Bitext Name Tagging for Cross-lingual Entity Annotation Projection

Dongxu Zhang[1], **Boliang Zhang**[2], **Xiaoman Pan**[2], **Xiaocheng Feng**[3],
Heng Ji[2], **Weiran Xu**[1]
[1]Beijing University of Posts and Telecommunications, Beijing, China
zhangdongxuu@gmail.com, xuweiran@bupt.edu.cn
[2]Rensselaer Polytechnic Institute, NY, USA
{zhangb8,panx2,jih}@rpi.edu
[3]Harbin Institute of Technology, Harbin, China
xcfeng@ir.hit.edu.cn

Abstract

Annotation projection is a practical method to deal with the low resource problem in incident languages (IL) processing. Previous methods on annotation projection mainly relied on word alignment results without any training process, which led to noise propagation caused by word alignment errors. In this paper, we focus on the named entity recognition (NER) task and propose a weakly-supervised framework to project entity annotations from English to IL through bitexts. Instead of directly relying on word alignment results, this framework combines advantages of rule-based methods and deep learning methods by implementing two steps: First, generates a high-confidence entity annotation set on IL side with strict searching methods; Second, uses this high-confidence set to weakly supervise the model training. The model is finally used to accomplish the projecting process. Experimental results on two low-resource ILs show that the proposed method can generate better annotations projected from English-IL parallel corpora. The performance of IL name tagger can also be improved significantly by training on the newly projected IL annotation set.

1 Introduction

Annotation projection task aims to deal with low resource issues where human annotations are limited or unavailable in incident languages or domains. Since supervised learning algorithms can not work without annotation sets, annotation projection methods could automatically generate annotations from another language or domain where rich annotation sets are available, such as English.

Yarowsky and Ngai (2001) proposed a method of using parallel text with word alignment results to project annotations. Fig. 1 shows an example of entity projection with word alignment results from English to Turkish. On English side, *Voice of America Radio* and *Congo* are tagged as an organization (ORG) and a location (LOC) respectively by an English name tagger. The dashed lines represent word alignment results generated by a word alignment tool. Following alignment results, we can project labels to *Amerikann Sesi* and *Congo* on Turkish side automatically. A major problem of this framework is that it suffers from noises produced by word alignment errors. Thus, some de-noising methods have been proposed (Kim et al., 2010; Wang and Manning, 2014).

Though promising, this framework has several disadvantages. One shortcoming is that it totally depends on word alignment results. In this case, noise propagation from word alignment errors is heavily troublesome. To alleviate this problem, there are post-processing methods for annotation correction (Kim et al., 2010) and soft expectations to make use of more probabilistic information inside word alignment results (Wang and Manning, 2014). Although post-processing correction is efficient to filter out wrong labels, it can hardly find back labels which have been lost in the word alignment step. The soft expectation method leverages probabilistic information from word alignment results instead of hard labels such as one and zero. It can revive some false negative cases where true answers still have quite high rankings (but not the highest one). But this method will fail if word alignment results are totally wrong. From

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 461–470, Osaka, Japan, December 11-17 2016.

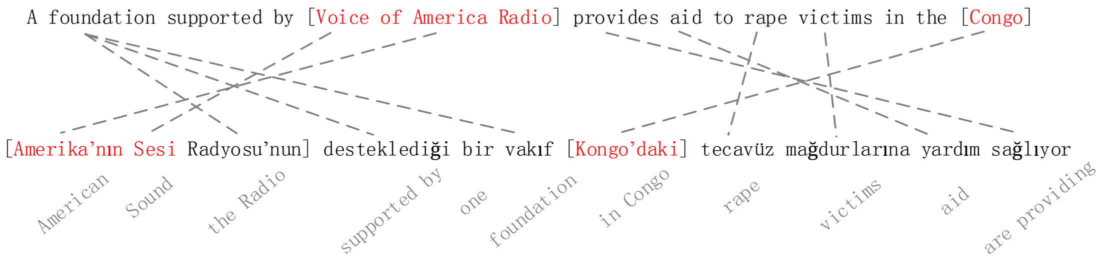
A foundation supported by [Voice of America Radio] provides aid to rape victims in the [Congo]
[Amerika'nın Sesi Radyosu'nun] desteklediği bir vakıf [Kongo'daki] tecavüz mağdurlarına yardım sağlıyor
American
Sound
the Radio
supported by
one
foundation
in Congo
rape
victims
aid
are providing

Figure 1: Errors of annotation projection with word alignment results using English and Turkish bitext.

another aspect, since word alignment task is mainly designed for the machine translation task (Och and Ney, 2003), it is not specifically tuned for other NLP tasks such as annotation projection.

Another disadvantage is that this framework depends on full-sequence word alignment where each alignment pair in the sequence will be taken into account for annotation projection. Then, the entity projection could always be disrupted by alignment errors on low-frequency word pairs and outliers, especially when the quality of bitexts cannot be guaranteed in low-resource languages. For intuition, in Fig. 1, the overall word alignment quality seems to be acceptable. But for entity projection, *Amerikann Sesi Radyosunun* is failed to be labeled completely caused by a single word alignment error on *Radyosunun*. In this circumstance, we should instead try to only focus on projecting meaningful tags, for example name tags.

In this paper, we focus on entity projection task on English-IL bitext and propose a weakly supervised framework to train a bitext name tagger and deal with issues mentioned above. The main contributions of this paper are as follows:

- We propose a new weakly-supervised framework for entity annotation projection. The framework contains two steps which can increase precision and recall step by step.
- The proposed model does not heavily depend on word alignment results. It bypasses the use of full word alignment results by taking the original English and IL data from the bitext as inputs and using training process to learn the projection. Also, the model only focuses on projecting name tags.
- We employ connections among hidden layers in recurrent neural networks to deal with sequence labeling tasks across parallel corpora.

2 Method

Given English-IL bitexts, we could start with labeling all sentences on the English side using a high-quality English name tagger [1] and find out entity names in each English sentence. For the rest of this section, our goal is to project these labeled entities from English to IL. To accomplish this goal, two separate steps are carried out. Firstly, in Sec 2.1, a **high-confidence annotation set** is generated using strict rules on parallel corpora. Secondly, in Sec 2.2, a supervised bitext name tagger is trained to recall those annotations which were lost during the first step. Besides, in Sec 2.3, we provide two different strategies to correct some errors made by the second step and give further improvements on the quality of projection.

2.1 High-confidence Annotation Projection

In order to supervise the training of the bitext name tagger, we need to firstly generate a high-quality training set out of the entire bitext. A naive way of projecting names from English to IL is to follow word alignment results [2]. Since word alignment task is not perfectly solved [3] and the performance could

[1] In our experiment, we use the Stanford NER tool (Manning et al., 2014).

[2] Here we use GIZA++ (Och and Ney, 2003)

[3] The poor alignment occurs especially when the parallel corpora is not enough (low resource issue) or the quality of bitext is poor.

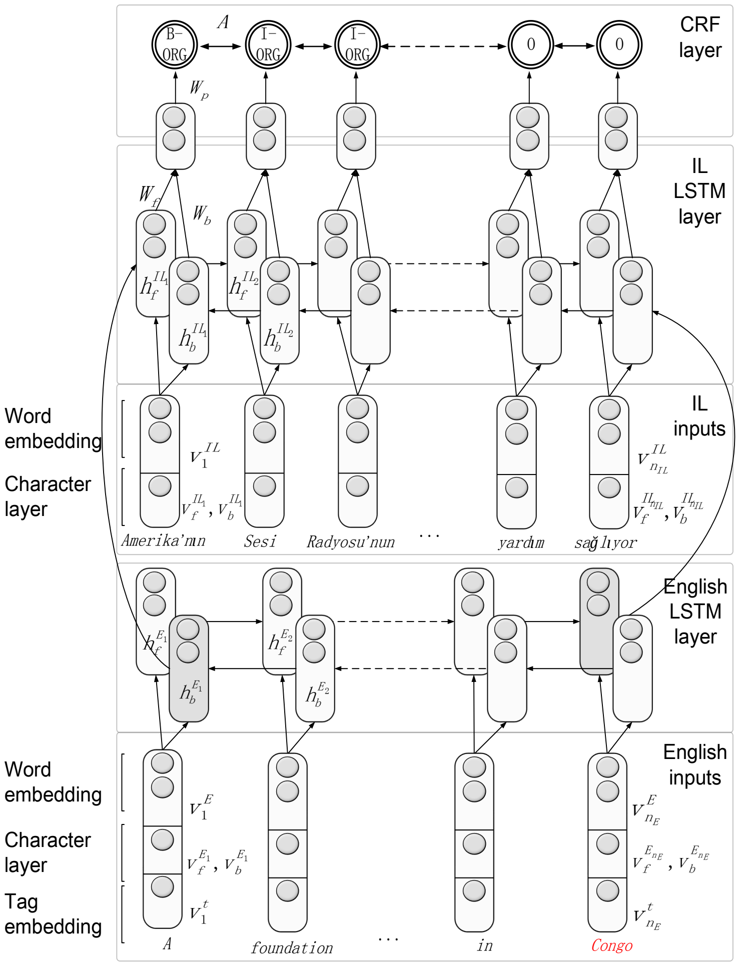

Figure 2: The framework of the bitext name tagger.

become much worse when considering the projection over low frequency phrases such as names, here we add some rules for name searching before using word alignment results.

For each labeled name in a English sentence, we search for the corresponding IL name on the IL side of the bitext as follows:

1. Firstly, if the Levenshtein distance between an IL word sequence and the English name is close enough, the word sequence will be labeled with the English name tag.

2. The second choice of measuring the similarity is to use Soundex (Raghavan and Allan, 2004).

3. If previous steps can not find the corresponding name in IL, a word-to-word translation table is used. The table is derived from GIZA++ and we only keep top 5 most credible translations for each word. And an exact word-to-word translation of entity names is carried out for string matching.

After searching steps, we use word alignment results to project names that has not been found in IL. Here we follow constraints that the number of words in a name should be the same between English and IL mentions, and words in the projected name should be contiguous. Finally, we only keep those sentence pairs where all the entity names labeled in English have been successfully projected by using previous methods.

2.2 Name Tagging on Parallel Text

In the previous section, a high-confidence annotation set is automatically generated. Using this annotation set with bitext inputs, we could train a bitext name tagger for name annotation projection. Figure

2 shows the structure of this neural-based model. Here we utilize the flexibility of recurrent neural networks to label IL sequences with the information flowing from English side. For both English and IL side, we employ bi-directional LSTM networks to handle sequence inputs of varied lengths.

Basically, our model follows the recipe of Lample et al. (2016), with several extensions designed for this task. First, there exist two sets of embedding and recurrent layers in order to handle inputs from both English and IL side. Second, to combine these two parts of signals, there are connections between two bi-directional LSTMs where the zero step of hidden layers on IL side is initialized by the last step of hidden layers on English side. Third, there are not only word and character level information, but tag sequences involved in the input signal, on both English and IL sides. Details of the bitext name tagger will be introduced in the rest of this section.

Input signal

We have three different types of input sequences for each sentence pair in the bitext:

- Word sequence $X_L = (x_1^L, x_2^L, ..., x_{n_L}^L)$, $x_i^L \in \{0, 1, ..., V_L - 1\}$, where $L \in \{E, IL\}$ represents English or IL, V_L is the size of vocabulary in L, and n_L is the number of words in the current word sequence of L.

- In order to let system know which names in English need to be projected, it is crucial to add the sequence of name types labeled by the English name tagger: $T_E = (t_1^E, t_2^E, ..., t_{n_E}^E)$, $t_i^E \in \{0, 1, ..., V_t - 1\}$, where V_t is the number of tag types. Here $V_t = 7$ since we follow the IOB format (*Inside, Outside, Beginning*) with three entity type *PERSON, LOCATION and ORGANIZATION.*

- To leverage character level information, for each word x_i^L, there is a character sequence $C^{L_i} = (c_1^{L_i}, c_2^{L_i}, ..., c_p^{L_i})$, $c_j^{L_i} \in \{0, 1, ..., V_{Lc} - 1\}$, where p is the number of characters in word X_i^L, and V_{Lc} is the number of different characters in language L.

Embeddings for each word

Then, we project these input signals from high dimensional space of token id into dense vector spaces using look-up tables. Thus, we have

$$v_i^L = W_{word}^L x_i^L,\ L \in \{E, IL\}$$

$$v_j^{L_i} = W_{char}^L c_j^{L_i},\ L \in \{E, IL\}$$

$$v_i^t = W_{tag}^E t_i^E$$

where W_{word}^L,W_{char}^L, W_{tag}^E are look-up tables for English/IL word, English/IL character, and English tag respectively.

Since character level information is helpful for name tagging task (Klein et al., 2003), we combine the word level embedding and character level embedding together by following Lample et al. (2016). For each work token, we derive a vector from the corresponding character embedding sequence using bidirectional LSTM networks as following:

$$v_f^{L_i} = LSTM_{for}^{LC}(\mathbf{v}^{L_i})[p-1],\ \ v_b^{L_i} = LSTM_{back}^{LC}(\mathbf{v}^{L_i})[0]$$

where $\mathbf{v}^{L_i} = (v_1^{L_i}, v_2^{L_i}, ..., v_p^{L_i})$, $LSTM_{for}^{LC}(\cdot)$ and $LSTM_{back}^{LC}(\cdot)$ are the forward and backward long short-term memory recurrent networks (LSTM-RNN) (Hochreiter and Schmidhuber, 1997) to encode $\mathbf{v}^{L_i}$. The output of $LSTM(\cdot)$ is a list of LSTM-RNN hidden layers.

Then, we concatenate v_i^E, $v_f^{E_i}$, $v_b^{E_i}$ and v_i^t into a vector d_i^E which represents the information of word i in each English sentence. Similarly, we concatenate v_i^{IL}, $v_f^{IL_i}$, $v_b^{IL_i}$ into a vector d_i^{IL} for each word in IL. In practice, to leverage information from rich unlabeled monolingual corpus, we initialize W_{word}^E and W_{word}^{IL} with pre-trained word embeddings using word2vec tool (Mikolov et al., 2013).

Two bi-directional LSTMs for bitext

After generating $\mathbf{d}^E = (d_1^E, d_2^E, ..., d_{n_E}^E)$ and $\mathbf{d}^{IL} = (d_1^{IL}, d_2^{IL}, ..., d_{n_{IL}}^{IL})$, we put them into English and IL bidirectional LSTM-RNNs separately:

$$\mathbf{h}_f^E = LSTM_{for}^E(\mathbf{d}^E), \;\; \mathbf{h}_b^E = LSTM_{back}^E(\mathbf{d}^E)$$

where $LSTM_{for}^E(\cdot)$ and $LSTM_{back}^E(\cdot)$ are the forward and backward LSTM-RNNs to encode $\mathbf{d}^E$. And $\mathbf{h}_f^E$ and $\mathbf{h}_b^E$ are the list of hidden layers. Similarly, for IL, we have

$$\mathbf{h}_f^{IL} = LSTM_{for}^{IL}(\mathbf{d}^{IL}, \mathbf{h}_b^E[0]), \;\; \mathbf{h}_b^{IL} = LSTM_{back}^{IL}(\mathbf{d}^{IL}, \mathbf{h}_f^E[n_E - 1])$$

To utilize information extracted from English side, we use $h_f^E[n_E - 1]$ and $h_b^E[0]$ from the last steps of bi-directional LSTM to initialize the starting states of LSTM hidden layers in IL side. Thus during training, errors can be back propagated to English side neural networks through these two vectors. The two hidden layers filled with gray color of background in Figure 2 show the intuition.

This idea is inspired by one of previous successful sequence-to-sequence deep learning method in machine translation (Sutskever et al., 2014), which encodes an input sequence into one single vector and then generates a new sequence with both the vector and the language model contained in its decoding networks. In our case, the task is much easier since we only need to predict a sequence of tag types instead of word tokens from a huge vocabulary. And the model also absorbs hints from the IL side of input sequence so it is not at all an auto-encoder.

After LSTM layers in IL, the score function is given to assess all the name types for each token in IL based on hidden layers of LSTM as follows:

$$P_i = W_p tanh(W_f \mathbf{h}_f^{IL}[i] + W_b \mathbf{h}_b^{IL}[i])$$

where P_i is a vector of dimension V_t representing scores of tags for x_i^{IL}.

Training and Projecting

Following the architecture of Lample et al. (2016), we add a first order transition matrix A on top of the previous model to simulate a CRF structure. The score function between input $\mathbf{X}$ and output tag sequence $\mathbf{y}$ is as following,

$$s(\mathbf{X}, \mathbf{y}) = s(\tilde{X}_E, \tilde{X}_{IL}, T_E, \mathbf{y}) = \sum_{i=1}^{n_{IL}} A_{y_i, y_{i+1}} + \sum_{i=1}^{n_{IL}} P_{i, y_i}$$

where $\tilde{X}_E$ and $\tilde{X}_{IL}$ represent X_E and X_{IL} with their character sequences. To normalize each score into probability, a softmax function is added,

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X}, \mathbf{y})}}{\sum_{\tilde{\mathbf{y}} \in \mathbf{Y_x}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})}}$$

Thus, the final output $\mathbf{y}_d$ for decoding is the tag sequence that has the highest probability among all possible tag sequences $\mathbf{Y_X}$.

During training, our goal is to maximize $\log p(\mathbf{y}_c|\mathbf{X})$ for the correct (high-confidence annotation) tag sequence $\mathbf{y}_c$. Stochastic gradient descent (SGD) is employed with dropout strategy [4].

Using the trained bitext tagger in this section, we can complete the name projection task by annotating the rest of sentence pairs not covered in the high-confidence annotation set, and generate the final annotation set in IL by combining both high-confidence annotation set and annotations found by the bitext tagger.

[4] For the detailed training strategy, please read the training section in Lample et al. (2016)

2.3 Error Correction Strategies

Sec 2.1 and Sec 2.2 have already introduced the main content of the proposed framework. In this section, we make use of an overlooked information to further improve the quality of the projection.

According to Sec 2.1, to generate the high-confidence annotation set, we omit those sentence pairs whose names labeled in English are not completely projected in IL. For example, we will exclude a sentence pair in the first step if the number of names in English is three while only two names are projected to IL. However, these two projected names are still *high-confidence*. So if these names can be properly utilized, the quality of projection could be further improved. Here we introduce two different strategies to make use of this information:

Post-processing strategy

Since there are two annotation results on each of these omitted sentence pairs: annotations generated by the first step and annotations generated by the second step, it is natural to implement a post-processing step and integrate these two annotation results. Compared with the second step, although the annotations from the first step is not complete, they should be more reliable if rules in Sec 2.1 are strict enough. So here we follow this assumption and force the final projected annotations to maintain names produced in the first step and add names from the second step only when there is no conflict between them.

High-confidence annotations as one of inputs

A disadvantage of the post-processing method is that the assumption that annotations from the first step are more trust-worthy is not always true. When the annotated names produced by the first step are wrong, they will not only introduce noise, but also ruin annotations from the second step if there are conflicts between them. So a better idea is to feed all the information we have to neural networks and let the model speaks the truth.

To provide information of annotations produced by the first step, we add another input signal $T_{IL} = (t_1^{IL}, t_2^{IL}, ..., t_{n_{IL}}^{IL})$ in IL side, where $t_i^{IL} \in \{0, 1, ..., V_t - 1\}$ represents the tag result from the first step, V_t is the number of tag types. Then T_{IL} will go through tag embedding layers, LSTM-RNN layers and influence the results of CRF tagging.

In order to simulate the scenario of prediction process where the number of high-confidence names found on IL side is less than the number on English side, it is crucial to randomly drop out some high-confidence annotation names in T_{IL} and replace them with *Outside* during the training time.

3 Experiments

3.1 Data and Experimental Setup

To simulate a practical scenario, we evaluate our model on two low-resource ILs: Turkish and Uzbek, using the ground-truth name tagging annotations from the DARPA LORELEI program [5]. Table 1 shows data statistics. The numbers of sentence pairs for training and development represent the high-confidence annotation sets we produced during our experiments. And we exclude all the ground truth data from bitext for training and validation. Since a small proportion of sentences in the ground truth do not have English bitext, the test sets are slightly different in Sec 3.2 and Sec 3.3.

In our experiment, we set learning rate=0.01 and dropout rate=0.5. Dimension of word embedding=300, dimension of hidden layer=100, dimension of character embedding and hidden layer=25, and dimension of tag embedding=25.

3.2 Results of Bitext Name Projection

Bitext name taggers are trained on high-confidence annotation sets. Since most of ground truth sentences also contain parallel data, we can project annotations on bitext of ground truth and directly evaluate the quality of the projection. Here we compare our results with names projected from pure word alignment and also method in Sec 2.1. Table 2 shows the result.

[5] http://www.darpa.mil /program/low-resource-languages-for-emergent-incidents

Category	**Turkish**	**Uzbek**
Full sentence pairs for projection	24,193	39,045
Sentence pairs for training	12,276	21,552
Sentence pairs for dev.	755	1,431
Ground truth sentence pairs on bitext	1759	2918
Ground truth names on bitext	1744	2894
Ground truth sentence pairs in total	2,121	3,040
Ground truth names in total	2,178	3,144

Table 1: Data statistics

Dataset	Turkish			Uzbek		
	P	R	F	P	R	F
WA	34.5	33.4	33.9	28.6	29.9	29.2
HA	66.6	38.3	48.7	**66.5**	39.6	49.6
BNT	**67.4**	**49.7**	**57.2**	62.7	**47.8**	**54.3**

Table 2: Performance of projected annotation sets with different projection strategies. *WA* represents projection results purely based on **w**ord **a**lignment with IOB constraint. *HA* represents the **h**igh-confidence **a**nnotation projection. *BNT* represents the annotation projection using **b**itext **n**ame **t**agger.

From the result, it demonstrates that, in the case of low resource languages, annotations with word alignment results perform poorly on both precision and recall rate. So it is hard to improve the projection based on word alignment framework. Even re-ranking strategies with word alignment results is not effective since searching-based method (HA) has significantly outperformed the word alignment results. Also, compared with high-confidence annotations, our final annotation set shows significant better recall rate without much precision loss. Table 3 also indicates that after the second step, there are a huge number of names (not necessarily correct names) been discovered. This indicates in another aspect the significance of the recalling process in the second step. When observing real labeled cases, our model shows stable performances without much influences from the quality of word alignment result. For instance, the bitext name tagger could successfully label *Radyosu'nun* as a I-ORG in Fig 1.

Lang.	High-confidence annotation set			Annotation set after recalling		
	PER	LOC	ORG	PER	LOC	ORG
Turkish	6469	6133	2329	13520	21346	8094
Uzbek	9889	11353	1696	20522	29769	5348

Table 3: Statistics of tag number before and after using bitext name tagger.

Further more, since in the low resource scenario, large size of parallel corpora is not always available, we do experiments on different sizes of training data to evaluate the capability of this framework with less amount of bitexts. The subset sentence pairs for training are randomly selected. Figure 3 shows the result on Turkish and Uzbek. Different from the case where RNN is employed on machine translation, we do not need a huge number of training data because the searching space of prediction is quite small, due to the small amount of tag types rather than the size of a vocabulary. From the curve, we can see that even with only 2000 parallel sentence pairs, our model shows around 50% F-value. Notice that since we can directly employ the word alignment method and also method in Sec 2.1 on the ground truth without training process, the curve of these two methods are flat.

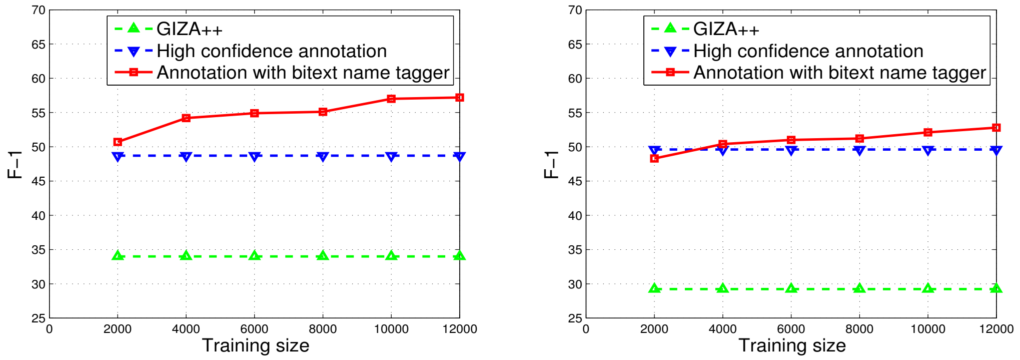

Figure 3: Performances of annotation projection with different size of training data on Turkish(left) and Uzbek(right) bitexts.

Model	Turkish			Uzbek		
	P	R	F	P	R	F
Clean	70.5	66.9	68.7	73.4	68.8	71.0
ExpDriv	45.8	**51.1**	48.3	38.3	41.0	39.6
HA *	62.3	45.3	52.5	59.4	45.2	51.3
BNT *	64.6	**51.1**	57.0	**62.6**	46.9	53.6
BNT + PP	65.6	49.7	56.5	59.5	**51.4**	55.2
BNT + HAI *	**66.4**	50.9	**57.6**	62.2	49.8	**55.3**

Table 4: Performance of IL name taggers. *Clean* represents the performance of LSTM-CRF model trained on human annotation data. *ExpDriv* refers to the baseline **Exp**ectation **driv**en method. *HA* represents LSTM-CRF model trained with **h**igh-confidence **a**nnotation set. *BNT* represents LSTM-CRF model trained with annotation set produced by the **b**itext **n**ame **t**agger. + *PP* represents *BNT* with **p**ost-**p**rocessing strategy. + *HAI* represents *BNT* with **h**igh-confidence **a**nnotations as one of **i**nputs. * indicates consistent improvement compared with results in the line above.

3.3 Results of IL Name Tagging

Since the final goal of annotation projection is to help IL tasks, we train and compare the performances of NER models on IL annotation sets produced with different projection methods. Here we choose the LSTM-CRF model proposed by Lample et al. (2016) to be the model of IL NER system because of its state-of-the-art performance on English NER task[6]. Table 4 shows the results.

From the results, the model trained with annotation set produced by the bitext name tagger outperforms both high-confidence annotations and also the state-of-the-art Expectation-driven learning method for IL name tagging (Zhang et al., 2016). Contrasting results between two different strategies in Sec 2.3, HAI shows consistent improvement across different languages while post-processing strategy fails in Turkish. Table 2 shows that, in Turkish, the precision of annotations from the second step outperforms the first step, which means in this case the assumption of the post-processing strategy is not valid.

4 Related Work

Most of related methods for annotation projection are based on word alignment results. Kim et al. (2010) employed both heuristic method and alignment correction with alignment dictionary of entity mentions. Das and Petrov (2011) designed a label propagation method to automatically induce a tag lexicon for the foreign language to smooth the projected labels. Wang and Manning (2014) project model expectations rather than labels, which facilities transfer of model uncertainty across language boundaries in word alignment projection.

[6] You can download the code of LSTM-CRF tagger from `https://github.com/glample/tagger`

Wang et al. (2013) also proposed a method to joint train word alignment and bilingual name tagging, which involves training process of word alignment. But this joint method is based on two assumptions. First, it requires a readily-trained name tagger in each languages. Even more, both taggers need to have competitive strengths so that they can correct each other. Unfortunately, in the case of low resource languages, no competitive name tagger is available.

One of most recent works linked to annotation projection was proposed by Fang and Cohn (2016) for the task of part of speech tagging (POS). Their work interestingly combined both gold annotations and projected ones by learning a global corrective matrix between gold annotation and projected annotation on IL side. The limitation is that gold annotation set on IL side must exist, which is not always the case especially in an incident language.

5 Conclusion and Future Work

We introduce a weakly-supervised framework for entity annotation projection. Our model takes original English and IL bitexts as inputs and does not heavily depend on word alignment results. Experiment results show that this method can provide significantly better annotations projected from English to IL.

Notice that in the first step of our framework, we do not consider the case where low-resource language does not use the Roman script. Although the method will still work because we also use word alignment results to measure the distance between text sequences, the performance could drop. So in the future work, we should plug in some transliteration techniques and gazetteers to make the system more robust.

Acknowledgments

We thank all the anonymous reviewers for their valuable comments and constructive suggestions. This work was supported by the 111 Project of China under Grant No. B08004, the key project of ministry of science and technology of China under Grant No. 2011ZX03002-005-01, the National Natural Science Foundation of China under Grant No. 61273217, the Natural Science Foundation of China under Grant No.61300080, the Ph.D. Programs Foundation of Ministry of Education of China under Grant No. 20130005110004, the U.S. DARPA LORELEI Program No. HR0011-15-C-0115 and ARL/ARO MURI W911NF-10-1-0533. The content is solely the responsibility of the authors and does not necessarily represent the official views of official policies or government.

References

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609. Association for Computational Linguistics.

Meng Fang and Trevor Cohn. 2016. Learning when to trust distant supervision: An application to low-resource pos tagging using cross-lingual projection. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Seokhwan Kim, Minwoo Jeong, Jonghoon Lee, and Gary Geunbae Lee. 2010. A cross-lingual annotation projection approach for relation detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 564–571. Association for Computational Linguistics.

Dan Klein, Joseph Smarr, Huy Nguyen, and Christopher D Manning. 2003. Named entity recognition with character-level models. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 180–183. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Hema Raghavan and James Allan. 2004. Using soundex codes for indexing names in asr documents. In *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*, pages 22–27. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Mengqiu Wang and Christopher D Manning. 2014. Cross-lingual pseudo-projected expectation regularization for weakly supervised learning. *Transactions of the Association for Computational Linguistics*, 2:55–66.

Mengqiu Wang, Wanxiang Che, and Christopher D Manning. 2013. Joint word alignment and bilingual named entity recognition using dual decomposition. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1073–1082.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proceedings of the 2001 Conference of the North American Chapter of the Association for Computational Linguistics*.

Boliang Zhang, Xiaoman Pan, Tianlu Wang, Ashish Vaswani, Heng Ji, Kevin Knight, and Daniel Marcu. 2016. Name tagging for low-resource incident languages based on expectation-driven learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*.