Name Translation in Statistical Machine Translation
Learning When to Transliterate

Ulf Hermjakob and **Kevin Knight**
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292, USA
{ulf,knight}@isi.edu

Hal Daumé III
University of Utah
School of Computing
50 S Central Campus Drive
Salt Lake City, UT 84112, USA
me@hal3.name

Abstract

We present a method to transliterate names in the framework of end-to-end statistical machine translation. The system is trained to learn when to transliterate. For Arabic to English MT, we developed and trained a transliterator on a bitext of 7 million sentences and Google's English terabyte ngrams and achieved better name translation accuracy than 3 out of 4 professional translators. The paper also includes a discussion of challenges in name translation evaluation.

1 Introduction

State-of-the-art statistical machine translation (SMT) is bad at translating names that are not very common, particularly across languages with different character sets and sound systems. For example, consider the following automatic translation:[1]

Arabic input موسيقيين مثل باخ وموزار وشوبان وبيتهوفن وشومان ورحمانينوف ورافيل وبروكوفييف

SMT output musicians such as Bach

Correct translation composers such as Bach, Mozart, Chopin, Beethoven, Schumann, Rachmaninoff, Ravel and Prokofiev

The SMT system drops most names in this example. "Name dropping" and mis-translation happens when the system encounters an unknown word, mistakes a name for a common noun, or trains on noisy parallel data. The state-of-the-art is poor for two reasons. First, although names are important to human readers, automatic MT scoring metrics (such as BLEU) do not encourage researchers to improve name translation in the context of MT. Names are vastly outnumbered by prepositions, articles, adjectives, common nouns, etc. Second, name translation is a hard problem — even professional human translators have trouble with names. Here are four reference translations taken from the same corpus, with mistakes underlined:

Ref1 composers such as Bach, *missing name* Chopin, Beethoven, Shumann, Rakmaninov, Ravel and Prokoviev

Ref2 musicians such as Bach, Mozart, Chopin, Bethoven, Shuman, Rachmaninoff, Rafael and Brokoviev

Ref3 composers including Bach, Mozart, Schopen, Beethoven, *missing name* Raphael, Rahmaniev and Brokofien

Ref4 composers such as Bach, Mozart, *missing name* Beethoven, Schumann, Rachmaninov, Raphael and Prokofiev

The task of transliterating names (independent of end-to-end MT) has received a significant amount of research, e.g., (Knight and Graehl, 1997; Chen et al., 1998; Al-Onaizan, 2002). One approach is to "sound out" words and create new, plausible target-language spellings that preserve the sounds of the source-language name as much as possible. Another approach is to phonetically match source-language names against a large list of target-language words

[1] taken from NIST02-05 corpora

Proceedings of ACL-08: HLT, pages 389–397,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

and phrases. Most of this work has been disconnected from end-to-end MT, a problem which we address head-on in this paper.

The simplest way to integrate name handling into SMT is: (1) run a named-entity identification system on the source sentence, (2) transliterate identified entities with a special-purpose transliteration component, and (3) run the SMT system on the source sentence, as usual, but when looking up phrasal translations for the words identified in step 1, instead use the transliterations from step 2.

Many researchers have attempted this, and it does not work. Typically, translation quality is degraded rather than improved, for the following reasons:

- Automatic named-entity identification makes errors. Some words and phrases that should not be transliterated are nonetheless sent to the transliteration component, which returns a bad translation.

- Not all named entities should be transliterated. Many named entities require a mix of transliteration and translation. For example, in the pair جنوب كاليفورنيا/jnub kalyfurnya/Southern California, the first Arabic word is translated, and the second word is transliterated.

- Transliteration components make errors. The base SMT system may translate a commonly-occurring name just fine, due to the bitext it was trained on, while the transliteration component can easily supply a worse answer.

- Integration hobbles SMT's use of longer phrases. Even if the named-entity identification and transliteration components operate perfectly, adopting their translations means that the SMT system may no longer have access to longer phrases that include the name. For example, our base SMT system translates رئيس الوزراء لي بنغ (as a whole phrase) to "Premier Li Peng", based on its bitext knowledge. However, if we force لي بنغ to translate as a separate phrase to "Li Peng", then the term رئيس الوزراء becomes ambiguous (with translations including "Prime Minister", "Premier", etc.), and we observe incorrect choices being subsequently made.

To spur better work in name handling, an ACE entity-translation pilot evaluation was recently developed (Day, 2007). This evaluation involves a mixture of entity identification and translation concerns—for example, the scoring system asks for coreference determination, which may or may not be of interest for improving machine translation output.

In this paper, we adopt a simpler metric. We ask: *what percentage of source-language named entities are translated correctly?* This is a precision metric. We can readily apply it to any base SMT system, and to human translations as well. Our goal in augmenting a base SMT system is to increase this percentage. A secondary goal is to make sure that our overall translation quality (as measured by BLEU) does not degrade as a result of the name-handling techniques we introduce. We make all our measurements on an Arabic/English newswire translation task.

Our overall technical approach is summarized here, along with references to sections of this paper:

- We build a component for transliterating between Arabic and English (Section 3).
- We automatically learn to tag those words and phrases in Arabic text, which we believe the transliteration component will translate correctly (Section 4).
- We integrate suggested transliterations into the base SMT search space, with their use controlled by a feature function (Section 5).
- We evaluate both the base SMT system and the augmented system in terms of entity translation accuracy and BLEU (Sections 2 and 6).

2 Evaluation

In this section we present the evaluation method that we use to measure our system and also discuss challenges in name transliteration evaluation.

2.1 NEWA Evaluation Metric

General MT metrics such as BLEU, TER, METEOR are not suitable for evaluating named entity translation and transliteration, because they are not focused on named entities (NEs). Dropping a comma or a *the* is penalized as much as dropping a name. We therefore use another metric, jointly developed with BBN and LanguageWeaver.

The general idea of the Named Entity Weak Accuracy (NEWA) metric is to

- Count number of NEs in source text: N
- Count number of correctly translated NEs: C
- Divide C/N to get an accuracy figure

In NEWA, an NE is counted as correctly translated if the target reference NE is found in the MT output. The metric has the advantage that it is easy to compute, has no special requirements on an MT system (such as depending on source-target word alignment) and is tokenization independent.

In the result section of this paper, we will use the NEWA metric to measure and compare the accuracy of NE translations in our end-to-end SMT translations and four human reference translations.

2.2 Annotated Corpus

BBN kindly provided us with an annotated Arabic text corpus, in which named entities were marked up with their type (e.g. GPE for Geopolitical Entity) and one or more English translations. Example:

فى <GPE alt=”Termoli”> تيرمولى</GPE>
<PER alt=”Abdullah II | Abdallah II”> عبد الله
الثانى</PER>

The BBN annotations exhibit a number of issues. For the English translations of the NEs, BBN annotators looked at human reference translations, which may introduce a bias towards those human translations. Specifically, the BBN annotations are sometimes wrong, because the reference translations were wrong. Consider for example the Arabic phrase مصنع بورتران فى تيرمولى (mSn‘ burtran fY tyrmulY), which means *Powertrain plant in Termoli*. The mapping from *tyrmulY* to *Termoli* is not obvious, and even less the one from *burtran* to *Powertrain*. The human reference translations for this phrase are

1. Portran site in Tremolo
2. Termoli plant *(one name dropped)*
3. Portran in Tirnoli
4. Portran assembly plant, in Tirmoli

The BBN annotators adopted the correct translation *Termoli*, but also the incorrect *Portran*. In other cases the BBN annotators adopted both a correct (Khatami) and an incorrect translation (Khatimi) when referring to the former Iranian president, which would reward a translation with such an incorrect spelling.

- <PER alt=”Khatami|Khatimi”> خاتمى</PER>
- <GPE alt=”the American”> الاميركية</GPE>

In other cases, all translations are correct, but additional correct translations are missing, as for “the American” above, for which “the US” is an equally valid alternative in the specific sentence it was annotated in.

All this raises the question of what **is** a correct answer. For most Western names, there is normally only one correct spelling. We follow the same conventions as standard media, paying attention to how an organization or individual spells its own name, e.g. Senator Jon Kyl, not Senator John Kyle. For Arabic names, variation is generally acceptable if there is no one clearly dominant spelling in English, e.g. Gaddafi|Gadhafi|Qaddafi|Qadhafi, as long as a given variant is not radically rarer than the most conventional or popular form.

2.3 Re-Annotation

Based on the issues we found with the BBN annotations, we re-annotated a sub-corpus of 637 sentences of the BBN gold standard.

We based this re-annotation on detailed annotation guidelines and sample annotations that had previously been developed in cooperation with LanguageWeaver, building on three iterations of test annotations with three annotators.

We checked each NE in every sentence, using human reference translations, automatic transliterator output, performing substantial Web research for many rare names, and checked Google ngrams and counts for the general Web and news archives to determine whether a variant form met our threshold of occurring at least 20% as often as the most dominant form.

3 Transliterator

This section describes how we transliterate Arabic words or phrases. Given a word such as رحمانينوف or a phrase such as موريس رافيل, we want to find the English transliteration for it. This is not just a romanization like *rHmanynuf* and *murys rafyl* for the examples above, but a properly spelled English name such as *Rachmaninoff* and *Maurice Ravel*. The transliteration result can contain several alternatives, e.g. *Rachmaninoff|Rachmaninov*. Unlike various generative approaches (Knight and Graehl, 1997; Stalls and Knight, 1998; Li et al., 2004; Matthews, 2007; Sherif and Kondrak, 2007; Kashani et al., 2007), we do not synthesize an English spelling from scratch, but rather find a translation in very large lists of English words (3.4 million) and phrases (47 million).

We develop a similarity metric for Arabic and English words. Since matching against millions of candidates is computationally prohibitive, we store the English words and phrases in an index, such that given an Arabic word or phrase, we quickly retrieve a much smaller set of likely candidates and apply our similarity metric to that smaller list.

We divide the task of transliteration into two steps: given an Arabic word or phrase to transliterate, we (1) identify a list of English transliteration candidates from indexed lists of English words and phrases with counts (section 3.1) and (2) compute for each English name candidate the cost for the Arabic/English name pair (transliteration scoring model, section 3.2).

We then combine the count information with the transliteration cost according to the formula:

score(e) = log(count(e))/20 - translit_cost(e,f)

3.1 Indexing with consonant skeletons

We identify a list of English transliteration candidates through what we call a *consonant skeleton* index. Arabic consonants are divided into 11 classes, represented by letters b,f,g,j,k,l,m,n,r,s,t. In a one-time pre-processing step, all 3,420,339 (unique) English words from our English unigram language model (based on Google's Web terabyte ngram collection) that might be names or part of names (mostly based on capitalization) are mapped to one or more skeletons, e.g.

Rachmaninoff → rkmnnf, rmnnf, rsmnnf, rtsmnnf

This yields 10,381,377 skeletons (average of 3.0 per word) for which a reverse index is created (with counts). At run time, an Arabic word to be transliterated is mapped to its skeleton, e.g.

رحمانينوف → rmnnf

This skeleton serves as a key for the previously built reverse index, which then yields the list of English candidates with counts:

rmnnf → Rachmaninov (186,216), Rachmaninoff (179,666), Armenonville (3,445), Rachmaninow (1,636), plus 8 others.

Shorter words tend to produce more candidates, resulting in slower transliteration, but since there are relatively few unique short words, this can be addressed by caching transliteration results.

The same consonant skeleton indexing process is applied to name bigrams (47,700,548 unique with 167,398,054 skeletons) and trigrams (46,543,712 unique with 165,536,451 skeletons).

3.2 Transliteration scoring model

The cost of an Arabic/English name pair is computed based on 732 rules that assign a cost to a pair of Arabic and English substrings, allowing for one or more context restrictions.

1. ق::q == ::0
2. وف::ough == ::0
3. ح::ch == :[aou],::0.1
4. ق::k == ,$:,$::0.1 ; ::0.2
5. ء:: == :,EC::0.1

The first example rule above assigns to the straightforward pair ق/q a cost of 0. The second rule includes 2 letters on the Arabic and 4 on the English side. The third rule restricts application to substring pairs where the English side is preceded by the letters a, o, or u. The fourth rule specifies a cost of 0.1 if the substrings occur at the end of (both) names, 0.2 otherwise. According to the fifth rule, the Arabic letter ء may match an empty string on the English side, if there is an English consonant (EC) in the right context of the English side.

The total cost is computed by always applying the longest applicable rule, without branching, resulting in a linear complexity with respect to word-pair length. Rules may include left and/or right context for both Arabic and English. The match fails if no rule applies or the accumulated cost exceeds a preset limit.

Names may have *n* words on the English and *m* on the Arabic side. For example, *New York* is one word in Arabic and *Abdullah* is two words in Arabic. The rules handle spaces (as well as digits, apostrophes and other non-alphabetic material) just like regular alphabetic characters, so that our system can handle cases like where words in English and Arabic names do not match one to one.

The French name *Beaujolais* (بوجوليه/bujulyh) deviates from standard English spelling conventions in several places. The accumulative cost from the rules handling these deviations could become prohibitive, with each cost element penalizing the same underlying offense — being French. We solve this problem by allowing for additional context in the form of *style flags*. The rule for matching eau/و specifies, in addition to a cost, an (output) style flag +fr (as in French), which in turn serves as an additional context for the rule that matches ais/يه at a much reduced cost. Style flags are also used for some Arabic dialects. Extended characters such as é, ö, and ş and spelling idiosyncrasies in names on the English side of the bitext that come from various third languages account for a significant portion of the rule set.

Casting the transliteration model as a scoring problem thus allows for very powerful rules with strong contexts. The current set of rules has been built by hand based on a bitext development corpus; future work might include deriving such rules automatically from a training set of transliterated names.

This transliteration scoring model described in this section is used in two ways: (1) to transliterate names at SMT decoding time, and (2) to identify transliteration pairs in a bitext.

4 Learning what to transliterate

As already mentioned in the introduction, named entity (NE) identification followed by MT is a bad idea. We don't want to identify NEs per se anyway — we want to identify things that our transliterator will be good at handling, i.e., things that should be transliterated. This might even include loanwords like *bnk (bank)* and *brlman (parliament)*, but would exclude names such as *National Basketball Association* that are often translated rather transliterated.

Our method follows these steps:

1. Take a bitext.
2. Mark the Arabic words and phrases that have a recognizable transliteration on the English side.
3. Remove the English side of the bitext.
4. Divide the annotated Arabic corpus into a training and test corpus.
5. Train a monolingual Arabic tagger to identify which words and phrases (in running Arabic) are good candidates for transliteration (section 4.2)
6. Apply the tagger to test data and evaluate its accuracy.

4.1 Mark-up of bitext

Given a tokenized (but unaligned and mixed-case) bitext, we mark up that bitext with links between Arabic and English words that appear to be transliterations. In the following example, linked words are underlined, with numbers indicating what is linked.

English The meeting was attended by Omani (1) Secretary of State for Foreign Affairs Yusif (2) bin (3) Alawi (6) bin (8) Abdallah (10) and Special Advisor to Sultan (12) Qabus (13) for Foreign Affairs Umar (14) bin (17) Abdul Munim (19) al-Zawawi (21).

Arabic (translit.) uHDr allqa’ uzyr aldule al‘manY (1) llsh’uun alkharjye yusf (2) bn (3) ‘luY (6) bn (8) ‘bd allh (10) ualmstshar alkhaS llslTan (12) qabus (13) ll‘laqat alkharjye ‘mr (14) bn (17) ‘bd almn‘m (19) alzuauY (21) .

For each Arabic word, the linking algorithm tries to find a matching word on the English side, using the transliteration scoring model described in section 3. If the matcher reaches the end of an Arabic or English word before reaching the end of the other, it continues to "consume" additional words until a word-boundary observing match is found or the cost threshold exceeded.

When there are several viable linking alternatives, the algorithm considers the cost provided by the transliteration scoring model, as well as context to eliminate inferior alternatives, so that for example the different occurrences of the name particle *bin* in the example above are linked to the proper Arabic words, based on the names next to them. The number of links depends, of course, on the specific corpus, but we typically identify about 3.0 links per sentence.

The algorithm is enhanced by a number of heuristics:

- English match candidates are restricted to capitalized words (with a few exceptions).
- We use a list of about 200 Arabic and English stopwords and stopword pairs.
- We use lists of countries and their adjective forms to bridge cross-POS translations such as *Italy's president* on the English and رئيس الايطالى (”*Italian president*”) on the Arabic side.
- Arabic prefixes such as ل/l- (”to”) are treated in a special way, because they are translated, not transliterated like the rest of the word. Link (12) above is an example.

In this bitext mark-up process, we achieve 99.5% precision and 95% recall based on a manual visualization-tool based evaluation. Of the 5% recall error, 3% are due to noisy data in the bitext such as typos, incorrect translations, or names missing on one side of the bitext.

4.2 Training of Arabic name tagger

The task of the Arabic name tagger (or more precisely, “transliterate-me” tagger) is to predict whether or not a word in an Arabic text should be transliterated, and if so, whether it includes a prefix. Prefixes such as و/u- (“and”) have to be translated rather than transliterated, so it is important to split off any prefix from a name before transliterating that name. This monolingual tagging task is not trivial, as many Arabic words can be both a name and a non-name. For example, الجزيرة (aljzyre) can mean both *Al-Jazeera* and *the island (or peninsula)*.

Features include the word itself plus two words to the left and right, along with various prefixes, suffixes and other characteristics of all of them, totalling about 250 features.

Some of our features depend on large corpus statistics. For this, we divide the tagged Arabic side of our training corpus into a *stat section* and a core training section. From the stat section we collect statistics as to how often every word, bigram or trigram occurs, and what distribution of name/non-name patterns these ngrams have. The name distribution bigram

الجزيرة الكورية 3327 00:133 01:3193 11:1

(aljzyre alkurye/“peninsula Korean”) for example tells us that in 3193 out of 3327 occurrences in the stat corpus bitext, the first word is a marked up as a non-name (”0”) and the second as a name (”1”), which strongly suggests that in such a bigram context, *aljzyre* better be **translated** as island or peninsula, and not be **transliterated** as Al-Jazeera.

We train our system on a corpus of 6 million stat sentences, and $500,000$ core training sentences. We employ a sequential tagger trained using the SEARN algorithm (Daumé III et al., 2006) with aggressive updates ($\beta = 1$). Our base learning algorithm is an averaged perceptron, as implemented in the MEGAM package[2].

Reference	Precision	Recall	F-meas.
Raw test corpus	87.4%	95.7%	91.4%
Adjusted for GS deficiencies	92.1%	95.9%	94.0%

Table 1: Accuracy of “transliterate-me” tagger

Testing on 10,000 sentences, we achieve precision of 87.4% and a recall of 95.7% with respect to the automatically marked-up Gold Standard as described in section 4.1. A manual error analysis of 500 sentences shows that a large portion are not errors after all, but have been marked as errors because of noise in the bitext and errors in the bitext markup. After adjusting for these deficiencies in the gold standard, we achieve precision of 92.1% and recall of 95.9% in the name tagging task.

5 Integration with SMT

We use the following method to integrate our transliterator into the overall SMT system:

1. We tag the Arabic source text using the tagger described in the previous section.

2. We apply the transliterator described in section 3 to the tagged items. We limit this transliteration to words that occur up to 50 times in the training corpus for single token names (or up to 100 and 150 times for two and three-word names). We do this because the general SMT mechanism tends to do well on more common names, but does poorly on rare names (and will

[2] Freely available at `http://hal3.name/megam`

always drop names it has never seen in the training bitext).

3. On the fly, we add transliterations to SMT phrase table. Instead of a phrasal probability, the transliterations have a special binary feature set to 1. In a tuning step, the Minimim Error Rate Training component of our SMT system iteratively adjusts the set of rule weights, including the weight associated with the transliteration feature, such that the English translations are optimized with respect to a set of known reference translations according to the BLEU translation metric.

4. At run-time, the transliterations then compete with the translations generated by the general SMT system. This means that the MT system will not always use the transliterator suggestions, depending on the combination of language model, translation model, and other component scores.

5.1 Multi-token names

We try to transliterate names as much as possible in context. Consider for example the Arabic name:
يوسف ابو صفية (”yusf abu Sfye”)
If transliterated as single words without context, the top results would be Joseph|Josef|Yusuf|Yosef|Youssef, Abu|Abo|Ivo|Apo|Ibo, and Sephia|Sofia|Sophia|Safieh|Safia respectively. However, when transliterating the three words together against our list of 47 million English trigrams (section 3), the transliterator will select the (correct) translation *Yousef Abu Safieh*. Note that *Yousef* was not among the top 5 choices, and that *Safieh* was only choice 4.

Similarly, when transliterating وموزار وشوبان /umuzar ushuban (”and Mozart and Chopin”) without context, the top results would be Moser|Mauser|Mozer|Mozart|Mouser and Shuppan|Shopping|Schwaben|Schuppan|Shobana (with *Chopin* way down on place 22). Checking our large English lists for a matching *name, name* pattern, the transliterator identifies the correct translation “, *Mozart, Chopin*”. Note that the transliteration module provides the overall SMT system with up to 5 alternatives, augmented with a choice of English translations for the Arabic prefixes like the comma and the conjunction *and* in the last example.

6 End-to-End results

We applied the NEWA metric (section 2) to both our SMT translations as well as the four human reference translations, using both the original named-entity translation annotation and the re-annotation:

Gold Standard	BBN GS	Re-annotated GS
Human 1	87.0%	85.0%
Human 2	85.3%	86.9%
Human 3	90.4%	91.8%
Human 4	86.5%	88.3%
SMT System	80.4%	89.7%

Table 2: Name translation accuracy with respect to BBN and re-annotated Gold Standard on 1730 named entities in 637 sentences.

Almost all scores went up with re-annotations, because the re-annotations more properly reward correct answers.

Based on the original annotations, all human name translations were much better than our SMT system. However, based on our re-annotation, the results are quite different: our system has a higher NEWA score and better name translations than 3 out of 4 human annotators.

The evaluation results confirm that the original annotation method produced a relative bias towards the human translation its annotations were largely based on, compared to other translations.

Table 3 provides more detailed NEWA results. The addition of the transliteration module improves our overall NEWA score from 87.8% to 89.7%, a relative gain of 16% over base SMT system. For names of persons (PER) and facilities (FAC), our system outperforms all human translators. Humans performed much better on Person Nominals (PER.Nom) such as *Swede*, *Dutchmen*, *Americans*. Note that name translation quality varies greatly between human translators, with error rates ranging from 8.2-15.0% (absolute).

To make sure our name transliterator does not degrade the overall translation quality, we evaluated our base SMT system with BLEU, as well as our transliteration-augmented SMT system. Our standard newswire training set consists of 10.5 million words of bitext (English side) and 1491 test sen-

NE Type	Count	Baseline SMT	SMT with Transliteration	Human 1	Human 2	Human 3	Human 4
PER	342	266 (77.8%)	280 (81.9%)	210 (61.4%)	265 (77.5%)	278 (81.3%)	275 (80.4%)
GPE	910	863 (94.8%)	877 (96.4%)	867 (95.3%)	849 (93.3%)	885 (97.3%)	852 (93.6%)
ORG	332	280 (84.3%)	282 (84.9%)	263 (79.2%)	265 (79.8%)	293 (88.3%)	281 (84.6%)
FAC	27	18 (66.7%)	24 (88.9%)	21 (77.8%)	20 (74.1%)	22 (81.5%)	20 (74.1%)
PER.Nom	61	49 (80.3%)	48 (78.7%)	61 (100.0%)	56 (91.8%)	60 (98.4%)	57 (93.4%)
LOC	58	43 (74.1%)	41 (70.7%)	48 (82.8%)	48 (82.8%)	51 (87.9%)	43 (74.1%)
All types	1730	1519 (87.8%)	1552 (89.7%)	1470 (85.0%)	1503 (86.9%)	1589 (91.8%)	1528 (88.3%)

Table 3: Name translation accuracy in end-to-end statistical machine translation (SMT) system for different named entity (NE) types: Person (PER), Geopolitical Entity, which includes countries, provinces and towns (GPE), Organization (ORG), Facility (FAC), Nominal Person, e.g. *Swede* (PER.Nom), other location (LOC).

tences. The BLEU scores for the two systems were 50.70 and 50.96 respectively.

Finally, here are end-to-end machine translation results for three sentences, with and without the transliteration module, along with a human reference translation.

Old: Al-Basha leads a broad list of musicians such as Bach.
New: Al-Basha leads a broad list of musical acts such as Bach, Mozart, Beethoven, Chopin, Schumann, Rachmaninoff, Ravel and Prokofiev.
Ref: Al-Bacha performs a long list of works by composers such as Bach, Chopin, Beethoven, Shumann, Rakmaninov, Ravel and Prokoviev.

Old: Earlier Israeli military correspondent turn introduction programme ”Entertainment Bui”
New: Earlier Israeli military correspondent turn to introduction of the programme ”Play Boy”
Ref: Former Israeli military correspondent turns host for ”Playboy” program

Old: The Nikkei president company De Beers said that ...
New: The company De Beers chairman Nicky Oppenheimer said that ...
Ref: Nicky Oppenheimer, chairman of the De Beers company, stated that ...

7 Discussion

We have shown that a state-of-the-art statistical machine translation system can benefit from a dedicated transliteration module to improve the translation of rare names. Improved named entity translation accuracy as measured by the NEWA metric in general, and a reduction in dropped names in particular is clearly valuable to the human reader of machine translated documents as well as for systems using machine translation for further information processing. At the same time, there has been no negative impact on overall quality as measured by BLEU.

We believe that all components can be further improved, e.g.

- Automatically retune the weights in the transliteration scoring model.
- Improve robustness with respect to typos, incorrect or missing translations, and badly aligned sentences when marking up bitexts.
- Add more features for learning whether or not a word should be transliterated, possibly using source language morphology to better identify non-name words never or rarely seen during training.

Additionally, our transliteration method could be applied to other language pairs.

We find it encouraging that we already outperform some professional translators in name translation accuracy. The potential to exceed human translator performance arises from the patience required to translate names right.

Acknowledgment

This research was supported under DARPA Contract No. HR0011-06-C-0022.

References

Yaser Al-Onaizan and Kevin Knight. 2002. Machine Transliteration of Names in Arabic Text. In *Proceedings of the Association for Computational Linguistics Workshop on Computational Approaches to Semitic Languages*.

Thorsten Brants, Alex Franz. 2006. Web 1T 5-gram Version 1. Released by Google through the Linguistic Data Consortium, Philadelphia, as LDC2006T13.

Hsin-Hsi Chen, Sheng-Jie Huang, Yung-Wei Ding, and Shih-Chung Tsai. 1998. Proper Name Translation in Cross-Language Information Retrieval. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*.

Hal Daumé III, John Langford, and Daniel Marcu. 2006. Search-based Structured Prediction. Submitted to the *Machine Learning Journal*. http://pub.hal3.name/#daume06searn

David Day. 2007. Entity Translation 2007 Pilot Evaluation (ET07). In proceedings of the Workshop on Automatic Content Extraction (ACE). College Park, Maryland.

Byung-Ju Kang and Key-Sun Choi. 2000. Automatic Transliteration and Back-transliteration by Decision Tree Learning. In *Conference on Language Resources and Evaluation*.

Mehdi M. Kashani, Fred Popowich, and Fatiha Sadat. 2007. Automatic Transliteration of Proper Nouns from Arabic to English. *The Challenge of Arabic For NLP/MT, 76-84*.

Alexandre Klementiev and Dan Roth. 2006. Named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*.

Kevin Knight and Jonathan Graehl. 1997. Machine Transliteration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*.

Li Haizhou, Zhang Min, and Su Jian. 2004. A Joint Source-Channel Model for Machine Transliteration. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.

Wei-Hao Lin and Hsin-Hsi Chen. 2002. Backward Machine Transliteration by Learning Phonetic Similarity. *Sixth Conference on Natural Language Learning*, Taipei, Taiwan, 2002.

David Matthews. 2007. Machine Transliteration of Proper Names. Master's Thesis. School of Informatics. University of Edinburgh.

Masaaki Nagata, Teruka Saito, and Kenji Suzuki. 2001. Using the Web as a Bilingual Dictionary. In *Proceedings of the Workshop on Data-driven Methods in Machine Translation*.

Bruno Pouliquen, Ralf Steinberger, Camelia Ignat, Irina Temnikova, Anna Widiger, Wajdi Zaghouani, and Jan Zizka. 2006. Multilingual Person Name Recognition and Transliteration. CORELA - COgnition, REpresentation, LAnguage, Poitiers, France. Volume 3/3, number 2, pp. 115-123.

Tarek Sherif and Grzegorz Kondrak. 2007. Substring-Based Transliteration. In *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*.

Richard Sproat, ChengXiang Zhai, and Tao Tao. 2006. Named Entity Transliteration with Comparable Corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting on Association for Computational Linguistics*.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating Names and Technical Terms in Arabic Text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*.

Stephen Wan and Cornelia Verspoor. 1998. Automatic English-Chinese Name Transliteration for Development of Multilingual Resources. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*. Montreal, Canada.