FEATURE-ENRICHED WORD EMBEDDINGS FOR NAMED ENTITY RECOGNITION IN OPEN-DOMAIN CONVERSATIONS

Yukun Ma, Jung-jae Kim†, Benjamin Bigot*, Tahir Muhammad Khan**

*Rolls-Royce@NTU Corporate Lab, Nanyang Technological University, Singapore
† Data Analytics Department, Institute for Infocomm Research, Singapore

ABSTRACT

Named entity recognition (NER) from open-domain conversation is challenging due to the informality of spoken language. Instead of increasing the size of labeled data, which is expensive and time-consuming, word embeddings learned from unlabeled data have been used by NER models to handle data sparsity. We propose a novel method for training the word embeddings specifically for the NER task. We show that our task-specific word embeddings outperform task-independent word embeddings when used as features of NER method.

Index Terms— word embedding, named entity, conversation,

1 Introduction

Named entity recognition (NER) has been studied and applied successfully to formal [1] and informal texts [2] for many years. Nevertheless the adaptation of NER methods to conversational speech remains challenging due to, for example, case insensitivity, lack of punctuations, ungrammatical structure, repetition, and presence of disfluencies inherent to conversations. In addition, there is not much spoken data annotated with named entities to cover the huge variety of named entity instances likely occurring in speech, and simply increasing the amount of manual annotation is not realistic for reasons of cost, evolution of new spoken terms and diversity.

Several works on NER from spoken contents have already explored the use of external resources like online gazetteers [3] and Wikipedia [4] to overcome the lack of annotations. Gazetteers, for instance, have successfully boosted NER performance for given entities (e.g. Location), but do not convey the information related with the context words surrounding the entity names that are also important for NER. Other lexical resources such as WordNet provide semantic relations like synonymy among common English words, but remain limited for names. A second category of approaches [5, 6, 7, 8, 9] tackling the sparseness of NER training data use unlabeled data to learn low dimensional vector representation of words, called word embeddings. Used either as continuous [5, 6, 9] or discrete features [7, 8], word embeddings have been shown effective in improving the generalization of NER.

For the last two years, an increasing number of studies have suggested that injecting application-specific information into the neural networks used to train word embeddings can further improve the performance of down-stream applications, e.g., dependency parsing [10], semantic relation classification [11], antonym detection [12], spoken language understanding [13]. The task-specific information used by these methods are injected into the training process of word embeddings mainly by expanding or replacing the input or output of the neural network. Passos *et al.* [14], to our knowledge, is the only work trying to learn word embeddings for NER by leveraging lexicons related to named entities. However, their approach uses only lexicon features indicating if the current word belonging to a limited number of semantic classes or not, and does not use additional information of context for learning word embeddings.

We thus propose an NER framework including a modified word embedding method based on Skip-gram models [15], in which we generalize the training objective to integrate features explicitly designed for NER task and grouped by types. Injecting such NER-specific information as part-of-speech tags, taxonomic relations, and self-training features [4] yields NER improvements over baseline (i.e., without using word embeddings), when evaluated against conversational speech transcripts (i.e., Switchboard Corpus). Furthermore, the results show that our feature-enriched word embeddings also outperform the task-independent word embeddings.

2 Methods

The proposed NER framework illustrated in Figure 1 is composed of two parts: 1) a task-specific word embedding learning integrating features specific to NER (Part I in Figure 1.), using not only unlabeled data but also resources like knowledge base and baseline NER tagger; and 2) a supervised model of linear-chain Conditional Random Field (CRF) trained with the NER features extracted from the labeled corpus and the resultant word embeddings (Part II in Figure 1). We denote the proposed word embedding method as Skip_{NER}.

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

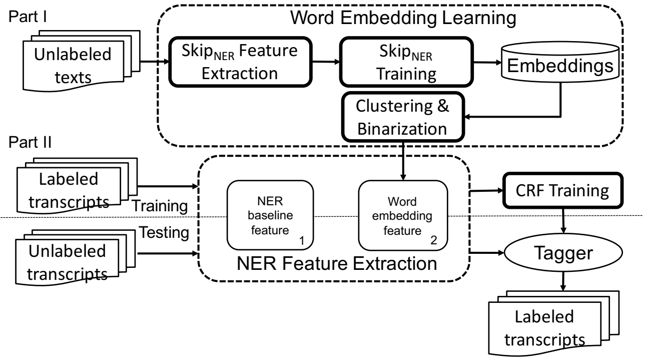

Fig. 1. NER Framework with NER-specific word embeddings

2.1 Feature extraction for word embeddings

We extract four types of features (i.e. neighboring words, part of speech (POS), taxonomic, self-trained) from the unlabeled data to train word embeddings.

Neighboring words and POS tags: They are acknowledged to be efficient for NER [3, 4]. In fact, we use these features not only for training our word embeddings, but also for training the baseline model of NER (Section 2.4). Their formal definitions can be found in the first 4 lines of Table 2.

Taxonomic features: The generalization of similar entities within a common category (or concept) in a taxonomy may be useful for NER to capture contexts shared by similar entities. We select 280 concepts with at least 100 instances from ConceptNet, which is a large, automatically created semantic graph including taxonomic relations. For each concept g, we add binary taxonomic features, which are defined as $\text{TX}_g(w_{i+k}), 2 \leq k \leq 2$, where i is the index of current word, and $\text{TX}_g(w)$ indicates if the word w belongs to the concept g or not.

Self-trained features: As in [16], these features are generated by automatically labeling the training data using a baseline NER tagger (Section 2.4). We use the resultant named entities hypotheses as additional features for word embeddings. The features are defined as T_{i+k}, with T_j the NER label of the j^{th} word.

2.2 Feature-enriched word embeddings

Our proposal is based-on the Skip-gram model [15], a neural language model that can be efficiently trained on a large corpus of billions of words. Its objective function is the sum of log probabilities $p(w_{i+j}|w_i)$ over the whole corpus,

$$\sum_{i=1}^{N}\sum_{j=-k}^{k}\log p(w_{i+j}|w_i)$$

, where w_{i+j} indicates a neighbor word of w_i, k the size of the context window and N is the length of the dataset. And, the basic Skip-gram formulation defines $p(w_{i+j}|w_i)$ using the softmax function as follows:

$$p(w_{i+j}|w_i) = \frac{\exp(v'^{\top}_{w_{i+j}} v_{w_i})}{\sum_{w\in W}\exp(v'^{\top}_{w} v_{w_i})}$$

, where W is the vocabulary of words, $v'_{w_{i+j}}$ and v_{w_i} are the embeddings of context word and current word, respectively. We modify the Skip-gram model to predict the set of features $F(w_i)$ extracted for the given word w_i at the i^{th} position of a corpus. The objective function can be rewritten:

$$\sum_{i=1}^{N}\sum_{f\in F(w_i)}\log p(f|w_i)$$

To estimate $p(w_{i+j}|w_i)$, the basic Skip-gram model assumes a single distribution over all the words. However, for $p(f|w_i)$, since the features we use for training application-specific word embeddings are heterogeneous, they may have different distributions. We thus split the whole set of features (S) into subsets (S_X), where X indicates one of the four feature types aforementioned and the relative position to the center word. For example, the subset $S_{pos:-1}$ includes all the features that tell the POS tags of the previous word. We define $C_S(f)$ as the function that returns the subset of S that contains the feature f. We define the probability of extracting f for a given word w from the training data as follows:

$$p(f|w) = \frac{\exp(v_f^{\top} v_w)}{\sum_{f'\in C_S(f)}\exp(v_{f'}^{\top} v_w)}$$

, where v_w and v_f are the vectors associated with center word w and feature f respectively, and both are the parameters to be learned. Note that, with the new definition, the objective function becomes a linear combination of the objective functions of multiple classifiers with equal weights. We optimize this objective function using stochastic gradient descent and negative sampling method previously proposed for Skip-gram [15], which rewrites the objective function as:

$$\sum_{i=1}^{N}\sum_{f\in F(w_i)}\left(\log\sigma(v_f^T v_{w_i}) + \sum_{f'\in Z(f)}\log\left(\sigma(-v_{f'}^T v_{w_i})\right)\right)$$

In this expression $\sigma(x) = 1/(1+\exp(-x))$, and the set of features $Z(f)$ is created by randomly selecting n negative samples from a unigram distribution over features in $C_S(f)$

2.3 Using word embeddings as NER features

We convert word embeddings to additional features of the CRF model as in [7, 8] by binarizing vector elements of word embeddings, and clustering of words based on similarity of word embeddings. The vectors are binarized using the following rules,

$$D_{mn} = \begin{cases} 1 & \text{if } W_{mn} \geq \overline{W^{+}}_{m\cdot} \\ -1 & \text{if } W_{mn} \leq \overline{W^{-}}_{m\cdot} \\ 0 & \text{else} \end{cases}$$

, where W is the original word embedding matrix, and D the binarized matrix, $\overline{W^{+}}_{m\cdot}$ the mean of all positive values of the m^{th} dimension, and $\overline{W^{-}}_{m\cdot}$ the mean of all negative values. Only non-zero values are added to the feature set. $\overrightarrow{V_D(w)}$ denotes the column of D corresponding to the word w. Then we cluster all words based on Euclidean distance using K-Means [8, 7]. We use different numbers of clusters, K, to let the clustering reflect different levels of granularity. $C_K(w)$ is the cluster of the word w, where K indicates the number of clusters in the K-Means (500≤K≤3000). The features learned from word embeddings are summarized in Table 1.

Binarized vector	$\overrightarrow{V_D(w_{i+k})}$	$-2 \leq k \leq 2$
Cluster (Unigram)	$C_K(w_{i+k})$	$-2 \leq k \leq 2$
Cluster (Bigram)	$C_K(w_{i+k}) \wedge C_K(w_{i+k+1})$	$-2 \leq k \leq 1$
Cluster (Disjunct)	$C_K(w_{i-1}) \wedge C_K(w_{i+1})$	

Table 1. Features learned from word embeddings for NER

2.4 Baseline NER method

In this section, we describe the baseline NER method, a conventional linear-chain CRF with BIO encodings. The features extracted from every word w_i of the training data and used to train the CRF are summarized in Table 2. The feature set consists of n-grams, part-of-speech (POS) tags, affixes, and the BIO tag of the i^{th} word (designated as y_i).

Context Features (CF)	
Unigram	$w_{i+k}, \ -2 \leq k \leq 2$
Bigram	$w_{i+k} \wedge w_{i+k+1}, \ -2 \leq k \leq 1$
POS	$t_{i+k}, \ -2 \leq k \leq 2$
POS bigram	$t_{i+k} \wedge t_{i+k+1}, \ -2 \leq k \leq 1$
Prefix	$Pre(w_{i+k}, l), \ -2 \leq k \leq 2, \ 0 \leq l \leq 4$
Suffix	$Suf(w_{i+k}, l), \ -2 \leq k \leq 2, \ 0 \leq l \leq 4$
Tag Feature	
Tag&Context	$y_i \wedge c\,, \quad c \in CF$
Tag Bigram	$y_{i-1}y_i$

Table 2. Templates of features used in the CRF baseline

3 Experiments

3.1 Corpora

We have used the **ukWaC corpus** [17] as unlabeled text dataset to train word embedding models. This is one of the largest English text corpora (about 1.8 billion tokens), crawled from the Web on the *co.uk* domains, and has been tagged with POS[1].

The labeled open-conversation dataset is the **Switchboard corpus** [18]. It is a large collection of 2-speaker conversational telephonic speech recordings. We use the NER annotations of the corpus including four classes – location (LOC), person (PER), organization (ORG) and miscellaneous (MISC), and the train/test partitions provided by Surdeanu *et al.* [3]. Detailed numbers are presented in Table 3.

	LOC	PER	ORG	MISC
Train	14,397	4,257	5,311	8,484
Test	631	342	296	687

Table 3. Named entity distribution in Switchboard

We compare the proposed Skip-gram model (Skip$_{\text{NER}}$) with non-task specific word embeddings models previously used for NER. We retrained the word embeddings for Skip-gram, CBOW [15], and Glove [9] with the ukWaC corpus, using two-word context windows for each side of the current word. We used the pre-trained word embeddings models[2] of HUANG [19] and SENNA [5], because their training is slow. Table 4 shows some statistics of the data for the models. Following [6, 5, 8, 9], the number of vector dimensions is set to 50 for all experiments.

Word embeddings	**Tokens**	**Vocab.**	**Dataset**
Skip$_{\text{NER}}$, Skip-gram	1.8B	167K	ukWaC
Glove, CBOW	1.8B	167K	ukWaC
HUANG	1.8B	100K	Wikipedia
SENNA	1.8B	130K	Wikipedia

Table 4. Details on baseline Word Embeddings preparation

3.2 Evaluations

We report in Table 5 the performance of the proposed NER method (Skip$_{\text{NER}}$) and the baseline method with the other word embeddings when tested against the manual transcripts of open-domain conversations. Our method outperforms all the other methods and achieves 2% absolute improvement of F-score in comparison to the original Skip-gram model.

System	**F-score**	**System**	**F-score**
Baseline	66.51		
Baseline + Skip	67.89	Baseline + Huang	67.88
Baseline + CBOW	68.45	Baseline + Senna	68.29
Baseline + Glove	67.44	Baseline + Skip$_{\text{NER}}$	**70.19**

Table 5. Performance of NER methods in terms of F-score

[1] http://www.cis.uni-muenchen.de/∼schmid/tools/TreeTagger/

[2] http://ai.stanford.edu/∼ehhuang/, http://ronan.collobert.com/senna/

In the second experiment, we investigate the impact of the amount of labeled training data used to train the NER model. As illustrated in Figure 2, the Skip$_{\text{NER}}$-based NER system always outperforms the other systems, independent of the amount of training data. Also, the NER systems with the generic (or non task-specific) word embeddings also perform better than the baseline. It suggests that the generalization brought by word embeddings consistently helps NER models in dealing with data sparsity. As more training data are used (80%-100% of training data), the gap of performance between all the baseline systems reduces. However, the gap between the baseline and Skip$_{\text{NER}}$ further increases even using more training data, which may mean that our method robustly handles data sparsity and extracts more discriminative information when more data are available.

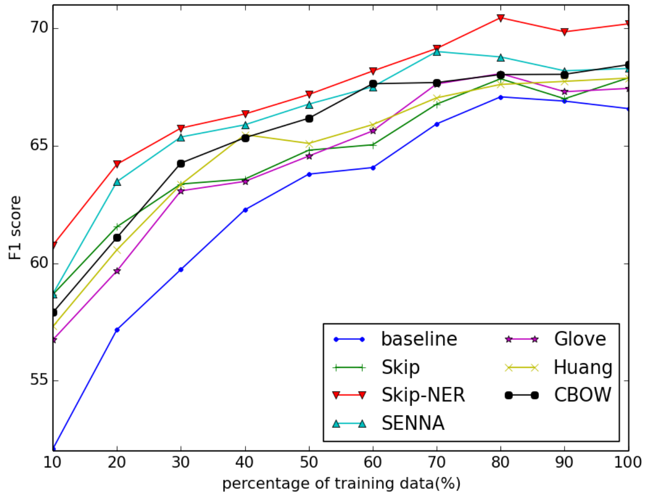

Fig. 2. Performance with varying size of NER training data

We also study the impact of each subset of features used to train the Skip$_{\text{NER}}$ model. As reported in Table 6, we find that using the whole set of features outperforms each of the feature subsets used alone. This result may indicate that the feature subsets are complementary to each other, thus supporting the proposed method of integrating NER-specific features for training word embeddings. Also, note that the Words only subset corresponds to the skip-gram model. Location and person names are particularly well recognized by using the POS subset, while self-trained features and taxonomic relations seem more effective to discriminate the class ORG.

Skip$_{\text{NER}}$ Feature	PER	LOC	ORG	MISC	All
Words	80.6	79.3	51.7	57.0	68.5
POS	**82.3**	80.6	51.1	57.8	69.4
self-trained	80.8	79.8	53.6	57.2	68.9
taxonomic	81.0	80.0	54.0	57.4	69.3
All	81.7	**80.9**	**55.7**	**57.9**	**70.2**

Table 6. Impact of feature subsets on Skip$_{\text{NER}}$

Word	texas	cowboys	batman
Skip-gram	lousiana spokane kansas sacramento biloxi	texans cowboy bandits cowgirls impersonators	superman superhero remake spiderman catwoman
Skip$_{\text{NER}}$	kentucky kansas lousiana florida minnesota	cheerleaders texans yankees redskins broncos	superman shrek starsky scooby-doo spiderman

Table 7. Top similar words returned by Skip-gram & Skip$_{\text{NER}}$

Table 7 lists the five words most similar to each of three example words (i.e. cowboys, texas, batman), which are computed using the Skip-gram and Skip$_{\text{NER}}$. For instance of the keyword *texas*, both Skip$_{\text{NER}}$ and Skip-gram models return locations in the United States (US), but Skip$_{\text{NER}}$ seems to produce more 'relevant' results than Skip-gram since *texas* and all its five most similar words (e.g. *kentucky*) are the names of states in US, while the results of Skip-gram include city names (e.g. *spokane*, *sacramento*), thus of different granularity. For ambiguous words, Skip$_{\text{NER}}$ seems to focus on the semantics of the words related to the NER task. For instance of the keyword *cowboys*, it may be the plural form of the noun 'cowboy' or may indicate the American football team "Dallas Cowboys". While its similar words from Skip-gram reflect the ambiguity, all the top results of Skip$_{\text{NER}}$ are related to the latter meaning of the keyword. We observe a similar difference in the results for the keyword *batman*. These observations suggest Skip$_{\text{NER}}$ is able to capture more discriminative information for NER compared to generic word embeddings.

4 Conclusion

In this paper, we have proposed a novel method to train task-specific word embeddings for NER by incorporating NER related features like neighboring words, POS tags, taxonomic relations and self-trained features into the training of word embeddings. Through several experiments, we have shown that, on the manual transcripts of open-domain conversations, how our proposed feature-enriched word embeddings can outperform baseline NER method and systems using task independent word embeddings.

In future, we will adapt our method to transcripts of conversations automatically generated by speech recognition system, and to the task of hierarchical classification of entities.

5 Acknowledgements

This work was conducted within the Rolls-Royce@NTU Corp Lab with support from the National Research Foundation Singapore under the Corp Lab@University Scheme.

6 References

[1] David Nadeau and Satoshi Sekine, "A survey of named entity recognition and classification," *Linguisticae Investigationes*, pp. 3–26, 2007.

[2] Chenliang Li, Aixin Sun, Jianshu Weng, and Qi He, "Tweet segmentation and its application to named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 558–570, 2015.

[3] Mihai Surdeanu, Jordi Turmo, and Eli Comelles, "Named entity recognition from spontaneous Open-Domain speech," in *Proceedings of Interspeech*, 2005, pp. 3433–3436.

[4] Frederic Bechet and Eric Charton, "Unsupervised knowledge acquisition for extracting named entities from speech.," in *Proceedings of ICASSP*, 2010, pp. 5338–5341.

[5] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of ICML*, 2008, pp. 160–167.

[6] Joseph Turian, Lev Ratinov, and Yoshua Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proceedings of ACL*, 2010, pp. 384–394.

[7] Mo Yu, Tiejun Zhao, Daxiang Dong, Hao Tian, and Dianhai Yu, "Compound embedding features for semi-supervised learning," in *Proceedings of HLT-NAACL*, 2013, pp. 563–568.

[8] Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu, "Revisiting embedding features for simple semi-supervised learning," in *Proc. of EMNLP*, 2014, pp. 110–120.

[9] Jeffrey Pennington, Richard Socher, and Christopher Manning, "Glove: Global vectors for word representation," in *Proceedings of EMNLP*, Doha, Qatar, 2014, pp. 1532–1543.

[10] Mohit Bansal, Kevin Gimpel, and Karen Livescu, "Tailoring continuous word representations for dependency parsing," in *Proceedings of ACL*, 2014, pp. 809–815.

[11] Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka, "Task-oriented learning of word embeddings for semantic relation classification," *CoRR*, 2015.

[12] Masataka Ono, Makoto Miwa, and Yutaka Sasaki, "Word embedding-based antonym detection using thesauri and distributional information," in *Proceedings of HLT-NAACL*, 2015, pp. 984–989.

[13] Tasos Anastasakos and Anoop Deoras, "Task specific continuous word representations for mono and multilingual spoken language understanding," in *Proceedings of ICASSP*, 2014, pp. 3246–3250.

[14] Alexandre Passos, Vineet Kumar, and Andrew McCallum, "Lexicon infused phrase embeddings for named entity resolution," in *Proceedings of CoNLL*, 2014, pp. 78–86.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS'13*, 2013, pp. 3111–3119.

[16] Yanjun Qi, Pavel P. Kuksa, Ronan Collobert, Kunihiko Sadamasa, Koray Kavukcuoglu, and Jason Weston, "Semi-supervised sequence labeling with self-learned features.," in *Proceedings of ICDM*, 2009, pp. 428–437.

[17] Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini, "Introducing and evaluating ukwac, a very large web-derived corpus of english," in *In Proceedings of WAC-4*, 2008.

[18] John Godfrey and Holliman Edward, "Switchboard-1 release 2 LDC97S62," 1993.

[19] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of ACL*, 2012, pp. 873–882.