PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A
MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Incorporating linguistic structure into statistical language models

Ronald Rosenfeld

Phil. Trans. R. Soc. Lond. A 2000 **358**, 1311-1324
doi: 10.1098/rsta.2000.0588

THE ROYAL SOCIETY

Incorporating linguistic structure into statistical language models

By Ronald Rosenfeld

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Statistical language models estimate the distribution of natural language for the purpose of improving various language technology applications. Ironically, the most successful models of this type take little advantage of the nature of language. I review the extent to which various aspects of natural language are captured in current models. I then describe a general framework, recently developed at our laboratory, for incorporating arbitrary linguistic structure into a statistical framework, and present a methodology for eliciting linguistic features currently missing from the model. Finally, I ponder our failure heretofore to integrate linguistic theories into a statistical framework, and suggest possible reasons for it.

Keywords: statistical language modelling; human language technologies; feature induction

1. Introduction

Statistical language models (SLMs) estimate the probability of sentences in natural language using large amounts of training data. SLMs are used in a variety of language technology applications, such as speech recognition, document classification, optical character recognitions, machine translation, and more. In speech recognition, for example, an incoming acoustic signal a is given. The goal is to find the sentence s^* that maximizes the posterior $P(s \mid a)$:

$$s^* = \arg\max_s P(s \mid a) = \arg\max_s P(a \mid s) \cdot P(s), \tag{1.1}$$

where the language model $P(s)$ plays the role of the prior.

A given language model M is often evaluated by its *perplexity*,

$$\text{perplexity}(M) = 2^{H(P;P_{\mathrm{M}})}, \tag{1.2}$$

where $H(P; P_{\mathrm{M}})$ is the cross entropy between the distribution P_{M} described by the model and P_{D}, the true distribution of the data.

Ironically, the most successful SLM techniques use very little knowledge of what language really is. Attempts to incorporate linguistic theories or even linguistic intuition into SLMs have met with very limited success. In what follows, § 2 lists various aspects of natural language, and reviews the extent to which they are captured in current models. Section 3 describes a general framework, recently developed at our laboratory, for integrating linguistic features into a statistical framework. Finally, in § 4 I ponder the SLM community's failure to integrate linguistic theories into a statistical framework, and suggest possible reasons for it.

Table 1. *Natural language sentences*
(Example average length sentences from the BN corpus.)

WANDILE ZOTHE DO YOU PERSONALLY KNOW PEOPLE WHO WERE ARRESTED AND TORTURED DURING THE APARTHEID ERA </s>
SO HE PROBABLY WILL HAVE TO HAVE THEM TAXED BECAUSE THEY'RE NOT A TRADITIONAL PENSION FUND </s>
BUT THE TOBACCO COMPANIES AND NASCAR OFFICIALS SAY THEIR FANS ARE WILDLY LOYAL TO RACE ADVERTISERS </s>
THERE ARE A LOT OF QUALITY SWEATERS IN THE MARKET RIGHT NOW CASHMERE AND CASHMERE BLENDS </s>
POLICE SAY THE MAN RAN FROM THE FRONT OF THE HOUSE AND CAME AROUND THIS CORNER </s>

2. Linguistic structure in statistical language models

(*a*) *Baseline: the n-gram*

Almost all language models estimate the probability of a sentence s by using the *chain rule* to decompose it into a product of conditional probabilities,

$$\Pr(s) \stackrel{\text{def}}{=} \Pr(w_1, \dots, w_n) = \prod_{i=1}^{n} \Pr(w_i \mid w_1, \dots, w_{i-1}) \stackrel{\text{def}}{=} \prod_{i=1}^{n} \Pr(w_i \mid h_i), \tag{2.1}$$

where $h_i \stackrel{\text{def}}{=} \{w_1, \dots, w_{i-1}\}$ is the *history* when predicting word w_i.

The most commonly used language model, the n-gram, makes the further simplifying assumption:

$$P(w_i \mid h_i) \approx P(w_i \mid w_{i-n+1}, \dots, w_{i-1}). \tag{2.2}$$

The n-gram captures correlations among nearby words reasonably well. Not surprisingly, it captures little else. This can be best appreciated by observing 'sentences' generated from this model. Table 1 lists example sentences from the Broadcast News (BN) corpus: a corpus of some 13 million sentences transcribed from TV and radio news-related programmes between 1992 and 1996 (Graff 1997). This complete corpus was used to train a state-of-the-art trigram language model, which was, in turn, used in generative mode to produce 'pseudo-sentences', examples of which are listed in table 2.

It is not difficult for people to tell these two language sources apart. In an informal blind study we conducted on Carnegie Mellon's Sphinx speech research group, classification accuracies of 95% were achieved. It is also easy to appreciate how such judgements are made, since just about every aspect of natural language (with the exception of short-distance dependences) are being violated by the pseudo-sentences. These include lexical relationships, topic and discourse coherence, syntax and semantics. One would expect that such glaring deficiencies in this simple model would be quickly remedied. Not so. We will now review these aspects of language and what attempts have been made to model them.

Table 2. *Trigram-generated pseudo-sentences*

(Average length pseudo-sentences generated by a trigram trained on the BN corpus.)

YOU CALL PORK MITCHELL IS THOSE THREE WIRE LUCK AFTER ATTENDANT S. COMPETITIVENESS AND KNOWS THAT </s>
ARE YOU REFERRING TO IS EXTREMELY RISKY BECAUSE I'VE BEEN TESTED WHOSE ONLY WITH A MAIN </s>
THE FIRST BLACK EDUCATORS CATACOMBS DOWN ROMAN GABRIEL SLEEP IN A WAY TO KNOW IS PROPER </s>
MY QUESTION TO YOU THOSE PICTURES MAY STILL NOT IN ROMANIA AND I LOOKED UP CLEAN </s>
YOU WERE GOING TO TAKE THEIR CUE FROM ANCHORAGE LIFTED OFF EVERYTHING WILL WORK SITE VERDI </s>

(*b*) *Lexical relations*

To an n-gram, the vocabulary is a long list of indistinguishable categories. But of course, words in a language form complex and not fully understood lexical relations. Surely TUESDAY is closer in some sense to WEDNESDAY than to, say, CHAIR.

The simplest attempt to consider lexical relations concentrates on part-of-speech (POS) information. The POS-based n-gram (Jelinek 1989) comes in several varieties. For example, for a trigram, one could try

$$\Pr(w_i \mid w_{i-2}, w_{i-1}) = \Pr(w_i \mid \mathrm{POS}_i) \cdot \Pr(\mathrm{POS}_i \mid \mathrm{POS}_{i-2}, \mathrm{POS}_{i-1}), \tag{2.3}$$

where POS_i is the POS class of w_i. The main motivation for such a model is to reduce the number of parameters and, hence, the variance of the estimation. One practical problem is that in a language as polysemous as English, the correct POS of each word token is often hard to determine. State-of-the-art POS taggers, boasting 95–97% accuracy under ideal conditions, can be helpful. Alternatively, a hidden-variable model can be used, in which all possible POSs are considered simultaneously. Nonetheless, these models are not usually very successful, as measured by perplexity improvement over the baseline word-based n-gram. Apparently, what is a useful linguistic distinction does not translate into a useful predictive distinction.

An improvement over the POS-based model is to use a class-based model, where classes may get their origin in POS categories, but are further optimized over the data. Several algorithms have been suggested for automatically clustering the vocabulary based on information-theoretic measures (e.g. Brown *et al*. 1991; Kneser & Ney 1993), in an either bottom-up or top-down fashion. In some of these, the algorithm yields not just a partition into classes but rather a word tree, namely a complete (usually binary) hierarchy of word types. These classes are then used by an n-gram similar to the one in equation (2.3). Yet another variation is to assume that each word type can belong to several different categories ('soft classes'), and use a hidden-variable model.

Examples of word classes derived by S. F. Chen (1998, unpublished work) using such an algorithm are shown in table 3. Note that although most of the members of a class seem appropriate, some are not. Not surprisingly, the 'misfits' are often rare word types, which only occurred a handful of times in the data on which the clustering algorithm was run. Ironically, it is exactly these word types, at the tail

Table 3. *Automatically derived word classes*

(Word classes derived automatically from data; notice the 'misfits' are infrequent words.)

MY THY JESSICA'S SARAH'S KEVIN'S CONGESTIVE KAREN'S HEIDI'S
THEN THEREFORE CONSEQUENTLY THIRDLY LASTLY BEHOLD FRO ABETTING
DOWN ASIDE ASHORE INS OVERBOARD IDLY... AFIRE ROUGHSHOD
LET EXCUSE FORGIVE PARDON TICKLE
STATE CENSUS COMMONWEALTH PROVISIONAL FOOTHILLS
WASHINGTON LONDON MOSCOW PARIS TOKYO... ISLAMABAD EDGEWISE
DONE RESOLVED ACCOMPLISHED ACHIEVED FORGOTTEN SOLVED TOLERATED UNDERTAKEN NOTS FORESEEN

end of the vocabulary distribution, that stood to benefit the most from clustering. This is true for all data-driven vocabulary-clustering algorithms: the more common the word is, the more reliably it can be assigned to an appropriate cluster, but the less it will benefit from such an assignment.

For this and other reasons, class-based n-gram models have only seen moderate success. For any amount of training data, these models do not perform as well as their word-based counterparts. When the two are interpolated together, a modest improvement is usually achieved, but only for large corpora.

The only circumstance where lexical relations are exploited successfully for language modelling is in very narrow discourse domains, where class-based n-gram models are used with hand-tailored classes. For example, in the Airline Travel Information System (ATIS) domain (Price 1990), classes consisting of city names, airline names, aircraft types, etc., proved very useful in the face of limited training data (see, for example, Ward 1990).

(*c*) *Syntactic structure*

Several attempts have been made to integrate theories of syntax into language modelling. We will mention three of them here.

(i) *Probabilistic context-free grammars*

Context-free grammars (CFGs) are inaccurate as models of natural language, yet can, arguably, serve as a first-order approximation. Probabilistic context-free grammars (PCFGs) are CFGs with a probability distribution defined over all productions that share their left-hand side. To use PCFGs to model unconstrained language, one must decide on both the CFG itself (set of non-terminals and production rules) and the (usually context-free) production probabilities. To date, no CFG has been suggested that sufficiently covers unconstrained English. Given a large parsed and annotated corpus such as the Penn Treebank (Marcus *et al.* 1993), a CFG can be created to cover it, although its coverage of new, unseen data will be more limited. Furthermore, given a CFG and annotated data, the 'inside–outside' algorithm (Baker 1979), an EM algorithm, can be used to find locally optimal context-free production probabilities. However, the local optima found by the algorithm are unlikely to be as good as the global optimum, which is computationally infeasible to find. Even if the global optimum were to be found, it is likely that context-free production probabilities do not have sufficient expressive power to capture the true distribution of parses. For these reasons, no PCFGs have been suggested that can compete (statistically) with the conventional n-gram, let alone surpass it.

An interesting attempt to combine n-grams and PCFGs was reported by Miller (1995). The CFG structure was formulated as a Markov random field (MRF), and a family of additional constraints was imposed on transitions between successive words, effectively capturing bigram information. This fusing of CFG and bigrams resulted in a model with size (number of parameters) comparable with a bigram, yet performance comparable with that of a trigram. However, no improvement over the state-of-the-art trigram has been reported.

(ii) *Probabilistic link grammar*

Link grammar is a lexicalized grammar formalism introduced by Sleator & Temperley (1991), where a specific link grammar for English has also been constructed by hand, with encouraging coverage. In a specialized form of the grammar known as 'grammatical trigrams' (Lafferty *et al*. 1992), a word can be predicted from any pair of adjacent words that precede it in the sentence. The choice of which such pair to use is encoded in the link grammar, which is trained automatically from a corpus. Grammatical trigrams have achieved a modest yet consistent perplexity improvement over the state-of-the-art trigram. Other promising forms of a dependency grammar were also attempted (Stolcke *et al*. 1997; Alshawi & Douglas, this issue).

(iii) *Structured language model*

Recently, Chelba & Jelinek (1999) introduced a model that predicts the next word based on a set of linguistic equivalence classifications of the history. Given a history, a lexicalized parser proposes several possible equivalence classifications, each with its own weight. The predictions from the various classifications are combined linearly. The parser uses a natural probabilistic parametrization of a push-down automaton, and an EM algorithm is used for training. Experiments on the Switchboard corpus (Godfrey *et al*. 1992) show modest improvements in both perplexity and word error rate over the baseline trigram.

(*d*) *Topic and semantic coherence*

One of the most striking aspects of the pseudo-sentences in table 2 is their lack of topic and semantic coherence. There is a strong sense in reading these sentences that they are not *about* anything.

(i) *Model interpolation*

The earliest attempts to capture topic coherence were through the use of interpolated language models. Typically, the training data were partitioned into multiple sets, each containing documents about a particular topic or set of topics. Each such set was used to create a separate topic-specific language model $P_t(w \mid h)$, and the various models were interpolated together at the word level,

$$P(w \mid h) = \sum_t \lambda_t \cdot P_t(w \mid h), \tag{2.4}$$

where the interpolation weights $\{\lambda_1, \lambda_2, \ldots\}$ varied based on the expected topic of the test data, and were generally determined from held-out data.

There are many variations on this general approach. The training data may be provided already classified into topics (e.g. Seymore & Rosenfeld 1997), or a clustering algorithm may need to be run to automatically derive such classification (e.g. Iyer & Ostendorf 1999). The topic classes themselves can be hard, soft (i.e. allow overlaps), or can even be arranged to form a hierarchy (Seymore & Rosenfeld 1997). Finally, interpolation can take place at the word level, as in equation (2.4) above, or at the sentence level,

$$P(s) = \sum_t \lambda_t \cdot P_t(s) = \sum_t \lambda_t \cdot \prod_i P_t(w_i \mid h_i), \tag{2.5}$$

or at both (Iyer & Ostendorf 1999). Generally speaking, topic interpolation results in moderate yet consistent reductions in perplexity, and often also in speech-recognition error rates.

However, interpolation is seriously deficient as a method for modelling topic coherence. This is because it fails to separate those aspects of language that vary from topic to topic from those that are invariant across all topics. As a result, the limited amount of training data in each topic means that the out-of-topic training data must be pulled in for more robust estimation, resulting in a dilution in the topicality of the interpolated model.

(ii) *Cache*

Another attempt to capture topic coherence and word correlations was through the use of an n-gram cache (Kuhn & De Mori 1990). Caches are easy to implement, and capture word auto-correlations, which are a very pronounced phenomenon across sentences. Both Kuhn & De Mori (1990) and Jelinek *et al.* (1991) report improvements in perplexity over the baseline trigram, and the latter group also reports a modest reduction in word-recognition error rate. Since then, caches have been implemented in many systems, with similar results, and have now become part of the 'baseline' in language modelling.†

(iii) *Word triggers*

A generalization of the cache idea to correlations between different words led to work on *word triggers* (Rosenfeld 1996; Beeferman *et al.* 1997). In principle, correlations between any pair of words or phrases can be captured and modelled. In practice, Rosenfeld (1996) showed that linear interpolation of the trigger component is suboptimal, and that an exponential model, trained using the maximum-entropy

† We did not use a cache in generating the sentences in table 2 because these sentences are evaluated in isolation, whereas the autocorrelations a cache is designed to capture are predominantly cross-sentence effects.

principle, is superior. Unfortunately, the computational requirements of training such a model grow supra-linearly with the number of independently modelled word trigger pairs, and are prohibitive even for a moderate number of such pairs. Although such a model achieves significant perplexity reduction over the baseline trigram, the computational difficulties render it impractical in most cases of interest.

(iv) *Dimensionality reduction*

An improvement over modelling individual word correlations can be achieved by using singular value decomposition (SVD) to reduce the dimensionality of the topic space. In Bellegarda (1998), a matrix of word-document occurrences is reduced to a relatively small size (100×100) via SVD. The resulting matrix succinctly captures the most salient correlations between groups of words on the one hand and clusters of documents on the other. The SVD process also provides the necessary projections from document-space and word-space into the new, combined space. As a result, any new document or partial document can be projected into the combined space, effectively being classified as a combination of the 100 underlying semantic dimensions. When combining SVD decomposition with an n-gram, significant reductions in perplexity are reported, as well as in speech-recognition errors (Bellegarda 2000).

3. A general framework for integrating linguistic structure

The modelling attempts described in the previous section suffer from two major deficiencies. First, the statistical methodology in these attempts varied greatly. Each such model was aimed at a specific linguistic phenomenon, which, in turn, affected the choice of model structure, parameter family, training algorithms, etc. In addition, a new method had to be found for combining the new model component with the existing n-gram baseline. If a new linguistic knowledge source were to suggest itself, a new modelling methodology would have to be developed and tested, and many practical estimation issues would have to be worked out.

Second, virtually all the models described above estimate the probability of a sentence s by using the chain rule, as in equation (2.1), to break it into a product of conditional probabilities (typically $P(w \mid h)$). While this practice is understandable from a historical perspective (n-gram modelling cannot be done on whole sentences), it is not desirable for capturing linguistic phenomena. Linguistic aspects of sentences—such as their grammar, syntax, semantics or pragmatics—are impossible or at best awkward to think about, let alone encode, in a conditional framework. Also, external influences on the sentence (e.g. the effect of preceding utterances, or dialogue-level variables) are equally hard to encode, and factoring them into the prediction of every word in the current sentence causes small but systematic biases in the probability estimation to be compounded.

We have recently introduced a new language-modelling framework that addresses these two deficiencies (Rosenfeld 1997). The exponential model we use directly models the probability of an entire sentence or utterance. By avoiding the chain rule, the model treats each sentence or utterance as a 'bag of features',† where features are arbitrary computable properties of the sentence. Furthermore, the unified structure of the model means that any linguistic theory can be incorporated without any

† Not to be confused with a *bag of words*: features may take account of sequentiality, if so desired.

change to the model itself. This solves the two problems mentioned above. In this section we describe the model and review the various features it has been used with so far.

(a) A whole-sentence exponential model

A whole-sentence exponential language model has the form

$$P(s) = \frac{1}{Z} \cdot P_0(s) \cdot \exp\left[\sum_i \lambda_i f_i(s)\right], \tag{3.1}$$

where the λ_i are the parameters of the model, Z is a universal normalization constant that depends only on the λ_i, and the $f_i(s)$ are arbitrary computable properties, or *features*, of the sentence s. The distribution $P_0(s)$ is an arbitrary probability distribution. It can be thought of as the starting point, or baseline, for further modelling improvements. Often, $P_0(s)$ will be simply derived from the baseline trigram.

The features $\{f_i(s)\}$ are selected by the modeller to capture those aspects of the data they consider appropriate or profitable. These can vary from conventional n-grams, longer-distance dependences, or simple global sentence properties, to more complex functions based on POS tagging, parsing, or other types of linguistic analysis (person and number agreement, semantic coherence, etc.).

For each feature $f_i(s)$, its expectation under $P(s)$ is constrained to a specific value K_i:

$$E_P f_i = K_i. \tag{3.2}$$

These target values are typically set to the expectation of that feature under the empirical distribution $\tilde{P}$ of the training corpus $T = \{s_1, \ldots, s_N\}$ (for binary features, this is simply the prevalence of that feature in the corpus). Then, the constraint (3.2) becomes

$$\sum_s P(s) \cdot f_i(s) = E_{\tilde{P}} f_i \equiv \frac{1}{N} \sum_{j=1}^{N} f_i(s_j). \tag{3.3}$$

If the constraints (3.2) are consistent, there exists a unique solution $\{\lambda_i\}$ within the exponential family (3.1) that satisfies them. Among all (not necessarily exponential) solutions to equation (3.2), the exponential solution is the one closest to the baseline $P_0(s)$ (in the Kullback–Liebler sense), and is thus called the minimum-divergence or minimum-discrimination-information (MDI) solution. If the baseline $P(s)$ is flat (uniform), this becomes the maximum-entropy (ME) solution. Furthermore, if the feature target values K_i are the empirical expectations over some training corpus (as in equation (3.3)), the MDI or ME solution is also the maximum-likelihood solution of the exponential family. For more information see Jaynes (1957), Berger *et al.* (1996) and Rosenfeld (1996).

It is instructive to compare this model with the conditional exponential model, which has seen considerable success recently in language modelling (Della Pietra *et al.* 1992; Lau *et al.* 1993; Berger *et al.* 1996; Rosenfeld 1996). The conditional model has the form

$$P(w \mid h) = \frac{1}{Z(h)} \cdot P_0(w \mid h) \cdot \exp\left[\sum_i \lambda_i f_i(h, w)\right], \tag{3.4}$$

where the features are functions of a specific word–history pair, and so is the baseline P_0. More importantly, Z is no longer a true constant: it depends on h and, thus, must be recomputed for each word in each sentence. The main drawbacks of the conditional model are the severe computational bottleneck of training (especially of computing $Z(h)$), and the difficulty in modelling whole-sentence phenomena.

(b) *Training the model*

The MDI or ME solution can be found by an iterative procedure such as the generalized-iterative-scaling (GIS) algorithm (Darroch & Ratcliff 1972). GIS starts with arbitrary λ_i. At each iteration, the algorithm improves the $\{\lambda_i\}$ values by comparing the expectation of each feature under the current P with the target value, and modifying the associated λ. In particular, we take

$$\lambda_i \qquad \lambda_i + F_i \log \frac{E_{\tilde{P}}[f_i]}{E_P[f_i]}, \tag{3.5}$$

where F_i is a parameter affecting the step size.

(i) *Sampling*

In training a whole-sentence maximum-entropy model, computing the expectations

$$E_P[f_i] = \sum_s P(s) \cdot f_i(s)$$

requires a summation over all possible sentences s, clearly an infeasible task. Instead, we estimate $E_P[f_i]$ by sampling from the distribution $P(s)$ and using the sample expectation of f_i. Sampling from an exponential distribution is a non-trivial task, and is the subject of intense research by statisticians, physicists and others. Sampling of sentences from an exponential distribution poses additional challenges, and is discussed in Chen & Rosenfeld (1999). Efficient sampling is crucial to successful training.

It is equally infeasible to compute the normalization constant

$$Z = \sum_s p_0(s) \cdot \exp\left(\sum_i \lambda_i f_i(s)\right).$$

Fortunately, this is not necessary for training, since sampling can be done without knowing Z. Using the model as part of a classifier (e.g. a speech recognizer) does not require knowledge of Z either, because the relative ranking of the different hypotheses is not changed by a single, universal, constant. Notice that this is not the case for conditional exponential models.

Even though the exact value of Z is not really needed, at times it may be desirable to approximate it, for example for perplexity calculation. This can be done to any desired accuracy by generating a large sample from $P(s)$, observing the frequency of one or more sentences that occur more than, say, 50 times, and making use of equation (3.1). For situations where no such sentences exist, or, in general, for a more efficient estimator, one could use

$$\hat{Z} = \frac{1}{\|T_0\|} \sum_{s \in T_0} \left[\exp\left(\sum_i \lambda_i f_i(s)\right)\right], \tag{3.6}$$

where T_0 is a sample of sentences generated from P_0. For more details, see Zhu *et al*. (1999).

(*c*) *Feature selection*

Once the general framework and training procedure have been worked out, attention can be concentrated on the art of modelling language. The goal is to choose features $f_i(s)$ that capture aspects of language that are not captured (or inadequately captured) by the current baseline-modelling technique. To this end, we have been using the following methodology for feature discovery and selection.

Given a corpus T of natural language sentences† with empirical distribution $\tilde{P}$, presumably representative of the unknown target distribution P, we use it to train our best baseline model P_0. Next, we use P_0 to generate a corpus T_0 of 'pseudo-sentences', like those in table 2. We then compare T_0 with T (or some other dataset from the same distribution P). We look for systematic differences between the two corpora. Any such difference we discover points to a deficiency in the way P_0 models the unknown target distribution P. Any such deficiency can now be readily fixed, by defining an appropriate feature $f(s)$ (or set of features) which have different expectations under P and P_0 (as evidenced by their respective samples T and T_0). The new feature is then added, resulting in a new model:

$$P_1(s) = \frac{1}{Z} P_0(s) \cdot \exp^{\lambda f(s)} . \tag{3.7}$$

Once P_1 is trained, the appropriate constraint (equation (3.3)) guarantees that it consistently captures the new feature, and the previously observed difference between our model and the target distribution has been eliminated.

The process can now be repeated by generating a corpus T_1 of 'pseudo-sentences' from the improved model P_1, and comparing it with the original corpus T, looking for new differences. The latter will be captured with new features, and so on. In practice, many features (or even sets of features) are added at each iteration.‡

As an example,¶ suppose we observe that the trigram-generated T_0 sentences are slightly shorter on average (as measured by number of words) than their T counterparts. We then define the simple feature

$$f_{\text{length}}(s) = \text{number of words in } s, \tag{3.8}$$

and observe that $E_{P_0}[f_{\text{length}}] \neq E_{\tilde{P}}[f_{\text{length}}]$. But once the new feature is incorporated, we are assured that $E_{P_1}[f_{\text{length}}] = E_{\tilde{P}}[f_{\text{length}}]$.

(*d*) *The search for features*

In Chen & Rosenfeld (1999), we searched for n-gram-style features that showed significant discrepancy between P and P_0. These included 4-grams and 5-grams (which

† Or, more generally, utterances. The model is equally suitable for direct estimation of any spoken utterance, whether or not it conforms to conventional linguistic boundaries.

‡ A process of iteratively incorporating the most information-bearing feature in a given candidate set into an exponential model was described in Della Pietra *et al*. (1997). The emphasis in our methodology, though, is on the manual inspection of two corpora and the linguistic analysis and 'detective work' of searching for and evaluating families of linguistically motivated features.

¶ A true one, it turns out: properly smoothed trigram models often do not accurately capture unigram marginals such as Pr(< /s >), the end-of-sentence probability.

were outside the range of the baseline P_0 trigram), class n-grams, and distance (non-contiguous) n-grams. All such features were ranked by a χ^2 significance test. Over 50 000 of these features were found to have a significance level of $\chi^2 > 15$. When incorporated into the language model, they resulted in a small improvement in recognition accuracy (perplexity was not computed). Although these n-gram features do not improve the linguistic plausibility of the model, they served to verify and demonstrate our methodology.

In Zhu *et al*. (1999), we used a shallow parser to map utterances from the Switchboard corpus into a flat list of variable length *constituents*. Features were then defined in terms of constituent sequences, constituent sets and constituent trigrams. Some 7000 such features were found to be statistically significant and added to the model. The perplexity of the new model was slightly lower than that of the baseline, and recognition accuracy was also slightly improved. Further analysis suggested that the potential of these features was limited due to their rarity.

We have subsequently refocused our attention on finding a small number of much more common features. For example, among the most glaring differences between true natural language and trigram-generated sentences is the lack of semantic and topic coherence in the latter. We have been working on modelling such coherence within this framework. As building blocks for the 'semantic coherence' feature, we use measures of association in 2×2 contingency tables based on pairs of content words in the same sentence. For more details, see Rosenfeld *et al*. (1999).

4. Discussion

Why has the language modelling community failed thus far to integrate formal linguistic theories into a statistical framework? Why do current practical language models lack any resemblance to even a rudimentary linguistic theory? Why did 20 years of research fail to yield practical and significant improvements over the trigram, which was proposed in its essential form by Jelinek & Mercer (1980)? In this last section, I propose a few answers to these questions.

(*a*) *Linguistic theories and statistical models have different goals*

Linguistic theories deal with *existence*. They are successful if they explain (and predict) which constructs are found in the language and which similar constructs are not. A theory is considered deficient if there are counter-examples to it. In contrast, statistical language models deal with *prevalence*. They are successful if they approximate reasonably well (in log space) the prevalence of the most common constructs found in the language. A model is considered deficient if there is a systematic bias, or discrepancy, between it and the phenomenon it purports to describe. Thus, a linguistic concept may be a useful tool in the context of a theory, yet prove far less useful when it comes to improving a statistical model. We have already seen an example of this in POS-based classes (§ 2 *b*).

(*b*) *Lack of general framework*

Until recently, we have lacked a general statistical framework for incorporating arbitrary aspects of language into our models. Without such a framework, accommodating each linguistic theory involves solving a (sometimes hard) statistical estimation problem. The model described in § 3 addresses this problem.

(c) Mental strait-jacket of the conditional formulation

Until recently, virtually all language modelling was done in the conditional framework, i.e. by estimating $P(w \mid h)$. As was argued earlier, this is not conducive to thinking about and modelling linguistic properties of the sentence as a whole (e.g. parsability). The model described in § 3 also addresses this problem.

(d) Impoverished priors

Viewed within a Bayesian framework, the problem may lie in our choice of priors. A prior is supposed to capture everything that is known about the domain before any data are observed. In our case, the prior should capture everything that we believe to be true about human languages in general, and about a specific language such as English in particular. The very large parameter space of language means that any feasible amount of training data is insufficient for overwhelming the prior. The choice of prior is therefore crucial. Yet the priors we currently use are impoverished: they do note take advantage of hardly anything we know about language.

As an example, consider the vocabulary clustering problem discussed in § 2 *b*: rare words stand to benefit the most from clustering, yet they do not occur often enough in corpora for reliable automatic clustering. However, much useful information can be provided manually about many semantic classes, such as named entities. If such information can be encoded in a 'soft' prior, automatic clustering methods may yet prove successful.

In summary, it could be argued that attempts to integrate linguistic knowledge into our models have so far failed because we do not yet know how to appropriately encode such knowledge, namely, how to optimally combine it with data. Put yet another way, we have not figured out how to simultaneously get the most out of both our knowledge and our data. Between knowledge without data and data without knowledge, the latter (witness the n-gram) is apparently more successful. But there is no inherent reason why we cannot have both.

I am grateful to Ciprian Chelba, Stanley Chen, Fred Jelinek, John Lafferty, Jerry Zhu and especially Mari Ostendorf for helpful discussions and suggestions. I am also grateful to Karen Spärck Jones and Gerald Gazdar for very useful feedback on a draft of this paper.

References

Baker, J. K. 1979 Trainable grammars for speech recognition. *Speech communication papers, 97th Mtg of the Acoustic Society of America* (ed. D. H. Klatt & J. J. Wolf), pp. 547–550.

Beeferman, D., Berger, A. & Lafferty, J. 1997 A model of lexical attraction and repulsion. In *Proc. ACL-EACL '97 Joint Conf.*, pp. 373–380.

Bellegarda, J. R. 1998 A multi-span language modeling framework for large vocabulary speech recognition. *IEEE Trans. SAP* **6**, 456–467.

Bellegarda, J. R. 2000 Large vocabulary speech recognition with multi-span statistical language models. *IEEE Trans. SAP* **6**, 76–84.

Berger, A., Della Pietra, S. & Della Pietra, V. 1996 A maximum entropy approach to natural language processing. *Comp. Ling.* **22**, 39–71.

Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C. & Mercer, R. L. 1991 Class-based N-gram models of natural language. *Comp. Ling.* **18**, 467–478.

Chelba, C. & Jelinek, F. 1999 Recognition performance of a structured language model. In *Proc. Eurospeech* **4**, 1567–1570.

Chen, S. F. & Rosenfeld, R. 1999 Efficient sampling and feature selection in whole sentence maximum entropy language models. In *Proc. ICASSP, Phoenix, AZ, March 1999*, pp. 549–552.

Darroch, J. & Ratcliff, D. 1972 Generalized iterative scaling for log-linear models. *Ann. Math. Stat.* **43**, 1470–1480.

Della Pietra, S. Della Pietra, V., Mercer, R. & Roukos, S. 1992 Adaptive language modeling using minimum discriminant estimation. In *Proc. ICASSP*, vol. I, pp. 633–636.

Della Pietra, S., Della Pietra, V. & Lafferty, J. 1997 Inducing features of random fields. *IEEE Trans. PAMI* **19**, 380–393.

Godfrey, J. J., Hilliman, E. C. & McDaniel, J. 1992 Switchboard: telephone speech corpus for research and development. In *Proc. ICASSP, San Francisco, March 1992*, pp. 517–520.

Graff, D. 1997 The 1996 Broadcast News speech and language model corpus. In *Proc. DARPA Workshop on Spoken Language Technology*, pp. 11–14.

Iyer, R. & Ostendorf, M. 1999 Modeling long distance dependence in language: topic mixture vs dynamic cache models. *IEEE Trans. SAP* **7**, 30–39.

Jaynes, E. T. 1957 Information theory and statistical mechanics. *Phys. Rev.* **106**, 620–630.

Jelinek, F. 1989 Self-organized language modeling for speech recognition. In *Readings in speech recognition* (ed. A. Waibel & K. F. Lee), pp. 450–506. San Matteo, CA: Morgan Kaufmann.

Jelinek, F. & Mercer, R. 1980 Interpolated estimation of Markov source parameters from sparse data. In *Pattern recognition in practice* (ed. E. S. Gelsema & L. N. Kanal), pp. 381–402. Amsterdam: North Holland.

Jelinek, F. Merialdo, B., Roukos, S. & Strauss, M. 1991 A dynamic language model for speech recognition. In *Proc. DARPA Workshop on Speech and Natural Language, February 1991*, pp. 293–295.

Kneser, R. & Ney, H. 1993 Improved clustering techniques for class-based statistical language modeling. In *Proc. Eurospeech* **2**, 973–976.

Kuhn, R. & De Mori, R. 1990 A cache-based natural language model for speech recognition. *IEEE Trans. PAMI* **12**, 570–583.

Lafferty, J. D., Sleator, D. & Temperley, D. 1992 Grammatical trigrams: a probabilistic model of link grammar. In *Proc. AAAI Fall Symp. on Probabilistic Approaches to Natural Language, October 1992, Cambridge, MA.*

Lau, R., Rosenfeld, R. & Roukos, S. 1993 Trigger-based language models: a maximum entropy approach. In *Proc. ICASSP* **II**, 45–48.

Marcus, M., Santorini, B. & Marcinkiewicz, M. A. 1993 Building a large annotated corpus of English: the Penn Treebank. *Comp. Ling.* **19**, 313–330.

Miller, M. 1995 Markov random fields on the graphs of natural language. In *Proc. Language Modeling Summer Workshop, Johns Hopkins University, August 1995.*

Price, P. 1990 Evaluation of spoken language systems: the ATIS domain. In *Proc. 3rd DARPA Speech and Natural Language Workshop, June 1990*, pp. 91–95.

Rosenfeld, R. 1996 A maximum entropy approach to adaptive statistical language modeling. *Computer Speech Language* **10**, 187–228. (Longer version available: Carnegie Mellon technical report CMU-CS-94-138.)

Rosenfeld, R. 1997 A whole sentence maximum entropy language model. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara, CA, December 1997*, pp. 230–237. IEEE.

Rosenfeld, R., Wasserman, L., Cai, C. & Zhu, X. 1999 Interactive feature induction and logistic regression for whole sentence exponential language models. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, Keystone, CO, December 1999.*

Seymore, K. & Rosenfeld, R. 1997 Using story topics for language model adaptation. In *Proc. Eurospeech* **4**, 1987–1990.

Sleator, D. & Temperley, D. 1991 Parsing English with a link grammar. Carnegie Mellon University Computer Science Department technical report CMU-CS-91-196.

Stolcke, A., Chelba, C., Engle, D., Jimenez, V., Mangu, L., Printz, H., Ristad, E., Rosenfeld, R. & Wu, D. 1997 Structure and performance of a dependency language model. *Proc. Eurospeech* **5**, 2775–2778.

Ward, W. 1990 The CMU air travel information service: understanding spontaneous speech. In *Proc. DARPA Speech and Natural Language Workshop, June 1990*, pp. 127–129.

Zhu, X. J., Chen, S. F. & Rosenfeld, R. 1999 Linguistic features for whole sentence maximum entropy language models. In *Proc. Eurospeech* **4**, 1807–1810.

Discussion

J. Cussens (*University of York, UK*). How does your framework compare to that of Della Pietra *et al.*'s 'Inducing features for random fields'?

R. Rosenfeld. There are two differences, relating to training and to feature selection.

In the quoted reference, the domain is modelling of the spelling of words. This domain is of moderate size and therefore Gibbs sampling can be used efficiently. However, in my own work on modelling sentences, the domain is far larger, and sampling is therefore more challenging.

Regarding feature selection, the quoted reference uses Kullback–Liebler distance to select features from a fully specified family of features. I make use of this as well, but place the emphasis on eliciting new families of features from specialists looking at the corpora rather than requiring a family of features to be available. The idea is to mesh the automatic procedure with human intervention at the right point.

D. B. James. Nouns and verbs are basic to language: why is it that an explicit noun–verb distinction is not being made?

R. Rosenfeld. There are other, more mundane, deficiencies in the model that are also not dealt with. This is beyond what we can currently achieve with statistical means.

P. A. Taylor (*University of Edinburgh, UK*). Are the problems with trigram models due mainly to data sparsity or to an inherent model limitation?

R. Rosenfeld. Model limitations are the main problem. Using a smaller vocabulary is infeasible since the task is to model unconstrained language. Back-off rates are already very low, so having more data would only affect a small percentage of sentences.