Mining Multi-word Named Entity Equivalents from Comparable Corpora

Abhijit Bhole
Microsoft Research India
Bangalore, India
v-abbhol@microsoft.com

Goutham Tholpadi
Indian Institute of Science
Bangalore, India
gtholpadi@gmail.com

Raghavendra Udupa
Microsoft Research India
Bangalore, India
raghavu@microsoft.com

Abstract

Named entity (NE) equivalents are useful in many multilingual tasks including MT, transliteration, cross-language IR, etc. Recently, several works have addressed the problem of mining NE equivalents from comparable corpora. These methods usually focus only on single-word NE equivalents whereas, in practice, most NEs are multi-word. In this work, we present a generative model for extracting equivalents of multi-word NEs (MWNEs) from a comparable corpus, given a NE tagger in only one of the languages. We show that our method is highly effective on three language pairs, and provide a detailed error analysis for one of them.

1 Introduction

NEs are important for many applications in natural language processing and information retrieval. In particular, NE equivalents, i.e. the same NE expressed in multiple languages, are used in several cross-language tasks such as machine translation, machine transliteration, cross-language information retrieval, cross-language news aggregation, etc. Recently, the problem of automatically constructing a table of NE equivalents in multiple languages has received considerable attention from the research community. One approach to solving this problem is to leverage the abundantly available comparable corpora in many different languages of the world (Udupa et al., 2008; Udupa et al., 2009a; Udupa et al., 2009b). While considerable progress has been made in improving both recall and precision of mining of NE equivalents from comparable corpora, most approaches in the literature are applicable only to single-word NEs, and particularly to transliterations (e.g. Tendulkar and तेन्डुलकर). In this work, we consider the more general problem of MWNE equivalents from comparable corpora.

In the MWNE equivalents mining problem, a NE in the source language could be related to a NE in the target language by, not just transliteration, but a combination of transliteration, translation, acronyms, deletion/addition of terms, etc. To give an example, Figure 1 shows a pair of comparable articles in English and Hindi. 'Sachin Tendulkar' and 'सचिन तेन्डुलकर' are MWNE equivalents, and both words have been transliterated. Another example is the pair 'Siddhivinayak Temple Trust' and 'सिद्धिविनायक मन्दिर siddhivinayak mandir'. Here, the first word has been transliterated, the second one translated, and the third omitted in Hindi. The task is to (a) identify these MWNEs as equivalents, (b) infer the word correspondence between the MWNE equivalents, and (c) identify the type of correspondence (transliteration, translation, etc.).

Such NE equivalents would not be mined correctly by the previously mentioned approaches as they would mine only the pair (Siddhivinayak, सिद्धिविनायक). In practice, most NEs are multi-word and hence it makes sense to address the problem of mining MWNE equivalents.

To the best of our knowledge, this is the first work on mining MWNEs in a language-neutral manner.

In this work, we make the following contributions:

- We perform an empirical study of MWNE occurrences, and the issues involved in mining (Section 2).
- We define a two-tier generative model for MWNE equivalents in a comparable corpus (Section 4).
- We propose a modified Viterbi algorithm for identifying MWNE equivalents, and

Proceedings of the 2011 Named Entities Workshop, IJCNLP 2011, pages 65–72,
Chiang Mai, Thailand, November 12, 2011.

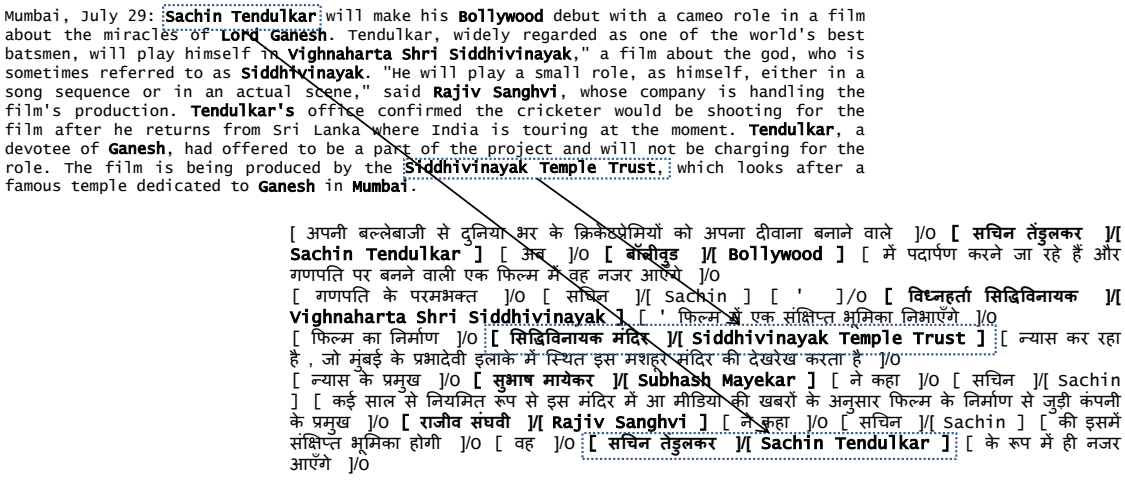

Figure 1: An example of MWNE mining.

for inferring correspondence information (Section 4.3).

- We evaluate the method on three language pairs (involving English (En), Arabic (Ar), Hindi (Hi) and Tamil (Ta)) (Section 6).

In our method, we assume the existence of the following linguistic resources: a NE tagger, a translation model, a transliteration model, and a language model. We show good mining performance for En-Hi and En-Ta. We perform error analysis for En-Ar, and identify sources of error (Section 6.5).

2 Empirical Study of Multi Word NE Equivalents

To understand the various issues in mining MWNE equivalents from comparable corpora, we took a random sample of 100 comparable En-Hi news article pairs from the Indian news portal WebDunia [1]. The English articles had 682 unique NEs of which 252 (37%) were person names, 130 (19%) were location names, and 300 (44%) were organization names. A substantial percentage of the names comprised of more than one word: locations 25%, person names 96%, and organizations 98%. For each English MWNE, we manually identified its equivalent (if any) in the comparable Hindi article. We observed that the MWNEs studied usually conformed to one/some of the following characteristics:

1. Each word in the Hindi MWNE is a transliteration of some word in the English MWNE. E.g. (Mahatma Gandhi, महात्मा गाँधी) where (Mahatma, महात्मा) and (Gandhi, गाँधी) are transliterations.

2. At least one word in the Hindi MWNE is a translation of some word in the English MWNE while the remaining words are transliterations. E.g. (New Delhi, नई दिल्ली `nai dillee`) where (New, नई) is a translation and (Delhi, दिल्ली) is a transliteration.

3. MWNEs contain abbreviations (initials). E.g. (M. K. Gandhi, एम. के. गाँधी) where (M, एम) and (K, के) are initials.

4. One-to-one correspondence between the words in the English and Hindi MWNEs. E.g. (New Delhi, नई दिल्ली)

5. One-to-many correspondence between the words in the English and Hindi MWNEs. E.g. (Card, प्रशस्ती पत्र `prashasti patr`).

6. Many-to-one correspondence between the words in the two MWNEs. E.g. (Air force, वायुसेना `vayusena`).

7. Sequential correspondence between words in the two MWNEs. E.g. (High Court, उच्चतम न्यायालय `ucchatam nyayalay`) where (High, उच्चतम) and (Court, न्यायालय) are equivalents.

8. Non-sequential correspondence between words in the two MWNEs. E.g. (Battle Honour Gurais, गुराइस युद्ध सम्मान `gurais`

[1] http://www.webdunia.com

yuddha sammaan) where the correspondence is (Battle, युद्ध), (Honour, सम्मान) and (Gurais, गुराइस).

9. Some words in the English MWNE do not have an equivalent in the Hindi MWNE. E.g. (Department of Telecommunication, दूरसंचार विभाग doorsanchaar vibhaag) where 'of' does not have an counterpart in the Hindi MWNE.

10. Acronym transliteration by transliterating each character separately. E.g. (IRRC, आईआरआरसी ai aar aar si) and (RBC, आर बी सी aar bi si).

11. Acronym transliteration by transliterating as a whole. E.g. (SAARC, सार्क saark) and (TRAI, ट्राई traai).

Our study revealed that each of the above characteristics is statistically important. Nearly 37% of location names and 77% of organization names involved both transliteration and translation. 12% of person names, 30% of location names and 45% of organization names had either one-to-many or many-to-one correspondence between words. 36% of organization names had non-sequential correspondence between words. These statistics clearly indicate that MWNEs need special treatment and any non-trivial MWNE equivalent mining technique must take into account the characteristics described above.

3 Problem Description

Given a pair of comparable documents in different languages, we wish to extract a set of pairs of MWNEs, one in each language, that are equivalent to each other. We are given a NE tagger in one of the languages, dubbed the *source* language, while the other language is called the *target* language (denoted with subscripts s and t). We are given a document pair (d_s, d_t) and the NEs in d_s i.e. $\{N_i\}_{i=1}^m$ and we want to find all possible NEs in d_t which are equivalent to some N_i. The problem now reduces to finding sequences of words in d_t that are equivalent to some N_i's.

In the example in Figure 1, $\{N_i\}_{i=1}^m$ = {(Sachin, Tendulkar), (Lord, Ganesh), (Siddhivinayak, Temple, Trust), ...}. We want to extract the set { (Sachin Tendulkar, सचिन तेंडुलकर), (Siddhivinayak Temple Trust, सिद्धिविनायक मंदिर), ...}.

4 Mining algorithm

4.1 Key idea

We model the problem of finding NE equivalents in the target sentence T using source NEs as a generative model. Each word t in the target sentence is hypothesized to be either part of a NE, or generated from a target language model (LM). Thus, in the generative model, the source NEs N's plus the target language model constitute the set of hidden states. The t's are the observations. We want to *align* states and observations, i.e. determine which state generated which observation, and choose the alignment that maximizes the probability of the observations. The probability of generating a target word t from a source NE state N is dependent on

- whether N is itself multi-word; if so, each word in N acts as a substate and can generate t.
- the context (the words preceding t in T); note that the length of the context window for t depends on the length of the source NE generating t, and is not a fixed parameter.
- the relationship (transliteration or translation) the state/substate and the target word.[2]

Dynamic programming (DP) approaches are usually used to compute the best alignment, but it fails here as the context size varies for each NE. Hence, we posit the generative model at two levels:

1. A sentence-level generative model (SGeM), where each word in the target sentence is generated either by the target LM or by one of the source NEs.
2. A generative model for the NE (NEGeM), where each word in the target NE is generated by one of the substates of the source NE.

This is illustrated by the example in Figure 2. The portions 'मंगलवार को' and 'के छात्रो ने अपने' of the Hindi sentence is generated by the language model. 'साउथेम्पटन युनिवर्सिटी' is generated by the English NE 'University of Southampton'. Note that without using the language model, 'के' would have been incorrectly aligned with 'of'. Another example is 'एम के

[2]We also use another relationship for letters in acronyms that are transliterated.

गाँधी ...' which is equivalent to the NE "M. K. Gandhi". Here, 'के' is likely to be a part of the NE. The language model not only reduces false positives but also disambiguates NE boundaries.

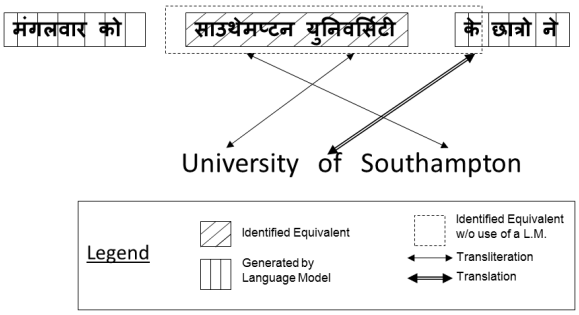

Figure 2: Generation of a Hindi sentence from an English NE.

4.2 Generative Model

SGeM Let $T = t_1 \dots t_n$ be the target sentence and $N = \{N_i\}_{i=0}^{m}$ be the hidden states (as before), where N_0 is the target LM state. In the SGeM, we want to predict the hidden state used to produce the next target term t_i. Let $a_i = j$ if t_i is generated by N_j. We find an alignment $A = a_1 \dots a_n$ which maximizes

$$P(T, A \,|N) = \prod_{i=1}^{n} P\left(a_i \left|a_1^{i-1}, N_{a_i}\right.\right) P\left(t_i \left|t_1^{i-1}, N_{a_i}\right.\right) \quad (1)$$

By choosing which source NE generates each target term, this model also controls the length of the target NE equivalent to a source NE.

Let $t_{k_i} \dots t_{i-1}$ be the *context* for t_i (all these terms are aligned to N_{a_i}). Then

$$P\left(t_i \left|t_1^{i-1}, N_{a_i}\right.\right) = P\left(t_i \left|t_{k_i}^{i-1}, N_{a_i}\right.\right)$$

NEGeM To model the generation of the target term t_i given the context t_1^{i-1} and the substates of the source NE N_j, we let $N_j = \left(n_{j1}, \dots, n_{jL_j}\right)$ where n_{jp} is a substate. The *internal* alignment $B = b_{k_i}, \dots, b_i$ is defined such that $b_p = s$ if t_p is generated by n_{js}. We get

$$P\left(t_i \left| t_{k_i}^{i-1}, N_{a_i}\right.\right) = \sum_{B} \prod_{p=k_i}^{i} P\left(b_p \left|b_{p+1}^{i}\right.\right) P\left(t_p \left|n_{a_i b_p}\right.\right) \quad (2)$$

To model the relationship between the source and target terms, we introduce variables in a fashion similar to the introduction of B in (2). Let $R = r_{k_i}, \dots, r_i$ where $r_p \in \{$transliteration, translation, acronym, none$\}$ such that t_p and n_{jb_p} have the relationship r_p. Then [3]

$$\begin{aligned} & P\left(t_j \left|n_{a_i b_j}, r_j\right.\right) \\ &= m_{tlat} P_{tlat}\left(t_j \left|n_{a_i b_j}\right.\right)^{r_{tlat}} && \text{if } r_j = \text{translation} \\ &= m_{tlit} P_{tlit}\left(t_j \left|n_{a_i b_j}\right.\right)^{r_{tlit}} && \text{if } r_j = \text{transliteration} \\ &= \delta\left[t_j \equiv n_{a_i b_j}\right] && \text{if } r_j = \text{acronym} \\ &= P_{lm}\left(t_j\right) && \text{if } r_j = \text{none} \end{aligned}$$

The four probability terms on the right are obtained, respectively, from a translation model, a transliteration model [4], an acronym model [5], and a language model.

Controlling target NE length In the SGeM, $P\left(a_i \left|a_1^{i-1}, N_{a_i}\right.\right)$ is the probability that N_{a_i} will generate t_i. To compute this, we first note that, for a given term t_i, either $a_i = a_{i+1}$ i.e. N_{a_i} continues to generate beyond t_i, or $a_i \neq a_{i+1}$ i.e. N_{a_i} terminates at t_i. The probability of continuation depends on the length L of N_{a_i} and the length l of the target NE generated so far by N_{a_i}. Based on empirical observations, we defined a function $f(l, L)$ as

$$\begin{aligned} f(l, L) &= 0 \text{ for } l \notin \{L-2, L+2\} \\ &= 1 - \epsilon \text{ for } l \in \{L-1, L\} \\ &= \epsilon \text{ for } l \in \{L+1, L+2\} \end{aligned}$$

where $f\left(l, L\right)$ is the probability of continuation, and $1 - f\left(l, L\right)$ is the probability of termination. ϵ is a very small number. We now define

$$\begin{aligned} & P\left(a_i \left|a_1^{i-1}, N\right.\right) \\ &= p_{NE} && \text{if } a_{i-1} = 0 \\ &= f\left(i - k_i, l_{a_i}\right) && \text{if } a_{i-1} \neq 0, k_i < i \\ &= 1 - f\left(i - k_{i-1}, l_{a_{i-1}}\right) && \text{if } a_{i-1} \neq 0, k_i = i \end{aligned}$$

where the probabilities on the right are for beginning an NE, continuing an NE, and terminating a previous NE, respectively.

[3] $\delta\left[\mathbf{x}\right] = 1$ if condition $\mathbf{x}$ is true

[4] A character-level extended HMM described in (Udupa et al., 2009a).

[5] A mapping from source language alphabets to target language transliterations of the alphabets.

4.3 Modified Viterbi algorithm

We use the dynamic programming framework to do the maximization in (1). For each target term t_i, for each source NE N_j, the subproblem is to find the best alignment $a_1 \ldots a_i$ such that $a_{i+1} \neq a_i$ i.e. t_i is the last term in the equivalent of N_j.

$$\texttt{subproblem}\,[i,j] = \max_{a_1^i} P\left(a_i = j \neq a_{i+1} \left| a_1^{i-1}, N_j\right.\right) P\left(t_i \left| t_1^{i-1}, N_j\right.\right)$$

Let l be the length of the target NE ending at t_i, based on the alignment so far. The first probability term becomes

$$P\left(a_{i-l-1} \neq a_{i-l}^{i} = j \neq a_{i+1} \left| N_j\right.\right) = \alpha \times f\left(l, L_j\right)\left(1 - f\left(l+1, L_j\right)\right)$$

This is non-zero only for certain values of l, for which we can construct the solution to $\texttt{subproblem}\,[i,j]$ using solutions for $i = l$. Denote $k = i - l$, then

$$\texttt{subproblem}\,[i,j] = \max_{j \neq i} \texttt{subproblem}\,[k-1, j] \times \texttt{negem}\left(t_k^p, N_i\right)$$

where the procedure `negem` computes the probability that a given sequence of target words is an equivalent of the given source NE. This procedure solves a second (independent) DP problem (for the NEGeM), constructed in a similar fashion. It also models conditions such as "If a target term is a transliteration, it cannot map to more than one source substate."

The output of the system is a set of MWNE pairs. For each pair, we also give the internal alignment between the words of the two NEs.

5 Parameter Tuning

The MWNE model has five user-set parameters. These need to be tuned appropriately in order to be able to compare probabilities from different models. In the following, we describe the parameters and a systematic way to go about tuning them.

- $p_{NE} \in (0, +\infty)$ specifies how likely are we to find an NE in a target sentence
- Given a probability p returned by the transliteration model, the probability value used for comparisons p'_{tlit} is calculated as $p'_{tlit} = m_{tlit}\ p^{r_{tlit}}$ where $r_{tlit} \in R$, $m_{tlit} \in (0, +\infty)$. r_{tlit} is tuned to boost/suppress p; m_{tlit} is also used similarly, but to get more fine-grained control.
- Similarly, for a probability p given by the translation model, we calculate $p'_{tlat} = m_{tlat}\ p^{r_{tlat}}$ where $r_{tlat} \in R$, $m_{tlat} \in (0, +\infty)$

In our experiments, we found that transliteration probabilities were quite low compared to the others, followed by the translation probabilities. So, we used the following procedure to tune these parameters use a small hand-annotated set of document pairs.

1. Initially set $p_{NE} = +\infty$, and all other parameters to zero.
2. Tune r_{tlit} to find as many of the transliterations as possible. Then, use m_{tlit} to fine-tune it to improve precision without losing too much on recall.
3. Next, tune r_{tlat} to find as many of the translations as possible. Then, use m_{tlat} to fine-tune it to improve precision without losing too much on recall.
4. The system is now finding as many NEs as possible, but it is also finding noise. Keep lowering p_{NE} to allow the language model LM to absorb more and more noise. Do this until NEs also begin to get absorbed by LM.

6 Empirical Evaluation

In this section, we study the overall precision and recall of our algorithm for three different language pairs. English (En) is the source language, and Hindi (Hi), Tamil (Ta) and Arabic (Ar) are the target languages. Hindi belongs to the Indo-Aryan family, Tamil belongs to Dravidian family, and Arabic belongs to the Semitic family of languages. The results show that the method is applicable for a wide spectrum of languages.

6.1 Linguistic Resources

Models We need four models (translation, transliteration, language, and acronym) in order to run the proposed algorithm. For a language pair, we learnt these models using the following kinds of data, which was available to us:

- A set of pairs of NEs that are transliterations, to train the transliteration model
- A set of parallel sentences, to learn a translation model

Lang. pairs	Translit. pairs	Word pairs	Monolin. corpus
En-Hi	15K	634K	23M words
En-Ta	17K	509K	27M words
En-Ar	30K	8.2M	47M words

(1K = 1 thousand, 1M = 1 million)

Table 1: Training data for the models.

- A monolingual corpus in the target language, to train a language model
- A dictionary mapping English alphabets to their transliterations in the target language.

One can get an idea of the scale of linguistic resources used by looking at Table 1.

Source language NER The Stanford NER tool (Finkel et al., 2005) was used for obtaining a list of English NEs from the source document.

6.2 Corpus for MWNE mining

For each language pair, a set of comparable article pairs is required. The article pairs each for En-Hi and En-Ta were obtained from news websites[6], where the article correspondence was obtained using a method described in (Udupa et al., 2009b). En-Ar article pairs were extracted from Wikipedia using inter-language links.

Preprocessing The Stanford NER tags each word in the source document as a person, location, organization or other. A continuous sequence of identical tags was treated as a single MWNE. Completely capitalized NEs were treated as acronyms. For each acronym (e.g. "FIFA"), both the acronym version ("FIFA") as well as the abbreviation version ("F I F A") were included in the list of source NEs. Each target document was sentence-separated and tokenized using simple rules based on the presence of newlines, punctuation, and blank spaces. If a word can be constructed by concatenating strings from the acronym model, it is treated as an acronym, and the acronym strings are separated out (e.g. 'एमके' `emke` is changed to 'एम के' `em ke`).

6.3 Experimental Setup

Annotation Given an article pair, a human annotator looks through the list of source NEs, and identifies transliterations in the target document. For MWNEs, the annotator also marks which word in the source corresponds to each word in the target MWNE. This constitutes gold standard data that can be used to measure performance. 120 article pairs were annotated for En-Hi, 120 for En-Ta, and 36 for En-Ar.

Evaluation The NEs mined from one article pair are compared with the gold standard for that pair, and one of three possible judgements is made:

- Fully matched (if it fully matches some annotated NE (both source and target)).
- Partially matched (if source NEs match, and the mined target NE is a subset of the gold target NE).
- Incorrect match (in all other cases).

The algorithm is agnostic of the type of the NE (*Person*, *Organization*, etc.). So, reporting the precision and recall for each NE type does not provide much insight into the performance of the method. Instead, we report at different levels of match—full or partial, and for different categories of MWNEs—single word transliteration equivalents (SW), multi word transliteration equivalents (including acronyms) (MW-Translit) and multi word NEs having at least one translation equivalent (MW-Mixed). We compute the numbers for each article pair and then average over all pairs.

Parameter Tuning Parameter tuning was done following the procedure described in Section 5. For En-Hi and En-Ta, the following values were used: $p_{NE} = 1$, $m_{tlit} = 100$, $r_{tlit} = 7$, $m_{tlat} = 1$, $r_{tlat} = 1$. For En-Ar, $m_{tlit} = 1$, $r_{tlit} = 14$ was used, the other parameters remaining the same. For the tuning exercise, 40 annotated article pairs were used for En-Hi, 40 pairs for En-Ta, and 26 pairs for En-Ar.

6.4 Results and Analysis

We evaluated the algorithm on 80 article pairs for En-Hi, 80 pairs for En-Ta, and 11 pairs for En-Ar. The results are given in Table 2.

We observe that the results for both types of precision (and recall) are nearly identical. This is so because, in most cases, the system is able to mine the entire NE. This validates our intuition of using

[6]En-Hi from *Webdunia*, En-Ta from *The New Indian Express*.

Lang Pair	Prec. (full)	Prec. (part.)	Recall (full)	Recall (part.)
En-Hi	0.84	0.86	0.89	0.89
En-Ta	0.78	0.80	0.61	0.63
En-Ar	0.42	0.44	0.63	0.66
En-Ar*	0.43	0.44	0.60	0.62

* including the data used for tuning

Table 2: Precision and recall of the system

Category	En-Hi	En-Ta	En-Ar
SW	0.90	0.82	0.69
MW - Translit	0.91	0.64	0.63
MW - Mixed	0.77	0.40	0.66

Table 3: Category-wise recall of the system

language models to disambiguate NE boundaries. (The false negatives are mostly due to limitations of transliteration model and the dictionary.) The precision is relatively low in Arabic, even when we include the tuning data. This suggests that the problem is not because of incorrect parameter values. The error analysis for Arabic is discussed in Section 6.5.

We also report recall of the system for various categories of NEs in Table 3.[7] Note that the MW cases and the SW case are mutually exclusive.

6.5 Error Analysis for Arabic

The system performed relatively poorly in Arabic than in the other languages. Detailed error analysis revealed the following sources of error.

Source NER The text of the English articles automatically extracted from Wikipedia was not very clean, as compared to the newswire text used for En-Hi and En-Ta. As a result, the source NER wrongly identified many words as NEs, which were mapped to words on the target side, affecting precision. E.g. words such as "best", "foxe" were marked as NEs, and words with similar meaning or sound were found in the target. But since the annotator had ignored these words, the evaluation marked them as false positives.

Translation model Many words were ignored by the translation model because of the presence of diacritics, or affixes (e.g. 'ال' `al` in Arabic is frequently prefixed to words; also, in Arabic, different sources of text may have different levels of diacritization for the same words). E.g. The target document contained الجمهوريه `al-jamhooriyah` "republic"; the dictionary contained الجمهوريات `al-jamhooriyat`, which has a different suffix, and hence was not found.

Transliteration model The non-uniform usage of diacritics and affixes (across training and test data) as mentioned above affected the performance of transliteration too. E.g. The model is trained on data where the 'ال' prefix usually occurs in the Arabic NE, but not in the English NE. As a result, it maps the 'new' in 'new york' to النيو `al-nyoo`. The annotator had mapped 'new' to نيو `nyoo` (i.e. without the prefix), causing the evaluation program to mark the system's output as a false positive.

Generative Model Some errors occurred due to deficiencies in the generative model. The model requires every word in the source NE to be mapped to a unique word in the target NE. This causes problems when there are function words in the source NE, or when two source words are mapped to the same target word. E.g. 'yale school of management' corresponds to the 3-word NE 'الاداره مدرسه ييل' where 'of' has no Arabic counterpart. 'al azhar' corresponds to the single word الازهر `al-azhar`(which can be split as ال ازهر `al azhar`, but is never done in practice).

7 Related work

Automatic learning of translation lexicons has been studied in many works. Pirkola et al. (Pirkola et al., 2003) suggest learning transformation rules from dictionaries and applying the rules to find cross lingual spelling variants. Several works (Fung, 1995; Al-Onaizan and Knight, 2001; Koehn and Knight, 2002; Rapp, 1999) suggest approaches to learn translation lexicons from monolingual corpora. Apart from single word approaches, some works (Munteanu and Marcu, 2006; Chris Quirk, 2007) focus on mining parallel sentences and fragments from 'near parallel' corpora.

On the other hand, out-of-vocabulary words are transliterated to the target language. Approaches have been suggested for automatically learning transliteration equivalents. Klementiev et al. (Klementiev and Roth, 2006) proposed the use of similarity of temporal distributions for identifying NEs

[7] Since we cannot determine the category of false positives, we do not report the precision here.

from comparable corpora. Tao et al. (Tao et al., 2006) used phonetic mappings for mining NEs from comparable corpora, but their approach requires language specific knowledge which limits it to specific languages. Udupa et al. (Udupa et al., 2008; Udupa et al., 2009b) proposed a language-independent mining technique for mining single-word NE transliteration equivalents from comparable corpora. In this work, we extend this approach for mining NE equivalents from comparable corpora.

8 Conclusion

Through an empirical study, we motivated the importance and non-triviality of mining multi-word NE equivalents in comparable corpora. We proposed a two-tier generative model for mining such equivalents, which is independent of the length of NE. We developed a variant of the Viterbi algorithm for finding the best alignment in our generative model. We evaluated our approach for three language pairs, and discussed the error analysis for English-Arabic.

Currently, unigram approaches are popular for most tasks in NLP, CLIR, MT, topic modeling, etc. tasks. Phrase-based approaches are limited by their efficiency and complexity, and also show limited improvement. We hope that this work will motivate researchers to explore principled methods that make use of NE phrases to significantly improve the state-of-the-art in these areas. The two-tier generative model is applicable to any problem where the context of an observed variable does not depend on a fixed number of past observed variables.

References

Yaser Al-Onaizan and Kevin Knight. 2001. Translating named entities using monolingual and bilingual resources. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 400–408, Morristown, NJ, USA. Association for Computational Linguistics.

Arul Menezes Chris Quirk, Raghavendra Udupa U. 2007. Generative models of noisy translations with applications to parallel fragments extraction. In *MT Summit XI*, pages 377–284. European Association for Machine Translation.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA. Association for Computational Linguistics.

Pascale Fung. 1995. A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In *IN PROCEEDINGS OF THE 33RD ANNUAL CONFERENCE OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, pages 236–243.

Alexandre Klementiev and Dan Roth. 2006. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 817–824, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, pages 9–16, Morristown, NJ, USA. Association for Computational Linguistics.

Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 81–88, Morristown, NJ, USA. Association for Computational Linguistics.

Ari Pirkola, Jarmo Toivonen, Heikki Keskustalo, Kari Visala, and Kalervo Järvelin. 2003. Fuzzy translation of cross-lingual spelling variants. In *SIGIR '03*, pages 345–352, New York, NY, USA. ACM.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *ACL*, pages 519–526.

Tao Tao, Su youn Yoon, Andrew Fister, Richard Sproat, and Chengxiang Zhai. 2006. Unsupervised named entity transliteration using temporal and phonetic correlation.

Raghavendra Udupa, K. Saravanan, A. Kumaran, and Jagadeesh Jagarlamudi. 2008. Mining named entity transliteration equivalents from comparable corpora. In *CIKM '08*, pages 1423–1424. ACM.

Raghavendra Udupa, K. Saravanan, Anton Bakalov, and Abhijit Bhole. 2009a. "they are out there, if you know where to look": Mining transliterations of oov query terms for cross-language information retrieval. In *ECIR*, volume 5478, pages 437–448. Springer.

Raghavendra Udupa, K. Saravanan, A. Kumaran, and Jagadeesh Jagarlamudi. 2009b. Mint: A method for effective and scalable mining of named entity transliterations from large comparable corpora. In *EACL*, pages 799–807.