

# Mitigating Problems in Analogy-based EBMT with SMT and vice versa: a Case Study with Named Entity Transliteration\*

Sandipan Dandapat, Sara Morrissey, Sudip Kumar Naskar, and Harold Somers  
Centre for Next Generation Localisation  
Dublin City University  
Glasnevin, Dublin 9, Ireland  
{sdandapat, smorri, snaskar, hsomers}@computing.dcu.ie

**Abstract.** Five years ago, a number of papers reported an experimental implementation of an Example Based Machine Translation (EBMT) system using proportional analogy. This approach, a type of analogical learning, was attractive because of its simplicity; and the paper reported considerable success with the method using various language pairs. In this paper, we describe our attempt to use this approach for tackling English–Hindi Named Entity (NE) Transliteration. We have implemented our own EBMT system using proportional analogy and have found that the analogy-based system on its own has low precision but a high recall due to the fact that a large number of names are untransliterated with the approach. However, mitigating problems in analogy-based EBMT with SMT and vice-versa have shown considerable improvement over the individual approach.

**Keywords:** Example Based Machine Translation, Proportional Analogy, Named Entity Transliteration

## 1 Introduction

In the EBMT workshop in Phuket, Thailand, Lepage and Denoual (2005a) presented “The ‘purest’ EBMT system ever built: no variable, no templates, no training, examples, just examples, only examples”. This purely data-driven approach to MT uses the notion of *proportional analogy* (PA, described below), a type of analogical learning, the very simplicity of which is its attraction. We have attempted to reimplement the method in order to verify the previous authors’ claims. It has been noted, that the PA-based system works well in the case of shorter sentences with similar structure (Langlais and Patry, 2007). Thus we have taken named entity (NE) transliteration as a case study because NEs are typically short. Furthermore, it was reported (Hermjakob et. al, 2008) that the state-of-the-art SMT system can’t handle NEs that are not found in the training parallel text. In this paper we describe an experiment in which we use the method to handle unknown named entities (NEs) in an English-Hindi transliteration task.

The idea introduced in Lepage and Denoual (2005a) is explained in considerably more detail in Lepage and Denoual (2005b, c). ALEPH system is an implementation of the research described in their three 2005 papers, and was tested on a corpus of 160K English, Japanese and Chinese sentences, from the C-STAR project’s BTEC, a travel and tourism domain corpus. The system did very well on data from the IWSLT 2004 competition, coming a close second to the competition winner on all measures. The ALEPH system evolved into a new system, named GREYC, with some modification described in Lepage and Lardilleux (2007). The GREYC system also incorporated new heuristics and had an additional refinement of non-determinism to generate all possible solutions for a single analogical equation which otherwise had only one solution in ALEPH, and is accordingly much slower. While Lepage and colleagues have had

---

\* The research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localization (<http://www.engl.ie>) at Dublin City University.

modest success using PA for a full translation task, the idea is adapted to translating unknown words in the context of another approach to MT as reported by Denoual (2007), Langlais and Patry (2007), and Langlais et al. (2009). Denoual’s (2007) experiments attempt to translate all unknown words in a Japanese–English task and have reported that translation adequacy (NIST score) improves but fluency (BLEU score) remains stable or is decreased. Langlais and Patry (2007) had more success in handling unknown words while the language pairs are quite close in morphological structure. Langlais and Yvon (2008) use PA to supplement the words and phrases for standard SMT when a word to be translated is not covered by the statistical model. Finally, Langlais et al. (2009) applied the method to the translation of medical terms and showed little improvement on purely statistical approaches. Since no off-the-shelf implementation is available for solving analogies, we have implemented our own EBMT system from scratch using PA based on the description in Lepage (2005c). It is often the case that a PA-based system suffers from low recall. First we try to improve the PA-based system by introducing new heuristics to overcome the low recall. Furthermore, we have improved the system accuracy using an SMT-based system with the PA-based system.

In Section 2 we describe the underlying process of EBMT using PA. Section 3 describes our implementation and different heuristics used for solving analogies. Section 4 describes a number of experiments carried out and empirically compares results against a standard SMT system with error analysis for the NE transliteration task. We conclude in Section 5.

## 2 The Underlying Idea

### 2.1 Proportional Analogy

PAs are global relationships between four objects -  $A : B :: C : D$ , read as “ $A$  is to  $B$  as  $C$  is to  $D$ ”. The symbol ‘ $::$ ’ is sometimes replaced with an equals sign (=) to denote an equation. This formulation as an equation can have zero, one, or more solutions if any of the objects (usually  $D$ ) is considered as a variable. PAs are often seen as a way of knowledge representation in Artificial Intelligence due to their power to represent world knowledge and the lexical relation encoded in them. In NLP, the analogies are used as an instrument to explain inflectional and derivational morphology (Lepage, 1998).

### 2.2 EBMT using Analogy

Lepage and Denoual (2005a,b,c) showed how an EBMT system can be built based on the algorithm proposed by Lepage (1998). Treating a sentence as a string of characters, they note that PAs can be handled as in (1).

$$(1) \quad \begin{array}{ccccccc} \textit{They swam in} & & \textit{They swam} & & \textit{It floated in the} & & \textit{It floated across} \\ & & & & \textit{sea} & & \textit{the river} \\ & & \textit{the sea} & : & \textit{across the river} & :: & \\ & & & & & & \end{array}$$

To build up an EBMT system, we must assume a database of example pairs, where each pair is a source and target language translation equivalent. For the first three sentences in (1), the translation equivalents in Spanish are given in (2).

$$(2) \quad \begin{array}{lll} \textit{a. Nadar3n en el mar.} & \textit{b. Atraversar3n el r3o nadando.} & \textit{c. Flot3 en el mar.} \end{array}$$

Suppose now that we want to translate the sentence *It floated across the river*. The translation process is as follows:

1. Find a pair ( $A, B$ ) of sentences in the example set that satisfies the PA in (3).

$$(3) \quad A : B :: C(?) : \textit{It floated across the river}$$

Solving this results in  $C = \textit{It floated in the sea}$ .

2. Take the translations corresponding to  $A, B$  and  $C$  (noted  $A', B', C'$ ).

3. Solve the equation in (4):  $x$  represents the desired translation.

$$(4) \quad A' : B' :: C' : x$$

Substituting the three sentences in (2) into (4), we have a solvable equation with  $x = \textit{Atraversaró el río flotando}$ , which is an acceptable translation.

However, due to the unconstrained nature of PA, there is always a possibility of solving “false analogies”, i.e. set of strings for which the analogy holds, but which do not represent a valid linguistic relationship. Example (5) illustrates this phenomenon.

(5) Yea : Yep :: At five a.m. : At five p.m.

An EBMT system using PAs shows quadratic time complexity. Thus, we only look for time-bounded solutions, i.e. allow the process to continue for a fixed amount of time. We also apply heuristics (describe in section 3.1) to filter out some of the PAs and to try better candidates first by ranking the equations.

### 3 Implementation

We have implemented the EBMT system using PAs based on Lepage (1998, 2005c). We distinguish between three main components in our system. These three components are used to solve both source- and target-side analogies. The three main components of the analogy-based EBMT, namely *Heuristics*, *Analogy Verifier* and *Analogy Solver*, are depicted in Figure 1.

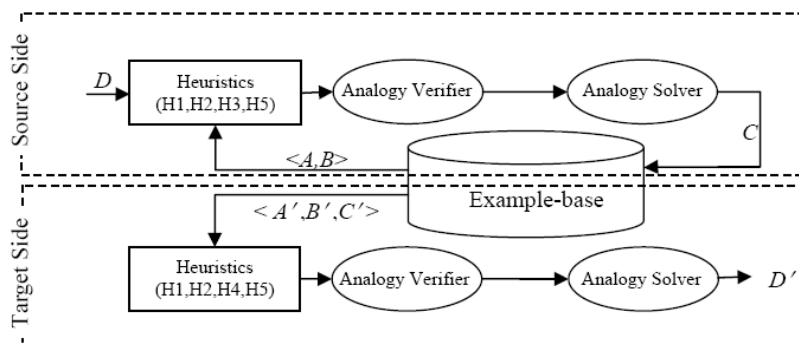


Figure 1: Analogy-based EBMT architecture

Firstly, the system requires some knowledge about choosing relevant  $\langle A, B \rangle$  pairs from the example-base to ensure that the better candidate analogical equations from the potential set of all possible analogies are solved first, and also to filter out some of the unsolvable analogies before verification. We adopt different *heuristics* to ensure this. Secondly, there is an *Analogy Verifier*, which decides the solvability of an analogical equation. The third component solves the analogy as in (4) based on the triplet  $\langle A, B, D \rangle$  and produces  $C$ . Note that  $D$  is the input sentence to be translated. We call this module the *Analogy Solver*. Once  $C$  is produced in the source side, we find the translation equivalents  $\langle A', B', C' \rangle$  in the target side for the source side  $\langle A, B, C \rangle$  triplet. Further, we apply the three components in the target side in the same order to obtain one candidate translation  $D'$  as in (4). Collecting all  $D'$ , we rank them by frequencies as different analogical equations might produce identical solutions.

#### 3.1 Heuristics

We adopted different heuristics from the literature to understand their relative performance in translation tasks under the time-constrained model. Note that when no heuristic is applied, to transliterate one input in our NE transliteration task, the average number of analogical equations within 1 second are around 600K equations in the source side and 40K equations in the target side. Out of these 40K target-side equations, the average number of analogical equations that generates the final solution is 0.692 only. As we will see, the various heuristics affect the number of equations solved or attempted, ideally cutting down effort wasted on computations which will not contribute to a useful solution. The heuristics do this in different ways, and with varying success. We first choose the heuristic from Lepage and Denoual (2005a,c) which selects a relevant pair  $\langle A, B \rangle$  based on a length comparison with the input  $D$ .

**H1:** Consider as candidates only sentences whose length is more than half and less than double the length of the input sentence. Formally,  $|D|/2 \leq |A|$ ,  $|B| \leq 2|D|$ , where  $|x|$  is the length of  $x$ .

With the help of H1, we are able to solve around 705K analogical equations in the source side and around 34K equations in the target side in 1 second. This heuristic is solving more equations on the source side but effectively reducing the number on the target side and the average number of equations that produce output is 0.335. This is reflected in the overall output of the experiments shown later in Table 1 (in Section 4).

Our second heuristic is based on that of Lepage and Lardilleux (2007), which speeds up the process of searching relevant  $\langle A, B \rangle$  pairs. This is done by sorting the corpus based on the sentence to be translated ( $D$ ), using edit distance for the selection of  $A$ s, and selecting  $B$ s based on *inclusion score* (Lepage, 1998:730) i.e. length of  $B$  minus its similarity to  $D$ .

**H2:** Consider as candidates primarily sentence pairs where  $A$  has a low edit distance w.r.t.  $D$ , and  $B$  has low inclusion score w.r.t.  $D$ .

We are able to solve around 788K and 42K analogical equations with the help of H2 within 1 second. We found that with this heuristic, the average number of analogical equations that lead to output are 176. Thus, this is expected to work well with our current experimental setup.

In the third heuristic, we adopt a “trick” described by Langlais and Yvon (2008), called S-TRICK based on a simple requirement of sharing the same first or last symbol.

**H3:** Consider a candidate pair where  $A$  shares the same first or last character with  $B$  or  $D$ .

$$A[1] \in \{B[1], D[1]\} \text{ and } A[S] \in \{B[S], D[S]\}$$

where  $S[1]$  and  $S[S]$  are respectively the first and last character in the string  $S$ .

The average numbers of source- and target-side analogical equations solved within 1 second with the help of H3 are around 791K and 10K respectively and the average number of analogical equations which produce output is 8.75.

Our fourth heuristic relates to the effort of solving target-side analogical equations  $A' : B' :: C' : x$  based on Langlais and Yvon’s (2008) character count property, called T-TRICK. Formally, it can be stated as:

**H4:** Whenever a symbol occurs more frequently in  $A'$  than it does in  $B'$  and  $C'$ , the analogical equation is bound to fail and need not be solved.

$$[A' : B' :: C' : x] \neq \phi \text{ if } |A'|_c \leq |B'|_c + |C'|_c \quad \forall c \in \{A', B', C'\}$$

The average numbers of source- and target-side analogical equations solved within 1 second with the help of H4 are around 703K and 33K respectively and the average number of analogical equations which produce output is 0.382.

Our final heuristic is based on the modification of H2. Here also, we speed up the process of searching relevant  $\langle A, B \rangle$  pairs. We choose  $\langle A, B \rangle$  pairs based on a smaller edit distance with respect to the input sentence to be translated ( $D$ ). This is done by sorting the examples based on the edit distance with respect to  $D$  and choosing the top two candidates as the  $\langle A, B \rangle$  pair from the sorted examples.

**H5:** Consider as candidate pair where  $A$  and  $B$  have a low edit distance w.r.t.  $D$  such that  $A \neq B$ .

We are able to solve around 673K and 10K analogical equations with the help of H5 within 1 second. However, we found that with this heuristic, the average number of analogical equations that lead to output are 1900 (was 176 with H2). Thus, this is expected to work best with our current experimental setup.

## 4 Experiments

We have tested our EBMT system using PA for a NE transliteration task from English to Hindi. Five different experiments were conducted based on our EBMT system using PA. We shall call

these analogy-based EBMT (**AEBMT**). The five experiments deal with the five different heuristics described in 3.1. Each of these five experiments was also tested with a time bound of one second and three seconds to understand the effect of time while using analogy-based system. In parallel, we have also used MaTrEx (Stroppa and Way, 2006), an open source statistical MT (**SMT**) system in order to estimate the relative performance of the models. Furthermore, the SMT system can be trained at character-, syllable- and word-level using appropriate example-bases.

We have found that there are cases where AEBMT correctly produces the transliteration but SMT fails and vice versa. In order to further improve the transliteration accuracy, we use a combination of AEBMT with SMT. We combine these two systems in two ways. We assume that the transliteration of a word  $w$  produced by AEBMT and SMT are respectively  $T_{\text{AEBMT}}(w)$  and  $T_{\text{SMT}}(w)$ . First, we back-off to the SMT system when AEBMT fails to transliterate some names to mitigate the problem of AEBMT with SMT (**AEBMT+SMT**). In order to do that, we combine the outputs of both the systems in order  $T_{\text{AEBMT}}(w)+T_{\text{SMT}}(w)$ , which automatically takes back-off when  $T_{\text{AEBMT}}(w)=\text{null}$ . On the other hand taking SMT as the base system, we collect transliteration from the AEBMT system to deal with the problem of SMT using analogy (**SMT+AEBMT**) with the ordered concatenation of  $T_{\text{SMT}}(w)+T_{\text{AEBMT}}(w)$ . Thus we have four systems (AEBMT, SMT, AEBMT+SMT, SMT+AEBMT) that are tested with five heuristics (H1, H2, H3, H4 and H5) and a situation when no heuristics are used.

#### 4.1 Data Used for the Experiment

All the experiments are conducted with the NEWS 2009 English–Hindi transliteration data (Kumaran and Kellner, 2007). The data consists of 10,000 NEs for training and 1,000 names for testing. The same examples are represented in three different ways as in word level {*spingarn* – सपिनगारन}, syllable level {*spi nga rn* - सपि न गा र्न} and in character level {*s p i n g a r n* - स प ि न ग ा र न}. All the experiments were tested on character, syllable and word level NEs as example-base.

#### 4.2 Results

We have evaluated the system with the NEWS’09 metrics (Li et al., 2009). The accuracy is defined as the ratio of correct transliteration in the first position to the total number of words to be transliterated. Table 1 summarizes the final accuracy achieved by different methods varying the allowable running time to transliterate a single name.

**Table 1:** transliteration accuracies (in %) with different models using different heuristics and with different allowable running time

Heuristics		Character Level System Acc (%)			Syllable Level System Acc (%)			Word Level System Acc (%)		
		<b>SMT = 31.8</b>			<b>SMT = 36.2</b>			<b>SMT = 8.7</b>		
		AEBMT	AEBMT +SMT	SMT +AEBMT	AEBMT	AEBMT +SMT	SMT +AEBMT	AEBMT	AEBMT +SMT	SMT +AEBMT
Running Time = 1s	No	13.7	32.6	31.8	14.2	36.5	36.6	15.7	15.7	17.2
	H1	9.4	32.3	31.8	13	35.8	36.6	11.2	14.1	15.3
	H2	22.2	32.5	31.8	21.4	32.6	36.4	20.6	20.6	20.9
	H3	14.1	32.4	31.8	15.4	36.2	36.7	15.3	15.3	15.6
	H4	9.4	32.2	31.8	13	35.8	36.6	11.2	14.1	15.3
	H5	<b>28.1</b>	<b>36.0</b>	31.8	<b>30.2</b>	<b>37.1</b>	<b>37.7</b>	<b>28.7</b>	<b>28.7</b>	<b>29.0</b>
Running Time = 3s	No	16.6	33.1	31.8	17.2	35.1	36.7	17.1	17.1	17.5
	H1	16.1	33	31.8	17.1	34.7	36.7	17	17	17.4
	H2	23.7	31.9	31.8	24.1	33.5	36.6	23.2	23.2	23.5
	H3	18.3	32.6	31.8	15.4	36.2	36.6	19.3	19.3	19.6
	H4	16	33	31.8	17.2	34.8	36.7	17.1	17.1	17.5
	H5	<b>28.9</b>	<b>35.7</b>	31.8	<b>30.3</b>	<b>37.0</b>	<b>37.6</b>	<b>29.3</b>	<b>29.3</b>	<b>29.6</b>

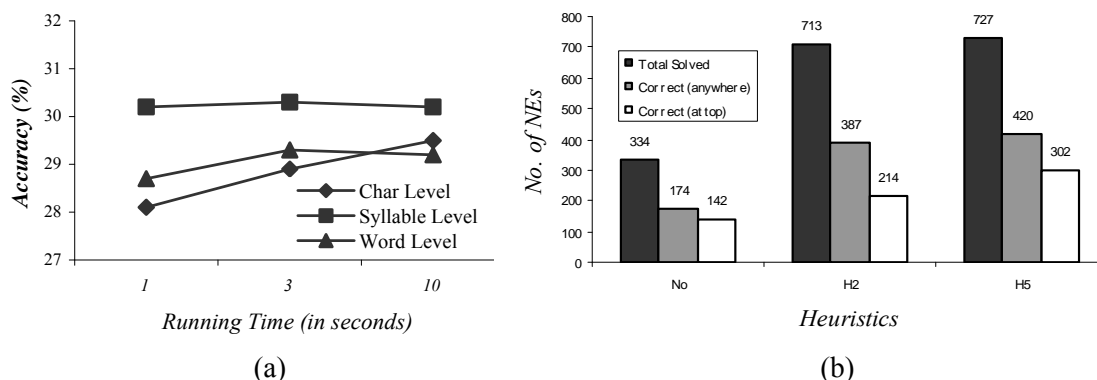
Note that the SMT baseline accuracies are 31.8%, 36.2% and 8.7% respectively for the character, syllable and word-level models. The highest accuracies achieved with EBMT using analogy are 28.9%, 30.3% and 29.3% respectively for the character, syllable and word-level models with the H5 heuristic. However, while combining the SMT with AEBMT (AEBMT+SMT) the highest accuracies obtained are 36.0%, 37.1% and 29.3% with a percentage of relative improvement over the baseline (SMT) of 13.2%, 2.5% and 236.8% respectively for the character, syllable and word-level models. Hereafter all the improvements essentially denote relative improvements. On the other hand, a combination of AEBMT with the SMT (SMT+AEBMT) shows results of 31.8%, 37.7% and 29.6% accuracy with the improvement of 0%, 4.1% and 240.2% respectively for three models.

### 4.3 Observations

We found that AEBMT has lower accuracy on its own for the character- and syllable-level model for the transliteration task. However, the word-level AEBMT models have a huge improvement over the SMT based models. The claim might be insignificant when transliterating NEs as a task on its own, as other models (character and syllable level) have higher accuracy. However, in the case of full text translation, SMT models are trained at the word/phrase level so can only transliterate names that are seen in the corpus. A similar effect has been observed in case of our word-level NE transliteration experiment. On the contrary, our AEBMT models inherently consider every word/sentence as a string of characters. Thus, a significant improvement has been obtained which might be relevant for considering an analogy-based MT system to address unknown words in the standard phrase-based SMT system.

Another significant observation is that AEBMT accuracy increases when a longer running time is allowed for the transliteration. This essentially solves more analogical equations which produce correct solutions for more NEs. This effect has been observed for all the heuristics applied in our system. Furthermore, we conducted experiments allowing a running time of 10 seconds, and we found significant improvement with AEBMT other than H5 but there is no improvement for the combined systems (AEBMT+SMT, SMT+AEBMT). The H5 heuristic is possibly able to capture the solvable analogy within 3 seconds thus there is no improvement with a running time of 10 seconds. Figure 2(a), shows the improvement in accuracy over time with AEBMT while H5 heuristic is in use. It is interesting to note that the use of heuristics improves the performance of the analogy-based MT with the exception of H1 and H4 heuristics. This is because some of the valid analogies are filtered out by the risky strategy of the heuristics which discount some  $\langle A, B \rangle$  pairs as in (6).

(6) a. *He dived.* [9 characters]    b. *He dived into the river.* [24 characters]



**Figure 2:** a) The effect of running time (1sec, 3sec, 10 sec) in analogy-based EBMT while H5 heuristic is in use with different models, b) Comparison of the total number of NEs transliterated, the total number of correct transliterations in the candidate output set and the correct number of transliterations at rank 1.

A combination of SMT with AEBMT (AEBMT+SMT), gives an improvement of 13.2%, 2.5% and 236.8% respectively for the character, syllable and word-level models compared to the standard SMT. More closely, we have seen improvement with AEBMT+SMT in the character-based model with all the heuristics compared to both AEBMT and SMT. However, the syllable-level model shows huge improvement (minimum of 51.9%) with AEBMT+SMT compared to AEBMT alone but only in two cases (no heuristic and H5) we have found a small improvement (0.8% and 2.5%) over SMT. Figure 2(b) gives a comparison of the total number of NEs transliterated, the number of NEs correctly transliterated irrespective of their rank in the output list and the number of NEs correctly transliterated at first position. Although, H2 is much better in all aspects over no heuristics, the percentage of names correctly transliterated at top position by H2 (30%) is much lower in comparison with no heuristics (42.5%). Thus we have seen in the combined system (AEBMT+SMT) no heuristics has a little improvement compared to H2. However, the H5 reduces the number of false positives and shows improvements for all possible combinations.

Finally, while combining AEBMT with SMT (SMT+AEBMT), if no transliteration is produced by SMT, back-off transliteration is taken using AEBMT. This gives an improvement of 4.1% and 240.2% in case of syllable and word-level models respectively over the standard SMT model. However, this combination has no improvement for the character-level model compared to the SMT model irrespective of the heuristic in use. This can be explained by the fact that in character-level SMT models observe all possible characters of the language when a reasonable amount of example-base is used to train the system. Thus, the character-based system produces transliterations for all the NEs and no back-off is being taken from the AEBMT system. Furthermore, in syllable and word-level SMT+AEBMT systems, similar effects of heuristics have been noticed as in the case of AEBMT+SMT. Heuristic H2 is better than H3 or no heuristics within the AEBMT system but has no or a negative effect in combination with SMT. The H5 heuristic has improvements for all with character-, syllable- and word-based models in case of SMT+AEBMT.

#### 4.4 Assessment of Error Types

The most common types of error encountered by the AEBMT models is that the correct output is often produced but not always in the first position. As we have seen in Figure 2(b), the heuristic H5, only 30.2% NEs are correctly transliterated with highest frequency in the output list although a total 42% NEs are transliterated correctly irrespective of their position in the output list. Thus, without this effect, the AEBMT could have been much better compared to the SMT models on its own.

The second type of error is spelling variation in the reference data. For example, the English input NE ‘*edinburgh*’ can be written as ‘ऐडिनबर्ग’ or ‘एडीनबर्ग’ in Hindi. The *matra* ‘ि’ becomes ‘ी’. With our system, we are able to produce ‘ऐडिनबर्ग’ but the reference translation has ‘एडीनबर्ग’, thus resulting in an incorrect transliteration. We have found 46 (4.6%) such cases where the output differs from the reference due to this spelling variation. Capturing these spelling variations could have increased the absolute accuracy by 4.6%.

Finally, we have seen cases where there is a tie in the top frequency of the output list. We choose one randomly in such cases. The effect is shown in Table 2 for the NEs ‘*pratima*’ and ‘*bhutti*’. In the case of ‘*pratima*’, the correct output as per the reference data is ‘प्रतिमा’ although all the three outputs have the same frequency of 1. In case of ‘*bhutti*’, there are two outputs which have the same frequency of 6 and ‘भुट्टी’ is the correct output.

**Table 2:** Example of transliteration with a tie in highest frequency output

Input NE	Output Transliterations
<i>Pratima</i>	प्रतीमा(1), प्रतिमा(1), प्रतिमै(1)
<i>bhutti</i>	भुट्टि(6), भुट्टी(6), भट्टुटी(2), भट्टई(2)

## 5 Conclusion

Unlike other approaches to EBMT, the PA-based approach seems to suffer badly when the size of the example-base is increased, with both processing times and numbers of solutions increasing. It is clear that heuristics must be introduced to handle the underlying problems of analogy-based translation. In this paper, we have described the effect of heuristics for automatic transliteration of NEs using analogy-based EBMT. We have found that the analogy-based approach has less recall (unable to find any solution in many cases) compared to SMT. However, a combination of both has achieved a reasonably good improvement over the individual model. The approach seems to have difficulty as a stand-alone translation model; its use for the special case of unknown words, particularly NEs seems much more promising.

## References

- Denoual, E. 2007. Analogical translation of unknown words in a statistical machine translation framework. In *Proceedings of the MT Summit XI*, Copenhagen, Denmark, pp.135-141.
- Hermjakob, U., Knight, K. and Daume III, H. 2008. Name translation in statistical machine translation learning when to transliterate. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, Columbus, Ohio, pp. 389-397.
- Kumaran, A. and Kellner, T. 2007. A generic framework for machine translation. In *Proceedings of the 30<sup>th</sup> Annual International SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, pp. 721-722.
- Langlais, P. and Patry, A. 2007. Translating unknown words using analogical learning. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL-2007*, Prague, Czech Republic, pp. 877-886.
- Langlais, P. and Yvon, F. 2008. Scaling up analogical learning. In *Proceedings of the 22<sup>nd</sup> International Conference on Computational Linguistics (COLING, 2008)*, Manchester, UK, pp. 51-54.
- Langlais, P., Yvon, F. and Zweigenbaum, P. 2009. Improvements in analogical learning: application to translating multi-terms of the medical domain. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL, 2009)*, Athens, Greece, pp. 487-495.
- Lepage, Y. 1998. Solving analogies on words: an algorithm. In *Proceedings of the 36<sup>th</sup> Annual Meeting of the Association of Computational Linguistics and 17<sup>th</sup> International Conference on Computational Linguistics*, Quebec, Canada, pp. 728-734.
- Lepage, Y. 2004. Lower and higher estimates of the number of “true analogies” between sentences contained in a large multilingual corpus. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, pp. 736-742.
- Lepage, Y. and Denoual, E. 2005a. The ‘purest’ EBMT system ever built: no variables, no templates, no training, examples, just examples, only examples. In *Proceedings of the MT Summit X, Second Workshop on Example-Based Machine Translation*, Phuket, Thailand, pp. 81-90.
- Lepage, Y. and Denoual, E. 2005b. ALEPH: an EBMT system based on the preservation of proportional analogies between sentences across languages. In *Proceedings of the International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation (IWSLT 2005)*, Pittsburgh, Pennsylvania, pp. 47-54.
- Lepage, Y. and Denoual, E. 2005c. Purest ever example-based machine translation: Detailed presentation and assessment. *Machine Translation* 19:251-282.
- Lepage, Y. and Lardilleux, A. 2007. The GREYC machine translation system for the evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2007)*, Trento, Italy, pp. 49-53.
- Li, H., Kumaran, A., Zhang, M. and Pervouchine, V. 2009. Whitepaper of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, Suntec, Singapore, pp. 19-26.
- Stroppa, N. and Way A. 2006. MaTrEx: the DCU machine translation system for IWSLT 2006. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2006)*, Kyoto, Japan, pp. 31-36.