Holy Moses! Leveraging Existing Tools and Resources for Entity Translation

Jean Tavernier[1], Rosa Cowan[1], Michelle Vanni[2]
[1]CACI International Inc., [2]US Army Research Laboratory
4831 Walden Lane, Lanham, MD 22207, USA
E-mail: {jtavernier,rcowan}@caci.com, mvanni@arl.army.mil

Abstract

Recently, there has been an emphasis on creating shared resources for natural language processing applications. This has resulted in the development of high-quality tools and data, which can then be leveraged by the research community as components for novel systems. In this paper, we reuse an open source machine translation framework to create an Arabic-to-English entity translation system. The system first translates known entity mentions using a standard phrase-based statistical machine translation framework, which is then reused to perform name transliteration on unknown mentions. In order to transliterate names more accurately, we introduce an algorithm to augment a names database with name origin and frequency information from existing data resources. Origin information is used to learn name origin classifiers and origin-specific transliteration models, while frequency information is used to select amongst n-best transliteration candidates. This work demonstrates the feasibility and benefit of adapting such data resources and shows how off-the-shelf tools and data resources can be repurposed to rapidly create a system outside their original domain.

1. Introduction

The entity translation problem has received increasing attention recently. In early 2007, the U.S. National Institute of Standards and Technology (NIST) added an entity translation (ET) pilot evaluation to its Automatic Content Extraction (ACE) evaluations (NIST, 2007b), in order to evaluate systems for cross-language information extraction. By accurately translating entity mentions, a number of applications benefit, including machine translation (MT) and information retrieval (Hassan, Fahmy & Hassan, 2007).

Entity translation (ET) systems translate or transliterate source language entity mentions into a target language. Typically, this is done in situations where the source and target language do not share a common script. The ET problem is non-trivial for a number of reasons. First, new names are continuously being introduced (Al-Onaizan & Knight, 2002), making complete coverage by a MT system impossible. Second, there may be multiple ways of rendering a source language name into a target name, as is the case with the name إسماعيل, which can be represented as Ismail or Ismael in English. Likewise, the same target language name can be represented by multiple source language renderings, as is the case with Jean, which can be written جان or جون in Arabic.

Most approaches to ET have used a statistical translation model to generate name transliterations, but these either use a custom translation formalism (Al-Onaizan & Knight, 2002; Jiang et al, 2007) or they leverage tools which are not publicly available (Ji et al, 2007). To deal with insufficient transliteration accuracy, previous approaches have leveraged online resources to generate additional translation candidates. Al-Onaizan and Knight (2002) and Hassan et al (2007) did this by searching for comparable corpora and then attempting to discover an appropriate alignment with the source entity. (Jiang et al, 2007) attempts to discover content where both the source and target entity exist on the same page while (Huang 2005) showed that classifying names by their likely origins and then generating the transliteration from an origin-specific model significantly improves transliteration accuracy.

Our approach makes use of similar techniques, but leverages existing tools and resources. This allows us to build a simple but competitive system in a short amount of time. We use an open source MT tool to learn phrase-based statistical MT (SMT) models to translate known entity mentions. We then reuse the same tool to learn character transliteration models to translate unknown mentions. Rather than looking to the web for additional translation candidates, we select amongst n-best variants by using freely available name frequency resources.

The remainder of the paper details this system and describes its performance characteristics: In section 2, we briefly give an overview of the system, followed by a section detailing the resources and tools used to build it. Section 4 outlines the runtime system. We describe an evaluation of our system in Section 5, followed by a discussion of the results.

2. Entity Translation System Overview

We view the ET task as requiring two major stages of processing: 1) translation of mentions and entities made up of "ordinary" words and phrases (known named, nominal and pronominal references), and 2) specialized entity rendering for entity mentions where MT is not able to provide a translation, which is often the case for named mentions. To address the first processing stage, we build phrase-based SMT models using Moses (Koehn et al, 2007), an open-source phrase-based SMT system and available data. To address the second processing stage, we construct a names database from existing entity dictionaries and parallel corpora that have been aligned at the sentence and entity level. The names database is then augmented with monolingual name origin information adapted from a Chinese-English entity list. Using this names database, a single generic model of Arabic-English name transliteration is built, again using Moses. To provide more precise translations, origin information is used to build name-origin classifiers and origin-specific name transliteration models. Both the generic and origin-specific transliteration models produce multiple English translation variants for a given Arabic name, which are re-ranked using monolingual frequency information for English names.

3. System components

This section describes the development of three primary components of our system: a phrase-based statistical machine translation model, a names database and name transliteration models.

3.1 Statistical Machine Translation

We trained a phrase-based SMT engine to translate known words and phrases using the training tools available with Moses (Koehn et al, 2007). Moses is trained using sentence aligned parallel text. One of the first steps in the training process is an alignment of words and phrases in parallel text, performed by GIZA++ (Och & Ney, 2003). The output of the training phase results in a four-part model, including a phrase translation table, an English language model, a reordering model to constrain distortion, and a word penalty to control the length of the translation.

3.2 Building a Names Database

The names database is used at training time and run time. At training time, it's used to provide training data for name transliteration training and at run time, it's used to compare an “n-best” list of name transliterations and select the most likely candidate. Because of this dual role, both bilingual and monolingual data resources are added to the names database. Table 1 provides a partial listing of data resources used to build the names database, where “Para.” indicates that the resource has parallel Arabic-English entries, “Freq.” means the resource contains frequency information and “Origin Names” gives a count of names with implicit or explicit origin labels for the names. Entries from the ISI and UN resources are added by performing named entity extraction using BBN Identifinder, a commercial NEE tool. This tool extracts named entities from the UN and ISI corpora, then entities are aligned using a custom alignment procedure.

Resource	Para.	Freq.	Origin Names	Total Names
NMSU	Y	N	851	2,186
1990 US Census	N	Y	94,293	94,293
Chinese <-> English Names Entity List	N	N	66,909	873,310
ISI and UN Sentence Aligned Text	Y	N	0	67,763

Table 1 - Summary of names database data resources.

For cases in which a name entry consists of multiple parts, GIZA++ is used to intra-align name parts. This alignment is carried out on the name parts stemmed to six characters, in order to reduce data sparseness. Following alignment, database entries are created for each of the aligned name parts with each of the new entries inheriting the origin information of the parent entry. For example, if the following names were collected from bilingual and monolingual sources:

Arabic	**English**	**Origin**
جون وغابي	John Ugabe	Kenya
جون وياثيرينغتون	John Weatherington	UK
	Jaoxping John Lin	China

The following entries would be created in the database:

Arabic	**English**	**Origin**
جون	John	Kenya,UK,China
وغابي	Ugabe	Kenya
وياثيرينغتون	Weatherington	UK

Following the intra-name alignment process, database entries with identical English and Arabic name parts are de-duplicated, retaining origin and identity information from the de-duplicated entries. Entries with identical Arabic strings but different English forms are retained, since they reflect the fact that there may be multiple ways of representing a single Arabic form in English. Likewise, entries with identical English string but different Arabic forms are retained.

The development of multiple origin-specific name transliteration models requires additional steps. First, data sources of name origin information must be identified. Sources of name origin information need not be Arabic-English bilingual; as long as the resource provides explicit or implicit origin information, and either Arabic or English name renderings, the source can provide origin information. Table 1 provides a listing of sources used that contain name origin information. Note that the 1990 US Census Corpus is monolingual and the Chinese <-> English Name Entity Lists v 1.0 database does not contain Arabic name renderings, but we can re-use the origin information by matching on the English names available in both sets of resources. Names from the former are given a default label of 'USA'. Roughly 65K names in the Chinese-English resource contain explicit labels; in addition to those, names in the “Chinese Who's Who” list are given a default label of 'China'. A labeling algorithm applies the labels from the annotated monolingual names by exact string match of name parts; that is, if a name part from the origin-annotated monolingual name list matches a name part in the bilingual corpus, the origin labels from the monolingual name are applied to the entire bilingual name entry.

At run-time, we use name frequency information to select from “n-best” variants produced by a name transliteration model. The only frequency information included in the names database comes from the 1990 Census Bureau source. The Census Bureau frequency, if greater than zero, is assigned to the matching name in the names database. If the frequency is 0, a frequency of 1/2 of the smallest non-zero frequency is assigned. For all other names in the names database, including non-person entities, a frequency of 1/4 of the smallest non-zero frequency is assigned.

3.3 Training Name Transliteration models

We create generic and origin-specific name transliteration models using Moses, the same tool used for the MT model. To train these models using the training script provided with Moses, Arabic-English name pairs constitute training "sentences" with each character as a vocabulary item. Person names are almost always translated phonetically and therefore provide accurate training data for transliterating names. As such, they are used as the basis for name transliteration model training.

We first build a single generic name transliteration model using all parallel names database entries. Once the Arabic/English bilingual portion of the names database has origin annotations, we cluster name origins, using the procedure proposed by Huang (2005). The clustering procedure takes an origin-labeled bilingual names database as input, a threshold parameter c indicating the minimum number of names that must be labeled with a particular origin in order for it to be included in training, and a value for parameter M defining the desired number of clusters. We re-use Huang's parameter settings of c=50 and M=45, which serves as a “best guess,” because time constraints did not permit optimizing the parameters for our collected dataset.

For each origin (and each cluster), Arabic and English language models are built using the SRI Language Modeling Toolkit, and character-level phrase tables are trained using the Moses training scripts. The language models and phrase tables are used during clustering and in the runtime system.

4. Runtime System

This section describes our runtime system. First, documents are preprocessed to find sentence and token boundaries. Sentences that contain entity mentions are translated using the phrase-based SMT system[1]. Mention tokens that are unknown, or out of vocabulary (OOV), to the MT component are identified and translated using name transliteration models. Name transliteration may occur using either a single generic model or one of many origin-specific name transliteration models. For run time configurations using a generic transliteration model, origin classification is not needed.

For run-time configurations using origin-specific name transliteration, the token's origin is determined using the Arabic character language models built during the origin clustering process. Following the classification process laid out in Huang (2005), each OOV Arabic token is assigned a likelihood, for each origin-specific model, of being a valid string generated by that model. Each of these likelihoods is multiplied by an a priori probability for the corresponding origin-specific cluster. The a priori probabilities are estimated from the database of names with known origins, dividing the number of database entries in an origin-specific cluster by the total number of entries. The origin-specific cluster that generates the greatest probability product determines the origin-specific name transliteration model to use for that token's character translation. Name variants are then generated from this origin-specific model using the Moses decoder. Using functionality within the Moses decoder for both the generic and origin-specific cases, the system can be configured to produce an n-best list of name transliteration variants. With this configuration, Moses produces the n translations with the highest probability. The probability $P(e/a)$ of an English translation, e , given an Arabic input , a , is a log-linear combination of four feature functions: the phrase translation model Φ , English language model M , reordering model O , and weight penalty model W . Each of the four models is weighted by a model-specific weight.

$$P(e/a)=\Phi(a/e)^{\omega_\Phi}\times \mathrm{M}(e)^{\omega_\mathrm{M}}\times \mathrm{O}(e,a)^{\omega_\mathrm{O}}\times W(e)^{\omega_W}$$

Given a list of n-best translations, the system uses frequency information from each name variant contained in the names database to select among them. If the name variant is not in the database, a default frequency is assigned. In order to incorporate the name variant frequency information into a n-best selection score, $S(e_i)$, we multiply the frequency information by the translation probability $P(e_i / a)$. Thus for each name variant e_i in the n-best list, a selection score can be calculated as

$$S(e_i) = P(e_i / a)\times f(e_i)^{\omega_f}$$

where $f(e_i)$ is the frequency of variant e_i . The name frequency measure can also be viewed as an additional feature function used to compute the translation probability $P(e/a)$. A default value of 1 was used for ω_f in the evaluation described below, although this weight, as well as the other weights, can be calculated using minimum error rate training (Och, 2003), to produce an optimized set of weights for a given translation evaluation measure. The name variant resulting in the greatest selection score is chosen from the n-best list when n > 1.

5. Evaluation

This work implicitly makes three hypotheses: (1) name transliteration improves name translation accuracy compared with performing just MT; (2) utilizing origin-specific models improves name translation accuracy compared with generic models; and (3) incorporating a measure of name frequency or popularity improves name translation accuracy. To test these hypotheses, we evaluated a number of system configurations using the NIST ACE ET Diagnostic evaluation data (NIST, 2007b) and scoring software[2]. Details of the unweighted and value-based scoring methods are described in The ACE 2007 Evaluation Plan

[1] Note that the system assumes that entity mentions have already been identified in the source language, Arabic.

[2] The scoring software may be downloaded at: http://www.nist.gov/speech/tests/ace/ace07/software.html

(NIST, 2007a).

6. Results

Table 2 shows unweighted (UW) and value-based (VB) F-scores for a number of system variants. Note that all of the system variants that incorporate name transliteration improve on the baseline, verifying our first hypothesis. The improvement over the baseline is statistically significant (p-value is 0.0158 for the two top performing systems), which was determined using Fisher's randomization test (Smucker, Allan & Carterette, 2007).

Transliteration Model	**# Variants**	**UW F-Score**	**VW F-Score**
Generic	1000	0.486	0.227
Origin-specific	1000	0.485	0.225
Origin-specific	100	0.484	0.225
Generic	100	0.483	0.224
Generic	10	0.483	0.221
Origin-specific	10	0.482	0.221
Generic	1	0.483	0.220
Origin-specific	1	0.482	0.220
Baseline[3]		0.478	0.208

Table 2 - System variants evaluated with NIST ACE ET Diagnostic test data, sorted by Value-weighted F-Score

Our second hypothesis, that utilizing origin information improves name translation, cannot be verified. While origin-specific model performance is comparable to the generic model, and slightly better in the n=100 case, they do not generally outperform the generic name transliteration models. The reason may be that the origin-specific character models are smaller than the generic model and do not produce as diverse a set of name variants (in particular, when *n* is large). Another possible cause is that the smaller size of the origin-specific models means that their performance can be disproportionately influenced by noise introduced during names database construction.

Note that in all cases, increasing the number of candidate variants (and letting the frequency information select amongst them) improves results, verifying our third hypothesis. An order of magnitude increase in the number of generated variants seems to result in a linear accuracy improvement, suggesting there is a logarithmic relationship between the number of variants and accuracy. However, because of the limited nature of our frequency information (a list of personal names occurring in the United States), we would expect accuracy improvements to stop as more-and-more unlikely variants are generated, some of which may be popular but incorrect.

Entity Type	**Baseline**	**Origins (n=1000)**	**Generic (n=1000)**
Facility	12	13.4	**14.1**
GPE	41.6	**42.8**	42.7
Location	**14.1**	**14.1**	13.9
Organization	13.3	**15.6**	15.4
Person	13.1	15.1	**15.6**
Vehicle	6.7	**9.4**	**9.4**
Weapon	10.8	**11.8**	11.7
Total	20.8	22.5	**22.7**

Table 3 - Value-weighted scores for the two best performing systems, and the baseline system, by entity type (with best scores in bold)

We also wanted to investigate the efficacy of utilizing transliteration models trained only on person names to correctly translate other types of entities. Table 3 shows the entity type specific performance (using the value-weighted metric) of the two best performing systems, versus the baseline system. Both origin-specific and generic transliteration models improve translation accuracy 10% or more for Facility, Organization, Person and Vehicle entity types, demonstrating that transliteration models built from personal name translations generalize well to other entity types. The effect for GPE, Location and Weapon types either does not exist or is dampened. One likely reason for this is that the primary data source for our MT system (United Nations documents) mentions these types of entities frequently, making it difficult for the transliteration models to improve upon the baseline accuracy. In addition, the nature of the GPE (e.g., nation names) and Location (e.g., mountain names) types is such that new names are introduced less frequently than other types, making a sufficiently trained MT system suitable for name translation.

7. Conclusion

This paper demonstrates how open source tools and available data resources can be repurposed to rapidly produce an ET system. We utilized GIZA++, an open-source alignment tool, to construct a names database from various existing name resources and Moses, an open-source SMT framework, to train both phrase-based SMT and name transliteration models. The resulting ET system demonstrated its utility by significantly improving name translation accuracy over the baseline SMT system. In addition, we showed how a names database containing frequency information from freely available monolingual resources aids in discriminating amongst n-best variants.

8. Acknowledgements

Thanks to Dr. Kristen Summers for reviewing this paper and for contributing to earlier drafts of this paper, and to Leila Saidi for reviewing and correcting name database entries used in this work. Thanks also to the anonymous reviewers for their helpful suggestions.

This work was funded in whole or in part under U.S.

[3] The Baseline system uses phrase-based SMT without generic or origin-specific name transliteration.

Army contract #DAAD19-03-C-0059. The opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the U.S.G.

9. References

Al-Onaizan, Y., Knight K. (2002). Translating Named Entities Using Monolingual and Bilingual Resources. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Morristown, NJ: Association for Computational Linguistics, pp. 400--408.

Hassan, A., Fahmy H., Hassan, H. (2007). Improving Named Entity Translation by Exploiting Comparable and Parallel Corpora. In *Proceedings of the 2007 Workshop on Acquisition and Management of Multilingual Lexicons*. Borovetz, Bulgaria: pp. 2--7.

Huang, F. (2005). Cluster-Specific Named Entity Transliteration. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Morristown, NJ: Association for Computational Linguistics, pp. 435--442.

Ji, H., Blume M., Freitag, D., Grishman, R., Khadivi, S., Zens, R. (2007). NYU-Fair Isaac-RWTH Chinese to English Entity Translation 07 System. In *Proceedings of ACE ET 2007 PI/Evaluation Workshop*. Washington, USA.

Jiang, L., Zhou, M., Chien, L., Niu, C. (2007). Named Entity Translation with Web Mining and Transliteration. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*. Menlo Park, CA: International Joint Conferences on Artificial Intelligence and AAAI Press, pp. 1629--1634.

Koehn, P., Hoang H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Morristown, NJ: Association for Computational Linguistics, pp. 177--180.

NIST (2007). The ACE 2007 Evaluation Plan. http://www.nist.gov/speech/tests/ace/ace07/doc/ace07-evalplan.v1.3a.pdf.

NIST (2007). The Evaluation Plan for the ACE 2007 Pilot Evaluation of Entity Translation. http://www.nist.gov/speech/tests/ace/ace07/doc/ET07-evalplan-v1.8.pdf.

Och, F. J., Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1), pp. 19--51.

Och, F.J. (2003), Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Morristown, NJ: Association for Computational Linguistics, pp. 160--167.

Smucker, M.D., Allan, J., Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the Sixteenth ACM Conference on Conference on information and Knowledge Management*. New York, NY: ACM Press, pp. 623--632.

Stolcke, A. (2002). SRILM -- an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*. Denver, CO: pp. 901--904.