# TASK SPECIFIC CONTINUOUS WORD REPRESENTATIONS FOR MONO AND MULTI-LINGUAL SPOKEN LANGUAGE UNDERSTANDING

*Tasos Anastasakos†, Young-Bum Kim‡, Anoop Deoras†*

†Microsoft Corporation, Sunnyvale, CA
‡Computer Science Department, University of Wisconsin Madison, Madison, Wisconsin
*Tasos.Anastasakos@microsoft.com, ybkim@cs.wisc.edu, Anoop.Deoras@microsoft.com*

**ABSTRACT**

Models for statistical spoken language understanding (SLU) systems are conventionally trained using supervised discriminative training methods. In many cases, however, labeled data necessary for these supervised techniques is not readily available necessitating a laborious data collection and annotation effort. This often results into data sets that are not expansive enough to cover adequately all patterns of natural language phrases that occur in the target applications. Word embedding features alleviate data and feature sparsity issues by learning mathematical representation of words and word associations in the continuous space. In this work, we present techniques to obtain task and domain specific word embeddings and show their usefulness over those obtained from generic unsupervised data. We also show how we transfer these embeddings from one language to another enabling training of a multilingual spoken language understanding system.

***Index Terms—*** spoken language understanding; natural language processing; word embedding; named entity recognition; vector space models

## 1. INTRODUCTION

State-of-the-art spoken language understanding (SLU) systems consist of models which are trained with various machine learning techniques to detect the domain of the user input, classify the user input into a set of known intents, and extract domain relevant entities using a sequence tagger (a task, we will refer to as slot filling). The training process uses supervised data to build these models. The training data corpus consists of natural language phrases that match the domain and tasks of interest and is annotated with labels for events that are relevant for modeling such as intents and slots. In many cases this data is not readily available for applications of language understanding (LU) and before model building, a laborious data collection and annotation effort is necessary to generate the required supervised data corpus. Due to effort and time constraints it is often the case that the data is not expansive enough to cover adequately all patterns of natural language phrases that occur in the target applications.

The state-of-the-art approaches for slot filling [1, 2, among others] use discriminative statistical models, such as conditional random fields, (CRFs) [3], Deep Neural Networks [4] for modeling. Typical set of features used in these classifiers include the identity of the current and past few words along with a few future words, named entity features provided as lexicons, part of speech tags etc. In all of these cases, the features used to learn the association of the word and the corresponding slot tag are either the identity of the word and its left and right context or some function of these. To elaborate, in an entertainment domain, such as Movies, if the user issues the query *"show me funny movies with brad pitt"*, then typically the model looks up all word combinations including the named entity *"brad pitt"* and identifies it positively as a celebrity name and uses this information as an additional feature. Similarly, if the word *"funny"* is recognized as a known feature from the training data and is tagged appropriately by the model with the slot type *"genre"*, then SLU system will eventually issue an action of searching a comedy movie with Brad Pitt as an actor. These additional features along with the identity of the words themselves will then be used to learn the association of the tag/slots with the words during supervised training and also predict these tag/slots during testing / evaluation.

The models learn associations among observations and features in the training data. In the previous example the models learn that *"brad pitt"* and *"matt damon"* are used in the same context *"show me movies with . . ."*. Learning the associations is facilitated further with the use of lexicons that contain groups of related entities. Lexicons offer a limited association capability through the presence or absence of entities in them. There is no additional detail about the degree or strength of this association. This limitation is more severe for rare or unseen entities. For rare entities the model may not build any associations and it relies on lexicons. For unseen entities that may also not be included in the lexicon there is very little evidence, taken primarily from the surrounding context about the relationship to known entities.

Previous work in SLU has employed features extracted from unsupervised data in order to expand existing lexical tokens and ngrams, for example using word sense disambiguation through WordNet [5] for named entity recognition. In [6] several methods are surveyed to augment labeled data with web search query logs in order to capture semantic components of utterances. Similarly, there is extended work in topic modeling using LSA and LDA approaches to extract underlying topics for words and phrases in an unsupervised way and use these topic clusters as features in NLP tasks [7, 8]. In this work, we describe an approach to obtain domain and task specific continuous valued vector of word embeddings i.e. a mathematical representation of words, virtue of which, association between words is formed in a latent space. Continuous representation of words, when used as additional features, help to overcome the limitation of the supervised models in dealing with rare and unseen words. We build on the work of Collobert et.al. [9] and Huang et.al. [10] and describe a scheme to derive task specific word representations (embeddings), which we then use in our slot sequence classification task. We show that task specific word embeddings outperform task independent embeddings when used as additional features in a slot sequence tagging task. The motivation behind obtaining task specific embeddings is that words form different associations depending upon the context they are in (here, by

context we mean the underlying domain). Word embeddings trained on some generic data may not capture domain specific word associations. In this paper, we address this problem and present a strategy to come up with domain and task specific word embeddings.

A related problem is the development of SLU systems in multiple languages, for example building a movie navigator system in French or German. Existing solutions follow the standard methodology of collecting sufficiently large amounts of relevant data in the target language, a practice that is very costly, time consuming, and not scalable. In this work, we present a novel embedding transfer technique to obtain word embeddings in target languages. Our proposed technique does not require us to obtain domain specific data in target languages, which may not be even possible to obtain.

The rest of the paper is organized as follows: we describe our proposed technique in Section 2. Experimental setup and results are discussed in Section 3. We finally summarize and conclude in Section 4.

## 2. MODEL DESCRIPTION

In this section we describe the neural language model used to derive the word representations. We employ the model described by Collobert et al [9] and then extended by Huang et al [10] with the use of global context. In this model each word $i$ is represented by a real vector in a $m$-dimensional space using a lookup table $LT_w(i) = W_i$, where $W_i$ is the $i$-th column of matrix $W \in \mathbb{R}^{m \times |V|}$, $|V|$ is the word vocabulary size and $m$ is the dimension of the word embeddings space. Each input text sequence $\{t_1, \ldots, t_j\}$ is transformed into a series of vectors $\{w_{t_1}, \ldots, w_{t_j}\}$. Using a fixed-length window of text, $l$ words, the model learns how related is the last word to its preceding local context, while also learning the word representations matrix $W$. We substitute the word at the end of the text window with a randomly selected word, and denote this as a corrupted sequence. The hypothesis is that the relevance of the last word to its context would be higher in the original sequence than in the corrupted sequence. A two-class classification task can be constructed using the original sequence as the positive examples and the corrupted sequence as the negative examples. The learning task is to discriminate between these two sequences by ranking the original sequence higher than the corrupted sequence.

Concretely, given a word sequence $\tau = \{t_1, \ldots, t_l\}$ that appears in the document $d$, we compute scores $g(\tau, d)$ and $g(\tau^w, d)$, where $\tau^w$ is the corrupted sequence produced by substituting $t_l$ with a randomly selected word and $g(\cdot, \cdot)$ is the neural network scoring function. The training algorithm minimizes the ranking loss:

$$c(\tau, d) = \sum_{w \in V} \max(0, 1 - g(\tau, d) + g(\tau^w, d)) \qquad (1)$$

for each $(\tau, d)$, that corresponds to ranking by the score $g(\cdot, \cdot)$ the correct word sequence higher than the corrupted sequence by a margin of 1. During training the cost is sampled with respect to $w$. Previous work on neural probabilistic language models [11, 12] aims at estimating the probability of a word given its local context (window of preceding words). This learning task learns a representation of words given its context in order to predict the relevance of the last word given its context. The resulting optimization problem is simpler to train because it does not compute probabilities and thus it is not burdened by normalization computations. In addition, it can be modified to incorporate additional context information. Huang et al [10] introduces a component that captures global context by parameterizing the document that the word sequence appears. The document is treated as a bag-of-words model and represented by the weighted average of the embeddings $W_i$ of the words that it contains. The weighting function is chosen to capture the importance of the words in the document. In [10] the inverse document frequency (IDF) is used as the weighting function and we follow this choice which is common in many information retrieval tasks.

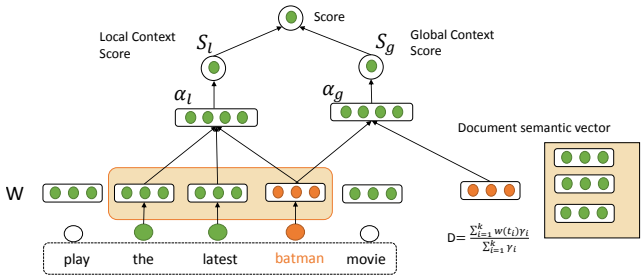


**Fig. 1**. An overview of the proposed neural network architecture (Adapted from [10])

Figure 1 depicts the proposed neural network architecture to compute the word embeddings. It uses a 3-word window and computes the local context score $s_l$ of the word "batman" in the context of the preceding two words. The local score captures syntactic information of the short length word sequence. It also computes the global context score $s_g$ by combining the embedding of the word "batman" with the average semantic vector of the document that the text window was sampled from, thus capturing broader semantics of the word. The neural network output is the sum of these two scores.

The two context scores are computed with two layer neural networks:

$$\alpha_l = f(W_1^{(l)}[x_l] + b_1^l) \qquad \alpha_g = f(W_1^{(g)}[x_g] + b_1^g) \qquad (2)$$

$$s_l = W_2^l \alpha_l + b_2^l \qquad s_g = W_2^g \alpha_g + b_2^g \qquad (3)$$

where $x_l$ is the concatenation of the word embeddings in the sequence $\tau$, $x_g$ is the concatenation of the last word embedding with the document semantic vector and $f$ is an element-wise activation function such as $\tanh$. The hidden layer of the local score consists of $h^{(l)}$ hidden nodes and $W_1^{(l)}$ and $W_2^{(l)}$ are respectively the hidden layers of the local context neural network. The training process estimates the parameters of the network: $\{W_1^{(l)}, b_1^{(l)}, W_2^{(l)}, b_2^{(l)}, W_1^{(g)}, b_1^{(g)}, W_2^{(g)}, b_2^{(g)}\}$ and the embedding matrix $W$.

The use of the document semantic vector captures long range semantics. By changing the definition of the document, we have control over the granularity of semantics that we model in the system. In SLU applications each user turn is relatively short, and is modeled primarily by the local context window. The document in this case can expand over a user session, where a user interacts with the system over multiple turns. In the case of unsupervised learning, where data is not annotated with session information or other labels, we employ self-training using existing domain classifiers to produce noisy labels for the data and create pseudo documents by grouping utterances with the same label. Words and phrases in these pseudo documents will exhibit similar syntactic and semantic properties.

### 2.1. Application in CRF

Conditional Random Fields (CRF) [3] models the posterior probability of a slot sequence given a word sequence as shown below:

$$P(\mathbf{C}|\mathbf{W}) = \frac{1}{Z(\mathbf{W})} \prod_{t=1}^{T} \psi_t\Big(c_t, \phi(c_1^{t-1}), \gamma_t(\mathbf{W})\Big) \qquad (4)$$

where, $\phi(c_1^{t-1})$ is an equivalence classification of the slot subsequence up to the $(t-1)^{th}$ position. In the first order linear chain CRFs, this function returns $c_{t-1}$. $\gamma_t(\mathbf{W})$ is a position specific equivalence classification of the entire word sequence. In practice, this function returns local features around the current word i.e. some representation of $w_{t-n+1}^{t+n-1}$ i.e., $n$-grams around the $t^{th}$ word. $\psi(\cdot)$ is the local potential function. This is nothing but an exponentiated weighted sum of active features. $Z(\mathbf{W})$ is the partition function that ensures the probability of all state sequences sum to 1. It is simply given as: $Z(\mathbf{W}) = \sum_{\mathbf{C}} \prod_{t=1}^{T} \psi_t\Big(c_t, \phi(c_1^{t-1}), \gamma_t(\mathbf{W})\Big)$.

In our baseline model, $\gamma_t(\mathbf{W})$ returns $n$-gram lexical tokens around the current word while the setup which uses word embeddings, this function returns not only the $n$-gram lexical tokens but also their continuous valued representations. Potential function encodes association of a feature (either the lexical token, $w_t$, or its continuous valued representation) with the current tag $c_t$ as well as the pair of tags $(c_{t-1}, c_t)$, thus modeling emission and transition probabilities.

In our work, we use state-of-the-art off the shelf CRF implementation: CRFsuite by Naoaki Okazaki [13]. This package supports numerical features thus allowing us to experiment with word embeddings. We modified its feature generation module so that each dimension of the embedding is treated as a separate feature.

### 2.2. Use in Multilingual setting

In this section we introduce a novel technique called *cross lingual word embedding projection* for learning domain dependent word embeddings for resource poor languages. *Resource poor* is an overloaded term, we refer to a language as resource poor only if finding huge amounts of domain dependent unsupervised data may become challenging. It may not be always possible to garner unsupervised domain specific data in a target language such as say Finnish or some Indian languages, say Hindi. So applying the above described domain dependent word embedding techniques to these low resource language could pose some serious limitations. Some approaches use automatic translation to convert existing English corpora to the target language where some quality is compromised by the translation process. The data sparsity as described earlier has significantly higher effect in the quality of the models in multilingual deployments. Developing the capability to compute word and phrase associations from unsupervised data and transfer of such embeddings from a resource rich to a resource poor language application is very important for multilingual system development. The continuous representations in multilingual SLU modeling address correlations between entities in a much richer way than lexicons do, mitigate the problem of unseen and rare entities as described in the single language modeling case and in addition provide rapid model building capabilities from the bilingual association from resource-rich to resource-poor languages. In this work, we obtain parallel corpus between English and low resource languages, a task relatively easy compared to finding domain specific data in target language. We align the parallel corpus and learn word pair mappings. We propose to transfer the domain specific word embeddings, learned for a resource rich language such as English, to a resource poor language by way of learned word pair mappings.

We applied these ideas to an SLU task in French. We obtain European parliament parallel corpus (Europarl) [14] of English and French and train a word alignment system. We use GIZA++ for the alignment purposes [15]. Once trained, for each word in the target language, $w^t$, we find that word from the source language (English) vocabulary, $w^s$, such that it has the maximum posterior probability of alignment with the target word. We then simply transfer the embedding of the source language to this target language. More formally, $\text{Embedding}(w^t) = \text{Embedding}(\text{argmax}_{w^s}\, P(w^s|w^t))$.

## 3. EXPERIMENTAL EVALUATION

We have focused on slot sequence tagging task for spoken language understanding on Microsoft's SLU system. Our SLU task pertains to three entertainment domains: movies, music and games. In this setup, a user issues a natural language query to retrieve movies, music titles, games and/or information there of. For instance, a user could say "`show me movies with brad pitt`" or "`find beyonce's music`". Our slot sequence tagger is trained with Conditional Random Fields (CRF) using lexical features with and without word embeddings. We used a window of 5 words around the current word to extract local features. The semantic space consists of 30 slots for movies, 23 slots for music and 24 slots for games. This semantic space consists of both named slots (game name, music title, movie name etc.) as well as unnamed slots (genre, description etc.).

Table 1 shows the properties of the data sets used in our experiments. In this study we used only written sentences (rather than spoken). On average, there are about 3.5 words and 1 slot per utterance. Table 2 shows the vocabulary sizes and data set coverage. The vocabulary size ranges from $4.3K$ words for games domain to $6.7K$ words for music to $8K$ words for movies.

| Domains | Training Utterances | Test Utterances |
|---|---|---|
| games | 26451 | 7196 |
| movies | 43784 | 12179 |
| music | 31853 | 8615 |

**Table 1**. Labeled data set size for games, movies and music domains partitioned into training and test subsets

| Domains | Training Vocabulary | Test Vocabulary | Test Coverage |
|---|---|---|---|
| games | 4359 | 2321 | 76% |
| movies | 8081 | 4589 | 82% |
| music | 6744 | 3539 | 78% |

**Table 2**. Labeled data set vocabulary size. The test coverage shows the test vocabulary fraction that is covered in the training data

In order to train domain dependent word embeddings, we obtained unlabeled (unlabeled in terms of domain information and the underlying intent and slot tags) voice search queries from the Bing query logs. We first tagged these queries with some baseline models predicting one of the entertainment domains (movies, music or

| Domains | Vocabulary | Sentences |
|---|---|---|
| games | 56434 | 221041 |
| movies | 29606 | 362343 |
| music | 56782 | 92283 |

**Table 3**. Unlabeled data set size for games, movies and music domains

| | Wikipedia | | Unlabeled | |
|---|---|---|---|---|
| Domains | Train | Test | Train | Test |
| games | 78.9% | 82.9% | 78.3% | 83.2% |
| movies | 80.9% | 86.4% | 65.1% | 74.2% |
| music | 74.5% | 81.6% | 74.2% | 82.6% |

**Table 4**. Vocabulary coverage of domain data with Wikipedia and unlabeled data. This table shows the fraction of vocabulary in the labeled training and test sets that is covered by the Wikipedia data set that consists of 130K words and the unlabeled data set

games), thus giving us domain labels, albeit noisy, for the huge unsupervised corpus. Table 3 shows the properties of the unlabeled data tagged with domain information using baseline domain classifiers. Table 4 shows the coverage of the training and test data sets with respect to the vocabulary extracted from the above described unlabeled corpus. As mentioned earlier, the baseline word embeddings of Collobert et.al [9] were trained on a large portion of Wikipedia data. Size of this Wikipedia vocabulary is $130K$. Table 4 shows Wikipedia vocabulary coverage for the training and test sets. While the vocabulary size of the "domain dependent" unlabeled data ($29K$ for movies and around $56K$ for games and music) is much smaller in comparison to that of Wikipedia, the vocabulary coverage, however, for the training and test sets is similar (except for the movies domain).

In the following experiments tabulated in Table 5 we investigate the effect of continuous word embeddings in the slot tagging performance task. Performance metrics are reported in terms of F-measure, a standard measure in sequence tagging tasks that combines precision and recall. Embeddings are used as additional features over a baseline CRF model that employs lexical ngram features and entity gazetteers. We first verify that the use of embeddings offers improved performance over the baseline system as reported previously for various NLP tasks [16]. The C&W embeddings are trained on the Wikipedia corpus according to the description in [9] and provided by the authors of the cited work. We then used the unlabeled data sets to create domain-specific embeddings using the neural network architecture described in section 2. We trained two embeddings matrices, one using only local context score and one using the complete architecture that combines local and global context in the estimation of the word embeddings. In both cases the performance is inferior to the generic Wikipedia data embeddings. Besides the data corpus differences, we should note that there are differences between the implementations of [9] and the implementation in [10]. The use of the global context in the experiment of row 4, does not offer any performance improvements as the definition of the document is the whole corpus. In the following experiment we created pseudo documents using the classification labels as described in section 2. These labels represent subtasks within each domain such as searching for a movie, or looking up information about an actor or a film. These new embeddings outperform the previous models of the generic corpus embeddings and the ones using a domain-based document definition. In the last of these experiments we employ phrase embeddings as features by using the product of $W_1^{(l)}[x_l]$ as defined in equation 2 in order to associate words with longer span features from their local context. These features provide vector space representations of the standard ngram features used in the baseline CRF.

| | games | movies | music |
|---|---|---|---|
| Baseline CRF<br>ngram + entity gazetteers | 85.77% | 87.11% | 84.14% |
| C&W embeddings<br>Wikipedia | 86.15% | 87.41% | 84.79% |
| local context<br>domain specific embedding | 85.45% | 86.64% | 83.53% |
| local and global context<br>domain specific embedding (1) | 85.85% | 86.56% | 83.5% |
| local and global context<br>domain specific embedding<br>intent-based global context | 88.53% | 88.89% | 88.49% |
| use of phrase embeddings | 88.8% | 88.99% | 89.24% |

**Table 5**. Experimental results using word and phrase embeddings computed from generic corpus (Wikipedia) and a task specific corpus

Finally, we present preliminary results of the application of embeddings in a multilingual setting. The experiments evaluate the use of embeddings computed in the target resource-poor language versus using cross-lingual embeddings that have been computed in a resource rich language and then using a machine translation method to transfer these embeddings to the target language.

| | games | movies | music |
|---|---|---|---|
| Baseline model | 82.88% | 75.26% | 81.18% |
| +Target embeddings | 81.94% | 74.56% | 81.43% |
| multilingual embeddings | 83.02% | 76.85% | 82.26% |

**Table 6**. FR multilingual embeddings

## 4. SUMMARY

In this paper, we demonstrated techniques to obtain task and domain specific word embeddings. We showed their usefulness over the embeddings obtained from task independent data. We showed that not only does in domain data help in obtaining meaningful word embeddings, but the models benefit even more if we further cluster the unsupervised data by their underlying intents. We also presented a novel technique to transfer word embeddings from one language to another enabling training of both mono and multilingual SLU systems using task and domain specific word embeddings. As part of future work, we plan to extend this work to obtain phrasal embeddings and use these embeddings in not only sequence classification but also other tasks such as intent detection and domain classification.

## 5. REFERENCES


[1] Y.-Y. Wang and A. Acero, "Discriminative models for spoken language understanding," in *Proceedings of the ICSLP*, Pittsburgh, PA, 2006.

[2] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proceedings of the Interspeech*, Antwerp, Belgium, 2007.

[3] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the ICML*, Williamstown, MA, 2001.

[4] Anoop Deoras and Ruhi Sarikaya, "Deep Belief Network based Semantic Taggers for Spoken Language Understanding," in *Proc. of the INTERSPEECH*, 2013.

[5] Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev, "A wordnet-based approach to named entity recognition," in *COLING-02 on SEMANET*, Morristown, NJ, USA.

[6] Dilek Hakkani-Tür, Gokhan Tür, and Asli Celikyilmaz, "Mining search query logs for spoken language understanding," in *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*, Stroudsburg, PA, USA, 2012, SDCTD '12, pp. 37–40, Association for Computational Linguistics.

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[8] Asli Celikyilmaz, Dilek Hakkani-Tur, Gokhan Tur, Ashley Fidler, and Dustin Hillard, "Exploiting distance based similarity in topic models for user intent detection," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, December 2011, pp. 425–430.

[9] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, New York, NY, USA, 2008, ICML '08, pp. 160–167, ACM.

[10] 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 the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, Stroudsburg, PA, USA, 2012, ACL '12, pp. 873–882, Association for Computational Linguistics.

[11] Holger Schwenk and Jean-Luc Gauvain, "Connectionist language modeling for large vocabulary continuous speech recognition," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, 2002, vol. 1, pp. I–765–I–768.

[12] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[13] Naoaki Okazaki, "Crfsuite: a fast implementation of conditional random fields (crfs)," 2007.

[14] Philipp Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *Conference Proceedings: the tenth Machine Translation Summit*. AAMT, pp. 79–86, AAMT.

[15] F. J. Och and H. Ney, "Improved statistical alignment models," Hongkong, China, October 2000, pp. 440–447.

[16] Joseph Turian, Lev Ratinov, and Yoshua Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA, 2010, ACL '10, pp. 384–394, Association for Computational Linguistics.