Bootstrapping Bilingual Data using Consensus Translation for a Multilingual Instant Messaging System

Srinivas Bangalore
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ
srini@research.att.com

Vanessa Murdock*
Computer Science Department
University of Massachusetts
Amherst, MA
vanessa@cs.umass.edu

Giuseppe Riccardi
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ
dsp3@research.att.com

Abstract

One of the primary issues in training statistical translation models is the paucity of bilingual data. In this paper, we propose techniques to alleviate the bilingual data bottleneck by creating a consensus from translations of monolingual data provided by several off-the-shelf translation engines. We compute the consensus alignment using a multi-sequence alignment algorithm used for DNA sequence alignment. We present an application of this technique to bootstrap bilingual data for the general domain of instant messaging. We train hierarchical statistical translation models on the bootstrapped bilingual data and show that the resulting statistical translation model outperforms each individual off-the-shelf translation system.

1 Introduction

In recent times, there has been an increased interest in translation systems that employ statistical models for mapping a source language sentence to a target language sentence (Brown et al., 1993; Ney, 1999; Alshawi et al., 1998b). The appeal of statistical translation approaches is that translation models can be trained directly from data without any human intervention, thus reducing the effort in building translation systems, in contrast to the effort involved in building hand-crafted translation systems. The models are trained using large bilingual corpora of parallel texts – sentence pairs in two languages that are known translations of each other. As a result, training a statistical translation model for a specific application requires the availability of a large parallel corpus for the application domain.

However, the data available for many applications is typically in only one language. Although there are bilingual corpora such as the English-French Hansard Corpus and the English-Chinese HongKong News Corpus for different language pairs that are known parallel texts, they may not be useful to a given application domain. Also, techniques that exploit parallel web pages (Resnik, 1999) to build a parallel corpus may not be well suited for translations in a given application domain. In previous work (Alshawi et al., 1998a), human translators translated application-specific monolingual data to create bilingual parallel data for training statistical translation models. However, such an approach is prohibitively expensive when large amounts of application-specific monolingual data needs to be translated as is the case in our application context.

An alternate resource for building parallel corpora is to use off-the-shelf translation systems to translate the monolingual data. It is to be expected that the quality of translation from such off-the-shelf translation engines on application-specific data is likely to be poor. Our approach in this paper is to combine the output of multiple such off-the-shelf translation engines (improving on the quality of each individual translation) to bootstrap a parallel corpus for an instant-messaging application.

In Bangalore et al. (2001), it has been shown that combining the translations provided by a variety of off-the-shelf machine translation (MT) engines on domain dependent spoken dialog transcriptions can improve upon the performance of each of the individual MT engines. In this paper, we apply the same techniques for data from a general domain obtained in the context of an instant messaging application, AT&T's instant messenger, Hubbub (Isaacs et

* Current address: Center for Intelligent Information Retrieval, University of Massachusetts

al., 2001a). Furthermore, we explore techniques to cope with the rapidly increasing amount of data available for training statistical translation models.

The layout of the paper follows. In Section 2 we describe the multilingual Hubbub instant messaging application and discuss other approaches to multilingual chat applications. In Section 3, we describe the multi-sequence alignment algorithm and its use in arriving at the consensus translation. The experimental set up and performance results from these experiments are presented in Section 4. We present our conclusions in Section 5.

2 Multilingual Instant Messaging Systems

Hubbub is an instant messenger developed at AT&T Labs-Research (Isaacs et al., 2001b; Isaacs et al., 2001a). It can be freely downloaded from http://www.hubbubme.com. Hubbub supports two-party conversations with the users conversing in the same language. We have extended Hubbub to allow for conversations between users of different languages, in particular English and Spanish, by incorporating a statistical translation system into Hubbub. In this paper, we discuss the process by which we bootstrapped the bilingual data needed for training statistical translation models from the 58,300 English sentences collected from conversations among users of Hubbub. These sentences are typically short and spontaneous and use a variety of slang and informal language. Furthermore, these conversations do not adhere to any particular topic. Also, due to an increase in the popularity of instant messengers and chat systems, there is a large quantity of monolingual data to be gleaned from these conversations.

Existing chat and instant-messaging systems (Lotus, 2001; Kellerman and Mayeur, 2000; GermanMart, 2001; Multicity, 2001b; Multicity, 2001a) that offer translation capability use a variety of technologies. One such system, Lotus Translation Services for SameTime (Lotus, 2001), connects users to a translation server of their choice within the Lotus Translation Community to provide chat translations. The translation server appears as another user participating in the chat conversation, under a user-defined alias. The user can connect to the translation service, which provides a URL link to the translation console, allowing the user to translate the chat using the translation console (Kellerman and Mayeur, 2000).

Several web sites offer translation for their chat rooms using the products developed by MultiCity.com. One such is GermanMart.com, which advertises multi-lingual chat rooms that use word-for-word translation (GermanMart, 2001). MultiCity.com offers six languages and uses Systran technology for both chat rooms and instant messaging (Multicity, 2001b; Multicity, 2001a).

There are other chat room translation services, including at least one that offers live human translation for chat conversations (LatinTrans, 2001). All of the translations by these systems are from an individual source, and can be improved upon using a consensus of translations from multiple sources.

3 Consensus Translation based on Multi-sequence Alignment

The combination of results from multiple systems performing the same task have been found to improve accuracy in a number of NLP tasks such as part-of-speech tagging (Roth and Zelenko, 1998) and text categorization (Larkey and Croft, 1996) and also in speech recognition (Fiscus, 1997). The underlying assumption is that each system being combined commits independent errors and the likelihood of all systems committing the same error is small.

Unlike part-of-speech tagging or text categorization tasks where the unit of consensus is given (either a word or a document), a unit of consensus in a translation task needs to be derived. The units for comparison across different translation systems are inferred by aligning the outputs of the translation systems. To compute a consensus string from the results of the different translation engines, we first need to align the strings with respect to each other. An alignment provides a representation that identifies common substrings among the different translations. For example, Figure 1 shows an example English sentence, the translations from five translation engines and a human translation. The result of aligning these sentences is shown in Figure 2. As can be seen from Figure 2, there are regions where the different translation sys-

English: give me driving directions please to middletown area
MT1: déme direcciones impulsoras por favor a área de middletown
MT2: déme direcciones por favor a área
MT3: déme direcciones conductores por favor al área middletown
MT4: déme las direcciones que conducen satisfacen al área de middletown
MT5: déme que las direcciones tend en cia a gradan al área de middletown
Reference: déme direcciones por favor al área de middletown

Figure 1: An example English sentence and its translation from five different translation systems

tems agree to a large extent on the words and their order and there are other regions where there is less or no agreement at all.

Multiple string alignment can be viewed as an extension of the pairwise string alignment. For pairwise string alignment, we define a profile as a string which records the insertion, deletion and substitution of tokens needed to transform one string into the other string. If L is the number of tokens in each string to be aligned, the time complexity of the pairwise alignment algorithm is $O(L^2)$. An extension of the pairwise string alignment algorithm, could be used for multiple string alignment, however, the time complexity is exponential ($O(L^N)$) in the number of strings(N) to be aligned.

An heuristic solution to multiple alignment, known as progressive multiple alignment is very popular in the biological sequencing literature (Feng and Doolittle, 1987). The algorithm is as follows:

1. *Compute the edit distance scores and their profiles for each of the $N(N-1)/2$ pairs of strings*
2. *Repeat the following until one profile remains*
   (a) *Select the profile for the least edit distance string-string, string-profile or profile-profile pair.*
   (b) *Compute the edit distance between the selected profile and the remaining strings and profiles.*

The result of the algorithm is a tree structure with the strings most similar appearing closer to the leaf level. The algorithm is greedy and is not guaranteed to find the global optimal solution. Details of this and other algorithms for multiple alignments can be found in (Durbin et al., 1998).

An implementation of the multiple string alignment called *CLUSTALW* (Thompson et al., 1994) is freely downloadable from (CLUSTALW, 2001). The implementation is specialized for aligning biological sequences. We adapted this implementation by changing the cost matrix so as to be more suitable for our purpose.

3.1 Retrieving the Consensus Translation

The result of alignment can be viewed as a lattice as shown in Figure 3. The lattice can be viewed in terms of a sequence of segments, where each segment contains the different translations for a word or a phrase. The fan out at a state indicates the disagreement in translation among the translation systems for that region. The arcs represent the words and phrases (possibly the empty word $< epsilon >$) and the weights on the arcs are the negative logarithm of the probability of each word or phrase in that segment. So if all the systems agree on a word or a phrase, the arc has a zero weight.

It is straightforward to observe that retrieving the least cost string from this lattice would correspond to selecting the majority translation for each segment. We refer to this model of consensus retrieval as consensus by majority vote (CMV).

However, note that there are segments of the lattice where there is no clear majority. Selection of a translation in such regions would be completely *ad hoc*. In order to improve selection in such regions, we employ a posterior n-gram language model (λ_M) that is built using the total translated corpus resulting from all the translation systems. The idea is to select those translations that best fit the n-gram context as given by a language model, when there is lack of information from the majority vote. We refer to this model of consensus retrieval as

déme			direcciones	impulsoras por favor	a	área	de	middletown
déme			direcciones	por favor	a	área		
déme			direcciones	conductores por favor	al	área		middletown
déme		las	direcciones	que conducen satisfacen	al	área	de	middletown
déme	que	las	direcciones	tend en cia a gradan	al	área	de	middletown
*****			**********			****		**********

Figure 2: Result of aligning different translations for the English sentence *give me driving directions please to middletown area*

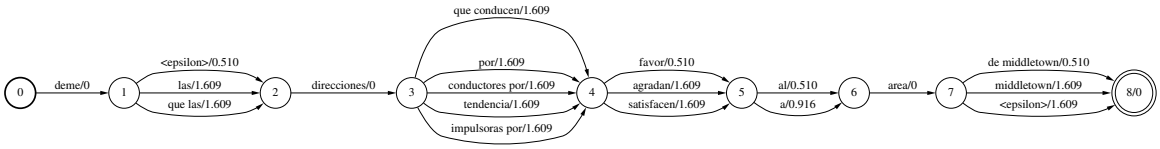

Figure 3: Lattice representation of the result of the multiple alignment. The weights on the arcs are negative logarithm of the probability that word.

$CMV{+}LM = min(\lambda_{CMV} \circ \alpha * \lambda_M)$.

In Bangalore et al. (2001) it was shown that the majority-vote-based consensus (CMV) was at least as successful as the best individual translation system for domain dependent spoken dialog transcriptions. Using a posterior trigram language model (CMV+LM) to retrieve the consensus provided translations better than those provided by any of the individual translation engines. The consensus translation system presented in Bangalore et al. (2001) was used in a domain-specific spoken language registration system. The corpus consisted of only 8000 utterances. In contrast, our corpus is largely domain independent and consists of 58,300 sentences. The style of conversation is chatty and informal with an average sentence length of about 8 words. We are interested in the applicability of consensus translation to create bilingual data for this domain.

4 Experiments and Results

The monolingual data we used was collected from Hubbub, an instant messenger developed at AT&T Labs-research (Isaacs et al., 2001b; Isaacs et al., 2001a). We received 58,300 sentences from Hubbub in English that consisted of conversations among AT&T researchers and developers and their associates.[1] We randomly divided the body of data into 58,000 sentences of training data, and 300 sentences of test data. The training data was translated into Spanish using three different translation engines, (ELingo, 2001; Systran, 2001; Promt, 2001). The three versions of translations were used to create a consensus translation for the training set of sentences. The consensus translation was paired with the corresponding English sentences to create the parallel corpus of 58,000 sentence pairs.

To create translations for the 300 sentence test set, we used the same three translation engines to obtain three different translations for each sentence. From these translations, we created a lexicon of words for each sentence. We then translated the English sentences by hand using the Spanish words for each sentence from the MT engines. The lexicon was presented in such a way that the origin of each word was not apparent. Thus we created the best possible human translation of our test set, using the vocabulary provided by the MT engines.

We then trained a statistical translation model that is based on a transfer paradigm between source and target dependency trees (Alshawi et al., 1998c). The dependency trees and the transfer lexicon are automatically inferred from the parallel corpus. We trained such translation models on the parallel corpus to produce English-Spanish and Spanish-English translation systems.

[1] Prior to our receiving the data, the sentences were randomized and all names were changed to protect the privacy of the participants.

	Off-the-shelf Translation	Model Translation
Consensus	n/a	.4137
Elingo	.3962	.3681
Systran	.4253	.4249
Promt	.4521	.4012

Table 1: Comparing the accuracy of translations from the off-the-shelf translation systems to translations from models trained on 15,000 sentence pairs obtained from the off-the-shelf translation systems

4.1 Evaluating Performance

In order to evaluate the performance of an MT system, we used the translation accuracy metric presented in Alshawi et al. (1998c). The metric is used to compare the resulting translation with the reference translation and is defined as in Equation 1.

$$Translation\ Accuracy = 1-(M+I+D+S)/R \quad (1)$$

where I is the number of insertions, D is the number of deletions, S is the number of substitutions, M is the number of moves, that are needed to transform the resulting translation to match the reference translation (R is the number of words in the reference).

4.2 Off-the-shelf translation system versus trained statistical translation system

Statistical translation models are trained directly from data in contrast to the off-the-shelf MT systems which have been carefully handcrafted over years and in some cases over decades. In order to compare the translation accuracy of statistically trained translation models against translations produced by MT systems, we trained translation models on 15,000 sentence pairs each, on data from each of the MT systems individually. Table 1 shows a comparison of the accuracy of translations produced by the statistically trained models and those produced by the MT systems on the test set of 300 sentences. It is interesting to note that with only 15,000 sentence pairs we were able to meet the accuracy of the translations produced by Systran and were only 3% lower than Elingo and 5% lower than Promt. Furthermore the accuracy of statistical translation models trained from the consensus translation (0.4137 from Table 1) outperforms the translation provided by Elingo (0.3962 from Table 1) on this data set.

4.3 Learning Curve

Figure 4 shows the improvement in translation accuracy on the test set of the models trained on increasingly large segments of the data. It also shows the accuracy of the off-the-shelf MT engines for the test set of 300 sentences.

Training on the 58,000 consensus-translated sentence pairs, we were able to exceed even the best of the off-the-shelf MT systems. Using only half of our available data we were able to meet the translation accuracy of Systran. Another interesting aspect to note is that the learning curve is still on the rise, with the potential to improve the accuracy further with the availability of more data.

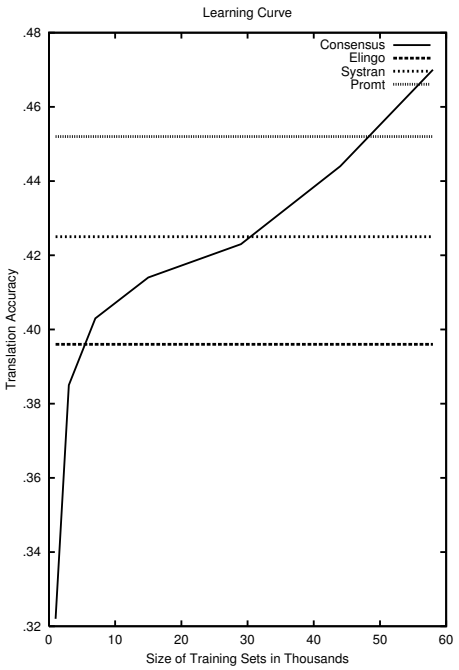

Figure 4: The translation accuracy of models trained on consensus data. Also shown are the performance of the output of the off-the-shelf MT engines.

We implemented a number of techniques to boost the performance beyond that demonstrated by the learning curve in figure 4. The English sentences were normalized for punctu-

ation and case. Normalizing for case improved the accuracy of the consensus translation by 4% on 44,000 sentence pairs. We also tried techniques such as example-based translation and achieved a small accuracy improvement (less than 1%). Using corpora of abbreviations and terminology, it is possible to achieve even bigger gains in accuracy.

4.4 Incremental Training of Statistical Translation Models

While the consensus algorithm is responsible for the accuracy improvements of the statistically trained translation system, it is clear that accuracy is directly related to the amount of data used to train the models. Using the consensus translation technique outlined above, we can create parallel corpora for large training sets. Fortunately, as mentioned before, there is an ever increasing amount of monolingual data for instant messaging application. However, retraining translation models over the entire corpus, when a new increment of parallel corpus becomes available, quickly becomes prohibitively expensive. In this section, we discuss training of the statistical models in an incremental process and investigate the decrease in performance resulting from incremental training.

We investigated the effect of training additively on translation accuracy and on training time. We trained a translation model using all of the 58,000 sentences and another translation model using the alignments from a corpus partitioned into 44,000 and 14,000 sentence pairs. We created the additive translation model by concatenating the alignments for the 44,000 and 14,000 sentence pairs and training the parameters of the translation model on the collective alignment. We found that the translation accuracy is slightly lower (0.467 with the entire 58,000 corpus versus 0.46 with the partitioned model) using the partitioned data set as compared to a translation model trained with the entire data set. However, the additive training regime allows us to use increasingly large sets of training data in the same amount of time as training the largest partition. The degree to which performance is decreased by dividing the data and training additively is directly related to the number of partitions, and the decrease in the translation accuracy could be compensated with the ability to train on very large training sets.

5 Conclusions

In this paper, we have applied the technique of consensus translation to bootstrap parallel data from off-the-shelf translation systems for the general domain of instant messaging conversations. We have trained hierarchical statistical translation models on the bootstrapped parallel data and demonstrated that the translation models trained on the consensus translation, with increasingly large amounts of data, clearly outperform each of the individual translation engines. We have also presented techniques to train translation models additively, in cases where training models with the entire parallel corpus would be prohibitively expensive. We have shown that additively trained models have only a slight loss in translation accuracy with the potential of being scalable to very large parallel corpora.

References

H. Alshawi, S. Bangalore, and S. Douglas. 1998a. Learning Phrase-based Head Transduction Models for Translation of Spoken Utterances. In *The fifth International Conference on Spoken Language Processing (ICSLP98)*, Sydney.

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. 1998b. Automatic acquisition of hierarchical transduction models for machine translation. In *Proceedings of the 36th Annual Meeting Association for Computational Linguistics*, Montreal, Canada.

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. 1998c. Automatic acquisition of hierarchical transduction models for machine translation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, volume 23 (3), pages 377–404.

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proceedings of ASRU 2001*.

P. Brown, S.D. Pietra, V.D. Pietra, and R. Mercer. 1993. The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*, 16(2):263–312.

CLUSTALW. 2001. http://www.at.embnet.org/embnet/progs/clustal/clustalw.htm.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. 1998. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press.

ELingo. 2001. http://www.elingo.com.

D-F Feng and RF Doolittle. 1987. Progressive sequence alignment as a a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360.

J Fiscus. 1997. A Post-Processing System To Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction(ROVER). In *IEEE Workshop on Automatic Speech Recognition and Understanding.*

GermanMart. 2001. http://www.germanmart.com/mainstore/content/GermanCorner/chat.asp.

Ellen Isaacs, Alan Walendowski, and Dipti Ranganathan. 2001a. Hubbub: A Wireless Instant Messenger that Uses Earcons for Awareness and for 'Sound Instant Messages'. *Supplement to the Proceedings of the Conference on Computer-Human Interaction.*

Ellen Isaacs, Alan Walendowski, and Dipti Ranganathan. 2001b. They're playing your song: Staying connected to distant and mobile colleagues through a mobile instant messenger. *Communications of the ACM*, December.

Seymour Kellerman and Thierry Mayeur. 2000. Using Machine Translation for Real-Time, Multilingual Collaboration. Demonstration at ACM Conference on Computer Supported Cooperative Work.

Leah S. Larkey and W. Bruce Croft. 1996. Combining Classifiers in Text Categorization. In *SIGIR-96.*

LatinTrans. 2001. http://hometown.aol.com/latintrans/pchat/chat1.html.

Lotus. 2001. http://www.lotus.com/home.nsf/welcome/international.

Multicity. 2001a. http://www.multicity.com/about/press/pressrelease/pr11.htm.

Multicity. 2001b. http://www.multicity.com/servlet/WebsiteServePage/webmasters/chat/index.

Herman Ney. 1999. Speech translation: Coupling of recognition and translation. In *Proceedings, IEEE ICASSP.*

Promt. 2001. http://www.softline.ru.

Philip Resnik. 1999. Mining the Web for Bilingual Text In *37th Annual Meeting of the Association for Computational Linguistics (ACL'99).*

Dan Roth and Dmitry Zelenko. 1998. Part of speech tagging using a network of linear separators. In *COLING-ACL*, pages 1136–1142.

Systran. 2001. http://www.systransoft.com.

J.D. Thompson, D.G. Higgins, and T.J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–80.