(ALMOST) ZERO-SHOT CROSS-LINGUAL SPOKEN LANGUAGE UNDERSTANDING

Shyam Upadhyay[*1] *Manaal Faruqui*[2] *Gokhan Tür*[2] *Dilek Hakkani-Tür*[2] *Larry Heck*[†3]

[1] University of Pennsylvania, Philadelphia, PA
[2] Google Research, Mountain View, CA
[3] Samsung Research, Mountain View, CA

shyamupa@seas.upenn.edu, {mfaruqui, gokhant, dilekh}@google.com, larry.heck@ieee.org

ABSTRACT

Spoken language understanding (SLU) is a component of goal-oriented dialogue systems that aims to interpret user's natural language queries in system's semantic representation format. While current state-of-the-art SLU approaches achieve high performance for English domains, the same is not true for other languages. Approaches in the literature for extending SLU models and grammars to new languages rely primarily on machine translation. This poses a challenge in scaling to new languages, as machine translation systems may not be reliable for several (especially low resource) languages. In this work, we examine different approaches to train a SLU component with little supervision for two new languages – Hindi and Turkish, and show that with only a few hundred labeled examples we can surpass the approaches proposed in the literature. Our experiments show that training a model bilingually (i.e., jointly with English), enables faster learning, in that the model requires fewer labeled instances in the target language to generalize. Qualitative analysis shows that rare slot types benefit the most from the bilingual training.

Index Terms— Spoken Language Understanding, Cross-Lingual, Slot-Filling, Intent Classification

1. INTRODUCTION

Goal-oriented dialogue systems rely on a Spoken Language Understanding (SLU) component to extract meaning from natural language used in conversation [1]. SLU models the semantics of a particular domain by parsing user utterances into semantic frames, which consists of intent and slots. Formally, given the input dialogue utterance $\vec{x}$ with n tokens $\vec{x} = (x_1, x_2, \cdots x_n)$, the slot filling task involves generating a sequence of n tags $\vec{y} = (y_1, y_2, \cdots y_n)$ which identify the kind and span of different slots, and the intent classification task assigns a intent label L to the utterance. For example, Fig. 1a shows an utterance and its slots in Begin-Inside-Outside (BIO) encoding with its intent label. To develop spoken dialogue systems in new languages, extending SLU systems to new languages is crucial. The cross-lingual SLU task poses the following problem – given an utterance in another language, the SLU model should generate predictions for slot-filling and intent classification. An example of a Hindi utterance with its slots and intent label is shown in Fig. 1b.

Developing a SLU system for a new language can be quite challenging. While datasets with labeled utterances for training an English model are plentiful, this is not the case for most other languages. Getting high quality human translations is costly for a new language, requiring native speakers for generating and verifying translations and then identifying the slots. Even collecting a few thousand examples per language becomes prohibitive, if one wants to scale to the most popular languages in the world. Ideally, we would like to minimize the amount of annotation effort required to achieve a reasonable performance. Existing approaches [2, 3, 4] for the cross-lingual SLU task use machine translation to either generate supervision in the target language automatically, or convert the test data to English. However, these approaches will fail on languages for which machine translation is not reliable, or even available.

Utt: find a one way flight from boston to atlanta on wednesday
Slots: O O B-RT I-RT O O B-FC O B-TC O B-DDN

(a) English Utterance

Utt: बुधवार को बोसटन से अटलांटा तक जाने वाली एकतरफ़ा उड़ाने खोजें
Slots: B-DDN O B-FC O B-TC O O O B-RT O O

(b) Hindi Utterance

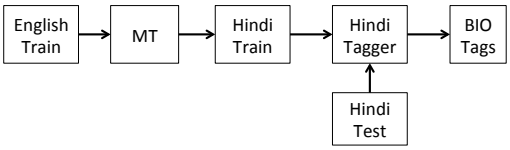
Fig. 1: English and corresponding Hindi utterance and their slots in BIO encoding. The correct intent label is "flight". Tags: RT - *round trip*, FC - *from city*, TC - *to city*, DDN - *departure day name*.

We develop a simple joint training approach which trains a SLU model for English and the target language jointly, without relying on machine translation. By using aligned word embeddings, our approach can perform zero-shot slot filling in the target language (with no training data). Further adding only a few 100 labeled examples from the target language improves performance dramatically, as our model benefits from the shared parameter space to achieve better performance. We experimentally show that to achieve the same performance, our model requires 2.5-3 times less training data in the target language compared to a naive approach which uses only the examples in the target language as supervision. We evaluate our approach on 2 relatively low resource languages, Hindi and Turkish, and show that our approach outperforms previous approaches in the literature with only a few 100 training examples.

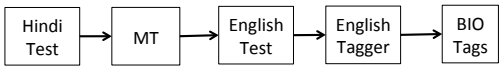
2. RELATED WORK

Existing work on extending SLU to new languages have relied primarily on Machine Translation (MT) systems, either at train or test time. In particular, two popular techniques have emerged – TEST ON SOURCE [4] and TRAIN ON TARGET [5]. In the former approach (Fig. 2a), the test data in the target language is translated using a MT system into English, for which a state-of-the-art SLU model is available. The English model is then run on the translated test data to identify frames and slots. In the latter approach (Fig. 2b), the

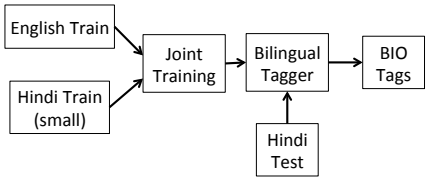
[*]Work done while the first author was an intern at Google Research.
[†]Work done while at Google Research.



(a) Test on Source



(b) Train on Target



(c) Our Approach

Fig. 2: Former approaches for Cross-lingual SLU compared to our approach of joint training across languages. Note that our approach does not rely on a Machine Translation (MT) step, which may be unreliable for relatively low resource languages.

training data available in English is translated using the MT system into the target language such that the annotations are preserved after translation. This translated data in the target language now serves as the training data for a new SLU model.

Variants of these approaches have also been proposed, notably the variant of TEST ON SOURCE presented in [4]. [4] made the English SLU model robust to translation inconsistencies by training on MT-distorted English back-translations via the target language. The SLU model was then trained on combination of the original English training data and the back-translated version. The intuition is that the back-translated version will have similar inconsistencies that the translated target language test data will exhibit, allowing the model to adaptive to translation errors. We call this approach ADAPTIVE TEST ON SOURCE in our experiments.

A major weakness of both these approaches is that they rely on machine translation. While machine translation is reliable for popular languages like Spanish, German etc., this is not the case for most languages. Indeed, previous work has focused more on high resource languages like Chinese and French, for which high quality machine translation is available. Machine translation also introduces test-time latency in approaches like TEST ON SOURCE. Directly tagging English utterances takes less than 10 ms per query (for our model), while translating from another language to English alone can introduce an order of magnitude larger latency ($\approx$ 100 ms), possibly resulting in reduced conversational experience quality in real use cases. Domain differences also has adverse effects on these approaches, as machine translation models are trained on written parallel text, instead of parallel dialog utterances.

3. OUR APPROACH

We first describe a naive model for joint slot filling and intent classification in the target language, which we build on later. This model is inspired by joint slot filling and intent classification approaches from [6, 7] and the success of RNNs on the SLU task [8].

In the following, we denote a training example in English as $(\vec{x^e}, \vec{y^e}, L^e)$ and an example in target language using $(\vec{x^f}, \vec{y^f}, L^f)$. Let $\{(\vec{x^e}, \vec{y^e}, L^e)\}^M$ denote English training data with M examples, and $\{(\vec{x^f}, \vec{y^f}, L^f)\}^N$ denote target language training data with N examples, where $M > N$. We use $\mathbf{\Phi}(x_i)$ to denote embedding of a token x_i and $\mathbf{\Phi}(\vec{x})$ as shorthand for $(\mathbf{\Phi}(x_1), \mathbf{\Phi}(x_2), \cdots)$.

Naive Model. The naive model uses $\{(\vec{x^f}, \vec{y^f}, L^f)\}^N$ to train a bidirectional RNN to predict both the intent and BIO tags (shown in Fig. 3). The hidden state at each timestep is used to predict the corresponding BIO tag, and the last hidden state is used to predict the intent label. Formally,

$$\vec{h} = BiRNN(\mathbf{\Phi}(\vec{x}))$$
$$\vec{y_i} = Softmax(W_y \vec{h_i}), \;\; L = Softmax(W_L \vec{h_n})$$

where $\vec{h}$ is the sequence of hidden states generated by the concatenation of forward and backward outputs from the bidirectional RNN ($BiRNN$), $\vec{h_n}$ is the last hidden state, W_y and W_L are model weights, and $Softmax$ is the softmax operation. The learning objective is the sum of the sequence-tagging loss and the intent classification loss, $\mathcal{L}_{naive} = \mathcal{L}_{seq}(\vec{y}, \vec{y_p}) + \mathcal{L}_{clf}(L, L_p)$, where $\vec{y_p}$ and L_p are current model predictions, averaged over all training examples. The model parameters, including the word embeddings $\mathbf{\Phi}(x_i)$, are learnt during training.

The naive approach only utilizes the little training data that might be available in the target language. However, for most SLU tasks, training data in English is available, therefore it is desirable to use it for improving generalization in a new language. We show how to achieve this by training a joint model for both languages, such that parameters are shared across languages.

Bilingual Embeddings. To encourage parameter sharing we need to ensure the features (viz. the word embeddings) in different languages lie in the same vector space. However, word embeddings trained monolingually for two different languages do not encode cross-lingual semantics appropriately. For instance, the embeddings for $\mathbf{\Phi}_e(atlanta)$ and its Hindi translation $\mathbf{\Phi}_f(\text{अटलांटा})$ need not have high cosine similarity. To achieve this, we first align the embeddings into a shared vector space.

Aligning word embeddings in different languages has been a popular research direction in natural language processing [9, 10, 11, 12, 13, inter alia]. A common approach is to learn linear transformations $\mathbf{W}$ and $\mathbf{V}$, such that vectors for semantically equivalent words are aligned (for instance, $\mathbf{W}\mathbf{\Phi}_e(atlanta)$ and $\mathbf{V}\mathbf{\Phi}_f(\text{अटलांटा})$ will have higher cosine score) and reside in a shared vector space, which we denote as $\mathbf{\Phi}_{e,f}$. We adopt this simple approach and use publicly available embeddings from [14] with the alignment matrices from [15] to project embeddings into a shared space. We also experimented with other approaches for aligning embeddings like CCA [16], but got the best results using off-the-shelf vectors.

3.1. Zero-Shot SLU

Aligned word embeddings also enable zero-shot SLU. For this, we first train an English SLU model on $\{(\vec{x^e}, \vec{y^e}, L^e)\}^M$ using $\mathbf{\Phi}_{e,f}$ to embed English tokens. To ensure the embeddings remain aligned across languages, they are not updated during training.

The model is then directly tested on the target language test utterances, using $\mathbf{\Phi}_{e,f}$ to embed the target language tokens. As $\mathbf{\Phi}_{e,f}$ ensures embeddings from different languages for semantically equivalent words are similar, the model parameters can still predict certain slots accurately. The approach is shown in Fig. 3, where the parameters enclosed in the grey box are pre-trained on English.

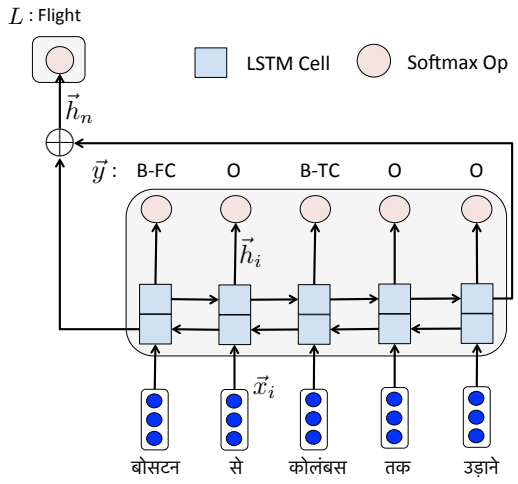


Fig. 3: **Naive Model**: Only target language examples are used during training. **Zero-Shot SLU**: The naive model is pre-trained on English, with fixed word embeddings from the shared vector space, and then tested directly on Hindi. Parameters enclosed in the grey box are pre-trained on English.

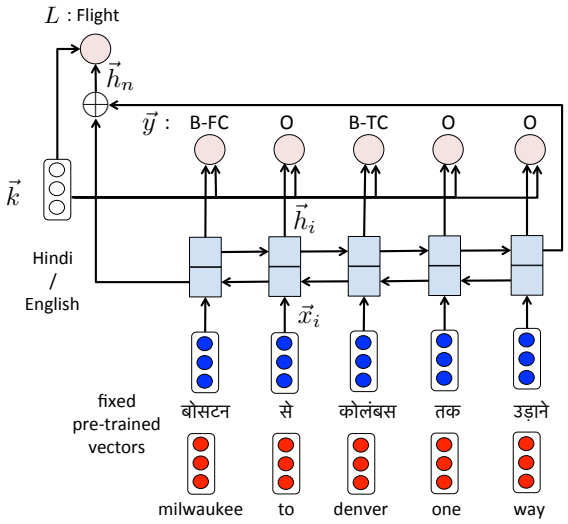


Fig. 4: The Bilingual Training Setup.

3.2. Bilingual Training

Using aligned embeddings $\mathbf{\Phi}_{e,f}$, we can modify the naive model to enable joint bilingual training (shown in Fig. 4). Formally,

$$\vec{h} = BiRNN(\mathbf{\Phi}_{e,f}(\vec{x}))$$

$$\vec{y_i} = Softmax(W_y(\vec{h}_i \oplus \vec{k})), \;\; L = Softmax(W_L(\vec{h}_n \oplus \vec{k}))$$

where $(\vec{x}, \vec{y}, L) \in \{(\vec{x^e}, \vec{y^e}, L^e)\}^M \cup \{(\vec{x^f}, \vec{y^f}, L^f)\}^N$. That is, the training examples come from either language. To aid the model learn language specific patterns, we also introduce a language indicator vector $\vec{k}$ which encodes which language the current training utterance belongs to, shown on the left in Fig. 4. Vector $\vec{k}$ is concatenated with $\vec{h_i}$ and fed into all softmax layers responsible for predicting either the intent or the slot tags. As before, we fix the word embeddings during training. The model is trained using a joint learning objective $\mathcal{L}_{joint} = \sum_{z \in \{e,f\}} \mathcal{L}_{seq}(\vec{y^z}, \vec{y_p}) + \mathcal{L}_{clf}(L^z, L_p)$ to optimize losses on both languages. Examples from either language are randomly mixed in a mini-batch.

4. EXPERIMENTS

Data Collection. Following previous work [4], we collected annotated utterances in two relatively low resource languages, Turkish and Hindi, by manually translating utterances in the English ATIS Corpus [17], a popular benchmark for SLU [18, 19, 20].

Manual translations were generated by native speakers of the target language, who were asked to ensure that the translations are faithful to the request expressed in the original English utterance. We used Amazon Mechanical Turk to generate the phrase level slot annotation on the manual translations. A total of 893 and 715 (randomly selected) utterances from the ATIS test split were translated and annotated for Hindi and Turkish evaluation respectively. We also translated and annotated 600 (randomly selected for each language separately) utterances from the ATIS train split to use as supervision. When training a joint bilingual model, or a model using the TEST ON SOURCE or TRAIN ON TARGET approach, we use the ATIS train split of 4978 utterances as supervision in English. Automatic translations for the TEST ON SOURCE and TRAIN ON TARGET approach were generated using Google Translate.

Evaluation and Training Setup. We compare the naive and the bilingual models by varying the amount of training data available in the other languages. We also include the TRAIN ON TARGET and TRAIN ON SOURCE approaches in the slot filling comparison to demonstrate their shortcomings.

We plot the evaluation metric (tagging F1 or intent accuracy) against the number of training examples. When using a fraction of the training data with the naive and the bilingual models, we subsample 5 times from the entire pool of training examples and report the average performance. We use the standard conlleval script [21] for evaluating the slot-filling, and classification accuracy for intent.

In all experiments, we used size 300 embeddings from [14, 15], normalized to unit norm. The RNN unit was a LSTM [22], with hidden state of size 100. The language indicator vector $\vec{k}$ of size 5 in the bilingual model was trained along with model parameters. All models were trained for a total of 10 epochs with a batch size of 5, using Adam [23] with a learning rate of 1e-3, and a word dropout rate of 0.5 [24]. All models were implemented using Tensorflow [25].

4.1. Experimental Results

The slot filling results are shown in Figure 5. TRAIN ON TARGET is worse on Hindi due to poor translation quality, even though it had ≈5k queries to train on. TEST ON SOURCE performs quite well on both languages, and improves substantially after adding adaptive training with back-translations, as shown by the ADAPTIVE TEST ON SOURCE curve. This is reasonable as translation quality is higher when translating from a foreign language to English (viz. TEST ON SOURCE) than the opposite direction (viz. TRAIN ON TARGET). In comparison, the zero-shot approach with a trained English SLU model performs relatively well, given that there was no training data in target language, but worse than all the existing approaches.

For the naive model, we approximately need around 600 examples in both languages to achieve a F1 of 75.0. In comparison, our bilingual model only requires ≈200 examples to achieve a F1 of 75.0. In fact, it beats the previous best approach, ADAPTIVE TEST ON SOURCE, with only 100 examples in both languages. Note that

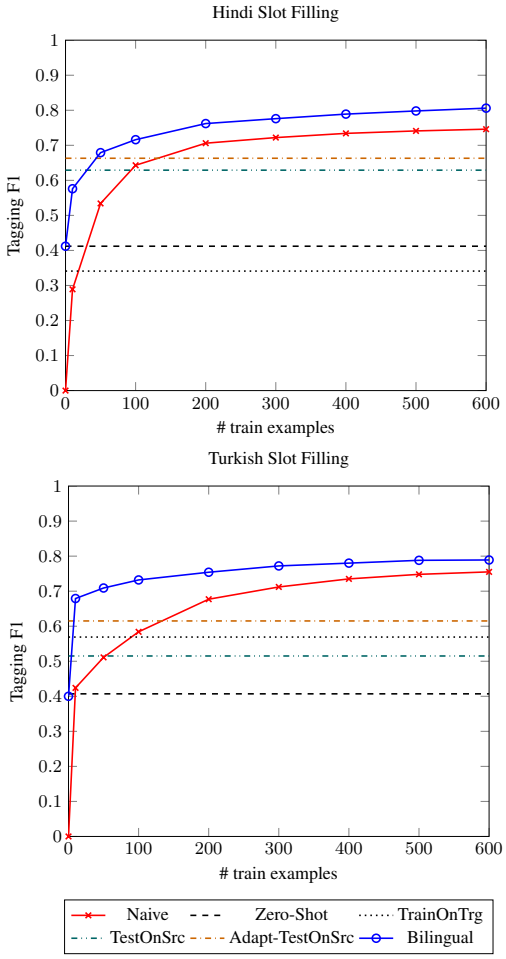


Fig. 5: Slot Filling F1 for Hindi (above) and Turkish (below), plotted against the number of training examples in the target language.

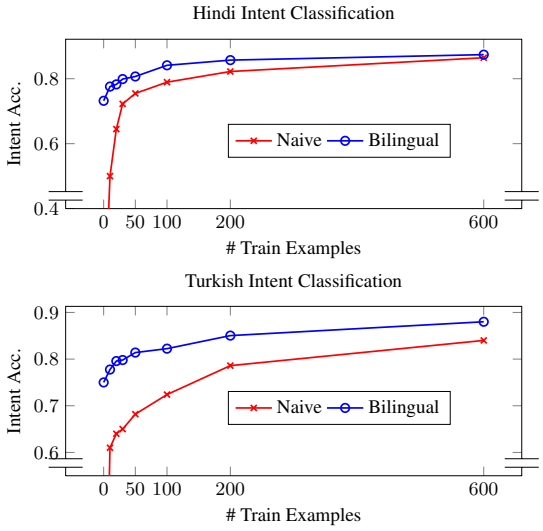


Fig. 6: Intent Accuracy for Hindi (above) and Turkish (below), plotted against the number of training examples in the target language.

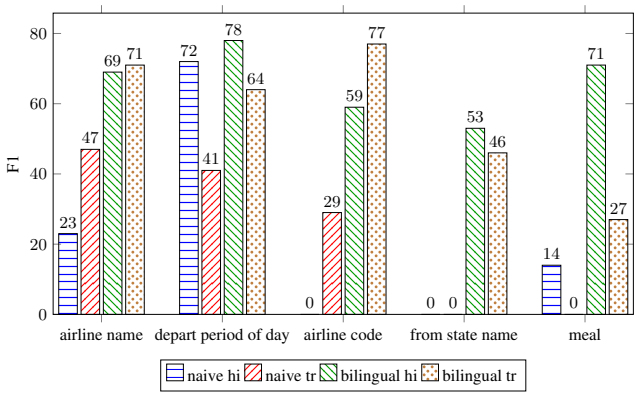


Fig. 7: F1 per slot type for Hindi and Turkish for naive and bilingual models when given only 100 examples in the target language.

our approach does not suffer from the latency introduced by machine translation that either of the TEST ON SOURCE approaches suffer. Overall, using all 600 training examples, the naive approach achieves F1 of 75.5 (Turkish) and 74.6 (Hindi), compared to 78.9 (Turkish) and 80.6 (Hindi) achieved by the bilingual approach. This suggests that not only does joint training reduce the amount of supervision required, it also improves generalization. The jointly trained model also performs competitively on English – with 93.2 and 94.9 F1 when trained jointly with Hindi and Turkish respectively.[1]

We also compare intent classification accuracy for the naive and the bilingual model in Fig. 6. A similar trend is observed, in that the naive model requires 600 (or more) examples to attain a accuracy of 80%, which the bilingual model attains with ≈50 examples.

Qualitative Analysis. We compare the per slot F1 for naive and the bilingual model for different slot types. We choose five slot types (out of 63) from ATIS based on frequency – two slots (*airline name*, *depart period of day*) are frequent in the dataset ($>$ 100 mentions) and three slots (*airline code*, *from state name*, *meal*) are rare ($<$50 mentions). We compare the F1 per slot type of the naive and bilingual model when given only 100 examples in the target language.

The profiles are shown in Fig. 7. Notice that for rare slots (like *meal*, *airline code*), there is a huge difference (over 40 F1 pts) between the bilingual model and the naive model. For more frequent slots like *depart period of day* and *airline name*, the bilingual model still performs better than the naive model, but the improvement is relatively less (over 20 F1 pts). This suggests that the bilingual training helps in learning patterns indicative of rare slot types with much less data compared to the naive model.

5. CONCLUSION

We proposed a simple bilingual training approach to train a SLU model in a new language jointly with English, without relying on machine translation. Our approach outperforms existing state-of-the-art approaches on new SLU benchmarks[2] in Hindi and Turkish, while maintaining competitive performance on English.

There are several avenues of future research. More parameter sharing can be achieved across languages by using character level embeddings in conjunction with word embeddings. A fully multilingual approach which trains the same model to handle three or more languages is also a natural extension of our work.

[1] training only on English achieves 95.2.

[2] Available at `github.com/google-research-datasets/dialogue/tree/master/multilingual-atis`

6. REFERENCES

[1] Gokhan Tür and Renato De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*, John Wiley & Sons, 2011.

[2] Evgeny A Stepanov, Ilya Kashkarev, Ali Orkan Bayer, Giuseppe Riccardi, and Arindam Ghosh, "Language style and domain adaptation for cross-language SLU porting," in *ASRU*, 2013.

[3] Fabrice Lefevre, François Mairesse, and Steve Young, "Cross-lingual spoken language understanding from unaligned data using discriminative classification models and machine translation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[4] Xiaodong He, Li Deng, Dilek Hakkani-Tür, and Gokhan Tür, "Multi-style adaptive training for robust cross-lingual spoken language understanding," in *ICASSP*. IEEE, 2013.

[5] Fernando García, Lluís F Hurtado, Encarna Segarra, Emilio Sanchis, and Giuseppe Riccardi, "Combining multiple translation systems for spoken language understanding portability," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 194–198.

[6] Bing Liu and Ian Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Interspeech*, 2016.

[7] Dilek Hakkani-Tür, Gokhan Tür, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Interspeech*, 2016.

[8] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tür, Xiaodong He, Larry Heck, Gokhan Tür, Dong Yu, et al., "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*.

[9] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai, "Inducing crosslingual distributed representations of words," in *Proc. of COLING*, 2012.

[10] Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum, "Trans-gram, fast cross-lingual word-embeddings," in *Proc. of EMNLP*, 2015.

[11] Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen, "Inverted indexing for cross-lingual nlp," in *Proc. of ACL*, 2015.

[12] Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith, "Massively multilingual word embeddings," *arXiv preprint arXiv:1602.01925*, 2016.

[13] Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth, "Cross-lingual models of word embeddings: An empirical comparison," in *ACL*, 2016.

[14] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, 2017.

[15] Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla, "Offline bilingual word vectors, orthogonal transformations and the inverted softmax," *ICLR*, 2017.

[16] Manaal Faruqui and Chris Dyer, "Improving vector space word representations using multilingual correlation," in *Proc. of EACL*, 2014.

[17] Patti J Price, "Evaluation of spoken language systems: The ATIS domain," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania*, 1990.

[18] Yulan He and Steve Young, "A data-driven spoken language understanding system," in *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*. IEEE, 2003, pp. 583–588.

[19] Christian Raymond and Giuseppe Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[20] Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck, "What is left to be understood in ATIS?," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2010.

[21] Erik F Tjong Kim Sang and Sabine Buchholz, "Introduction to the CoNLL-2000 shared task: Chunking," in *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*. Association for Computational Linguistics, 2000, pp. 127–132.

[22] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.

[24] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.