Learning Better Name Translation for Cross-Lingual Wikification

Chen-Tse Tsai
Bloomberg LP[*]
New York, NY
ctsai54@bloomberg.net

Dan Roth
University of Pennsylvania[*]
Philadelphia, PA
danroth@seas.upenn.edu

Abstract

A notable challenge in cross-lingual wikification is the problem of retrieving English Wikipedia title candidates given a non-English mention, a step that requires translating names[1] written in a foreign language into English. Creating training data for name translation requires significant amount of human efforts. In order to cover as many languages as possible, we propose a probabilistic model that leverages indirect supervision signals in a knowledge base. More specifically, the model learns name translation from title pairs obtained from the inter-language links in Wikipedia. The model jointly considers word alignment and word transliteration. Comparing to 6 other approaches on 9 languages, we show that the proposed model outperforms others not only on the transliteration metric, but also on the ability to generate target English titles for a cross-lingual wikifier. Consequently, as we show, it improves the end-to-end performance of a cross-lingual wikifier on the TAC 2016 EDL dataset.

Introduction

Cross-lingual wikification is the problem of grounding entity mentions written in a foreign language to the English Wikipedia. That is, given a mention from a document written in language L, different than English, the goal is to find its corresponding title in the English Wikipedia. Figure 1 shows an example. This task is driven by the will of English readers to partly understand documents in low resource languages and, in particular, in those languages with no English translation technology. Instead of translating the whole document to English, grounding the important entity mentions in the English Wikipedia may provide a good partial solution that could at least capture the key message of the text.

One of the key challenges for wikification is the *candidate generation* step – generating title candidates given a mention. Since there are millions of entries in the English Wikipedia, this step aims at quickly producing a manageable number of title candidates, so that a more sophisticated algorithm can be applied to rank them. The candidate generation step is typically done by indexing titles in Wikipedia using strings that could be used to refer to the titles.

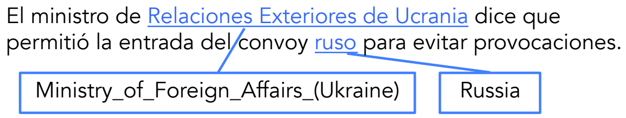
El ministro de Relaciones Exteriores de Ucrania dice que permitió la entrada del convoy ruso para evitar provocaciones.
Ministry_of_Foreign_Affairs_(Ukraine)
Russia

Figure 1: An example of cross-lingual wikification. The Spanish mention "ruso" is grounded to the English Wikipedia title "Russia", and "Relaciones Exteriores de Ucrania" is grounded to the title "Ministry_of_Foreign_Affairs_(Ukraine)".

	Current upper bound	Proposed method
Spanish	40.44%	64.62%
Bengali	20.82%	65.18%
Tagalog	16.23%	73.23%

Table 1: Paper Impact: Improved title candidate generation coverage. The left column gives the fraction of Wikipedia titles that are covered by the inter-language links. This is the *upper bound for current cross-lingual wikification methods*. The right column gives the fraction of these mentions that have the correct English title in the candidate set using the method developed in this paper.

In the cross-lingual setting, this problem becomes more challenging. There are two intuitive ways to retrieve English title candidates given a foreign mention: 1) Querying the English titles' index directly using a foreign mention. 2) Querying the foreign language titles' index using the foreign mention, and then converting the foreign titles to English using the inter-language links in Wikipedia. The first approach only works if the target language is very close to English, so that names in the two languages are expressed almost identically. The second approach depends heavily on the size of the foreign language Wikipedia. That is, this approach only works if the target entity exists in the foreign language Wikipedia, and there is a link pointing to the corresponding English page. In Figure 1, the first approach does not work for any of the two mentions since they are expressed differently in English. The second approach will

[*] Most of this work was done while the authors were at University of Illinois at Urbana-Champaign.

[1] We use the term *name translation* to refer to the process of converting names from one language to another, which includes the transliteration problem for some type of names.

work on the mention "ruso", because we can get to the target entity "Russia" in the Spanish Wikipedia based on the given mention "ruso", and further reach Russia's English page via the inter-language links in Wikipedia. However, since "Ukraine's Ministry of Foreign Affair" does not exist in the Spanish Wikipedia, this approach fails to find the correct English title for the mention "Relaciones Exteriores de Ucrania". Table 1 shows an upper bound on the coverage of the second approach. These numbers are estimated from the anchor texts in Wikipedia articles. For example, only 20.82% of the Bengali mentions are linked to English titles. In our experiments, we show that our model can retrieve the target English title for 65.18% of the Bengali mentions.

In this paper, we focus on cases that cannot be addressed by these two approaches. One solution to this problem is to use a transliteration or translation model. We can translate foreign names into English and then use it to query the index of the English titles. However, the use of standard transliteration and translation models in this context suffers from two problems. First, the traditional setting of transliteration focuses only on single-token names of people or locations, but for wikification, the entities that we want to ground are often longer (e.g., names of organizations). Since multi-token names of locations and organizations typically require a mixture of translation and transliteration, they are excluded from "standard" transliteration studies. Second, the transliteration models are usually learned from word pairs, which could be manually created or mined from large amounts of parallel text. These are also required to train machine translation models. However, with the goal of solving cross-lingual wikification for all languages in Wikipedia, including many low resource languages, we cannot assume large amounts of high quality parallel data.

We propose a probabilistic model which learns name translation from Wikipedia title pairs. Using the inter-language links in Wikipedia, we can obtain foreign-to-English title pairs for different types of entities and for all languages in Wikipedia. Since we learn from phrase pairs rather than word pairs, we extend a transliteration model to jointly model word alignment and word-to-word transliteration. It is clear that if we can align words in a phrase pair well, we can learn word transliteration better. On the other hand, a good transliteration model can help to improve word alignment performance, because the transliteration model may provide better word generation probability if the word pair appears infrequently in the training data, but the subwords pairs in the word pair are frequent enough.

We compare the proposed model with six strong approaches from the literature, including 4 transliteration models and 2 character-based neural machine translation models. When these models are trained on Wikipedia title pairs of 9 languages, we show that our model outperforms these approaches not only on the standard string similarity-based metric, but also on the candidate generation performance of cross-lingual wikification. Finally, we show that our model improves an end-to-end cross-lingual wikification system on the TAC 2016 EDL dataset.

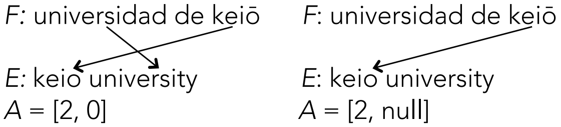

Figure 2: Examples of the word alignment variable A in Eq. (1).

The Joint Model

In this section, we present our model for learning name translation from Wikipedia title pairs.

Given a title pair (F, E), where F is the foreign title and E is the target English title, let the number of words in F and E be m and l respectively, we model the title generation probability as

$$P(F, A|E, m) = P(A|m) \prod_{(f,e)\in A} P(f|e), \qquad (1)$$

where A is an alignment assignment of words $f \in F$ and $e \in E$. The alignment A is a list of size $|E| = l$, where $A[j]$ could either be null, or the index of the word in F which is aligned with the j-th target word. Figure 2 shows two examples. Given a Spanish-English title pair ("universidad de keiō", "keio university"), the word alignments variable A in the left example is $[2, 0]$. The 2 in the first position means that "keio" is aligned with "keiō", and the 0 indicates "university" is aligned with "universidad". In the right example, since the word "university" is not aligned with any source word, the second element in A becomes *null*. In order to reduce the number of possible alignments A, we assume that a source word (in F) can be aligned with up to one target word, and no two source words can be aligned with the same target word. Note that this way, in contrast to word alignment models in machine translation which usually have independence assumption between words, our model jointly determines the alignment of all words in the pair of strings. Training and inference in our model are tractable since the number of words in a title is typically small.

The last term of Eq. (1) is the word generation probability given the word alignment, where $(f, e) \in A$ is a word pair according to the alignment A. In the left example of Figure 2, there are two word pairs, (universidad, university) and (keiō, keio), therefore $\prod_{(f,e)\in A} P(f|e) = P(\text{universidad}|\text{university})P(\text{keiō}|\text{keio})$. This is where we use a transliteration model to better estimate the word generation probability. We adopt the model proposed by Pasternack and Roth (2009) in which the word generation probability is modeled as

$$P(f, a|e) = \prod_{(u,v)\in a} P(v|u), \qquad (2)$$

where a indicates sub-word alignment between the foreign word f and the English word e. For instance, given $(f, e) = (\text{keiō}, \text{keio})$, one possible sub-word alignment is (ke-iō, ke-io), namely, "ke" is aligned with "ke" and "iō" is aligned with "io", therefore $P(f, a|e) = P(\text{ke}|\text{ke})P(\text{iō}|\text{io})$. This model will try all sub-word alignments that segment both source and target words into the same number of sub-words. Using Eq. (2), we can compute the word generation probability by:

$$P(f|e) = \sum_{a} \prod_{(u,v)\in a} P(v|u), \tag{3}$$

which sums over all possible sub-word alignments between the word pair (f, e). Combining Eq. (1) and (2), we have our final likelihood:

$$\begin{aligned} &P(F, A, a_A|E, m) = \\ &P(A|m) \prod_{(f,e)\in A} \prod_{(u,v)\in a_f^e} P(v|u). \end{aligned} \tag{4}$$

We use a_A to represent sub-word alignments of all word pairs according to the word alignment A, and use a_f^e to represent the sub-word alignments between a particular word pair (f, e).

To summarize, given a title pair (F, E), our model uses a latent variable A to indicate word alignments between the two titles and uses a_A to describe sub-word alignments. We use the EM algorithm (Dempster, Laird, and Rubin 1977) to maximize the likelihood of training pairs, and update the parameters $P(A|m)$ and $P(v|u)$ iteratively.

After training the model, we can generate the target English phrase given a foreign phrase F using Bayes rule:

$$\begin{aligned} E^* &= \arg\max_E P(E|F) = \arg\max_E \frac{P(F|E)P(E)}{P(F)} \\ &= \arg\max_E \sum_A \sum_{a_A} P(F, A, a_A|E, m)P(E), \end{aligned}$$

where $P(F, A, a_A|E)$ is from Eq. (4) and $P(E)$ is obtained from an English language model. The second summation over sub-word alignments can be computed using dynamic programming efficiently (Pasternack and Roth 2009). However, since our word alignments are joint assignments and there could be crossing edges (Figure 2), we actually iterate through all possible word alignments for the first summation in the above equation. This is feasible because most names have no more than three words and due to the assumption made at the beginning of this section, which guarantees that the space of legitimate alignments is small enough.

Expectation Maximization

In this section, we derive update rules for the parameters in the proposed model. In Eq. (4), two types of parameters are $P(A|m)$ and $P(v|u)$. For each possible length of foreign title, m, there is a distribution of word alignments to the English titles. $P(v|u)$ is the probability of a foreign sub-word u aligns to an English sub-word v.

Given training title pairs (F_i, E_i), $i = 1, \cdots, n$, the expected log likelihood of Eq. (4) is

$$\begin{aligned} &E_A E_{a_A}[\sum_{i=1}^{n} (\log P(A_i|m_i) + \\ &\sum_{(f,e)\in A_i} \sum_{(u,v)\in a_f^e} \log P(v|u))] \\ =&E_A[\sum_{i=1}^{n} \log P(A_i|m_i)] + \\ &E_A E_{a_A}[\sum_{i=1}^{n} \sum_{(f,e)\in A_i} \sum_{(u,v)\in a_f^e} \log P(v|u)] \end{aligned} \tag{5}$$

The expectation with respect to word alignment A and sub-word alignment a_A are computed using the current parameters, which are updated in the previous iteration. The first term in Eq. (5) can be expanded:

$$\begin{aligned} &E_A[\sum_{i=1}^{n} \log P(A_i|m_i)] \\ =&\sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \log P(A_i|m_i), \end{aligned}$$

Adding the constraints that $\sum_A P(A|m) = 1, \forall m$, and using the Lagrangian multipliers method, we have the following objective:

$$\begin{aligned} &\sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \log P(A_i|m_i) - \\ &\sum_{m} \alpha_m (\sum_{A} P(A|m) - 1). \end{aligned}$$

To maximize this function, we take the partial derivative with respect to $P(\bar{A}|\bar{m})$, a particular word alignment $\bar{A}$ given $\bar{m}$ source tokens, and set the result to 0.

$$\sum_{i:|F_i|=\bar{m}} \frac{P(\bar{A}|F_i, E_i)}{P(\bar{A}|\bar{m})} - \alpha_{\bar{m}} = 0$$

The update rule of $P(\bar{A}|\bar{m})$ is

$$P(\bar{A}|\bar{m}) = \frac{\sum_{i:|F_i|=\bar{m}} P(\bar{A}|F_i, E_i)}{\sum_{i:|F_i|=\bar{m}} \sum_A P(A|F_i, E_i)}, \tag{6}$$

where $P(A|F_i, E_i)$ can be computed from the current parameters:

$$\begin{aligned} P(A|F_i, E_i) &= \frac{P(A, F_i|E_i)}{P(F_i|E_i)} \\ =&\frac{\sum_{a_A} P(A, F_i, a_A|E_i, m_i)}{\sum_A \sum_{a_A} P(A, F_i, a_A|E_i, m_i)} \\ =&\frac{P(A|m_i) \prod_{(f,e)\in A} \sum_{a_f^e} \prod_{(u,v)\in a_f^e} P(v|u)}{\sum_A P(A|m_i) \prod_{(f,e)\in A} \sum_{a_f^e} \prod_{(u,v)\in a_f^e} P(v|u)} \\ =&\frac{P(A|m_i) \prod_{(f,e)\in A} P(f|e)}{\sum_A P(A|m_i) \prod_{(f,e)\in A} P(f|e)} \end{aligned}$$

The last equation is derived by using Eq. (3).

For the second term in Eq. (5),

$$E_A E_{a_A}[\sum_{i=1}^{n} \sum_{(f,e)\in A_i} \sum_{(u,v)\in a_f^e} \log P(v|u)]$$

$$= \sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \times \sum_{(f,e)\in A_i} \sum_{a_f^e} P(a_f^e) \sum_{(u,v)\in a_f^e} \log P(v|u),$$

where $P(a_f^e)$ is the word generation probability of the foreign word f and the English word e given a particular sub-word alignment a, which is exactly Eq. (2).

Adding the constraints that $\sum_v P(v|u) = 1, \forall u$, and using the Lagrangian multiplier method, we obtain

$$\sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \times \sum_{(f,e)\in A_i} \sum_{a_f^e} P(a_f^e) \times \sum_{(u,v)\in a_f^e} \log P(v|u) - \sum_u \beta_u (\sum_v P(v|u) - 1).$$

To maximize this expectation, we take the partial derivative with respect to the generation probability of a particular pair of sub-words $P(\bar{v}|\bar{u})$, and set the result to 0.

$$\sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \sum_{a_f^e \in A_i} \frac{n_{\bar{u},\bar{v}|a_f^e} P(a_f^e)}{P(\bar{v}|\bar{u})} - \beta_u = 0,$$

where $n_{\bar{u},\bar{v}|a_f^e}$ is number of times the sub-word $\bar{v}$ is aligned with the sub-word $\bar{u}$ under the word alignment A_i and the sub-word alignment a_f^e. The update rule of $P(\bar{v}|\bar{u})$ becomes

$$P(\bar{v}|\bar{u}) = \sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \sum_{a_f^e \in A_i} \frac{n_{\bar{u},\bar{v}|a_f^e}}{n_{\bar{u},*|a_f^e}} P(a_f^e), \quad (7)$$

where $n_{\bar{u},*|a_f^e}$ is the number of times the sub-word $\bar{u}$ occurs in any target word under the word alignment A_i and the sub-word alignment a_f^e. The term $P(a_f^e)$ can be computed from the current parameters using Eq. (2).

We have derived the update rules Eq. (6) and (7) for the parameters in the proposed model (Eq. (4)).

Note that for the frequent foreign words in the training data, we memorize their translation by taking the most probable alignment in each iteration. That is, during updating $P(A|m)$ using Eq. (6), we also compute word translation probabilities $P(e|f)$ for each foreign word f. These word pairs are excluded in updating sub-word generation probabilities (Eq. (7)), since these words are usually translated instead of transliterated. More specifically, when we iterate through word pairs in the third summation of Eq. (7), we simply skip the frequent word pairs. For example, in Turkish, "ili" means prefecture and "adaları" means islands. Since the sub-word alignments of these word pairs are very different from the words that are transliterated, using word pairs (ili, prefecture) and (adaları, islands) to update sub-word generation probabilities may result in worse estimation. In our experiments, we choose the number of words to exclude using development sets. The top 10 frequent foreign words are usually selected.

Experiments

We compare our model with six other approaches. The first four approaches are the standard transliteration models which are designed to learn from transliteration word pairs:

- **DirecTL+** (Jiampojamarn, Cherry, and Kondrak 2008) is a discriminative string transduction tool, which was successfully applied to transliteration in the NEWS shared tasks. Given sub-word aligned word pairs, DirecTL+ views transliteration problem as a sequence tagging problem. We use m2m-aligner (Jiampojamarn, Kondrak, and Sherif 2007) to segment and align the input word pairs.
- **Sequitur** (Bisani and Ney 2008) is a probabilistic model for grapheme-to-phoneme conversion. Unlike DirecTL+, which requires sub-word alignment, Sequitur directly trains a joint n-gram model from unaligned word pairs. Higher order n-gram models are trained iteratively from lower order models. We train up to 5-gram models.
- **P&R** (Pasternack and Roth 2009) is the model which our model is based on. The probability of the source word given the target word is modeled as in Eq. (2), where sub-word alignments are described by the latent variable a.
- **JANUS** (Liu et al. 2016) trains a character-based left-to-right and a right-to-left LSTM model on the input word pairs. The prediction is based on the agreement between the outputs of these two models. We use 500 dimensional embeddings and 100 training epochs.

We also compare with the following two character-level neural machine translation models:

- **NMT-bpe** (Chung, Cho, and Bengio 2016) segments the source words into sub-words using byte pair encoding (Sennrich, Haddow, and Birch 2015), and encodes these sub-words by gated recurrent units (GRUs). When decoding, a newly designed character-level bi-scale recurrent neural network is applied.
- **NMT-char**[2] is inspired by NMT-bpe. This model not only decodes at character-level, but also encodes the source side at character-level. When encoding, the model applies a series of convolutional, pooling, and highway layers. The results are fed into a bi-directional GRU. At the decoding stage, a single feed-forward neural network is used to compute attention scores of every source segments. A two-layer character-level decoder is applied to predict the target characters.

For the methods which are designed for transliterating word pairs, we apply two word alignment methods to make word pairs from the title pairs.

- **p-align** takes title pairs that contain the same number of words on each side, and aligns the words by their position.

[2] https://github.com/nyu-dl/dl4mt-c2c

That is, the i-th source word is aligned to the i-th word of the target phrase.

- **f-align** applies a word alignment (Dyer, Chahuneau, and Smith 2013) model on the training title pairs to produce word pairs.

At test time, after translating each word in the test phrase, we apply a bigram language model to reorder the predicted words. The language model is trained on all articles in the English Wikipedia.

Name Translation Performance

The first experiment evaluates the performance of each model by a standard transliteration metric: fuzzy F1 of the top-1 prediction. This metric is based on the longest common subsequence between the gold and generated names, and has been used for several years in the NEWS transliteration workshops (Li et al. 2009; 2010; Banchs et al. 2015).

We create training, development, and test title pairs from the inter-language links in Wikipedia. For a test language L, we take all the titles in L's Wikipedia which have a link pointing to the corresponding English page, and then use FreeBase types to classify them into one of the three entity types: person, location, and organization, or discard a title if it is not of any of these types. Since different types of entities may be translated differently, we find that it is better to train a model for each entity type separately. For each entity type, we take at most 10k pairs for training and 5k pairs for both development and test. The numbers of title pairs for each language are shown in the column "#Title Pairs" of Table 2.

The results are listed in Table 3. The last block of rows shows the average performance on all 9 languages. The bold-faced numbers are the highest numbers of each row, and the underlined numbers are the second highest.

For the four transliteration models, using a word alignment model (f-align) to preprocess title pairs is better than aligning words according to their position in the titles (p-align). However, for person names, sometimes the performance of using f-align is worse than of p-align. Since word reordering does not change much in person names across languages, using f-align may create more incorrect training word pairs than p-align.

Liu et al. (2016) report that JANUS performs very well on transliterating between Japanese and English names, but it is not as strong on our dataset. The reason could be that this model is not so robust to noisy input, since it is designed for training on clean word pairs.

For the neural machine translation methods, the full character-based model (NMT-char) performs much better than NMT-bpe which only decodes at the character-level. NMT models tend to perform better on European languages which the models are developed on and have more training pairs. From their low performance on lower-resource languages (TL and BN), it can be concluded that they may need more training data in order to generalize better. We have tried to use a state-of-the-art NMT-char model which is trained on a MT corpus, but the performance is worse than using the model trained on our name pairs.

		#Title Pairs			#Mentions
Lang.	Type	Train	Dev	Test	
ES	LOC	10,000	5,000	5,000	4,953
	ORG	4,120	1,640	2,471	1,311
	PER	10,000	5,000	5,000	2,094
DE	LOC	10,000	5,000	5,000	5,426
	ORG	5,025	2,042	3,048	1,882
	PER	10,000	5,000	5,000	1,876
TR	LOC	7,738	3,084	4,644	2,569
	ORG	1,556	642	962	1,539
	PER	3,646	1,451	2,181	2,011
TL	LOC	1,538	609	903	931
	ORG	132	48	73	117
	PER	845	334	501	282
BN	LOC	3,151	1,262	1,893	2,229
	ORG	634	248	379	337
	PER	3,999	1,597	2,388	1,684
HE	LOC	8,861	3,557	5,000	5,891
	ORG	2,909	1,131	1,747	2,996
	PER	10,000	5,000	5,000	9,768
FR	LOC	10,000	5,000	5,000	5,271
	ORG	6,318	2,517	3,739	1,805
	PER	10,000	5,000	5,000	2,305
IT	LOC	10,000	5,000	5,000	4,405
	ORG	3,861	1,502	2,302	1,423
	PER	10,000	5,000	5,000	2,702
AR	LOC	10,000	5,000	5,000	4,743
	ORG	4,820	1,920	2,894	1,051
	PER	10,000	5,000	5,000	2,796

Table 2: Statistics of the data used in our experiments. ES: Spanish, DE: German, TR: Turkish, TL: Tagalog, BN: Bengali, HE: Hebrew, FR: French, IT: Italian, AR: Arabic.

Our model outperforms all other approaches in most cases, especially on LOC and ORG where word alignment is required. P&R with f-align often gets the second highest numbers, and Sequitur is usually slightly worse than P&R.

Candidate Generation Performance

The fuzzy F1 score in the previous section evaluates string similarity between the predicted name and the gold translation. It does not directly show the ability of retrieving the target English title given a foreign mention. In this experiment, we use the translated English names to generate English title candidates, and evaluate how often a model can produce the correct English title in the candidate set.

Following the way Tsai and Roth (2016b) created a dataset for cross-lingual wikification, we use articles in Wikipedia to make a dataset which only contains named entity mentions. For each anchor text (hyperlinked string) in Wikipedia articles, we get its entity type (or non-entity) based on the FreeBase types of its target title, and only keep the mentions that belong to one of the three types (PER, ORG, and LOC). We use 30,000 articles for Turkish, Tagalog, and Bengali, and 10,000 articles for the other languages.

		DirecTL+		Sequitur		P&R		JANUS		NMT	NMT	Ours
		p-align	f-align	p-align	f-align	p-align	f-align	p-align	f-align	-bpe	-char	
ES	LOC	50.85	61.29	52.28	64.59	52.41	65.76	48.14	55.84	73.24	**73.56**	71.72
	ORG	56.35	62.14	61.57	67.46	61.69	68.23	55.09	60.13	58.75	61.27	**73.51**†
	PER	80.61	80.39	80.37	80.86	81.34	81.61	78.84	77.03	67.87	75.81	**81.85**†
	Avg	63.87	69.12	65.38	71.68	65.85	72.60	61.83	65.19	68.22	72.03	**76.14**†
DE	LOC	57.87	64.07	57.37	63.08	61.33	66.38	49.65	53.64	67.06	**68.81**	68.57
	ORG	62.59	63.65	64.90	66.48	68.49	69.38	55.90	56.79	54.94	58.43	**70.01**†
	PER	78.69	78.83	78.48	79.10	79.79	79.94	76.38	75.52	70.75	79.08	**80.53**†
	Avg	66.95	69.63	67.22	70.01	70.08	72.28	61.36	62.76	65.64	70.32	**73.49**†
TR	LOC	74.22	73.56	79.15	78.12	78.93	78.72	75.61	73.28	68.42	70.87	**80.33**†
	ORG	72.95	72.92	74.88	74.37	75.32	75.23	68.61	65.61	59.33	59.97	**76.34**†
	PER	80.51	80.09	80.03	79.69	81.30	81.42	76.01	75.37	60.10	66.10	**81.59**†
	Avg	75.83	75.31	78.87	78.09	79.15	79.05	74.85	72.92	64.97	68.19	**80.19**†
TL	LOC	64.10	62.17	64.26	65.06	65.76	66.65	57.86	57.25	56.84	59.63	**74.41**†
	ORG	54.79	53.76	57.02	56.25	55.78	56.54	55.81	55.60	55.89	56.49	**71.63**†
	PER	84.24	81.97	83.36	82.66	83.37	82.78	77.30	74.51	61.37	64.09	**85.88**†
	Avg	70.47	68.47	70.38	70.59	71.24	71.62	64.35	63.02	58.33	60.99	**78.16**†
BN	LOC	80.70	79.69	89.69	89.10	89.20	89.15	86.45	83.88	68.26	72.02	**90.02**†
	ORG	75.71	76.38	85.93	85.14	84.93	84.39	79.28	76.86	57.35	57.52	**86.37**
	PER	84.50	85.22	90.49	90.13	89.86	89.74	88.73	88.32	70.43	73.45	**90.87**†
	Avg	82.24	82.26	89.80	89.31	89.19	89.07	87.04	85.59	68.48	71.57	**90.16**†
HE	LOC	66.21	65.38	68.84	69.52	66.71	67.78	64.01	62.71	60.96	61.46	**71.20**†
	ORG	63.12	62.64	64.73	65.03	63.43	65.05	56.98	56.87	54.57	56.22	**68.02**†
	PER	77.61	79.43	**88.08**†	87.88	86.68	86.51	87.60	84.75	77.77	80.00	87.59
	Avg	70.60	70.95	76.42	76.66	74.72	75.35	73.01	71.22	67.17	68.57	**77.70**†
FR	LOC	54.37	59.56	52.22	62.13	57.11	63.96	46.11	54.68	69.79	**71.15**†	70.25
	ORG	58.15	62.46	61.85	67.51	65.64	68.78	54.94	62.80	60.89	65.46	**71.73**†
	PER	81.34	80.73	81.21	81.09	82.22	82.23	77.03	80.54	66.60	73.50	**82.42**†
	Avg	65.21	68.05	65.40	70.49	68.57	71.92	59.77	66.30	66.21	70.46	**75.08**†
IT	LOC	55.37	61.12	55.57	63.90	55.57	64.92	45.88	60.61	72.79	**74.12**†	73.45
	ORG	58.96	62.96	62.45	67.77	63.98	68.65	54.22	59.46	56.61	59.38	**71.09**†
	PER	80.26	80.10	80.06	80.27	81.09	81.16	78.72	78.87	67.54	76.71	**81.80**†
	Avg	66.16	69.18	66.81	71.28	67.51	72.22	60.79	67.82	67.63	72.41	**76.40**†
AR	LOC	65.16	65.18	68.44	68.07	66.86	68.14	63.34	64.51	64.22	66.19	**69.78**†
	ORG	58.34	60.95	64.67	68.19	61.80	66.70	58.50	62.84	57.81	59.86	**69.54**†
	PER	81.03	81.28	87.54	87.35	86.68	86.51	87.25	86.38	76.35	81.21	**87.56**
	Avg	69.78	70.47	75.00	75.57	73.41	74.94	71.53	72.61	67.49	70.59	**76.62**†
Avg	LOC	63.21	65.78	65.31	69.29	65.99	70.16	59.67	62.93	66.84	68.65	**74.41**†
	ORG	62.33	64.21	66.44	68.69	66.78	69.22	59.93	61.88	57.35	59.40	**73.14**†
	PER	80.98	80.89	83.29	83.23	83.59	83.54	80.87	80.14	68.75	74.44	**84.45**†
	Avg	70.12	71.49	72.81	74.85	73.30	75.45	68.28	69.71	66.02	69.46	**78.22**†

Table 3: Wikipedia title translation results. Given a Wikipedia title in a foreign language, we translate it into English using various models. The numbers are fuzzy F1 scores between the top-1 translation and the gold English title. The highest number of each row is bold-faced and the second highest is underlined. A bold-faced number with a dagger indicates the difference between it and the runner-up is statistically significant. We use approximate randomization (Noreen 1989) with p-value < 0.05.

Note that we exclude the mentions which appear in the training pairs, and the mentions which are identical to the target title. For example, if a Spanish mention "Barack Obama" is linked to the English title "Barack_Obama", we will exclude this mention. Since this trivial case will be handled without a name translation model in practice. The number of test mentions for each language is listed in the column "#Mentions" of Table 2.

The title candidate generation algorithm is as follows. We collect all anchor text and its corresponding title from all articles in the English Wikipedia. We then build three dictionaries from this collection. The first one simply maps each anchor text (the entire string) to all possible titles. The second dictionary breaks each anchor text into words and maps each word to all possible titles. The third dictionary further breaks words into character 4-grams and maps each char-

		DirecTL+		Sequitur		P&R		JANUS		NMT	NMT	Ours
		P-align	f-align	p-align	f-align	p-align	f-align	p-align	f-align	-bpe	-char	
ES	LOC	4.93	27.05	20.51	37.03	17.34	61.72	2.40	15.06	26.43	27.86	**65.90**†
	ORG	8.39	22.43	21.28	31.05	34.25	58.05	5.34	13.20	7.40	7.70	**61.25**†
	PER	31.18	31.66	35.24	35.24	63.80	**63.94**	30.28	28.99	11.22	23.26	63.71
	Avg	12.05	27.48	24.32	35.64	31.63	61.70	9.85	18.26	19.63	23.55	**64.62**†
DE	LOC	15.41	39.46	28.92	42.44	40.88	61.21	3.80	17.34	16.51	21.84	**61.65**
	ORG	18.28	24.97	28.53	34.33	49.73	58.13	8.66	13.07	4.41	10.15	**58.55**
	PER	33.53	34.22	34.01	35.18	65.35	**66.36**†	27.93	28.68	11.99	29.90	63.38
	Avg	19.70	35.42	29.88	39.30	47.69	**61.63**	9.72	18.78	13.11	21.09	61.37
TR	LOC	49.94	47.92	51.50	50.60	60.65	61.35	38.07	42.62	21.29	24.45	**63.60**†
	ORG	43.92	41.33	39.18	39.64	56.27	57.76	9.23	10.98	3.96	4.35	**61.99**†
	PER	33.71	33.71	31.92	33.76	62.31	**64.69**	23.92	25.61	6.02	7.26	63.85
	Avg	43.10	41.59	41.97	42.31	60.09	61.55	26.16	29.07	11.91	13.74	**63.28**†
TL	LOC	38.45	30.61	42.53	43.39	68.42	71.00	13.96	17.51	8.27	4.73	**78.09**†
	ORG	0.85	4.27	12.82	8.55	11.97	19.66	0.00	2.56	1.71	1.71	**35.04**†
	PER	32.98	33.69	35.11	33.33	64.89	67.73	15.60	18.44	5.67	6.74	**73.05**†
	Avg	33.98	28.95	38.35	38.20	62.71	65.79	13.08	16.39	7.14	4.89	**73.23**†
BN	LOC	39.03	36.11	64.87	64.87	66.67	66.67	51.77	48.77	18.26	19.25	**68.37**†
	ORG	18.69	17.21	39.76	39.47	54.90	**56.08**	24.93	24.93	9.20	5.64	54.60
	PER	24.58	25.71	39.31	39.61	62.29	61.70	35.04	37.05	15.62	17.46	**63.66**†
	Avg	31.69	30.49	52.75	52.85	64.00	63.86	43.01	42.24	16.49	17.46	**65.41**†
HE	LOC	11.54	10.25	23.60	27.75	27.28	32.54	16.28	17.64	11.19	11.17	**42.32**†
	ORG	6.78	4.91	14.85	18.86	20.09	28.81	7.51	12.45	5.94	5.61	**36.95**†
	PER	6.70	6.82	25.66	26.10	47.85	49.09	23.88	21.71	25.57	25.26	**50.60**†
	Avg	8.24	7.60	23.27	25.46	36.90	40.61	18.85	18.94	17.88	17.65	**45.66**†
FR	LOC	16.35	30.73	24.78	39.50	39.39	63.33	2.49	22.82	20.72	22.75	**64.24**†
	ORG	13.68	27.92	24.60	36.23	48.31	61.94	5.26	21.88	11.41	13.07	**64.10**†
	PER	37.96	37.27	39.13	40.39	65.64	**66.46**	29.20	37.18	10.54	20.87	65.77
	Avg	21.15	31.80	28.27	39.09	47.55	63.83	9.58	26.17	16.43	20.42	**64.59**†
IT	LOC	11.06	27.83	24.36	41.00	25.24	62.29	2.29	32.85	26.88	30.85	**67.51**†
	ORG	24.31	39.63	29.52	43.85	46.31	63.25	4.08	20.31	6.04	4.01	**69.08**†
	PER	36.27	35.31	39.93	39.56	61.40	59.88	33.49	35.05	10.99	25.39	**63.06**†
	Avg	21.25	32.17	30.15	41.02	40.21	61.69	12.47	31.45	18.37	24.64	**66.37**†
AR	LOC	13.92	14.34	18.09	22.35	22.14	27.26	10.58	16.21	7.82	8.92	**31.86**†
	ORG	3.52	5.14	13.42	21.12	21.31	36.16	8.18	16.37	15.13	12.08	**40.72**†
	PER	8.91	9.66	24.50	25.61	42.92	44.81	24.07	23.86	20.71	25.43	**45.06**
	Avg	11.01	11.69	19.60	23.26	28.80	34.06	14.68	18.72	12.91	14.68	**37.24**†
Avg	LOC	22.29	29.37	33.24	40.99	40.89	56.37	15.74	25.65	17.49	19.09	**60.36**†
	ORG	15.38	20.87	24.88	30.34	38.13	48.87	8.13	15.08	7.24	7.15	**53.92**†
	PER	27.31	27.56	33.87	34.31	59.61	60.52	27.05	28.51	13.15	20.17	**60.99**†
	Avg	22.46	27.47	32.06	37.46	46.62	57.19	17.49	24.45	14.87	17.57	**60.12**†

Table 4: Wikipedia title candidate generation experiment. Given a mention, we translate it into English using different models, and then a candidate generation algorithm is applied to the translated names. The numbers indicate percentage of mentions that have the gold English title in the candidate set.

acter 4-gram to all possible titles. In other words, the first dictionary has the highest precision but lowest recall. In contrast, in the third dictionary, each character 4-gram is likely to be mapped to many titles, thus has the highest recall. We sort the titles by $P(\text{title}|\text{key})$ in each dictionary, where "key" is the key of each dictionary (phrases, words, or character 4-grams).

For each mention (translated English name), we will generate at most 30 candidate titles. We query the first dictionary by the entire mention string to retrieve the top 30 titles. If there are less than 30 titles, we then query the second dictionary by each word in the mention. The third dictionary is used in a similar way if the total number of candidates is still less than 30. It is true that generating more than 30 candidates can make the coverage higher for all models. However, as Tsai and Roth (2016a) pointed out, generating too many candidates will result in worse ranking performance in the later steps of the wikification pipeline.

	Spanish		
	Precision	Recall	F1
Base system	56.33	51.33	53.71
+Proposed model	**63.62**	**57.98**	**60.67**†
	Chinese		
Base system	69.95	58.53	63.62
+Proposed model	**72.05**	**60.10**	**65.53**†

Table 5: End-to-end wikification performance on TAC 2016 EDL task. Incorporating the proposed name translation model into the base system improves the overall performance for both languages.

The results are shown in Table 4. The numbers indicate the percentage of mentions which have gold English title in the candidate set. Note that we use the same candidate generation algorithm for all models, and we do not evaluate the ranking performance in this experiment since this problem is not the focus of this paper. To compare different transliteration and translation models, we only want to see if the name translation can help to retrieve the target English title.

We can see that although the relative performance between models is similar to the trend shown in Table 3, the range of the numbers in Table 4 is much wider. This indicates that even if two strings are pretty similar in terms of fuzzy F1 score, they could generate very different sets of candidates. More importantly, a string which has a higher fuzzy F1 score does not always retrieve the correct title. For example, NMT-char gets the best score on Spanish location names in Table 3, but it only ranked the fourth in generating candidates. We notice that NMT-char tend to generate tokens of popular location names in the training pairs. This behavior may not hurt the fuzzy F1 score much, but when generating candidates, it will generate candidates which are totally unrelated to the mention. On the other hand, Sequitur fails to transliterate several tokens of the test mentions, so the predictions tend to be short. Again, although the fuzzy F1 score of Sequitur looks good, it fails to generate the correct title since some key words in the foreign mentions are not translated.

End-to-end Wikification Performance

To evaluate the impact of using the proposed model in a cross-lingual wikification system, we add our name translation model to one of the top systems (Tsai et al. 2016)[3] in the TAC 2016 Entity Discovery and Linking shared task (Ji, Nothman, and Dang 2016) in which the two target languages are Chinese and Spanish. This system simply uses the approach which relies on the inter-language links to generate English title candidates. As discussed in introduction, if the target entity does not exist in the target-language Wikipedia or it is not linked to the corresponding English page, this approach will fail to retrieve the correct title.

We augment this base system in the following way: if the base system does not generate any candidate for a mention, we use our model to translate the mention into English and then query the English title index. Note that the test documents were written after 2011, and many entities were added into Wikipedia after the events happened. To simulate a more challenging and realistic situation, we remove the entities in the target-language Wikipedia which were created after 2011 in this experiment. The results are shown in Table 5. A predicted mention is considered correct if and only if the mention boundary, entity type, and the FreeBase ID (which can be derived from Wikipedia titles) are all identical to a gold mention. We can see that the scores on both languages have improved significantly by incorporating the proposed model. The smaller improvement on Chinese indicates that Chinese-to-English name translation is harder than Spanish-to-English, but the smaller gap is also due to the fact that the naive candidate generation approach works better on Chinese.

Related Work

In addition to the transliteration models introduced in the experimental section, we note that Irvine, Callison-Burch, and Klementiev (2010) mined training word pairs from inter-language links in Wikipedia. Although they only work on person names in which the words can be easily aligned, they conduct careful analysis on 13 languages and show the effect of the amount of training data on transliteration performance. Several works (Tao et al. 2006; Yoon, Kim, and Sproat 2007; Klementiev and Roth 2008; Goldwasser et al. 2009) propose to discover name transliteration from comparable corpora or temporally aligned documents. Although these resources may not be available for low-resource languages, these methods could be used for generating more training phrase pairs for our model. In TAC EDL (Ji, Nothman, and Hachey 2014), several teams tried to mine name translation pairs from comparable corpora in order to improve cross-lingual wikification performance.

Conclusion

We proposed a probabilistic model that learns name translation from Wikipedia titles. Using inter-language links in Wikipedia, we can collect training title pairs for more than 250 languages. The proposed model jointly considers word alignments and word transliteration, and thus has an advantage in learning location and organization names, where words are more likely to be ordered differently across languages. We show that our model outperforms 6 other transliteration and translation models not only on a string similarity metric, but also on the ability to generate title candidates for the cross-lingual wikification problem.

Acknowledgments

This work was supported by Contract HR0011-15-2-0025 with the US Defense Advanced Research Projects Agency (DARPA). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

[3] `https://github.com/CogComp/cross-lingual-wikifier`.

References

Banchs, R. E.; Zhang, M.; Duan, X.; Li, H.; and Kumaran, A. 2015. Report of NEWS 2015 machine transliteration shared task. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, 10.

Bisani, M., and Ney, H. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication* 50(5):434–451.

Chung, J.; Cho, K.; and Bengio, Y. 2016. A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B* 39:1–38.

Dyer, C.; Chahuneau, V.; and Smith, N. A. 2013. A simple, fast, and effective reparameterization of ibm model 2. Association for Computational Linguistics.

Goldwasser, D.; Chang, M.-W.; Tu, Y.; and Roth, D. 2009. Constraint driven transliteration discovery. In Nicolov, N., ed., *Proc. of the Conference on Recent Advances in Natural Language Processing*. John Benjamins.

Irvine, A.; Callison-Burch, C.; and Klementiev, A. 2010. Transliterating from all languages. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

Ji, H.; Nothman, J.; and Dang, H. T. 2016. Overview of TAC-KBP2016 tri-lingual EDL and its impact on end-to-end cold-start KBP. In *Text Analysis Conference (TAC)*.

Ji, H.; Nothman, J.; and Hachey, B. 2014. Overview of TAC-KBP2014 entity discovery and linking tasks. In *Text Analysis Conference (TAC)*.

Jiampojamarn, S.; Cherry, C.; and Kondrak, G. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *ACL*, 905–913.

Jiampojamarn, S.; Kondrak, G.; and Sherif, T. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *NAACL*.

Klementiev, A., and Roth, D. 2008. Named entity transliteration and discovery in multilingual corpora. In Goutte, C.; Cancedda, N.; Dymetman, M.; and Foster, G., eds., *Learning Machine Translation*. MIT Press.

Li, H.; Kumaran, A.; Pervouchine, V.; and Zhang, M. 2009. Report of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, 1–18.

Li, H.; Kumaran, A.; Zhang, M.; and Pervouchine, V. 2010. Report of NEWS 2010 transliteration generation shared task. In *Proceedings of the 2010 Named Entities Workshop*, 1–11.

Liu, L.; Finch, A.; Utiyama, M.; and Sumita, E. 2016. Agreement on target-bidirectional lstms for sequence-to-sequence learning. In *AAAI*.

Noreen, E. W. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Pasternack, J., and Roth, D. 2009. Learning better transliterations. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*.

Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Tao, T.; Yoon, S.-Y.; Fister, A.; Sproat, R.; and Zhai, C. 2006. Unsupervised named entitly transliteration using temporal and phonetic correlation. 250–257.

Tsai, C.-T., and Roth, D. 2016a. Concept grounding to multiple knowledge bases via indirect supervision.

Tsai, C.-T., and Roth, D. 2016b. Cross-lingual wikification using multilingual embeddings. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Tsai, C.-T.; Mayhew, S.; Peng, H.; Sammons, M.; Mangipundi, B.; Reddy, P.; and Roth, D. 2016. Illinois CCG entity discovery and linking, event nugget detection and co-reference, and slot filler validation systems for tac 2016. In *Text Analysis Conference (TAC)*.

Yoon, S.-Y.; Kim, K.-Y.; and Sproat, R. 2007. Multilingual transliteration using feature based phonetic method. 112–119. Prague, Czech Republic: Association for Computational Linguistics.