of the Test1 set, so the ST RNN gets lower F-score overall compared with the MT RNN, which performs best on the other two domains (90% of the Test1 set). On all domains, the MT RNN greatly improves precision compared to the ST RNN (at the expense of recall), and it improves recall compared to the BOW + ME approach (at the expense of precision).

In order to study the effectiveness of using external data, we also train all models on an enlarged training set including extra name-tagged sentences from Reddit. The improvement for each domain due to also using the Reddit data is shown in Fig. 2 for the three best configurations. (There is no benefit in the unsupervised learning cases.) Com-

pared with training only on the BOLT data, all three of these models get substantial overall performance improvement by utilizing the external domain training data. Since the external training data covers mostly topics in the General domain, the performance gain in that domain is most significant. For the MT RNN and BOW + ME classifiers, the additional training data primarily benefits recall, particulary for the General domain. In contrast, the ST RNN sees an improvement in precision for all domains. One reason that the added training text also benefits the models on the TRANSTAC domain is that over 90% of the words in the speech recognizer vocabulary are not seen in the small set of name-labeled speech training data, which means that the embeddings for these words are random in the RNNs trained only this data and the BOW + ME classifier will never use these words.

4.3 Word-level Name Error Detection

We next assess the usefulness of the resulting MT RNN model for word-level name error detection on ASR hypotheses. Here, several word-level name error detection approaches are compared, including direct name error prediction and factored name/OOV prediction approaches.

- **OOV thresholding**. This system simply uses the OOV prediction, but with the posterior threshold tuned for word-level name error based on the Dev1 set.
- **Word Context**. This is the baseline ME system described in Section 2 and also used as a baseline in (He et al., 2014; Marin et al., 2015), which directly predicts the word-level name error using WCN structural features, current word Brown class, and up-to trigram left and right word context information.
- **Word Class**. This system, from (Marin et al., 2015), also directly predicts the word-level name error, but replaces the word n-gram features with a smaller number of word class features to address the sparse training problem. The word classes are based on seed words learned from sentence-level name prediction which are expanded to classes using a nearest neighbor distance with RNN embeddings trained on the BOLT data.
- **Word Context + OOV**. This system uses an ℓ_2 regularized ME classifier to predict the word-level name error using two posteriors as features: a word-dependent posterior from the Word Context system and one from the OOV detection system.
- **MT RNN + OOV**. This system also uses an ℓ_2 regularized ME classifier to predict the word-level name error using two posteriors as features: the same word-dependent OOV posterior as above, and the posterior from the sentence-level name prediction using the MT RNN model described in Section 4.2, which is constant for all positions in the sentence.

All of the above models are trained on BOLT-Train and tuned on Dev1. The ME classifier used in the Word Context + OOV and the MT RNN + OOV systems are trained on Dev2, with regularization weights and decision boundaries tuned on Dev1.

Name error detection results (F-scores) are summarized in Table 4. The three systems that use discrete, categorical lexical context features (word or word class context) in direct name error prediction have worse results than the OOV thresholding approach overall, as well as on the TRANSTAC subset. Presumably this is due to over-fitting associated with the lexical context features. The factored MT RNN + OOV system, which uses a continuous-space representation of lexical context, achieves a gain in overall performance compared to the other systems and a substantial gain in performance on the TRANSTAC domain on which it is trained. Using the Reddit data further improves the overall F-score, with a slight loss on the TRANSTAC subset but substantial gains in the other domains. Although the performance improvement in the general domain is relatively small compared with sentence-level name prediction, utilizing external data makes the resulting representation more transferable across different domains and tasks. The best MT RNN + OOV system obtains 26% relative improvement over the baseline Word Context system (or 17% improvement over the simple OOV thresholding approach).

Looking at performance trade-offs in precision and recall, we find that the use of the RNN systems mainly improves precision, though there is also a small gain in recall overall. The added training data benefits recall for all domains, with a small loss in precision for the TRANSTAC and HADR sets. The use of the OOV posterior improves precision but limits recall, particularly for the general domain where recall of the OOV posterior alone

Model	TRANSTAC	HADR	General	All
OOV thresholding	41.6	31.2	16.5	30.9
Word Context	37.6	29.0	16.3	28.6
Word Class	33.6	35.9	11.9	27.9
Word Context + OOV	40.2	26.0	14.4	28.4
MT RNN + OOV	47.9	32.8	13.5	34.1
MT RNN† + OOV	46.4	37.5	18.0	36.2

Table 4: F-scores on Test2 for word-level name error prediction. MT RNN is trained on BOLT-Train, whereas MT RNN† is trained on BOLT-Train + Reddit.

is only 18% vs. 25% for the word context model with the OOV posterior information.

To better understand some of the challenges of this task, consider the following examples (R=reference, H=hypothesis):

R1: i'm doing good my name is captain **rodriguez**

H1: i'm doing good my name is captain *road radios*

R2: well it's got flying **lizards** knights and **zombies** and shit

H2: well it's gotta flying lives there it's nights and some bees and shia

R3: i live in a city called **omaha**

H3: i live in a city called *omar*

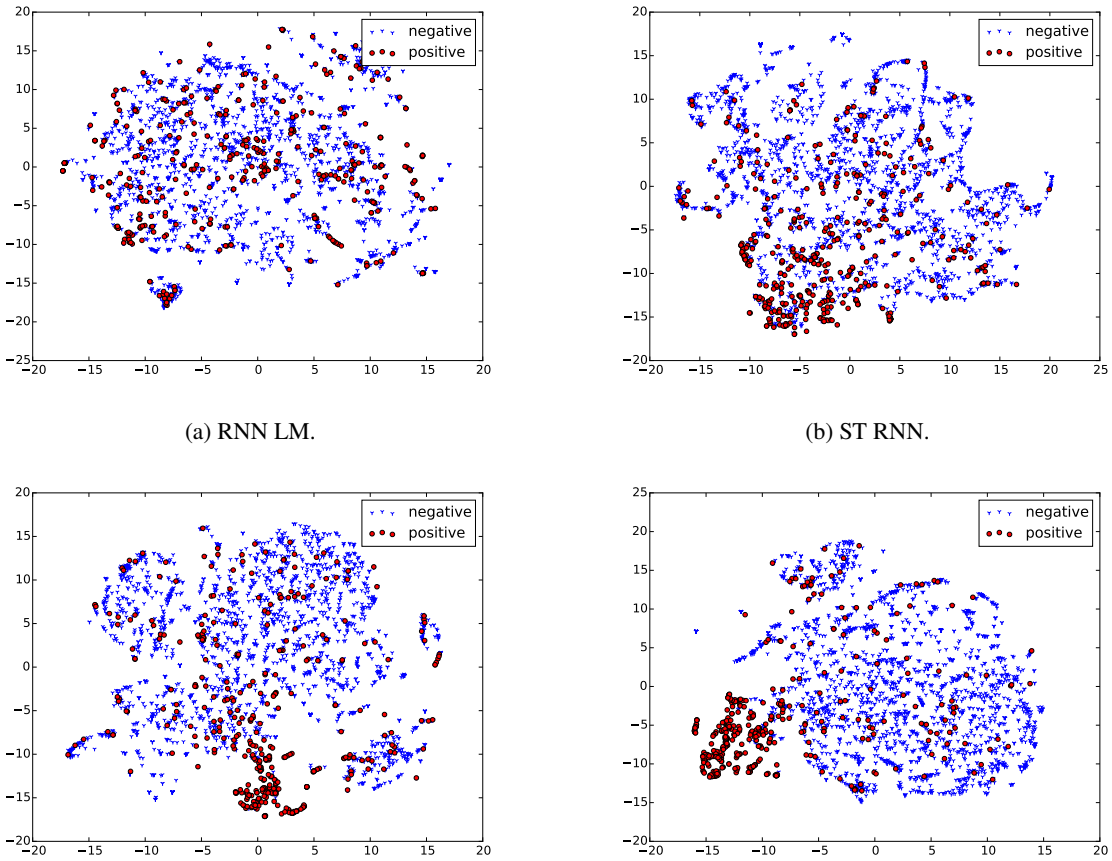
ASR tokens associated with name errors are underlined and italicized; tokens associated with non-name OOV errors are simply underlined. Name errors have a similar character to OOV erors in that they often have anomalous word sequences in the region of the OOV word (examples 1 and 2), which is why the OOV posterior is so useful. However, too much reliance on the OOV posterior leads to wrongly detecting general OOV errors as name errors ('lizards' and 'zombies' in example 2) and missed detection of name errors where the confusion network cues indicate a plausible hypothesis ('omaha' in example 3). Examples 1 and 3 illustrate the importance of lexical cues to names ('name ... captain', 'city called'), but word-based cues are unreliable for the systems trained only on the small amount of domain-specific data. Leveraging the reddit data allowed the MT RNN system to detect the error in example 1 (HADR domain) that was missed by the word context system. Example 3 was only detected by the word context system when no OOV posterior is used. Though this example was from the General domain, city names represent an important error class in the TRANSTAC data, so the term 'city' is learned as a useful cue.

4.4 Sentence Embedding

As discussed in Section 3, we postulate that by modeling words in a sentence in sequential order and simutaneously predicting sentence categorical information, the resulting hidden vector of the last word should be a good representation of the whole sentence, i.e., a sentence embedding. To provide support for this hypothesis and show the impact of external data, we present the sentence embeddings learned by the different RNN variants on Test2 using the t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization (van der Maaten and Hinton, 2008) in Fig. 3. Four models are compared, including three RNNs trained on the BOLT data, and the MT RNN model trained on BOLT + Reddit data. Note the visualization method t-SNE does not take the label information into account during the learning.

As we can see in Fig. 3a, the positive and negative sentence embeddings learned by RNN LM are randomly scattered in the space, indicating that embeddings learned via unsupervised training (e.g., solely on word context) may fail to capture the sentence-level indicators associated with a particular task, in this case presence of a name. When the results are ploted in terms of domains, the embeddings are similarly broadly scattered – there is no obvious topic representation in the embeddings.

When comparing Fig. 3b to Fig. 3a, which is associated with the RNN using a sentence-final name indicator, we can see that a lot of positive vectors have moved to the bottom-left of the space, though there is still a relatively large overlap between positive and negative embeddings. In Fig. 3c, corresponding to the word-level MT RNN, there forms a separable subgroup of positive embeddings and the overlapping seems to be reduced as well in contrast to Figs. 3a and 3b. Finally, most of the positive sentence embeddings produced by the MT RNN model trained with external Reddit data gather at the botton-left of Fig. 3d. In general, the overlap between positive and negative sentences decreases from single task models to multi-task model. The external data make the proposed model produce more well-shaped groupings of sentence embeddings.



(a) RNN LM.

(b) ST RNN.

(c) MT RNN.

(d) MT RNN using extra Reddit data.

Figure 3: t-SNE visualization of sentence embeddings learned from different RNN models.

5 Related Work

Recently, due to the success of continuous representation methods, much work has been devoted to studying methods for learning word embeddings that capture semantic and syntactic meaning. Although these word embeddings are shown to be successful in some word-level tasks, such as word analogy (Mikolov et al., 2013b; Jeffery Pennington, 2014) and semantic role labeling (Collobert and Weston, 2008), it is still an open question how best to compose the word embeddings effectively and make use of them for text understanding.

Recent work on learning continuous sentence representations usually compose the word embeddings using either a convolutional neural network (CNN), a tree-structured recursive NN, or a variant of an RNN. A typical CNN-based structure composes word embeddings in a hierachical fashion (alternating between convolutional layers and pooling layers) to form the continuous sentence representation for sentence-level classification tasks (Kim, 2014; Kalchbrenner et al., 2014; Lai et al., 2015). These models usually build up the sentence representation directly from the lexical surface representation and rely on the pooling layer to capture the dependencies between words. Another popular method for continuous sentence representation is based on the recursive neural network (Socher et al., 2012; Socher et al., 2013; Tai et al., 2015). These models use a tree structure to compose a continuous sentence representation and have the advantages of capturing more fine-grained sentential structure due to the use of parsing trees. Note that the RNN-based sequential modeling used in this paper can be viewed as a linearized tree-structure model.

In this paper, we train the neural network model with a multi-task objective reflecting both the probability of the sequence and the probability that the sequence contains names. The general idea of multitask learning dates back to (Caruana, 1997), and is shown to be effective recently for neural network models in different natural language

processing tasks. Collobert and Weston (2008) propose a unified deep convolutional neural network for different tasks by using a set of task-independent word embeddings together with a set of task-specific word embeddings. For each task, it uses a *unique* neural network with its own layers and connections. Liu et al. (2015) propose a different neural network structure for search query classification and document retrieval where lower-level layers and connections are all shared but the high-level layers are task-specific. For tasks considered in (Collobert and Weston, 2008) and (Liu et al., 2015), training samples are task-dependent. Thus, both models are trained following the SGD manner by alternating tasks for each training samples with task-dependent training objectives. In this paper, we combine the language modeling task with the sentence-level name prediction task, and each training sample has labels for both tasks. Therefore, the SGD training can be done with the weighted sum of the task-specific objectives for each training sample, and the language model objective can be thought of as a regularization term. Similar settings of multitask learning for neural network models are employed in phoneme recognition for speech (Seltzer and Droppo, 2013) and speech synthesis (Wu et al., 2015) as well, but both of them use equal weights for all tasks.

6 Conclusion

In this paper, we address an open domain rare event detection problem, specifically, name error detection on ASR hypotheses. To alleviate the data skewness and domain mismatch problems, we adopt a factored approach (sentence-level name prediction and OOV error prediction) and propose an MT RNN for sentence-level name prediction. The factored model is shown to be more robust to the sparse training problem. For the problem of sentence-level name prediction, the proposed method of combining the language modeling and sentence-level name prediction objectives in an MT RNN achieves the best results among studied models for the domain represented by the training data as well as in the open-domain scenario. Visualization of sentence-level embeddings show how both the multi-task and the word-level name label update are important for achieving good results. The use of unrelated external training text (which can only be used in sentence-level name prediction) improves all models, particularly for the highly-mismatched general domain data.

The improvement in performance associated with using the external text is much smaller on the word-level name error detection task than on the sentence-level name prediction. This seems to be due to the high weight learned for the word-level posterior. For future work, it is worthwhile looking into whether continuous word and sentence representations can be combined in the name error detector to achieve further improvement.

In this work, we proposed a model for learning sentence representations that might be useful for other sentence classification tasks, such as review and opinion polarity detection, question type classification and so on. As discussed in Section 5, there are other models that have been found useful for obtaining continuous sentence embeddings. It would be of interest to investigate whether other structures are more or less sensitive to data skew and/or useful for incorporating multi-domain training data.

Acknowledgments

The authors would like to thank Alex Marin, Ji He, Wen Wang and all reviewers for useful discussion and comment. This material is based on work supported by DARPA under Contract No. HR0011-12-C-0016 (subcontract 27-001389). Any opinions, findings and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of DARPA.

References

N.F. Ayan, A Mandal, M. Frandsen, Jing Zheng, P. Blasco, A Kathol, F. Bechet, B. Favre, A Marin, T. Kwiatkowski, M. Ostendorf, L. Zettlemoyer, P. Salletmayr, J. Hirschberg, and S. Stoyanchev. 2013. "Can you give me another word for hyperbaric?" Improving speech translation using targeted clarification questions. In *Proc. ICASSP*, pages 8391–8395.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28, July.

Wei Chen, Sankaranarayanan Ananthakrishnan, Rohit Prasad, and Prem Natarajan. 2013. Variable-span out-of-vocabulary named entity detection. In *Proc. Interspeech*, pages 3761–3765.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. ICML*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Machine Learning Research*, 12.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. ACL*.

Ji He, Alex Marin, and Mari Ostendorf. 2014. Effective data-driven feature learning for detecting name errors in automatic speech recognition. In *Proc. SLT*.

Christopher Manning Jeffery Pennington, Richard Socher. 2014. Glove: Global vectors for word representations. In *Proc. EMNLP*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proc. ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proc. EMNLP*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proc. AAAI*.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proc. NAACL*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proc. ACL*.

Alex Marin, Mari Ostendorf, and Ji He. 2015. Learning phrase patterns for asr error detection using semantic similarity. In *Proc. Interspeech*.

Alex Marin. 2015. *Effective use of cross-domain parsing in automatic speech recognition and error detection*. Ph.D. thesis, University of Washington.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proc. Interspeech*.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*.

Tomáš Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proc. NAACL-HLT*.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proc. ICML*.

David Palmer and Mari Ostendorf. 2005. Improving out-of-vocabulary name resolution. *Computer Speech and language*, 19(1):107–128.

Carolina Parada, Mark Dredze, and Frederick Jelinek. 2011. OOV sensitive named-entity recognition in speech. In *Proc. Interspeech*, pages 2085–2088.

Michael L. Seltzer and Jasha Droppo. 2013. Multitask learning in deep neural networks for improved phoneme recognition. In *Proc. ICASSP*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector space. In *Proc. EMNLP-CoNLL*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. EMNLP*.

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. 2006. Incorporating speech recognition confidence into discriminative named entity recognition of speech. In *Proc. ACL*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proc. ACL*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Machine Learning Research*, 9, November.

Zhizheng Wu, Cassia Valentini-Botinhao, Oliver Watts, and Simon King. 2015. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *Proc. ICASSP*.