INTERLINGUA BASED STATISTICAL MACHINE TRANSLATION

*Manuel Kauers**, *Stephan Vogel*†, *Christian Fügen**, *Alex Waibel**†

*University of Karlsruhe
Fakultät für Informatik
D-76131 Karlsruhe

† Carnegie Mellon University
School of Computer Science
Pittsburgh, PA 15221, USA

ABSTRACT

In goal oriented spoken language translation, an interlingua based approach has proven quite useful as it (1) reduces overall effort when multiple language pairs are required, (2) can provide a paraphrase of semantic equivalence in the input language, (3) abstracts away from the disfluencies of spoken language to express the speaker's intention. On the other hand, interlingua based systems are cumbersome to develop as semantic grammars have to be laboriously prepared for each input language. In this paper, we demonstrate that mappings from input text to interlingua can be learned automatically and that new input languages can be added by language projection. We show that the resulting system also delivers competitive performance.

1. INTRODUCTION

The use of an interlingua [1] in machine translation has frequently been viewed as the best method of translation, when multiple language pairs and directions have to be provided and when a need and desire to abstract away from the surface form is given. This is particularly the case when we develop spoken language translation systems, where spontaneous dialogs and conversations are ill-formed, fragmentary and incomplete. Yet, such dialogs are frequently goal driven and domain limited so that the intent of an utterance can be determined uniquely within this setting. As a result, a number of spoken language systems (see, for example, www.c-star.org) have been proposed and demonstrated successfully using interlingua for domain limited spontaneous speech translation. While interlingua based systems gain in attractiveness when many language pairs are to be hooked up with each other (O(n) instead of O(n^2) language directions), they have required the development of handwritten semantic grammars for each input and output language. Automatic training of semantic mappings has been proposed before [2, 3], but most methods use linear chains of concepts instead of trees (as desired in an interlingua representation) and relatively simple semantic concepts for use in human-machine interaction (e.g., ATIS).

In this paper, we develop a method to automatically train a mapping between source text and a *tree structured* interlingua for use in a somewhat more complex human-to-human travel planning task. We show that this can be done, given a corpus of semantically tagged data. Experimental evaluation of both the automatically trained and handwritten systems result in spoken language translation performance that is comparable.

Based on a workable analyzer from a natural language A to interlingua IF, we then develop a projection scheme by which we attempt to infer the input-to-interlingua mapping (the analysis stage) for a second input language B. This is to avoid laborious hand tagging for every new language. This projection can be achieved using human translations of data in language A to language B. Such parallel data can be collected more easily than the semantic tags or treebanks we require initially for the domain in language A. The resulting analyzers from A to IF can thus be projected for use in B to IF, C to IF, etc. and afterwards provide translators in arbitrary directions and language pairs. Only a small degradation in performance is observed using the new system in the new language. For the experiments presented here we are still using a handwritten generator, but similar training methods for generation can be devised as well.

2. STATISTICAL TRANSLATION INTO SEMANTIC TREES

Statistical machine translation (SMT) [4] is based upon the noisy channel paradigm. Translating $\mathbf{f}$ to $\hat{\mathbf{e}}$ is regarded as the search process

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} \left(p(\mathbf{e}) \cdot p(\mathbf{f}|\mathbf{e})\right). \qquad (1)$$

Models to estimate $p(\mathbf{e})$ (the language model probability) and $p(\mathbf{f}|\mathbf{e})$ (the translation model probability) have been widely studied for the case that both $\mathbf{e}$ and $\mathbf{f}$ are linear sequences of tokens (i.e. natural language sentences). For our purposes, translation into a tree-structured interlingua, we need models which allow the $\mathbf{e}$'s to be trees.

The fact that linear sequences might be considered as special trees consisting of exactly one path leads us to models that fall down to the usual ones when applied to trivial trees.

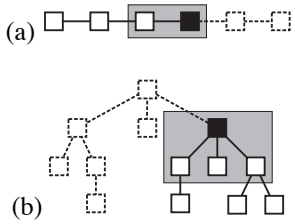

Fig. 1. The concept of n-grams (a) in sequences (b) in trees

2.1. A Language Model for Trees

In the usual situation where $\mathbf{e} = (e_1, \ldots, e_l)$, language modelling is typically based on the decomposition

$$p(\mathbf{e}) = \prod_{i=1}^{l} p(e_i|e_1, \ldots, e_{i-1})$$

where the $p(e_i|e_1, \ldots, e_{i-1})$ are approximated by the relative frequencies of n-grams seen in the training corpus. While $\mathbf{e}$ may in this case be over-formally defined as some token e together with a subsequence $\mathbf{e}'$, a tree $\mathbf{e}$ may be defined as consisting of some token e together with a set of $a \geq 0$ subtrees $\mathbf{e}_1, \ldots, \mathbf{e}_a$ (a is the arity of the tree). This leads to the decomposition

$$p(\mathbf{e}) = p(e|\mathbf{e}_1, \ldots, \mathbf{e}_a) \cdot \prod_{i=1}^{a} p(\mathbf{e}_i|\mathbf{e}_1, \ldots, \mathbf{e}_{i-1})$$

which corresponds to a bottom-up decoding in the order $\mathbf{e}_1, \ldots, \mathbf{e}_a, e$.

It is a special feature of the IF that the ordering of subtrees is unimportant for the semantics they cover, i.e. the term $a(b, c)$ is semantically equivalent to $a(c, b)$. This justifies the assumption that the probabilities $p(\mathbf{e}_i|\mathbf{e}_1, \ldots, \mathbf{e}_{i-1})$ are independent of $\mathbf{e}_1, \ldots, \mathbf{e}_{i-1}$, giving the recursive formula

$$p(\mathbf{e}) = p(e|\mathbf{e}_1, \ldots, \mathbf{e}_a) \cdot \prod_{i=1}^{a} p(\mathbf{e}_i)$$

in which $p(\mathbf{e}_i)$ is to be decomposed further in the same way as $p(\mathbf{e})$. To approximate $p(e|\mathbf{e}_1, \ldots, \mathbf{e}_a)$ with relative frequencies "tree-n-grams" are used (Figure 1).

2.2. Translation Models for Trees

The standard translation models as described in [4] use the concept of word alignment: each word in the source sentence is aligned to a word in the target language. Words which have no correspondence in the target sentence are aligned to the so-called empty word added to the sentences at position 0. This concept of alignment can also be used when translating into IF, as illustrated in Figure 2.

We are now generalizing the translation models IBM 1 and IBM 2 proposed in [4] to the case where the $\mathbf{e}$'s are

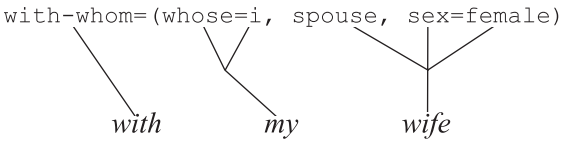

Fig. 2. An alignment between an English phrase and its corresponding IF representation

trees. For the IBM 1 translation model, this is straightforward because the model makes no assumptions which are specific to sequences. In the model's formula

$$p(\mathbf{f}|\mathbf{e}) = \frac{1}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i),$$

we just need to assign an index to each of the l nodes in the tree $\mathbf{e}$ in some arbitrary way.

The IBM 2 model does also include alignment probabilities $a(i|j, m, l)$, resulting in the estimation formula

$$p(\mathbf{f}|\mathbf{e}) = \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) a(i|j, m, l)$$

in the sequential case. The idea is that the jth token of the source sentence $\mathbf{f}$ is aligned to the ith token of the target sentence $\mathbf{e}$ with probability $a(i|j, m, l)$ provided that m is the length of $\mathbf{f}$ and l is the length of $\mathbf{e}$.

Now indexing the nodes of a tree is no longer arbitrary because it affects the values of the $a(i|j, m, l)$. Using again the fact that the IF is commutative, it seems appropriate to consider only the depth of a particular node. Its position within the level does not contribute any information. Taking d as the depth of $\mathbf{e}$ and $\#e_i$ as the number of nodes in level i, this leads to

$$p(\mathbf{f}|\mathbf{e}) = \prod_{j=1}^{m} \sum_{i=0}^{d} \Big(\frac{a(i|j, m, d)}{\#e_i} \sum_{e_i} t(f_j|e_i) \Big)$$

where the e_i in the right sum run over all nodes of level i.

2.3. Decoding Trees

For generating $\mathbf{e}$ from $\mathbf{f}$ given a language model $p(\mathbf{e})$ as well as a translation model $p(\mathbf{f}|\mathbf{e})$, a stack decoder is used similar to the one described in [5] but adapted to generate trees rather than linear sequences. To cope with the huge search space, "bad" hypotheses are pruned after each iteration.

The algorithm starts with an empty hypothesis. In the case of linear sequences new hypotheses are generated by iteratively appending new target words to existing hypotheses. The tree decoder takes a *set* of existing hypotheses $\mathbf{h}_1, \ldots, \mathbf{h}_n$ and forms a new tree with the $\mathbf{h}_i$ as subtrees

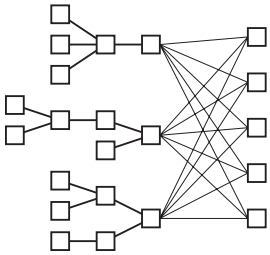

Fig. 3. Decoding trees: A new hypothesis is generated by appending a new node to a set of existing hypotheses.

and an additional target token as its root (Fig. 3). If the algorithm is restricted to choose only sets of size 1, it reduces to the sequential version.

Generating trees gives a much larger search space than generating linear sequences. In fact, while n hypotheses and k words to append lead to $n \cdot k$ new hypotheses for the next iteration of the sequence decoder, the tree decoder generates $2^n \cdot k$ new hypotheses in the same situation. This is because a set of size n has exactly 2^n subsets.

Three methods are applied to reduce the search space. First, hypotheses which are not legal IFs according to the IF specification are not generated. Second, the branching factor of generated trees in restricted to three, the depth to four. Finally, standard pruning is used. The decoder generated for less than 5% of the test sentences an IF which had a lower score than the reference IF, indicating that the number of search errors due to pruning is small.

3. TRANSLATION MODEL PROJECTION TO NEW LANGUAGE PAIRS

The motivation behind our efforts was to avoid the need to handcraft grammars. Instead, the mapping from source language to IF is learned from an annotated corpus. This means that a corpus with ideal human generated Interlingua annotations is required.

However, once we have a well-trained translation model to generate IF from some source language F, there is an elegant way to get a model for the translation of some other language G to IF by "merging" a model for G to F with the model for F to IF. This process, called translation model projection, is illustrated in Fig. 4. The assumption is that generating a training corpus for G to F is much cheaper than building a corpus for G to IF.

Assume that we have translations model for G to F and F to E and we want to build a model for G to E. The composition of the translation probability distributions $p(g|f)$ and $p(e|f)$ to $p(g|e)$ is done via

$$p(g|e) = \sum_f p(f)p(g|f)p(f|e)$$

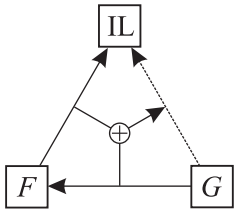

Fig. 4. Projection of one translation model (F to IF) to another (G to IF) by use of a third model (G to F).

where $p(f)$ is the unigram probability of f. The formula is justified by the following calculation, where Chapman-Kolmogorov is used in the third step:

$$\sum_f p(f)p(g|f)p(f|e) = \sum_f p(f) \cdot \frac{p(g,f)}{p(f)} \cdot \frac{p(f,e)}{p(e)}$$
$$= \frac{1}{p(e)} \sum_f p(g,f)p(f,e) = \frac{p(g,e)}{p(e)} = p(g|e).$$

The combination of the alignment probabilities is more involved because they depend on two parameters per language. Let m, l, n be the respective lengths of sentences in F, E, G, and use j, i, k as index variables for F, E, G, respectively. In addition to the probabilities $a(k|j,n,m)$ and $a(i|j,m,l)$ which to combine to form the probabilities $a(i|k,n,l)$, we first need probabilities $a(j|n,m)$ and $a(i|m,l)$ which can easily be trained using inverse IBM 2 models. The other needed distributions can be obtained trivially.

In the first step, we need to project $a(j|n,m)$, $a(i|m,l)$ to $a(k|n,l)$, which can be done by

$$a(k|n,l) = \frac{1}{p(n|l)} \sum_m p(m)p(m|l)p(n|m)a(k|n,m)$$

after $p(m|l)$ has been obtained by combining $p(n|l)$ and $p(m|n)$. Using this auxiliary distribution, we get

$$a(i|k,n,l) = \frac{1}{p(l|n)a(k|n,l)} \cdot \sum_m \Big(p(m)p(m|n)p(l|m)a(k|n,m) \cdot \sum_j a(j|k,n,m)a(j|m,l)a(i|j,m,l) \Big)$$

which only depends on known values. The proofs of the two formulas are omitted here due to space restrictions.

4. EVALUATION

We evaluated the new system on a speech-to-speech translation task and also compared it to a grammar based translation system which uses the IF as Interlingua [6]. Dialogs

	Training		Test	
Language	Ger	Eng	Ger	Eng
Sentences	2,427	2,427	194	194
Tokens	11,236	11,729	889	955
Vocabulary	1,196	1,010	269	241
Singletons	566	429	152	123

Table 1. Some statistics about our training and test set.

	G	S	S_P
Perfect	18.56%	15.20%	19.72%
Okay	38.02%	31.19%	22.81%
Bad	43.43%	53.61%	57.48%
Acceptable	**56.57%**	**46.39%**	**42.52%**

Table 2. Scores of the translations generated by system G, system S, and system S on the projected model (S_P). The values are averages of 4 independent graders.

in the travel planning domain have been collected, transcribed, and annotated with IF representations. From this database, we extracted a trilingual corpus of about 2,500 triples German-English-IF as a training set. 194 German sentences were held out to use them as a test set. Detailed corpus statistics is given in Table 1.

For each German test sentence three IF representations were generated using (1) the grammar-based system G, (2) the statistical system S with a model trained on G to IF, and (3) the system S_P using a model obtained by projection of a G-to-E model and a E-to-IF model. The IF expressions were then all converted into English using the same IF to E generation grammar.

The final results were then presented to four human evaluators. Each translation was assigned one of three grades: "perfect"—translation is semantically complete and grammatically correct, "okay"—the main part of the original semantics is covered and expressed understandably, and "bad" otherwise. "perfect" and "okay" translations form the class "acceptable."

The evaluation results are given in Table 5. The statistical system is not quite as good as the grammar-based system. Given the very small training corpus, with about 40% of all words seen only once during training, this is not surprising. On the contrary, the results show the potential of the proposed approach. Surprisingly, the "perfect" score for the projected model is even better than the respective value for the grammar based system. It seems that the detour over a second natural language provides some beneficial smoothing feature, at least on small training sets as our one is.

5. CONCLUSIONS

In this paper, we have extended methods in statistical machine translation and applied them to tree languages, regarding language understanding as "translation" from some natural language into a treelike interlingua. A projection mechanism allows the reuse of bilingual corpora of two natural languages once an initial translation model for the translation to interlingua is available.

A human evaluation shows that even with very limited training data and simple first models, performance can be achieved that is comparable to a handwritten grammar based system. Beyond performance, our approach delivers substantial reduction in cost and development time. Statistical interlingua translation therefore appears to hold promise for numerous goal oriented speech translation system applications.

6. REFERENCES

[1] L. Levin, D. Gates, A. Lavie, and A. Waibel, "An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues," in *Proceedings of ICSLP*, 1998.

[2] W. Minker, M. Gavalda, and A. Waibel, "Hidden understanding models for machine translation," in *European Speech Communication Association Tutorial and Research Workshop on Interactive Dialogue in Multi-Modal Systems (ESCA-1999)*, June 1999.

[3] K. Macherey, F. J. Och, and H. Ney, "Natural Language Understanding Using Statistical Machine Translation," in *Proceedings of Eurospeech 2001*, September 2001, pp. 2205–2208.

[4] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer, "Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[5] Ye-Yi Wang and Alex Waibel, "Decoding Algorithm in Statistical Translation," in *Proceedings of the ACL/EACL '97, Madrid, Spain*, July 1997, pp. 366–372.

[6] A. Lavie, C. Langley, A. Waibel, F. Pianesi, G. Lazzari, P. Coletti, L. Taddei, and F. Balducci, "Architecture and Design Considerations in Nespole!," in *Proceedings of HLT: Human Language Technology*, San Diego, CA, March 2001.