Fertility Models for Statistical Natural Language Understanding

Stephen Della Pietra*, Mark Epstein, Salim Roukos, Todd Ward
IBM Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598, USA
(*Now With Renaissance Technologies, Stonybrook, NY, USA)
sdella@rentec.com
[meps/roukos/tward]@watson.ibm.com

Abstract

Several recent efforts in statistical natural language understanding (NLU) have focused on generating clumps of English words from semantic meaning concepts (Miller et al., 1995; Levin and Pieraccini, 1995; Epstein et al., 1996; Epstein, 1996). This paper extends the IBM Machine Translation Group's concept of *fertility* (Brown et al., 1993) to the generation of clumps for natural language understanding. The basic underlying intuition is that a single concept may be expressed in English as many disjoint clump of words. We present two fertility models which attempt to capture this phenomenon. The first is a Poisson model which leads to appealing computational simplicity. The second is a general nonparametric fertility model. The general model's parameters are bootstrapped from the Poisson model and updated by the EM algorithm. These fertility models can be used to impose clump fertility structure on top of preexisting clump generation models. Here, we present results for adding fertility structure to unigram, bigram, and headword clump generation models on ARPA's *Air Travel Information Service (ATIS)* domain.

1 Introduction

The goal of a natural language understanding (NLU) system is to *interpret* a user's request and respond with an appropriate action. We view this interpretation as translation from a natural language expression, E, into an equivalent expression, F, in an unambigous *formal language*. Typically, this formal language will be hand-crafted to enhance performance on some task-specific domain. A *statistical* NLU system translates a request E as the most likely formal expression $\hat{F}$ according to a probability model p,

$$\hat{F} = \underset{over\ all\ F}{\arg\max}\, p(F|E) = \underset{over\ all\ F}{\arg\max}\, p(F,E).$$

We have previously built a fully automatic statistical NLU system (Epstein et al., 1996) based on the source-channel factorization of the joint distribution $p(F,E)$

$$p(F,E) = p(F)p(E|F).$$

This factorization, which has proven effective in speech recognition (Bahl, Jelinek, and Mercer, 1983), partitions the joint probability into an a priori intention model $p(F)$, and a translation model $p(E|F)$ which models how a user might phrase a request F in English.

For the ATIS task, our formal language is a minor variant of the NL-Parse (Hemphill, Godfrey, and Doddington, 1990) used by ARPA to annotate the ATIS corpus. An example of a formal and natural language pair is:

- F: List flights from New Orleans to Memphis flying on Monday departing early_morning
- E: do you have any flights going to Memphis leaving New Orleans early Monday morning

Here, the evidence for the formal language concept 'early_morning' resides in the two disjoint clumps of English 'early' and 'morning'. In this paper, we introduce the notion of concept fertility into our translation models $p(E|F)$ to capture this effect and the more general linguistic phenomenon of embedded clauses. Basically, this entails augmenting the translation model with terms of the form $p(n|f)$, where n is the number of clumps generated by the formal language word f. The resulting model can be trained automatically from a bilingual corpus of English and formal language sentence pairs.

Other attempts at statistical NLU systems have used various meaning representations such as *concepts* in the AT&T system (Levin and Pieraccini, 1995) or *initial semantic structure* in the BBN system (Miller et al., 1995). Both of these systems require significant rule-based transformations to produce disambiguated interpretations which are then used to generate the SQL query for ATIS. More recently, BBN has replaced handwritten rules with decision trees (Miller et al., 1996). Moreover, both systems were trained using English *annotated by hand* with segmentation and labeling, and both systems produce a semantic representation which is forced to preserve the time order expressed in the English. Interestingly, both the AT&T and BBN systems generate words within a clump according to bigram models. Other statistical approachs to NLU include decision trees (Kuhn and Mori, 1995) and neural nets (Gorin et al., 1991).

In earlier IBM translation systems (Brown et al., 1993) each English word would be generated by, or "aligned to", exactly one formal language word. This mapping between the English and formal language expressions is called the "alignment". In the simplest case, the translation model is simply proportional to the product of word-pair translation probabilities, one per element in the alignment. In these models, the alignment provides all of the structure in the translation model. The alignment is a "hidden" quantity which is not annotated in the training data and must be inferred indirectly. The EM algorithm (Dempster, Laird, and Rubin, 1977) used to train such "hidden" models requires us to sum an expression over all possible alignments.

These early models were developed for French to English translation. However, in NLU there is a fundamental asymmetry between the natural language and the unambiguous formal language. Most notably, one formal language word may frequently correspond to whole English phrases. We added the "clump", an extra layer of structure, to accomodate this phenomenon (Epstein et al., 1996). In this paradigm, formal language words first generate a clumping, or partition, of the word slots of the English expression. Then, each clump is filled in according to a translation model as before. The alignment is defined between the formal language words and the clumps. Then, both the alignment and the clumping are hidden structures which must be summed over to train the models.

Already, these models represent significant progress. They learn automatically from a bilingual corpus of English and formal language sentences. They do not require linguistically knowledgeable experts to tediously annotate a training corpus. Rather, they rely upon a group of translators with significantly less linguistic knowledge to produce a bilingual training corpus. The fertility models introduced below maintain these benefits while slightly improving performance.

2 Fertility Clumping Translation Models

The rationale behind a clumping model is that the input English can be *clumped* or *bracketed* into phrases. Each clump is then generated from a single formal language word using a translation model. The notion of what constitutes a natural clumping depends on the formal language. For example, suppose the English sentence were:

I want to fly to Memphis please.

If the formal language for this sentence were:

LIST FLIGHTS TO LOCATION,

then the most plausible clumping would be:

[I want] [to fly] [to] [Memphis] [please],

for which we would expect "[I want]" and "[please]" to be generated from "LIST", "[to fly]" from "FLIGHTS", "[to]" from "TO, and "[Memphis]" from LOCATION. Similarly, if the formal language were:

LIST FLIGHTS DESTINATION_LOC

then the most natural clumping would be:

[I want] [to fly] [to Memphis] [please],

in which we would now expect "[to Memphis]" to be generated by "DESTINATION_LOC".

Although these clumpings are perhaps the most natural, neither the clumping nor the alignment is annotated in our training data. Instead, both the alignment and the clumping are viewed as "hidden" quantities for which all values are possible with some probability. The EM algorithm is used to produce a maximum likelihood estimate of the model parameters, taking into account all possible alignments and clumpings.

In the discussion of fertility models we denote an English sentence by E, which consists of $\ell(E)$ words. Similarly, we denote the formal language by F, a tuple of order $\ell(F)$, whose individual elements are denoted by f_i. A clumping for a sentence partitions E into a tuple of clumps C. The number of clumps in C is denoted by $\ell(C)$, and is an integer in the range $1 \cdots \ell(E)$. A particular clump is denoted by c_i, where $i \in \{1 \cdots \ell(C)\}$. The number of words in c_i is denoted by $\ell(c_i)$. c_1 begins at the first word in the sentence, and $c_{\ell(C)}$ ends at the last word in the sentence. The clumps form a proper partition of E. All the words in a clump c must align to the same f. An alignment between E and F determines which f generates each clump of E in C. Similarly, A denotes the alignment, with $\ell(A) = \ell(C)$, and the a_i denote the formal language word to which each e in c_i align. The individual words in a clump c are represented by $e_1 \cdots e_{\ell(c)}$.

For all fertility models, the *fundamental parameters* are the joint probabilities $p(E, C, A, F)$. Since the clumping and alignment are hidden, to compute the probability that E is generated by F, one calculates:

$$p(E \mid F) = \sum_{C,A} p(E, C, A \mid F)$$

3 General and Poisson Fertility

In the general fertility model, the translation probability with "revealed" alignment and clumping is

$$p(E,C,A \mid F) = \frac{1}{L!}\prod_{i=1}^{\ell(F)} p(n_i \mid f_i) n_i! \prod_{j=1}^{\ell(C)} p(c_j \mid f_{a_j}) \quad (1)$$

$$p(c \mid f) = p(\ell(c) \mid f) \prod_{i=1}^{\ell(c)} p(e_i \mid f_c) \quad (2)$$

where $p(n_i \mid f_i)$ is the fertility probability of generating n_i clumps by formal word f_i. Note that $\sum n_i = L$. The factorial terms combine to give an inverse multinomial coefficient which is the uniform probability distribution for the alignment A of F to C.

It appears that the computation of the likelihood, which is the sum of $\ell(F)(\ell(F)+1)^{\ell(E)-1}$ product terms, is exponential. Although dynamic programming can reduce the complexity, there remain an exponentially large number of terms to evaluate in each iteration of the EM algorithm. We resort to a top-N approximation to the EM sum for the general model, summing over candidate clumpings and alignments proposed by the Poisson fertility model developed below.

If one assumes that the fertility is modeled by the Poisson distribution with mean fertility λ_f

$$p(n \mid f) = \frac{e^{-\lambda_f} {\lambda_f}^n}{n!} \quad (3)$$

then a polynomial time training algorithm exists. The simplicity arises from the fortuitous cancellation of $n!$ between the Poisson distribution and the uniform alignment probability. Substituting equation 3 into equation 1 yields:

$$\begin{aligned} p(E,C,A \mid F) &= \frac{1}{L!}\prod_{i=1}^{\ell(F)} e^{-\lambda_{f_i}} {\lambda_{f_i}}^{n_i} \prod_{j=1}^{\ell(C)} p(c_j \mid f_{a_j}) \quad (4)\\ &= \frac{1}{L!}\prod_{i=1}^{\ell(F)} e^{-\lambda_{f_i}} \prod_{j=1}^{\ell(C)} q(c_j \mid f_{a_j}) \quad (5)\\ q(c \mid f) &= \lambda_f \, p(c \mid f), \quad (6)\end{aligned}$$

where ${\lambda_f}^n$ has been absorbed into the effective clump score $q(c \mid f)$. In this form, it is particularly simple to explicitly sum over all alignments A to obtain $p(E,C \mid F)$ by repeated application of the distributive law. The resulting polynomial time expressions are:

$$\begin{aligned} p(E,C \mid F) &= \frac{1}{L!}\prod_{i=1}^{\ell(F)} e^{-\lambda_{f_i}} \prod_{j=1}^{\ell(C)} \hat{q}(c_j \mid F) \quad (7)\\ \hat{q}(c \mid F) &= \sum_{f \in F} q(c \mid f) \quad (8)\end{aligned}$$

The $\hat{q}(C \mid F)$ values for all possible clumpings can be calculated in $O(\ell(E)^2\ell(F))$ time if the maximum clump size is unbounded, and in $O(\ell(E)\ell(F))$ if bounded. The Viterbi decoding algorithm (Forney, 1973) is used to calculate $p(E \mid L, F)$ from these expressions. The Viterbi algorithm produces a score which is the sum over all possible clumpings for a fixed L. This score must then normalized by the $\exp(-\sum_{i=1}^{\ell(F)} \lambda_{f_i})/L!$ factor. The EM count accumulation is done using an adaptation of the Baum-Welch algorithm (Baum, 1972) which searches through the space of all possible clumpings, first considering 1 clump, then 2, and so forth.

Initial values for $p(e \mid f)$ are bootstrapped from Model 1 (Epstein et al., 1996) with the initial mean fertilities λ_f set to 1. We also fixed the maximum clump size at 5 words. Empirically, we found it beneficial to hold the $p(e \mid f)$ parameters fixed for 20 iterations to allow the other parameters to train to reasonable values. After training, the translation probabilities and clump lengths are smoothed using deleted interpolation (Bahl, Jelinek, and Mercer, 1983).

Since we have been unable to find a polynomial time algorithm to train the general fertility model, we use the Poisson model to "expose" the hidden alignments. The Poisson fertility model gives the most likely 1000 clumpings and alignments, which are then rescored according to the current general fertility model parameters. This gives fractional counts for each of the 1000 alignments, which are then used to update the the general fertility model parameters.

4 Improved Clump Modeling

In both the Poisson and general fertility models, the computation of $p(c|f)$ in equation 2 uses a unigram model. Each English word e_i is generated with probability $p(e_i|f_c)$. Two more powerful modeling techniques for modeling clump generation are n-gram language models (Miller et al., 1995; Levin and Pieraccini, 1995; Epstein, 1996), and headword language models (Epstein, 1996). A bigram language model uses:

$$p(c \mid f) = p(\ell(c) \mid f) p(e_1 \mid bdy, f_c) p(bdy \mid e_{\ell(c)}, f_c) \times \prod_{i=2}^{\ell(c)} p(e_i \mid e_{i-1}, f_c)$$

where bdy is a special marker to delimit the beginning and end of the clump.

A headword language model uses two unigram models, a headword model and a non-headword model. Each clump is required to have a headword. All other words are non-headwords. The identity of a clump's headword is hidden, hence it is necessary

Word	λ	$p(n=0)$	$p(n=1)$	$p(n=2)$	$p(n>=3)$
late	1.49	.00	.62	.28	.10
early	1.55	.00	.89	.03	.08
morning	1.40	.01	.85	.11	.03
afternoon	1.62	.00	.85	.12	.03
early_morning	2.50	.00	.16	.69	.15

Table 1: Trained Poisson and General Fertility

Word	Top $p(e\|f)$	Score	Top $p_{head}(e\|f)$	Score	Top $p_{nonhead}(e\|f)$	Score
early	early	.37	early	.68	an	.30
	an	.22	i	.23	flight	.29
	i	.09	day	.06	would	.10
morning	morning	.63	morning	.75	the	.43
	in	.12	leaving	.06	in	.37
	leaving	.05	flights	.05	of	.08
List	the	.21	show	.49	me	.45
	me	.19	what	.17	the	.19
	show	.18	give	.07	all	.12
	what	.17	you	.06	are	.05
	please	.04	list	.05	please	.05

Table 2: Trained Translation Probabilities using Poisson Fertility

Model	DEC93	DEC93a
1	75.00	75.22
Clump	74.78	77.01
Clump-HW	75.89	78.35
Clump-BG	76.79	78.12
Poisson	78.12	81.25
Poisson-HW	78.12	81.25
Poisson-BG	78.12	81.25
General	79.91	82.59
General-HW	79.91	79.91
General-BG	73.21	83.04

Table 3: Class A CAS on Patterns for DEC93

to sum over all possible headwords:

$$p(c \mid f) = \frac{p(\ell(c) \mid f)}{\ell(c)} \sum_{i=1}^{\ell(c)} p_{head}(e_i \mid f_c) \prod_{j \neq i} p_{nonhead}(e_j \mid f_c)$$

5 Example Fertilities

To illustrate how well fertility captures simple cases of embedding, trained fertilities are shown in table 1 for several formal language words denoting time intervals. As expected, "early_morning" dominantly produces two clumps, but can produce either one or three clumps with reasonable probability. "morning" and "afternoon" train to comparable fertilities and preferentially generate a single clump. Another interesting case is the formal language token "List" which trains to a λ of 0.62 indicating that it frequently generates no English text. As a further check, the λ values for "from", "to", and the two special classed words "CITY-1" and "CITY-2" are near 1, ranging between 0.96 and 1.17.

Some trained translation probabilities are shown for the unigram and headword models in table 2. The formal language words have captured reasonable English words for their most likely translation or headword translation. However, "early" and "morning" have fairly undesirable looking second and third choices. The reason for this is that these undesirable words are frequently adjacent to the English words "early" and "morning"; hence the training algorithm includes contributions with two word clumps containing these extraneous words. This is the price we pay for not using supervised training data. Intriguingly, the headword model is more strongly biased towards the likely translations and has a smoother tail than the unigram model.

6 Results

The translation models were trained with 5627 context-independent ATIS sentences and smoothed with 600 sentences. In addition, 3567 training sentences were manually aligned and included in a separate training experiment. This allows comparison between an unannotated corpus and a partially annotated one.

We employ a trivial decoder and language model since our emphasis is on evaluating the performance of different translation models. Our decoder is a simple *pattern matcher*. That is, we accumulate the different formal language patterns seen in the training set, and score each of them on the test set. The language model is just the unsmoothed unigram probability distribution of the patterns. This LM has a 10% chance of not including a test pattern and its use leads to pessimistic performance estimates. A more general language model for ATIS is presented in (Koppelman et al., 1995). Answers are generated by an SQL program which is a deterministically constructed from the formal language of our system. The accuracy of these database answers is measured using ARPA's Common Answer Specification (CAS) metric.

The results are presented in table 3 for ARPA's December 1993 blind test set. The column headed DEC93 reports results on unsupervised training data, while the column entitled DEC93a contains the results from using models trained on the partially annotated corpus. The rows correspond to various translation models. Model 1 is the word-pair translation model used in simple machine translation and understanding models (Brown et al., 1993; Epstein et al., 1996). The models labeled "Clump" use a basic clumped model without fertility. The models labeled "Poisson" and "General" use the Poisson and general fertility models presented in this paper. The "HW" and "BG" suffixes indicate the results when $p(c|f)$ is computed with a headword or bigram model.

The partially annotated corpus provides an increase in performance of about 2-3% for most models. For General-LM, results increased by 8-10%. The Poisson and general fertility models show a 2-5% gain in performance over the basic clump model when using the partially annotated corpus. This is a reduction of the error rate by 10-20%. The unannotated corpus also shows a comparable gain.

Acknowledgement: This work was sponsored in part by ARPA and monitored by Fort Huachuca HJ1500-4309-0513. The views and conclusions contained in this document should not be interpreted as representing the official policies of the U.S. Government.

References

Bahl, Lalit R., Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190, March.

Baum, L.E. 1972. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8.

Brown, Peter F., Stephen A. DellaPietra, Vincent J. DellaPietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, pages 19(2):263–311, June.

Dempster, A.P., N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(B):1–38.

Epstein, M. 1996. *Statistical Source Channel Models for Natural Language Understanding.* Ph.D. thesis, New York University, September.

Epstein, M., K. Papineni, S. Roukos, T. Ward, and S. Della Pietra. 1996. Statistical natural language understanding using hidden clumpings. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 176–179, Atlanta, Georgia, May.

Forney, G. David. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61:268–278, March.

Gorin, A., S. Levinson, A. Gertner, and E. Goldman. 1991. Adaptive acquisition of language. *Computer Speech and Language*, 5:101–132.

Hemphill, C., J. Godfrey, and G. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 96–101, Hidden Valley, PA, June. Morgan Kaufmann Publishers, Inc.

Koppelman, J., S. Della Pietra, M. Epstein, and S. Roukos. 1995. A statistical approach to language modeling for the ATIS task. In *Proceedings of the Spoken Language Systems Workshop*, pages 1785–1788, Madrid, Spain, September.

Kuhn, R. and R. De Mori. 1995. The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):449–460, May.

Levin, E. and R. Pieraccini. 1995. Chronus, the next generation. In *Proceedings of the Spoken Language Systems Workshop*, pages 269–271, Austin, Texas, January.

Miller, S., M. Bates, R. Bobrow, R. Ingria, J. Makhoul, and R. Schwartz. 1995. Recent progress in hidden understanding models. In *Proceedings of the Spoken Language Systems Workshop*, pages 276–279, Austin, Texas, January.

Miller, S., D. Stallard, R. Bobrow, and R. Schwartz. 1996. A fully statistical approach to natural language interfaces. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 55–61, Santa Cruz, CA, June. Morgan Kaufmann Publishers, Inc.