Cheap Translation for Cross-Lingual Named Entity Recognition

Stephen Mayhew Chen-Tse Tsai Dan Roth
University of Illinois, Urbana-Champaign
201 N. Goodwin
Urbana, Illinois, 61801
{mayhew2, ctsai12, danr}@illinois.edu

Abstract

Recent work in NLP has attempted to deal with low-resource languages but still assumed a resource level that is not present for most languages, e.g., the availability of Wikipedia in the target language. We propose a simple method for cross-lingual named entity recognition (NER) that works well in settings with *very* minimal resources. Our approach makes use of a lexicon to "translate" annotated data available in one or several high resource language(s) into the target language, and learns a standard monolingual NER model there. Further, when Wikipedia is available in the target language, our method can enhance Wikipedia based methods to yield state-of-the-art NER results; we evaluate on 7 diverse languages, improving the state-of-the-art by an average of 5.5% F1 points. With the minimal resources required, this is an extremely portable cross-lingual NER approach, as illustrated using a truly low-resource language, Uyghur.

1 Introduction

In recent years, interest in the natural language processing (NLP) community has expanded to include multilingual applications. Although this uptick of interest has produced diverse annotated corpora, most languages are still classified as low-resource. In order to build NLP tools for low-resource languages, we either need to annotate data (a costly exercise, especially for languages with few native speakers), or find a way to use annotated data in other languages in service to the cause. We refer to the latter techniques as cross-lingual techniques.

In this paper, we address cross-lingual named entity recognition (NER). Prior methods (described in detail in Section 2) depend heavily on limited and expensive resources such as Wikipedia or large parallel text. Concretely, there are about 3800 written languages in the world.[1] Wikipedia exists in about 280 languages, but most versions are too sparse to be useful. Parallel text may be found on an ad-hoc basis for some languages, but it is hardly a general solution. Religious texts, such as the Bible and the Koran, exist in many languages, but the unique domain makes them hard to use. This leaves the vast majority of the world's languages with no general method for NER.

	German	Spanish	Dutch	Avg
Baseline	22.61	45.77	43.10	37.27
Previous SOA	48.12	60.55	61.56	56.74
Cheap Translation	57.53	65.18	66.50	62.65

Table 1: We show dramatic improvement on 3 European languages in a low-resource setting. More detailed results in Table 2 show that this improvement continues to a wide variety of languages. The baseline is a simple direct transfer model. The previous state-of-the-art (SOA) is Tsai et al. (2016)

We propose a simple solution that requires only minimal resources. We translate annotated data in a high-resource language into a low-resource language, using just a lexicon.[2] We refer to this as *cheap translation*, because in general, lexicons are much cheaper and easier to find than parallel text (Mausam et al., 2010).

One of the biggest efforts at gathering lexicons is Panlex (Kamholz et al., 2014), which has lexicons for 10,000 language varieties available to download today. The quality and size of these dic-

[1] https://www.ethnologue.com/enterprise-faq/how-many-languages-world-are-unwritten-0

[2] We use the terms 'lexicon' and 'dictionary' interchangeably.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2526–2535
Copenhagen, Denmark, September 7–11, 2017.

tionaries may vary, but in Section 5.3 we showed that even small dictionaries can give improvements. If there is no dictionary, or if the quality is poor, then the Uyghur case study outlined in Section 6 suggests that effort is best spent in developing a high-quality dictionary, rather than gathering questionable-quality parallel text.

We show that our approach gives non-trivial scores across several languages, and when combined with orthogonal features from Wikipedia, improves on state-of-the-art scores. Table 1 compares a simple direct transfer baseline, the previous state-of-the-art in cross-lingual NER, and our proposed algorithm. For these languages, we beat the baseline by 25.4 points, and the state-of-the-art by 5.9 points. In addition, we found that translating from a language related to the target language gives a further boost. We conclude with a case study of a truly low-resource language, Uyghur, and show a good score, despite having almost no target language resources.

2 Related Work

There are two main branches of work in cross-lingual NLP: projection across parallel data, and language independent methods.

2.1 Projection

Projection methods take a parallel corpus between source and target languages, annotate the source side, and push annotations across learned alignment edges. Assuming that source side annotations are of high quality, success depends largely on the quality of the alignments, which depends, in turn, on the size of the parallel data.

There is work on projection for POS tagging (Yarowsky et al., 2001; Das and Petrov, 2011; Duong et al., 2014), NER (Wang and Manning, 2014; Kim et al., 2012; Ehrmann et al., 2011; Ni and Florian, 2016), mention detection (Zitouni and Florian, 2008), and parsing (Hwa et al., 2005; McDonald et al., 2011).

For NER, the received wisdom is that parallel projection methods work very well, although there is no consensus on the necessary size of the parallel corpus. Most approaches require millions of sentences, with a few exceptions which require thousands. Accordingly, the drawback to this approach is the difficulty of finding any parallel data, let alone millions of sentences. Religious texts (such as the Bible and the Koran) exist in a large number of languages, but the domain is too far removed from typical target domains (such as newswire) to be useful. As a simple example, the Bible contains almost no entities tagged as 'organization'. We approach the problem with the assumption that little to no parallel data is available.

2.2 Language Independent

The second common tool for cross-lingual NLP is to use language independent features. This is often called direct transfer, in the sense that a model is trained on one language and then applied without modification on a dataset in a different language. Lexical or lexical-derived features are typically not used unless there is significant vocabulary overlap between languages.

Täckström et al. (2012) experiments with direct transfer of dependency parsing and NER, and showed that using word cluster features can help, especially if the clusters are forced to conform across languages. The cross-lingual word clusters were induced using large parallel corpora.

Building on this work, Täckström (2012) focuses solely on NER, and includes experiments on self-training and multi-source transfer for NER.

Tsai and Roth (2016) link words and phrases to entries in Wikipedia and use page categories as features. They showed that these wikifier features are strong language independent features. We build on this work, and use these features in our experiments.

Bharadwaj et al. (2016) build a transfer model using phonetic features instead of lexical features. These features are not strictly language-independent, but can work well when languages share vocabulary but with spelling variations, as in the case of Turkish, Uzbek, and Uyghur.

2.3 Others

In a technique similar to ours, Carreras et al. (2003) use Spanish resources for Catalan NER. They translate the features in the weight vector, which has the flavor of a language independent model with the lexical features of a projection model. Our work is a natural extension of this paper, but explores these techniques on many more languages, showing that with some modifications, it has a broad applicability. Further, we experiment with orthogonal features, and with combining multiple source languages to get state of the art results on standard datasets.

Irvine and Callison-Burch (2016) build a machine translation system for low-resource languages by inducing bilingual dictionaries from monolingual texts.

Koehn and Knight (2001) experiment with varying knowledge levels on the task of translating German nouns in a small parallel German-English corpus. A lexicon along with monolingual text can correctly translate 79% of the nouns in the evaluation set. They reach a score of 89% when a parallel corpus is available along with a lexicon, but also comment on the scarcity of parallel corpora.

The main takeaways from the viewpoint of our work are a) word level translation can be effective, at least for nouns, and b) obtaining the correct word pair is more difficult than choosing between a set of options.

3 Our method: Cheap Translation

We create target language training data by translating source data into the target language. It is effectively the same as standard phrase-based statistical machine translation systems (such as MOSES (Koehn et al., 2007)), except that the translation table is not induced from expensive parallel text, but is built from a lexicon, hence the name *cheap translation*.

The entries in our lexicon contain word-to-word translations, as well as word-to-phrase, phrase-to-word, and phrase-to-phrase translations. Entries typically do not have any further information, such as part of speech or sense disambiguation. The standard problems related to ambiguity in language apply: a source language word may have several translations, and several source language words may have the same translation.

We are mostly concerned with the problem of multiple translations of a source language word. For example, in the English-Spanish lexicon, the English word *woman* translates into about 50 different words, with meanings ranging from *woman*, to *female golfer*, to *youth*. Although all candidates might be technically correct, we are interested in the most prominent translation. To estimate this, we gathered cooccurrence counts of each source-target word pair in the lexicon. For Spanish, in the case of *woman*, the most probable translation is *mujer*, because it shows up in other contexts in the dictionary, such as *farm woman* or *young woman*, whereas translations such as *joven* cooccur infrequently with *woman*. We normalize these cooccurrence counts in each candidate set, and call this the *prominence score*.

Algorithm 1 Our translation algorithm

Input
D_E: Annotated data in E
L: Lexicon between E–F
Returns
D_F: Annotated data in F

for $\forall w_i \in D_E$ **do**
  $p = w_i w_{i+1} ... w_{i+j}$ ▷ Window of size j
  while p not in L and $j \geq 0$ **do**
    Decrement j
    $p = w_i w_{i+1} ... w_{i+j}$
  end while
  if p in L **then**
    if $|L[p]| > 1$ **then**
      resolve with LM and prominence
    end if
    D_F add ($L[p]$, labels of p)
  else
    D_F add (w_i, label of w_i)
  end if
  Increment i by length of p
end for

With these probabilities in hand, we have effectively constructed a phrase translation table. We use a simple greedy decoding method (as shown in Algorithm 1) where options from the lexicon are resolved by a language model multiplied by the prominence score of each option. We use SRILM (Stolcke et al., 2002) trained on Wikipedia (although any large monolingual corpus will do).

During decoding, once we have chosen a candidate, we copy all labels from the source phrase to the target phrase. Since the translation is phrase-to-phrase, we can copy gold labels directly,[3] without worrying about getting good alignments. The result is annotated data in the target language.

Notice that the algorithm allows for no reordering beyond what exists in the phrase-to-phrase entries of the lexicon. Compared to phrase-tables learned from massive parallel corpora, our lexicon-based phrase tables are not large enough or expressive enough for robust reordering. We leave explorations of reordering to future work.

See Figure 1 for a representative example of translation from English to Turkish, with a human translation as reference. There are sin-

[3] We use a standard BIO labeling scheme.

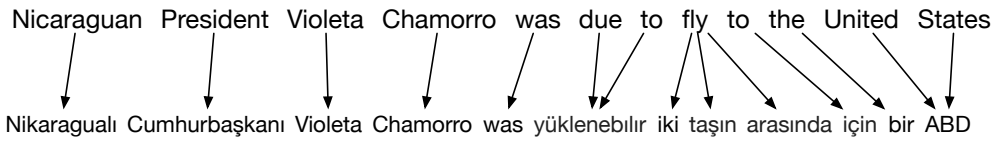

Figure 1: Demonstration of word translation. The top is English, the bottom is Turkish. Lines represent dictionary translations (e.g. *the* translates to *bir*). **Correct** is the correct translation. This illustrates congruence in named entity patterns between languages, as well as some errors we are prone to make.

gle words translated into phrases, named entities copied over verbatim, and phrases translated into single words. Some words are translated correctly (*President* into *Cumhurbaşkanı*) and some incorrectly (*fly* into *iki taşın arasında*, which loosely translates to 'between two stones'). We see ignorance of morphology (seen in translation of United States), and confused word order. But in spite of all these mistakes, the context around the entities, which is what matters for NER, is reasonably well-preserved. Notably, the word *President/Cumhurbaşkanı* is a strong context feature for both LOC (Nicaragua) and PER (Violeta Chamorro) in both languages.

4 Experimental Setup

Before we describe our experiments, we describe some of the tools we used.

4.1 Lexicons

We use lexicons provided by (Rolston and Kirchhoff, 2016), which are harvested from PanLex, Wiktionary, and various other sources. There are 103 lexicons, each mapping between English and a target language. These vary in size from 56K entries to 1.36M entries, as shown in the second row of Table 2. There are also noisy translations. Some entries consist of a single English letter, some are morphological endings, others are misspellings, others are obscure translations of metaphors, and still others are just wrong.

4.2 Datasets

We use data from CoNLL2002/2003 shared tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003). The 4 languages represented are English, German, Spanish, and Dutch. All training is on the train set, and testing is on the test set (TestB). The evaluation metric for all experiments is phrase level F1, as explained in Tjong Kim Sang (2002).

In order to experiment on a broader range of languages, we also use data from the REFLEX (Simpson et al., 2008), and LORELEI projects. From LORELEI, we use Turkish and Hausa [4] From REFLEX, we use Bengali, Tamil, and Yoruba.[5] We use the same set of test documents as used in Tsai et al. (2016).

We also use Hindi and Malayalam data from FIRE 2013,[6] pre-processed to contain only PER, ORG, and LOC tags.

While several of these languages are decidedly high-resource, we limit the resources used in order to show that our techniques will work in truly low-resource settings. In practice, this means generating training data where high-quality manually annotated data is already available, and using dictionaries where translation is available.

4.3 NER Model

In all of our work we use the Illinois NER system (Ratinov and Roth, 2009) with standard features (forms, capitalization, affixes, word prior, word after, etc.) as our base model. We train Brown clusters on the entire Wikipedia dump for any given language (again, any monolingual corpus will do), and include the multilingual gazetteers and wikifier features proposed in Tsai et al. (2016).

5 Experiments

We performed two different sets of experiments: first translating only from English, then translating from additional languages selected to be similar to the target language.

5.1 Translation from English

We start by translating from the highest resourced language in the world, English. We first show that

[4] LDC2014E115,LDC2015E70

[5] LDC2015E13,LDC2015E90,LDC2015E83, LDC2015E91

[6] http://au-kbc.org/nlp/NER-FIRE2013/

our technique gives large improvement over a simple baseline, then combine with orthogonal features, then compare against a ceiling obtained with Google Translate.

Baseline Improvement

To get the baseline, we trained a model on English CoNLL data (train set), and applied the model directly to the target language, mismatching lexical features notwithstanding. We did not use gazetteers in this approach. For the non-Latin script languages, Tamil and Bengali, we transliterated the entire English corpus into the target script. These results are in Table 2, row "Baseline".

In our approach ("Cheap Translation"), for each test language, we translated the English CoNLL data (train set) into that language. The first row of Table 2 shows the coverage of each dictionary. For example, in the case of Spanish, 90.94% of the words were translated into Spanish. This gives an average of 14.6 points F1 improvement over the baseline. This shows that simple translation is surprisingly effective across the board. The improvement is most noticeable for Bengali and Tamil, which are languages with non-Latin script. This mostly shows that the trivial baseline doesn't work across scripts, even with transliteration. Spanish shows the least improvement over the baseline, which may be because English and Spanish are so similar that the baseline is already high.

We found that we needed to normalize the Yoruba text (that is, remove all pronunciation symbols on vowels) in order to make the data less sparse. Since the training data for Bengali and Tamil never shares a script with the test data, we omit using the word surface form as a feature. This is indicated by the † in Table 2. Brown clusters, which implicitly use the word form, are still used.

Wikifier Features

Now we show that our approach is also orthogonal to other approaches, and can be combined with great effect. Wikifier features (Tsai et al., 2016) are obtained by grounding words and phrases to English Wikipedia pages, and using the categories of the linked page as NER features for the surface text. Our approach can be naturally combined with wikifier features. We show results in Table 2, in the row marked 'Cheap Translation+Wiki'.

Using wikifier features improves scores for all 7 languages. Further, for all languages we beat Tsai et al. (2016), with an average of 3.92 points F1 improvement. For the three European languages (Dutch, German, and Spanish), we have an average improvement of 4.8 points F1 over Tsai et al. (2016). This may reflect the fact that English is more closely related to European languages than Indian or African languages, in terms of lexical similarities, word order, and spellings and distribution of named entities. This suggests that it is advantageous to select a source language similar to the target language (by some definition of similar). We explore this hypothesis in Section 5.2.

Google Translate

Since we are performing translation, we compared against a high-quality translation system to get a ceiling. We used Google Translate to translate the English CoNLL training data into the target language, sentence by sentence. We aligned the source-target data using fast align (Dyer et al., 2013), and projected labels across alignments.[7] Since this is high-quality translation, we treat it as an upper bound on our technique, but with the caveat that the alignments can be noisy given the relatively small amount of text. This introduces a source of noise that is not present in our technique, but the loss from this noise is small compared to the gain from the high-quality translation. As with the other approaches, we found that Brown cluster features were an important signal.

Surprisingly, Google Translate beats our basic approach with a margin of only 4.3 points. Despite the naïvete of our approach, we are relatively close to the ceiling. Further, Google Translate is limited to 103 languages, whereas our approach is limited only by available dictionaries. In low-resource settings, such as the one presented in Section 6, Google Translate is not available, but dictionaries are available, although perhaps only by pivoting through a high-resource language.

5.2 Translation from Similar Languages

Observing that English as a source works well for European languages, but not as well for non-European languages, we form a key hypothesis: *cheap translation between similar languages should be better than between different languages*. There are several reasons for this. First, similar languages should have similar word orderings. Since we do no reordering in translation, this means the target text has a better chance of a

[7] Google Translate does not output alignments. If we had an in-house translation system, we could avoid this step.

Method	Dutch	German	Spanish	Turkish	Bengali	Tamil	Yoruba	Avg
Lexicon Coverage	88.01	89.97	90.94	83.80	83.34	73.84	74.60	–
E-L Dict size	961K	1.36M	1.25M	578K	217K	182K	334K	–
Baseline	43.10	22.61	45.77	34.63	6.40	4.60	37.70	27.83
Google Translate ceiling	65.71	56.65	53.65	45.63	37.84	29.11	39.18	46.82
Wiki (Tsai et al., 2016)	61.56	48.12	60.55	47.12	43.27	29.64	36.65	46.70
Cheap Translation	53.94	50.96	51.82	46.37	30.47†	25.91†	37.58	42.43
Cheap Translation+Wiki	63.37	57.23	64.10	51.79	46.28†	33.10†	38.52	50.62
Best Combination	64.48	57.53	65.95	48.50	31.70†	27.63†	39.12	47.84
Best Combination+Wiki	66.50	59.11	65.43	53.44	45.70†	34.90†	40.88	52.28

Table 2: Baseline is naive direct transfer, with no gazetteers. 'Cheap Translation' translates from English into the target. Google Translate translates whole sentences, and does not use gazetteers. 'Cheap Translation+Wiki' incorporates wikifier features. 'Best Combination' uses language combinations from Table 3 for source training data. † denotes that this run does not use word features.

Target	Train lang
Dutch	English, German
German	English, Dutch
Spanish	English, Dutch
Turkish	English, Uzbek
Bengali	English, Hindi
Tamil	English, Malayalam
Yoruba	English, Hausa

Table 3: This shows the language selection results. In each row, we see the target language, and the languages used for training. For example, when testing on Dutch, we train on German and English. These scores came from WALS.

coherent ordering. Second, in case of dictionary misses, vocabulary common between languages will be correct in the target language.

This requires two new resources: annotated data in a similar language S, and a lexicon that maps from S to T, the target language.

Data in other languages

For most target languages, English is not the closest language, and it is likely that there exists an annotated dataset in a closer language. There are annotated datasets available in many languages with a diversity of script and family. We have datasets annotated in about 10 different languages, although more exist.

One caveat is that the source dataset must have a matching tagset with the target dataset. At present, we accept this as a limitation, with the understanding that there is a common set of coarse-grained tags that is widely used (PER, ORG, LOC). We leave further exploration to future work.

Pivoting Lexicons

Although we cannot expect to find lexicons between all pairs of languages, we can usually expect that a language will have at least one lexicon with a high-resource language. Often that language is English. We can use this high-resource language as a pivot to transitively create an S–T dictionary, although perhaps with some loss of precision.

Assume we want a Turkish-Bengali lexicon and we have only English-Bengali and English-Turkish lexicons. We collect all English words that appear in both dictionaries. Each such English word has two sets of candidate translations, one set in Turkish, the other in Bengali. To create transitive pairs, we take the Cartesian product of these two sets of candidate translations. This will create too many entries, some of which will be incorrect, but usually the correct entry is there.

Notice also that the resulting dictionary contains only those English words that appear in both original dictionaries. If either of the original dictionaries is small, the result will be smaller still.

Source Selection and Combination

To choose a related source language, we used syntactic features of languages retrieved from the World Atlas of Language Structures (WALS) (Dryer and Haspelmath, 2013). Each language is represented as a binary vector with each index indicating presence or absence of a syntactic feature in that language. We used the feature set called *syntax_knn*, which includes 103 syntactic features, such as *subject before verb*, and *possessive suffix*, and uses k-Nearest Neighbors to predict missing values. We measure similarity as cosine distance between language vectors.

In the absence of criteria for a similarity cutoff, we chose to include only the top most similar language as source for that target language. The results of this similarity calculation are shown in Table 3. For example, when the target language is Dutch, German is the closest. We also included English in the training, as the highest resource language, and with the highest quality dictionaries.

Results

Our results are in Table 2, in the row named 'Best Combination'. The average over all languages surpasses the English-source average by 5.4 points, and also beats (Tsai et al., 2016). We also add wikifier features, and report results in row 'Best Combination+Wiki.' This shows improvement on all but Spanish, with an average improvement of 5.58 points F1 over Tsai et al. (2016). To the best of our knowledge, these are the best cross-language settings scores for all these datasets.

While these scores are lower than those seen on typical NER tasks (70-90% F1), we emphasize first that cross-lingual scores will necessarily be much lower than monolingual scores, and second that these are the best available given the setup.

5.3 Dictionary Ablation

The most expensive resource we require is a lexicon. In this section, we briefly explore what effect the size of the lexicon has on the end result. Using Turkish, we vary the size of the dictionary by randomly removing entries. The sizes vary from no entries to full dictionary (rows 'Baseline' and 'Cheap Translation' in Table 2, respectively), with several gradations in the middle. With each reduced dictionary, we translate from English to generate Turkish training data as in Section 5.1. As before, we train an NER model on the generated data, and test on the Turkish test data. Results are shown in Figure 2.

Interestingly, we see improvement over the baseline even with only 500 entries. This improvement continues until 125K entries. It is important to note that only a small number of dictionary entries – words that typically show up in the contexts of named entities, such as *president*, *university* or *town* – are likely to be useful. The larger the dictionary, the more likely these valuable entries are present. Further, our random removal process may unfairly prioritize less common words, compared to a manually compiled dictionary which would prioritize common words. It is likely that a small but carefully constructed manual dictionary could have a large impact.

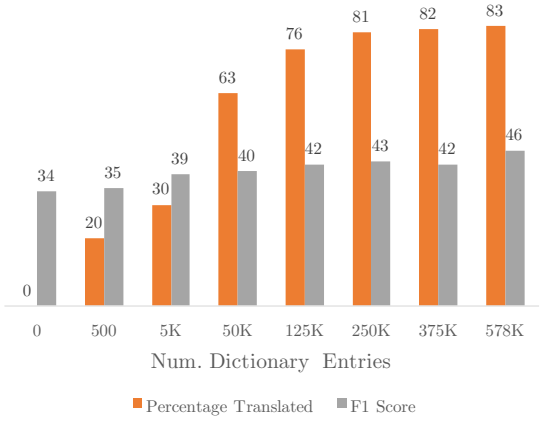

Figure 2: Effect of dictionary size on F1 score for Turkish. Each column is an experiment with a randomly reduced dictionary. The orange bars represent how much of the corpus is translated.

6 Case study: Uyghur

We have shown in the previous sections that our method is effective across a variety of languages. However, all of the tested languages have some resources, most notably, Google Translate and reasonably sized Wikipedias. In this section, we show that our methods hold up on a truly low-resource language, Uyhgur.

Uyghur is a language native to northwest China, with about 25 million speakers.[8] It is a Turkic language, and is related most closely to Uzbek, although it uses an Arabic writing system. Uyghur is not supported by Google Translate, and the Uyghur Wikipedia has less than 3,000 articles. In contrast, the smallest Wikipedia size language in our test set is Yoruba, with 30K articles. Because of the small Wikipedia size, we do not use any wikifier features.

We did this work as part of the NIST LoReHLT evaluation in the summer of 2016. The official evaluation scores were calculated over a set of 4500 Uyghur documents. Each team was given the unannotated version of those documents, with the task being to submit annotations on that set. Our official scores are reported in Table 4, and compared with Bharadwaj et al. (2016).

After the evaluation, NIST released 199 of the annotated evaluation documents, called the *unse-*

[8] https://en.wikipedia.org/wiki/Uyghur_language, accessed July 21, 2017

Method	All	Unseq.
Bharadwaj et al. (2016)	51.2	–
Ours	55.6	51.32

Table 4: F1 scores of official submissions in LoReHLT16. The numbers in the "All" column are the scores on the entire evaluation data reported from NIST. We evaluate our submissions on the unsequestered data in order to compare with the results in Table 5.

Method	F1
Monolingual	69.92
Standard Translation	
Train: English	27.20
Train: Turkish	33.02
Train: Uzbek	27.88
Language Specific (stemmed)	
Train: English	30.84
Train: Turkish	40.04
Train: Uzbek	40.15
Induced dictionaries	43.46
Manual annotations	42.51
All Lang. Spec.	51.32

Table 5: F1 scores for Uyghur. Monolingual scores are on the 41 document test set. All other scores are on the full unsequestered data. We omit forms or gazetteers but use Brown clusters. 'Standard Translation' uses the same resources as the scores in Table 2 (e.g. without stemming)

questered set. In this section, we will drill into the various methods we used to build the transfer model, and report finer-grained results using the unsequestered set.

The following are some of the language-specific techniques we employed.

- **Dictionary** The dictionary provided for Uyghur from Rolston and Kirchhoff (2016) had only 5K entries, so we augmented this with the dictionary provided in the LORELEI evaluation, which resulted in 116K entries.

- **Name Substitution** As with Bengali and Tamil, very few names were translated. We found transliteration models were too noisy, so instead, we gathered a list of gazetteers from Uyghur Wikipedia, categorized by tag type (PER, LOC, GPE, ORG). Upon encountering an untranslatable NE, we replaced it with a randomly selected NE from the gazetteer list corresponding to the tag. This led to improbable sentences like *John Kerry has joined the Baskin Robbins*, but it meant that NEs were fluent in the target text.

- **Stemming** We created a very simple stemmer for Uyghur. This consists of 45 common suffixes sorted longest first. For each Uyghur word in a corpus, we removed all possible suffixes (Uyghur words can take multiple suffixes). We stemmed all train and test data.

We report results in Table 5. The first row is from a monolingual model trained on 158 documents in the unsequestered set, and tested on the remaining 41. All other rows test on the complete unsequestered set. The next section, 'Standard Translation', refers to the method described above. Notably, we do not use stemming for train or test data here. As with Bengali and Tamil, we omit form features.

We translate from English, Turkish, and Uzbek, which are the closest languages predicted by WALS. Next, we incorporated language specific methods. The scores we get from training on English, Turkish and Uzbek all go up because the stemming makes the features more dense. Next we generated dictionaries using observations over Uyghur and Uzbek, and we used non-native speakers to annotate Uyghur data.

6.1 Language Specific Dictionary Induction

We began by romanizing Uyghur text into the Uyghur Latin alphabet (ULY) so we could read it. We noticed that Uzbek and Uyghur are very similar, sharing a sizable amount of vocabulary, and several morphological rules. However, while there is a shared vocabulary, the words are usually spelled slightly differently. For example, the word for "southern" is "janubiy" in Uzbek and "jenubiy" in Uyghur.

We tried several ideas for gathering a mapping for this shared vocabulary: manual mapping, edit-distance mapping, and cross-lingual CCA with word vectors.

Manual mapping: We manually translated about 100 words often found around entities, such as *president*, and *university*

Edit-distance mapping: We gathered (Uyghur, Uzbek) word pairs with low-edit distance, using a modified edit-distance algorithm that allowed cer-

tain substitutions at zero cost. For example, this discovered such pairs as *pokistan-pakistan* and *telegraph-télégraf*.

Cross-lingual CCA with word vectors: We projected Uyghur and Uzbek monolingual vectors into a shared semantic space, using CCA (Faruqui and Dyer, 2014). We used the list of low edit-distance word pairs as the dictionary for the projection. Once all the vectors were in the same space, we found the closest Uyghur word to each Uzbek word.

6.2 Results

Scores are in Table 5. Interestingly, the language specific methods evaluated individually did not improve much over the generic word translation methods. But with all language specific methods combined, 'All Lang. Spec.', the score increased by nearly 10 points, suggesting that the different training data covers many angles.

To the best of our knowledge, there are no published scores on the unsequestered data set. Our best score is comparable to the score of our evaluation submission on the unsequestered dataset.

7 Conclusion

We have shown a novel cross-lingual method for generating NER data that gives significant improvement over state-of-the-art on standard datasets. The method benefits from annotated data in many languages, combines well with orthogonal features, and works even when resources are virtually nil. The simplicity and minimal use of resources makes this approach more portable than all previous approaches.

Acknowledgments

This material is based on research sponsored by the US Defense Advanced Research Projects Agency (DARPA) under agreement numbers FA8750-13-2-0008 and HR0011-15-2-0025. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

References

Akash Bharadwaj, David R. Mortensen, Chris Dyer, and Jaime G. Carbonell. 2016. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *EMNLP*.

Xavier Carreras, Lluís Màrquez, and Lluís Padró. 2003. Named entity recognition for catalan using spanish resources. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 43–50. Association for Computational Linguistics.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Portland, OR. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird, and Paul Cook. 2014. What can we get from 1000 tokens? A case study of multilingual POS tagging for resource-poor languages. In *EMNLP*, pages 886–897. Citeseer.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. Association for Computational Linguistics.

Maud Ehrmann, Marco Turchi, and Ralf Steinberger. 2011. Building a multilingual named entity-annotated corpus using annotation projection. In *RANLP*.

M. Faruqui and C. Dyer. 2014. Improving vector space word representations using multilingual correlation. Association for Computational Linguistics.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara I. Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:311–325.

Ann Irvine and Chris Callison-Burch. 2016. End-to-end statistical machine translation with zero or small parallel texts. *Natural Language Engineering*, 22:517–548.

David Kamholz, Jonathan Pool, and Susan M Colowick. 2014. Panlex: Building a resource for panlingual lexical translation. In *LREC*, pages 3145–3150.

Sungchul Kim, Kristina Toutanova, and Hwanjo Yu. 2012. Multilingual named entity recognition using parallel data and metadata from Wikipedia. In *ACL*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran,

Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2001. Knowledge sources for word-level translation models.

Stephen Mausam, Soderland, Oren Etzioni, Daniel S Weld, Kobi Reiter, Michael Skinner, Marcus Sammer, Jeff Bilmes, et al. 2010. Panlingual lexical translation via probabilistic inference. *Artificial Intelligence*, 174(9-10):619–637.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Jian Ni and Radu Florian. 2016. Improving multilingual named entity recognition with wikipedia entity type mapping.

L. Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

Leanne Rolston and Katrin Kirchhoff. 2016. Collection of bilingual data for lexicon transfer learning. Technical report, University of Washington.

Heather Simpson, Christopher Cieri, Kazuaki Maeda, Kathryn Baker, and Boyan Onyshkevych. 2008. Human language technology resources for less commonly taught languages: Lessons learned toward creation of basic language resources. *Collaboration: interoperability between people in the creation of language resources for less-resourced languages*, page 7.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *Interspeech*, volume 2002, page 2002.

Oscar Täckström. 2012. Nudging the envelope of direct transfer methods for multilingual named entity recognition. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 55–63. Association for Computational Linguistics.

Oscar Täckström, Ryan T. McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *NAACL*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. 2016. Cross-lingual named entity recognition via wikification. *CoNLL 2016*, page 219.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *NAACL*.

Mengqiu Wang and Christopher D Manning. 2014. Cross-lingual projected expectation regularization for weakly supervised learning. In *TACL*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics.

Imed Zitouni and Radu Florian. 2008. Mention detection crossing the language barrier. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 600–609. Association for Computational Linguistics.