FEATURE-BASED LANGUAGE UNDERSTANDING

K. A. Papineni *S. Roukos* *R. T. Ward*
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA

ABSTRACT

We consider translating natural language sentences into a formal language using a system that is data-driven and built automatically from training data. We use features that capture correlations between automatically determined key phrases in both languages. The features and their associated weights are selected using a training corpus of matched pairs of source and target language sentences to maximize the entropy of the resulting conditional probability model. Given a source-language sentence, we select as the translation a target-language candidate to which the model assigns maximum probability. We report results in Air Travel Information System (ATIS) domain.

10.21437/Eurospeech.1997-417

1. INTRODUCTION

Our interest is in developing a statistical translation system that translates source language sentences into target language sentences. The main consideration is that the system be fully data-driven and be built automatically from the training data. Such systems can be ported easily to new domains since they do not use domain-specific rules developed by experts.

Principally, we consider the case when the source language is a natural language in a restricted domain and the target language is an artificial (formal) language. Such cases arise in building natural language interfaces to applications such as word-processors, email-managers, data-bases, or automatic teller machines. The formal language expresses operations that the applications can perform.

We apply our techniques to Air Travel Information System (ATIS) domain. In ATIS, one is interested in translating English queries on air travel information (flights, fares, airlines, ground-transport etc) into a formal language that can be translated deterministically into a database query. The data for ATIS was collected in an ARPA-sponsored program [1].

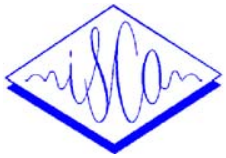
We have at our disposal several thousands of English queries and their man-made formal language translations. These pairs of English and formal sentences form what is called the training corpus. Some examples from the training corpus follow:

S_1: show me all the nonstop flights from city-1 to city-2 leaving city-1 after time-1 on date-1 .
T_1: LIST FLIGHTS NONSTOP DEPARTING AFTER TIME-1 FLYING-ON DATE-1 FROM:CITY CITY-1 TO:CITY CITY-2

S_2: what are the available flights on air-1 from city-1 to city-2 the evening of date-1 .
T_2: LIST FLIGHTS AIR-1 EVENING FLYING-ON DATE-1 FROM:CITY CITY-1 TO:CITY CITY-2

Statistical translation models are used to translate a new (unseen in the training corpus) source sentence S in the following natural way: Evaluate conditional probability $P(T|S)$ for all T in the target language space and select as the translation that T which maximizes $P(T|S)$. Parameters of these models are "trained" from the training corpus.

Statistical translation models were invented at IBM by Brown, et al [2] in the context of French to English translation. These models are based on a source-channel paradigm. The source-channel paradigm uses two component models: 1. $P(S|T)$ called the channel model, and 2. $P(T)$ called the language model (or source model). The two component models are then used to compute $P(T|S) = P(S|T)P(T)/P(S)$; then that T which maximizes the product $P(S|T)P(T)$ is chosen as the translation of the input sentence S. The channel model can also be thought of a translation model, *but from target to source*. Each of the component models is estimated independently. These *a priori* models were first applied to natural language understanding in [3] for extracting the full meaning of a context-independent sentence in the ATIS domain. The system was built automatically from the training data.



EUROSPEECH '97
5th European Conference on Speech Communication and Technology
Rhodes, Greece, September 22-25, 1997

source-channel model for automatically constructing a language understanding system from training data; we build a *direct* model of the *a posteriori* conditional distribution $P(T|S)$. The direct model uses features that capture translation effects and language model effects in a unified framework; the selection of features is fully data-driven. The model is powerful in that it can handle a variety of features involving phrases, words, parses, and long-distance relations in the source and target sentences. Substantially more general alignments, which treat the two languages more symmetrically, are possible here than in [3]. Neither explicit manual labeling of important words nor explicit intrasentence segmentation of the training data nor rule-based transformations are required, unlike in [4] - [5]. Our approach is also dissimilar to the decision-tree based approach for language understanding [6].

The model we describe here is built on top of a given prior distribution $P_0(T|S)$. The prior could be uniform, or could be a decision tree, or any probabilistic model. The model can be seen as a correction to the prior relative to a set of feature functions.

2.1. Features

Although the method we describe is applicable when feature functions are real-valued, we consider only binary-valued feature functions here. So a feature maps the product set of source and target language sentences to 0 or 1. We now give concrete examples of features that we consider. To this end, first consider some sample English and Formal sentences. The formal sentences are not translations of the English sentences.

E_1: what are least expensive flights from city-1 to city-2.
E_2: what flights do you have from city-1 to city-2.

F_1: LIST FLIGHTS MORNING EARLIEST-ARRIVING FROM:CITY CITY-1 TO:CITY CITY-2
F_2: LIST FLIGHTS CHEAPEST FROM:CITY CITY-1 TO:CITY CITY-2

One of type of feature we considered is a phrase-feature of the form

$$f_{s,t}(S,T) = \begin{cases} 1 & \text{if } s \in S, t \in T \\ 0 & \text{else.} \end{cases}$$

Such features model the fact that certain phrases in source language sentences tend to co-occur with

$$f_{\text{least expensive, CHEAPEST}}$$

fires on (E_1, F_2), but not on (E_2, F_2) or (E_1, F_1).

A special case with a null s-phrase results in features that effect target language modeling. Such features obviate the need to estimate target language model separately, as is the case with source-channel models [2] -[3]. A variation of a phrase feature is one which ignores the order of words in s-phrase and t-phrase. Another is a long-distance bigram feature.

Given an English query, there are often competing formal candidates that differ with each other minimally: one may have an extra word (phrase) relative to another. Of course, conversely, a phrase may be missing in one relative to the other. We need features that differentiate such candidates. A feature that looks for existence of words in the target sentence that do not have an "informant" in the source sentence serves this purpose. Such words are deemed to be spurious in a formal candidate. An example feature is one that looks for the word "CHEAPEST" in Formal in the absense of "lowest" and "cheapest" in English. This feature fires on (E_2, F_2) but not on (E_1, F_2). Another type of a feature looks for absence of words or phrases in the target sentence that ought to explain "informants" in the source sentence. An example feature is one that fires if the word "CHEAPEST" is absent in Formal while "lowest" or "cheapest" or "least expensive" is present in English. This feature fires on (E_1, F_1) but not on (E_1, F_2).

2.2. Feature Selection and Optimization

We described a variety of features so far. Let $\phi(S,T)$ be a vector feature of dimension n. Clearly, how many times the component features are true on the training data is of interest. Let $\tilde{P}$ be the empirical distribution of the training corpus and define

$$d := \sum_{S,T} \tilde{P}(S,T)\phi(S,T).$$

We are then interested in a conditional distribution $P(T|S)$ that satisfies

$$\sum_S \tilde{P}(S) \sum_T P(T|S)\phi(S,T) = d,$$

and is as close to the prior as possible.

The following Kullback-Leibler-like pseudometric $D(\cdot,\cdot)$ quantifies the notion of closeness between any two conditional distributions P_1 and P_2:

$$\sum_S \tilde{P}(S) \sum_T P(T|S) \log \frac{P(T|S)}{P_2(T|S)}$$

We have the following (primal) optimization problem:

$$\min_P D(P, P_0)$$

subject to

$$\sum_S \tilde{P}(S) \sum_T P(T|S)\phi(S,T) = d.$$

This optimization problem gives rise to models P of the form

$$P_{\phi,\lambda}(T|S) := \frac{P_0(T|S)e^{\lambda\phi(S,T)}}{Z(S)}$$

with the normalization factor

$$Z(S) := \sum_T P_0(T|S)e^{\lambda\phi(S,T)}.$$

The (dual) optimization problem is posed in R^n and the optimal solution is described by the optimal $\lambda_* \in R^n$. This (convex) optimization problem is standard. We use Improved Iterative Scaling [7] to solve it. When the prior is uniform, the optimal solution to this problem also maximizes the likelihood of training data and the entropy of the model.

With $\alpha_i := e^{\lambda_i}$, we can rewrite the above as

$$P(T|S) = \frac{P_0(T|S)\Pi\alpha_i^{\phi_i(S,T)}}{Z(S)}.$$

In this formulation, we see that each feature that is true (i.e. takes the value 1) gets a multiplicative "vote" α_i to modify the prior score $P_0(T|S)$.

We now describe feature selection. First, we assume that a set of n features ϕ have already been selected somehow. We then solve the optimization problem described above. Then, D_*, the minimum achieved by λ_*, is a figure of merit of the feature set $\{\phi_1, \cdots, \phi_n\}$. Once the set $\{\phi_1, \cdots, \phi_n\}$ is selected, we compute D_* for $\{\phi_1, \phi_2, \cdots, \phi_n, f\}$ for all features f in the remaining pool and rank the features by the new D_*. We can then add the top-ranking feature as ϕ_{n+1} to the set of features already selected and find the optimal weights $\lambda_1, \lambda_2, \cdots, \lambda_{n+1}$. Thus, in principle, we can start with $n = 0$ and build a good feature set by increasing the set by 1 in each batch. In practice, we add top k-ranking new features in each batch to the features already selected. The figure of merit D_* increases monotonically with the size of the feature set. We stop feature selection when the increment is marginal.

English queries. We used 5627 pairs of context-independent sentences from the ATIS training data. Examples of features that our system selected are shown below along with their near-optimal weights.

Source Phrase	Target Phrase	α
arrive	FLIGHTS ARRIVING-ON	39
about	AROUND	2900
late-afternoon	LATE-AFTERNOON	56000
cheapest round-trip	FARES CHEAPEST ROUND-TRIP	43
including	ALONG-WITH	280

ATIS data also contains test sets, which are outside the training corpus. These are DEV94 (to test models during development) and DEC93 and DEC94 (actual evaluation sets). We report results on context-independent queries from these test sets. Translation performance is measured by Common Answer Specification, a metric defined by ARPA in terms of response from air travel database.

While D_* increases monotonically with the number of features, performance of the translator on a held-out test set may not. A typical movement of DEV94 performance with number of features is shown below:

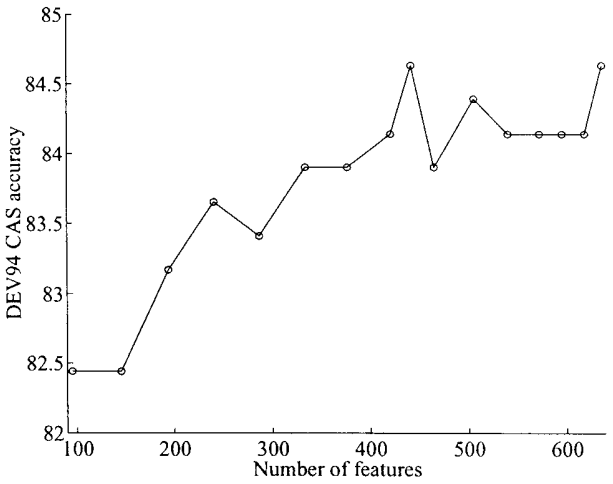


Figure 1. Performance vs number of features

The following results are a good improvement over those of previous automatic statistical translation systems.

DEV94	DEC93	DEC94
84.63	86.83	85.84

a Spoken Language Corpus", Proceedings of the DARPA Speech and Natural Language Workshop, pp 7-14, Harriman, NY, Feb 1992.

[2] P. F. Brown et al. "The mathematics of statistical machine translation: Parameter estimation", *Computational Lingustics*, 19 (2), 263-311, June 1993.

[3] M. Epstein, K. Papineni, S. Roukos, T. Ward, and S. Della Pietra, "Statistical Natural Language Understanding", IEEE ICASSP Proceedings, vol I, pp. 176-179, May 1996.

[4] E. Levin et al, "Chronus, the next generation," Proceedings of the Spoken Language Systems Workshop, pp 269-271, Austin, Jan 1995.

[5] S. Miller et al, "Recent progress in hidden understanding models," Proceedings of the Spoken Language Systems Workshop, pp 269-271, Austin, Jan 1995.

[6] R. Kuhn et al, "The application of semantic classification trees to natural language understanding," IEEE Trans. Pattern Analysis and Machine Intelligence, 17 (5):449-460, May 1995.

[7] S. Della Pietra et al, "Inducing features of random fields," CMU Technical Report CMU-CS-95-144, 1995.