Improved Named Entity Recognition using Machine Translation-based Cross-lingual Information

Sandipan Dandapat[1], Andy Way[2]

[1] Microsoft,
India

[2] ADAPT Centre, Dublin City University,
Ireland

sadandap@microsoft.com, away@computing.dcu.ie

Abstract. In this paper, we describe a technique to improve named entity recognition in a resource-poor language (Hindi) by using cross-lingual information. We use an on-line machine translation system and a separate word alignment phase to find the projection of each Hindi word into the translated English sentence. We estimate the cross-lingual features using an English named entity recognizer and the alignment information. We use these cross-lingual features in a support vector machine-based classifier. The use of cross-lingual features improves F_1 score by 2.1 points absolute (2.9% relative) over a good-performing baseline model.

Keywords. Named entity recognition, machine translation, cross-lingual information.

1 Introduction

Named Entity Recognition (NER) is an essential task for natural language understanding to identify the names in a given sentence. A Named Entity (NE) primarily refers to the name of a person, location or organization, but sometimes a larger set of names have to be considered. The set of names used in NER is often considered as the NE tagset. In sum, NER is a multi-class classification problem.

A lot of work has been done in the area of NER [23].[1] Researchers primarily use machine learning-based techniques to address the NE classification task. Almost all the work in this area of research requires a substantial amount of linguistic expertise. The linguistic information is required either to produce linguistic rules for a rule-based system or to produce NE-annotated data to train a statistical model.

The performance of a machine learning-based NER system depends on the amount of data used to train the system and the features used to build the model. Some languages of the world have large amounts of annotated data to train a reasonably good NER system. However, there remain a number of languages which suffer from the scarcity of large NE-annotated data. In fact, training data for NER only exists for restricted combination of domains and genres (e.g. written news) even for the most resource-rich languages.

In this work, we use information from a resource-rich language (English) to improve the NER task of a relatively less-resourced language (Hindi). Although a large amount of NE-annotated data is not always readily available for a language, a large amount of parallel data may exist between that language and English to obtain cross-lingual information without needing to avail of linguistic expertise. If such parallel text is unavailable, a large number of third party freely available MT systems might be found between the less-resourced language and English. For example, Google Translate[2] and Bing translator[3] includes 8100 and 2704 possible source–target

[1] http://www.clips.ua.ac.be/conll2003/ner/

[2] https://translate.google.com

[3] https://www.bing.com/translator/

Computación y Sistemas, Vol. 20, No. 3, 2016, pp. 495–504
ISSN 1405-5546
doi: 10.13053/CyS-20-3-2468

translation systems, respectively. In our work, first we adopt Google Translate to translate the Hindi NE-annotated text into English. Furthermore, we use an English language NE recognizer to identify different NE tags in the translated English text. English NER has a very high accuracy [12]. We incorporate English NER information into different features of the source Hindi word using alignment information. Finally, we use these cross-lingual features along with monolingual features to build our NER model.

The rest of the paper is organized as follows. The next section presents related research in the area. Section 3 details our particular approach. Section 4 describes the cross-lingual feature extraction process with an illustrative example. Section 5 presents the experimental set-up, the data and the results obtained from the different experiments conducted. Section 6 presents our observations along with an error analysis. We conclude in Section 7 with some avenues for future work.

2 Related Work

Prior work on NER mostly use either a rule-based [14] or a machine learning (ML) approach [4, 5, 18, 27, 11], with the ML-based approach being by far the most prevalent of the two. A wide range of ML techniques are used for NER of which Hidden Markov Model (HMM) [4], Maximum Entropy (MaxEnt) [5], Conditional Random Field (CRF) [18] and Support Vector Machines (SVM) [11] are quite popular. Researchers have also applied hybrid approaches for the NER task [27]. The ML-based techniques primarily rely on the NE-annotated text as its main knowledge-base. However, researchers often use additional source of knowledge such as *gazetteer lists* or grammatical information within a ML technique [4, 5, 27].

More recently, the focus of NER has shifted to multilingual NER. Richman and Schone [24] proposed a technique to build large multilingual NE-annotated data from Wikipedia using the underlying multilingual characteristics. Researchers also have been using parallel data to improve NER systems. Developing annotated data (NE, part-of-speech (POS) etc.) involves a lot of time, money and other resources. In contrast, parallel data may be available for many language pairs due to the rapid growth of multilingual content on the web. Yarowsky et al. [30] used bilingual text corpora and English text analysis tools for automatic NE-tagging in a foreign language. Kim et al. [17] used a combination of Wikipedia metadata and English–foreign language parallel Wikipedia sentences to produce NE-labelled multilingual data. Parallel data has also been used to improve monolingual natural language processing (NLP) models [7] or to improve models for both languages simultaneously [6]. Parallel data has also been used in unsupervised NLP models using a projection from the resource-rich language to the resource-poor language [9, 29].

Resource-poor languages may not have publicly available parallel data (between the resource-poor and a resource-rich language) to help in NLP tasks. Thus instead of using parallel data, we use MT systems to translate the resource-poor language into a resource-rich language sentence in order to use the information from the resource-rich language [26]. Note that compared to (say) European language pairs, MT is still in its infancy and the quality is still poor for the language pair English-to-Hindi. Thus we are projecting information from noisy parallel data to try to improve NER performance.

Basic NLP tools are often used to improve translation quality [28, 15]. NER is used within an MT framework to improve the MT system by transliterating the names or by using a fixed translation for the names [1, 16]. Significant research work was carried out to improve MT quality using NER. However, very little work has been done in the reverse direction, i.e. to improve NER using MT.

Shah et al. [26] used machine-translated data to develop an NER system (SYNERGY) for Swahali and Arabic. They use an online MT system to translate the Swahali text into English, and English NER to find list of NEs in English. Furthermore, different alignment techniques were used to map Swahali words to the English NEs. Our approach is similar to their work with the following differences: (i) SYNERGY uses only two NE classes (*name*

and *not name*) while we use 15 different NE classes, and (ii) we use translated text to adopt cross-lingual features into a classification problem, while SYNERGY uses purely projection-based techniques to build an NER system.

A significant amount of work has been done previously on NER for Hindi. Hindi is the main language spoken in India, and the fourth most commonly spoken language in the world. Most of this research uses machine learning-based techniques and different monolingual features to build an NER system [11, 25]. Some recent work has developed an NER system using customizable rules automatically created via rule induction [21]. However, no work has ever used cross-lingual features using either parallel data or an MT system to reduce the data sparsity problem of Hindi. Recently conducted NLP tool contests[4] on NER report very low accuracy for Hindi NER using 15 NE classes, with the wining team achieving an accuracy of just 77.4%.

3 Our Approach

The NER task can be formally defined as follows: given a sentence $S = w_1 \dots w_n$, we want to find the possible NE tag t_i for each word w_i in S. The NE tag for a particular word w_i is assigned from a predefined NE tagset T. Thus, NER can be considered as a classification problem or a sequence-labelling problem. We use an SVM model [8] to build our NER system. SVM is a discriminative model of learning which uses both positive and negative examples to learn the distinction between two classes. Like all other discriminative approaches, an SVM model also uses feature vectors for each training instance to learn the classifier. In our approach, we use the YamCha[5] toolkit to train the model and to classify new instances. We used TinySVM[6] within YamCha for NER training and classification. In this paper, we do not aim to explore the best configuration of the SVM classifier; rather we explore how an MT system can be used to improve state-of-the-art NER systems.

[4] http://ltrc.iiit.ac.in/icon/2013/nlptools/

[5] http://chasen.org/~taku/software/yamcha/

[6] http://cl.naist.jp/~taku-ku/software/TinySVM/

3.1 System Architecture

In our system, we use both monolingual and cross-lingual features to build the SVM model. Monolingual features are estimated from the NE-annotated data (cf. Section 3.2). Central to our approach is the *Cross-lingual Feature Estimator*, as shown in Figure 1. We use Google Translate, the Stanford English NER toolkit[7] [12] and an unsupervised word aligner GIZA++ [22] to estimate the cross-lingual features. First, we extract the raw Hindi text (H_R) from the Hindi NE-annotated data (H). Google Translate is used to translate the Hindi text H_R into English (E). The unsupervised word aligner GIZA++ takes both the corpus H_R and E, and produces an alignment ($a : i \rightarrow j$) between each pair of sentences: the Hindi sentence $h \in H_R$ and its translation $e \in E$. The alignment function $a : i \rightarrow j$ indicates that the i-th word of the Hindi sentence h maps to the j-th word of the English sentence e. Note that one word in h may map to multiple words in e. Furthermore, we use the Stanford English NER toolkit to estimate the NE tag for every word in the English translated text E. After obtaining the alignment between h and e, and the NE-annotation of e for all Hindi sentences in the corpus (H), we estimate the cross-lingual features for each Hindi word in H. We illustrate the process with a running example in Section 4.

3.2 Monolingual Features

We use state-of-the-art monolingual features which are often used for Hindi NER [25] including both static and dynamic features. The static features include information from words and POS context. The static features also include prefix and suffix information for all words. The term prefix/suffix is a sequence of the first/last few characters of a word, which does not necessarily imply a linguistically meaningful prefix and suffix. The dynamic features include the NE tags of the previous two words. Table 1 lists all the features used in our SVM model. A combination of these features is used to conduct two baseline experiments for the NER task.

[7] http://nlp.stanford.edu/software/CRF-NER.shtml

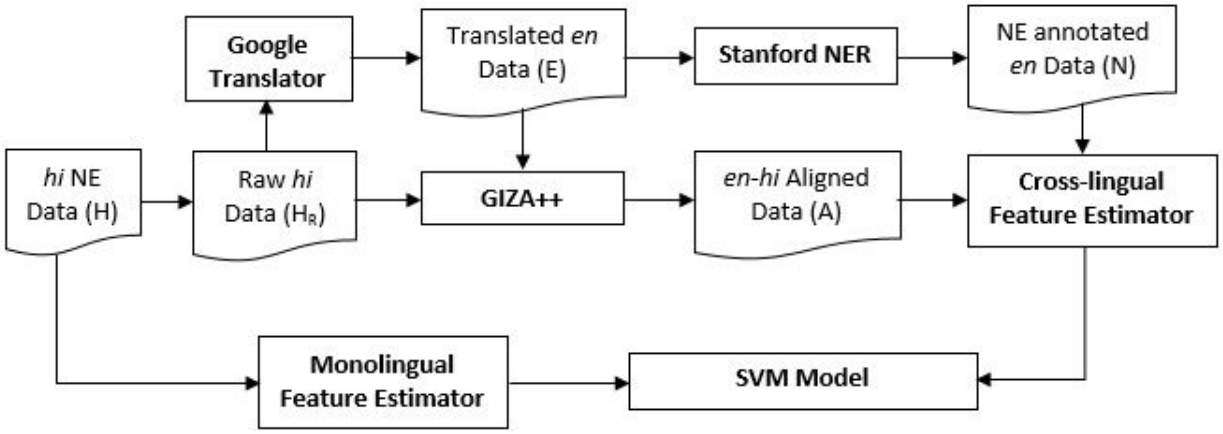

Fig. 1. System Architecture of the NER System. *hi*: Hindi and *en*: English

Static Features	
Type	Features
Word	w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}
POS	p_i, p_{i-1}, p_{i-2}
Affixes	$\|pref\| \leq 4$, $\|suff\| \leq 4$
Dynamic Features	
NE-tag	t_{i-1}, t_{i-2}

Table 1. Monolingual Features Used for NER

3.3 Cross-lingual Features

We use cross-lingual features along with monolingual features to improve the NER task. The cross-lingual features are extracted from a resource-rich language for which we already have a reasonably good NER system. In our case we consider English as the resource-rich language. In our approach, we assume the availability of an MT system from the language of interest into the resource-rich language. We adopt the Google Hindi-to-English MT system.

It is important to note that the correctness of the cross-lingual features largely depends on the translation quality of the MT system. We could not conduct the automatic evaluation to estimate the translation quality for our particular data as we do not have reference translation for the NE annotated corpus, so we carried out a small human evaluation. While manually evaluating the MT systems, we assign values from two five-point scales representing *fluency* and *adequacy* [20]. We performed a manual evaluation of randomly selected 100 sentences of the Hindi-to-English MT output by 2 evaluators. The average fluency and adequacy for the Hindi-to-English MT output are 2.69 and 2.73, respectively (inter annotator agreement [13] of 0.51 and 0.46, respectively). This indicate the overall translation quality is still in infancy for Hindi-to-English MT however, much of the meaning is conveyed by the MT system [20].

During cross-lingual feature extraction, we try to find whether the translation of a Hindi word belongs to a particular NE in the resource-rich language. Note that a Hindi word may correspond to several words in English as in example (1). Thus we consider cross-lingual features as a vector of integers(=count) to accumulate cues from English. If the translation of the Hindi word belongs to a particular NE then that information is projected into the feature vector. It is likely that NEs remain in the same class across languages. The main issue is that the aligner (GIZA++) may not find the correct alignment. Thus, cross-lingual projections are used as features where otherwise English NEs could have been used as NE tags for the Hindi words; indeed, in Section 5 we use such a model to demonstrate indicative performance.

Another issue is that the number of tags may differ between two languages. Our cross-lingual features use the number of NE tags available in the

Algorithm 1 Cross-lingual feature extraction algorithm

Require: H = List of Hindi NE-annotated sentences
Ensure: Cross-lingual feature set
$F = \langle$ Is Person?, Is Location?, Is Organization?, Other NE? $\rangle$
1: $E \leftarrow \mathrm{GoogleTranslator}(H_R)$ //H_R is the list of raw Hindi sentences
2: $A \leftarrow$ align H_R and E using GIZA++
3: **for all** $h \in H_R$ and $a : \{h, e\} \in A$ **do**
4: $N \leftarrow \mathrm{EnglishNER}(e)$
5: **for all** $w_i \in h$ **do**
6: $F_{w_i} = \langle 0, 0, 0, 0 \rangle$
7: Find English words $T(= \{w_j\} \in e)$ based on alignment function $a : i \rightarrow j$
8: **for all** $w_j \in T$ **do**
9: Update F_{w_i} based on the NE tag of $w_j \in N$
//Add 1 if the NE tag of w_j matches with any of the tags in the feature vector
10: **end for**
11: return F_{w_i}
12: **end for**
13: **end for**

resource-rich language regardless of the number of tags available in the Hindi NE-annotated data, i.e. the number of features is equal to the number of tags available in English. We use two variants of the Stanford NE recognizer which uses 4 and 7 NE classes and accordingly generates 4 and 7 cross-lingual features in our system, respectively. The detail of our cross-lingual feature extraction process is given in Algorithm 1 when using 4 cross-lingual features.

Lines 1-2 of the algorithm translate raw Hindi sentences from the NE-tagged data into English (E) using Google Translator and aligns H_R with E. In line 4, we estimate the NE-tags for an English sentence e. In steps 5-7, we find the English words that map to a source Hindi word and initialize the feature vector to all 0s. In steps 8-10, we update the feature vector based on the NE tags associated with the mapped English words using the OR operation (in line 9). This is to ensure that if any of the mapped English words (in case of multiple words aligned to a single Hindi word) indicate an NE tag, we consider that the Hindi word is likely to belong to the same NE category.

4 An Illustrative Example

We describe below the cross-lingual feature extraction process with a running example from our corpus. Consider the Hindi NE-tagged sentence from the annotated corpus in (1a). All the words are represented in word/POS-tag/NE-tag format. Expansion of POS tags can be found in [3].

The Hindi raw sentence from (1a) is translated into English in (1b) and aligned in (1c). Note that (1b) is a machine-translated sentence.

(1) a. अनुष्का/NN/B-PERSON को/PSP/O-NE खासतौर/NN/O-NE पर/PSP/O-NE ब्राजील/NNP/B-LOCATION बहुत/QF/O-NE पसंद/NN/O-NE है/VAUX/O-NE ./SYM/O-NE

b. e: Anushka is very much like particularly Brazil .

c. h: अनुष्का ({ 1 }) को ({ }) खासतौर ({ 6 }) पर ({ }) ब्राजील ({ 4 5 7 }) बहुत ({ 3 }) पसंद ({ }) है ({ 2 }) . ({ 8 })

The Hindi sentence in (1c) is listed word by word with reference to the aligned English word(s) in e. For example, the word 'अनुष्का ({ 1 })' is aligned to the first English word *Anushka*, the word 'को ({ })' is not mapped to any English word and 'ब्राजील ({ 4 5 7 })' is mapped to three English words {*much*{4}, *like*{5} and *Brazil*{7}}.

The NE-tagged output using the Stanford tagger is shown in (2) for the translated English sentence in (1b). Example (2) represents the sentence with word/NE-tag format where 'O' indicates *not a name*.

(2) N: Anushka/PERSON is/O very/O much/O like/O particularly/O Brazil/LOCATION ./O

For each word in h_i, we initialize the cross-lingual feature vector to $\langle 0, 0, 0, 0 \rangle$ based on step 6 of Algorithm 1. The four fields of the feature vector indicate $\langle$Is Person?, Is Location?, Is Organization?, Other NE?$\rangle$ (4 NE tags of the Stanford tagger). For example, initially अनुष्का $\equiv \langle 0, 0, 0, 0 \rangle$ and ब्राजील $\equiv \langle 0, 0, 0, 0 \rangle$. Based on (2), the word 'अनुष्का' is projected to 'Anushka/PERSON'

using the mapping from (1c). Thus the word 'अनुष्का' is a potential candidate for PERSON name and we update the feature vector to $\langle 1, 0, 0, 0 \rangle$. Similarly, the word ब्राजील is mapped to three words (*much*, *like* and *Brazil*). We find only one of these words (*Brazil*) belongs to LOCATION type and the remaining two words (*much* and *like*) are not NEs. Thus the cross-lingual feature vector for the word ब्राजील is $\langle 0, 1, 0, 0 \rangle$. Note that more than one field in the feature vector can be '1' if the mapped English words point to different NE types. We combine the above cross-lingual features with monolingual features to produce the training instances for the SVM-based classifier.

5 Experimental Set-up

First we conduct two different experiments to estimate the baseline accuracy of our approach for the Hindi NER task. We use two different sets of monolingual features to train the baseline systems and compare the results with our cross-lingual feature-based approach. The following are the feature vectors for the two baseline systems:

Baseline1: $\{w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, p_i, |pref| \leq 4, |suff| \leq 4, t_{i-1}, t_{i-2}\}$

Baseline2: $\{w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, p_i, p_{i-1}, p_{i-2}, |pref| \leq 4, |suff| \leq 4, t_{i-1}, t_{i-2}\}$

We conduct the second set of experiments adding the cross-lingual features (cf. Section 3.3) with the monolingual features used in the two baseline experiments. We call them **Baseline$i$+CL**. We conduct two different experiments within the Baseline$i$+CL experiments.

— We use 4 different cross-lingual features ($\langle$ *Is Person*?, *Is Location*?, *Is Organization*?, *Other NE*? $\rangle$) (cf. Algorithm1) based on the 4 different NE classes of the Stanford English NER. We call this system **Baseline$i$+CL-4**. Note that, the Hindi NE-data has 15 different NE classes.

— Moreover, instead of considering only 4 classes, we consider the 7 NE-tags from the Stanford NE recognizer to annotate the English text. This generates a feature vector of size 7. The four additional features included here are $\langle$ *Is Money*?, *Is Date*?, *Is Time*?, *Is Percent*? $\rangle$ and there is no *Other NE* type. We anticipate that the use of a larger number of classes for the English NER will help to improve the Hindi NER task using 15 NE types. We call this system **Baseline$i$+CL-7**.

Furthermore, we assume that an equal number of NE-tags for both Hindi and English may have a higher impact while projecting information from the resource-rich to the resource-poor language. Thus, we merge the 15 NE classes from Hindi into the 4 classes (Person, Location, Organization and Others) of the Stanford NER tool. This gives us equivalent tagsets for both the Hindi task and the Stanford tagger. We conduct a third set of experiments using the 4 cross-lingual features and using 4 NE classes for Hindi. We call this experiment **Baseline$i$+CL-4$_{eq}$**. Note that the Baseline systems also change (in accuracies) in this setting.

Finally, we conduct another experiment to understand the performance of the direct projection of NEs between two languages based on GIZA++ alignment. This indeed justify the need of using cross-lingual features in a classifier instead of directly identifying NEs based on the alignment. This direct mapping require equal number of NE types between two languages. The number of NE classes in Hindi NER task is different from the Stanford English NE recognizer. Thus we conduct this experiment only in the CL-4$_{eq}$ setup, where English and Hindi NEs refer to an equivalent tagset of 4 NE types. We shall call this **Projection Baseline**. In this process, we assign the most likely NE type to a Hindi word based on the alignment information and the English NEs corresponding the alignment. If multiple NE types are equally likely for a Hindi word based on alignment function and English-side NE types, we randomly select one from them.

5.1 Data

For all experiments we used the Hindi NER data from ICON2013 NLP tools contest.[8] The training

[8] http://ltrc.iiit.ac.in/icon/2013/nlptools/index.html

System	Precision	Recall	F_1-score
Baseline1	78.27	67.46	72.46
Baseline1+CL-4	79.50	70.16	74.54
Baseline1+CL-7	78.46	69.37	73.63
Baseline2	82.32	73.17	77.48
Baseline2+CL-4	82.83	74.29	78.33
Baseline2+CL-7	82.11	74.29	78.00

Table 2. NER accuracy using cross-lingual features.

data consists of 3,583 sentences (approximately 70k words). We used 449 sentences from ICON2013 test data to evaluate our system. The test data contains a total of 630 NEs. All the data is represented in Shakti Standard Format (SSF) [2]. For our experiments, we transformed the data from SSF to *BIO* format where *B–X* indicates the first word of an NE type *X*, *I-X* indicates the intermediate word of an NE type *X* and *O* indicates a word outside a NE. Note that the best reported system performance achieved for Hindi in the ICON2013 contest with this data set is 77.44% [10] using both linguistic and word-based features along with a gazetteer list and post-processing rules.

6 Results and Observations

We measure tagging accuracy in terms of *Precision*, *Recall* and F_1-score. F_1-score is the harmonic mean of precision and recall: $F_1 = 2.precision.recall/(precision + recall)$. Table 2 shows the results obtained with different systems for the first two sets of experiments. We evaluate our NER systems using the CONLL-2000[9] shared task evaluation strategy. Table 3 shows the accuracy obtained from our third set of experiments using an equal number of NE classes for Hindi and English.

The effect of cross-lingual features on different NE classes is given in Table 4. We compare the *Baseline1* system with the *Baseline1+CL-4* system.

[9] http://conll.cemantix.org/2011/task-description.html

System	Precision	Recall	F_1 score
Baseline1	78.08	68.41	72.93
Baseline2	82.84	75.08	78.77
Projection Baseline	36.36	30.14	33.04
Baseline1+CL-4$_{eq}$	78.95	71.43	75.00
Baseline2+CL-4$_{eq}$	83.36	75.56	79.27

Table 3. NER accuracy using cross-lingual features and equal number of NE classes in both languages.

	Baseline1	Baseline1+CL-4
PERSON(58)*	61.86	66.02
LOCATION(377)*	78.68	80.68
ORGANIZATION(15)*	50.00	52.17
MONEY(3)	50.00	66.67
DISTANCE(21)	87.80	92.68
COUNT(15)	46.67	50.00
LIVTHINGS(35)	55.56	58.18
ARTIFACT(25)	42.42	42.42
DISEASE(10)	80.00	80.00
ENTERTAINMENT(28)	79.17	79.17
LOCOMOTIVE(4)	66.67	66.67
MATERIALS(20)	50.00	50.00
PLANTS(6)	50.00	50.00
QUANTITY(5)	80.00	80.00

Table 4. Comparison of F_1-score for different NE types. The first column represents different NE tags and their frequency in the test data. '*' indicates the NE types that are common between the Hindi task and English NER.

6.1 Summary of the Results

We found that the inclusion of cross-lingual features projected from a resource-rich language improves the NER accuracy (cf. Table 2). We found that Baseline1+CL-4 gives an improvement of 2.08 points F_1-score over the Baseline1 model (2.9% relative). Furthermore, when a larger monolingual feature set is used in Baseline2 model, we found an improvement of 0.85 points (1.1% relative) in F_1-score in Baseline2+CL-4 system.

The use of 7 NE types gives an improvement of 1.17 points (1.6% relative) and 0.52 points (0.7% relative) F_1-score for Baseline1+CL-7 and Baseline2+CL-7 system, respectively, compared to their relative baseline scores. These improvements are lower compared to the improvement from Baseline1+CL-4 and Baseline2+CL-4 systems.

In Table 4, we find that there are significant improvements in F_1-score for PERSON, LOCATION and ORGANIZATION types. These three NE types are common in both the Hindi NE tagset and Stanford 4 NE tags. Note that 71% of the NEs in the test document belong to these three NE types. Thus an improvement in these three NE types gives a significant improvement in the overall accuracy. Only 4 tag types (MONEY, DISTANCE, COUNT and LIVTHINGS) show some improvement out of a total of 11 tags that are not common between the two tagsets. However, these tags occur less frequently in the corpus compared to PERSON and LOCATION. Thus these tags have a lesser contribution to the overall accuracy. Most interestingly, we found that the accuracy does not drop for any of the tag type.

We expected the use of an equal number of tags in both the resource-rich and resource-poor language to improve NER accuracy. This is reflected in Table 3. We found 2.07 points (2.8% relative) and 0.50 points (0.6% relative) improvement in F_1-score with Baseline1+CL-4_{eq} and Baseline2+CL-4_{eq} systems, respectively, compared to the relative baseline system. This improvement is comparable to the improvement we obtained in our second set of experiments (cf. Table 4). Note that the direct projection of NEs has very low score (F_1=33.04%) which essentially indicates direct cross-lingual projection is not effective for NE recognition in Hindi using English-to-Hindi MT system. Altogether, in all our experiments we found that use of cross-lingual features projected from the resource-rich language to the resource-poor language improves the NER accuracy regardless of the feature set used.

6.2 Assessment of Error Types

Errors are propagated mostly due to errors in the GIZA++ alignment and incorrect NE recognition in the English text. Due to alignment errors, some potential Hindi NE words do not map to the actual corresponding word in the English sentence. This produces misleading features for the wrongly aligned Hindi word. In example (3b), the word मुंबई does not map to any word in (3a) despite the correct aligning word (*Bombay*) being present in e.

(3) a. e: Royal Bombay continued into the 20th century .

b. h: राजसी ({ 1 }) मुंबई ({ }) का ( ) निर्माण ({ }) २०वीं ({ 2 3 4 6 }) शताब्दी ({ 7 }) में ({ }) भी ({ }) रहा ({ }) . ({ 8 })

Sometimes the potential Hindi NE word is aligned to the correct word in the translated English sentence e but the English NER produces an incorrect NE tag for the English word. In example (4b) the word दीव is mapped to the correct English word *Diu* in (5a) but the Stanford NER marks it as *Diu/O* (not a name).

(4) a. e: It/O is/O also/O the/O story/O of/O Diu/O ./O

b. h: ऐसा ({ 1 }) किस्सा ({ 5 }) दीव ({ 7 }) का ({ 6 }) भी ({ 3 }) है ({ 2 }) . ({ 8 })

Finally, we use an MT system to translate the Hindi sentence into English. The translation system sometimes fails to produce an accurate enough translation to allow the correct translated word to be found for a given potential Hindi NE word.

7 Conclusion

Our experiments show that MT systems can be used to project information from resource-rich languages to resource-poor ones. These projections can be used as cross-lingual features in the classification problem. We have shown that NER for a resource-poor language Hindi can be improved using a Hindi-to-English MT system and English NER. Our best performance improvement results in 2.1 (2.9% relative) F_1 score improvement over the baseline.

So far our system has been tested for just one classification problem, namely NER. In order to test the effectiveness of our approach, we plan to use our approach for other NLP classification problems (viz. POS labelling, NP chunking). We have tested our approach using one learning algorithm and we plan to test our approach over a wide range of classification algorithms using state-of-the-art features. We also plan to use different word aligners (e.g. [19]) to compare the effect of alignment in our work.

Acknowledgments

This research is supported by Science Foundation Ireland through the ADAPT Centre (Grant 13/RC/2106) (www.adaptcentre.ie) at Dublin City University and Trinity College Dublin, and by Grant 610879 for the Falcon project funded by the European Commission.

References

1. **Babych, B. & Hartley, A. (2003).** Improving machine translation quality with automatic named entity recognition. *Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT*, Association for Computational Linguistics, pp. 1–8.
2. **Bharati, A., Sangal, R., & Sharma, D. M. (2007).** Ssf: Shakti standard format guide. *Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India*, pp. 1–25.
3. **Bharati, A., Sangal, R., Sharma, D. M., & Bai, L. (2006).** Anncorra: Annotating corpora guidelines for pos and chunk annotation for Indian languages. Technical report, Technical Report (TR-LTRC-31), LTRC, IIIT-Hyderabad.
4. **Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. (1997).** Nymble: a high-performance learning name-finder. *Proceedings of the fifth conference on Applied natural language processing*, Association for Computational Linguistics, pp. 194–201.
5. **Borthwick, A. (1999).** *A maximum entropy approach to named entity recognition*. Ph.D. thesis, Citeseer.
6. **Burkett, D., Blitzer, J., & Klein, D. (2010).** Joint parsing and alignment with weakly synchronized grammars. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 127–135.
7. **Burkett, D., Petrov, S., Blitzer, J., & Klein, D. (2010).** Learning better monolingual models with unannotated bilingual text. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, pp. 46–54.
8. **Cortes, C. & Vapnik, V. (1995).** Support-vector networks. *Machine learning*, Vol. 20, No. 3, pp. 273–297.
9. **Das, D. & Petrov, S. (2011).** Unsupervised part-of-speech tagging with bilingual graph-based projections. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, pp. 600–609.
10. **Devi, S. L., Malarkodi, C., Marimuthu, K., & Chrompet, C. (2013).** Named entity recognizer for Indian languages. *ICON NLP Tool Contest*.
11. **Ekbal, U. K. S. A. & Saha, S. (2012).** Differential evolution based feature selection and classifier ensemble for named entity recognition.
12. **Finkel, J. R., Grenager, T., & Manning, C. (2005).** Incorporating non-local information into information extraction systems by gibbs sampling. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, pp. 363–370.
13. **Fleiss, J. L. (1971).** Measuring nominal scale agreement among many raters. *Psychological bulletin*, Vol. 76, No. 5, pp. 378.
14. **Grishman, R. (1995).** The NYU system for MUC-6 or where's the syntax? *Proceedings of the 6th conference on Message understanding*, Association for Computational Linguistics, pp. 167–175.
15. **Haque, R., Kumar Naskar, S., Van Den Bosch, A., & Way, A. (2010).** Supertags as source language context in hierarchical phrase-based smt. Association for Machine Translation in the Americas.
16. **Hermjakob, U., Knight, K., & Daumé III, H. (2008).** Name translation in statistical machine translation-learning when to transliterate. *ACL*, pp. 389–397.
17. **Kim, S., Toutanova, K., & Yu, H. (2012).** Multilingual named entity recognition using parallel data and metadata from wikipedia. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*,

Association for Computational Linguistics, pp. 694–702.

18. **Li, W. & McCallum, A. (2003).** Rapid development of Hindi named entity recognition using conditional random fields and feature induction. *ACM Transactions on Asian Language Information Processing (TALIP)*, Vol. 2, No. 3, pp. 290–294.
19. **Liang, P., Taskar, B., & Klein, D. (2006).** Alignment by agreement. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, Association for Computational Linguistics, pp. 104–111.
20. **Ma, X. & Cieri, C. (2006).** Corpus support for machine translation at LDC. *Proceedings of LREC*.
21. **Nagesh, A., Ramakrishnan, G., Chiticariu, L., Krishnamurthy, R., Dharkar, A., & Bhattacharyya, P. (2012).** Towards efficient named-entity rule induction for customizability. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, pp. 128–138.
22. **Och, F. J. & Ney, H. (2003).** A systematic comparison of various statistical alignment models. *Computational linguistics*, Vol. 29, No. 1, pp. 19–51.
23. **Ratinov, L. & Roth, D. (2009).** Design challenges and misconceptions in named entity recognition. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, pp. 147–155.
24. **Richman, A. E. & Schone, P. (2008).** Mining Wiki resources for multilingual named entity recognition. *ACL*, pp. 1–9.
25. **Saha, S. K., Mitra, P., & Sarkar, S. (2008).** Word clustering and word selection based feature reduction for MaxEnt based Hindi NER. *ACL*, pp. 488–495.
26. **Shah, R., Lin, B., Gershman, A., & Frederking, R. (2010).** SYNERGY: a named entity recognition system for resource-scarce languages such as Swahili using online machine translation. *Proceedings of the Second Workshop on African Language Technology (AfLaT 2010)*, pp. 21–26.
27. **Srihari, R., Niu, C., & Li, W. (2000).** A hybrid approach for named entity and sub-type tagging. *Proceedings of the sixth conference on Applied natural language processing*, Association for Computational Linguistics, pp. 247–254.
28. **Ueffing, N. & Ney, H. (2003).** Using pos information for statistical machine translation into morphologically rich languages. *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, Association for Computational Linguistics, pp. 347–354.
29. **Wang, M. & Manning, C. D. (2014).** Cross-lingual pseudo-projected expectation regularization for weakly supervised learning. *Transactions of the Association for Computational Linguistics*, Vol. 2, pp. 55–66.
30. **Yarowsky, D., Ngai, G., & Wicentowski, R. (2001).** Inducing multilingual text analysis tools via robust projection across aligned corpora. *Proceedings of the first international conference on Human language technology research*, Association for Computational Linguistics, pp. 1–8.

Sandipan Dandapat is a Senior Applied Researcher at Microsoft India. He has been working in the field of NLP for about 10 years and have more than 30 publications in reputed international conferences and journals.His primary research area is Machine Translation. Apart from Machine Translation, he has also worked on morphological analyzer and generator, POS Taggers, intelligent linguistic annotation framework, MWE's.

Andy Way is Professor in Computing at Dublin City University (DCU). He is also Deputy Director of the ADAPT Centre for Digital Content Technology (formerly CNGL). His research interests include all areas of machine translation, which he has applied to a career that has spanned academia and industry. In 2015 Professor Way received the DCU Presidents Research Award in recognition of his contribution to the field of computing. From 200915, Professor Way was President of the European Association for Machine Translation, and from 201113 President of the International Association for Machine Translation. He has been Editor of the leading journal, Machine Translation, since 2007.

Article received on 07/01/2016; accepted on 28/02/2016.
Corresponding author is Sandipan Dandapat.