TWO DECADES OF STATISTICAL LANGUAGE MODELING: WHERE DO WE GO FROM HERE?

Ronald Rosenfeld

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
USA
roni@cs.cmu.edu

ABSTRACT

Statistical Language Models estimate the distribution of various natural language phenomena for the purpose of speech recognition and other language technologies. Since the first significant model was proposed in 1980, many attempts have been made to improve the state of the art. We review them here, point to a few promising directions, and argue for a Bayesian approach to integration of linguistic theories with data.

1. OUTLINE

Statistical language modeling (SLM) is the attempt to capture regularities of natural language for the purpose of improving the performance of various natural language applications. By and large, statistical language modeling amounts to estimating the probability distribution of various linguistic units, such as words, sentences, and whole documents.

Statistical language modeling is crucial for a large variety of language technology applications. These include speech recognition (where SLM got its start), machine translation, document classification and routing, optical character recognition, information retrieval, handwriting recognition, spelling correction, and many more.

In machine translation, for example, purely statistical approaches have been introduced in [1]. But even researchers using rule-based approaches have found it beneficial to introduce some elements of SLM and statistical estimation [2]. In information retrieval, a language modeling approach was recently proposed by [3], and a statistical/information theoretical approach was developed by [4].

SLM employs statistical estimation techniques using language training data, that is, text. Because of the categorical nature of language, and the large vocabularies people naturally use, statistical techniques must estimate a large number of parameters, and consequently depend critically on the availability of large amounts of training data.

Over the past twenty years, successively larger amounts of text of various types have become available online. As a result, in domains where such data became available, the quality of language models has increased dramatically. However, this improvement is now beginning to asymptote. Even if online text continues to accumulate at an exponential rate (which it no doubt will, given the growth rate of the web), the quality of currently used statistical language models is not likely to improve by a significant factor. One informal estimate from IBM shows that bigram models effectively saturate within several hundred million words, and trigram models are likely to saturate within a few billion words. In several domains we already have this much data.

Ironically, the most successful SLM techniques use very little knowledge of what language really is. The most popular language models (n-grams) take no advantage of the fact that what is being modeled is language – it may as well be a sequence of arbitrary symbols, with no deep structure, intention or thought behind them.

A possible reason for this situation is that the knowledge impoverished but data optimal techniques of n-grams succeeded too well, and thus stymied work on knowledge based approaches.

But one can only go so far without knowledge. In the words of the premier proponent of the statistical approach to language modeling, Fred Jelinek, we must 'put language back into language modeling' [5]. Unfortunately, only a handful of attempts have been made to date to incorporate linguistic structure, theories or knowledge into statistical language models, and most such attempts have been only modestly successful.

In what follows, section 2 introduces statistical language modeling in more detail, and discusses the potential for improvement in this area. Section 3 overviews major established SLM techniques. Section 4 lists promising current research directions. Finally, section 5 suggests both an interactive approach and a Bayesian approach to the integration of linguistic knowledge into the model, and points to the encoding of such knowledge as a main challenge facing the field.

2. STATISTICAL LANGUAGE MODELING

2.1. Definition and use

A statistical language model is simply a probability distribution $P(s)$ over all possible sentences s.[1]

It is instructive to compare statistical language modeling to computational linguistics. Admittedly, the two fields (and communities) have fuzzy boundaries, and a great deal of overlap. Nonetheless, one way to characterize the difference is as follows. Let S be the word sequence of a given sentence, i.e. its *surface* form, and let H be some *hidden* structure associated with it (i.e. its parse tree, word senses, etc.). Statistical language modeling is mostly about estimating Pr(S), whereas computational linguistics is mostly about estimating Pr$(H|S)$. Of course, if one could estimate well the joint Pr(S,H), both Pr(S) and Pr$(H|S)$ could be derived from it. In practice, this is usually not feasible.

Statistical language models are usually used in the context of a Bayes classifier, where they can play the role of either the prior or the likelihood function. For example, in **automatic speech recognition**, given an acoustic signal a, the goal is to find the sentence s that is most likely to have been spoken. Using a Bayesian framework, the solution is:

$$s^* = \arg\max_s P(s|a) = \arg\max_s P(a|s) \cdot P(s) \qquad (1)$$

where the language model $P(s)$ plays the role of the prior. In contrast, in **document classification**, given a document d, the goal is to find the class c to which it belongs. Typically, examples of documents from each of the (say) k classes are given, from which k different language models $\{P_1(d)$, $P_2(d)$, $\ldots$, $P_k(d)\}$ are constructed. Using a Bayes classifier, the solution c^* is:

$$c^* = \arg\max_c P(c|d) = \arg\max_c P(d|c) \cdot P(c) \qquad (2)$$

where the language model $P_c(d)$ plays the role of the likelihood. In a similar fashion, one can derive the role of language models in Bayesian classifiers for the other language technologies listed above.

2.2. Measures of progress

To assess the quality of a given language modeling technique, the likelihood of new data is most commonly used. The average log likelihood of a new random sample is given by:

$$\text{Average-Log-Likelihood}(D|M) = \frac{1}{n}\sum_i \log P_M(D_i) \qquad (3)$$

where $D = \{D_1, D_2, \ldots, D_n\}$ is the new data sample, and M is the given language model. This latter quantity can also be viewed as an empirical estimate of the *cross entropy* of the true (but unknown) data distribution P with regard to the model distribution P_M:

$$\text{cross-entropy}(P;P_M) \;=\; -\sum_D P(D) \cdot logP_M(D) \qquad (4)$$

Actual performance of language models is often reported in terms of *perplexity* [6]:

$$\mathrm{perplexity}(P;P_M) = 2^{\text{cross-entropy}(P;P_M)} \qquad (5)$$

Perplexity can be interpreted as the (geometric) average branching factor of the language according to the model. It is a function of both the language and the model. When considered a function of the model, it measures how good the model is (the better the model, the lower the perplexity). When considered a function of the language, it estimates the entropy, or complexity, of that language.

Ultimately, the quality of a language model must be measured by its effect on the specific application for which it was designed, namely by its effect on the error rate of that application. However, error rates are typically non-linear and poorly understood functions of the language model. Lower perplexity usually result in lower error rates, but there are plenty of counterexamples in the literature. As a rough rule of thumb, reduction of 5% in perplexity is usually not practically significant; a 10%–20% reduction is noteworthy, and usually (but not always) translates into some improvement in application performance; a perplexity improvement of 30% or more over a good baseline is quite significant (and rare!).

Several attempts have been made to devise metrics that are better correlated with application error rate than perplexity, yet are easier to optimize than the error rate itself. These attempts have met with limited success. For now, perplexity continues to be the preferred metric for practical language model construction. For more details, see [7].

2.3. Known weaknesses in current models

Even the simplest language model has a drastic effect on the application in which it is used (this can be observed by, say, removing the language model from a speech recognition system). However, current language modeling techniques are far from optimal. Evidence for this comes from several sources:

[1] Or spoken utterances, documents, or any other linguistic unit.

Brittleness across domains: Current language models are extremely sensitive to changes in the style, topic or genre of the text on which they are trained. For example, to model casual phone conversations, one is much better off using 2 million words of transcripts from such conversations than using 140 million words of transcripts from TV and radio news broadcasts. This effect is quite strong even for changes that seem trivial to a human: a language model trained on Dow-Jones newswire text will see its perplexity *doubled* when applied to the very similar Associated Press newswire text from the same time period ([8, p. 220]).

False independence assumption: In order to remain tractable, virtually all existing language modeling techniques assume some form of independence among different portions of the same document. For example, the most commonly used model, the n-gram, assumes that the probability of the next word in a sentence depends only on the identity of the last n-1 words. Yet even a cursory look at any natural text proves this assumption patently false. False independence assumptions in statistical models usually lead to overly sharp distributions. This is precisely what is happening in language modeling, as can be seen for example in document classification: the posterior computed by equation 2 is usually extremely sharp, reaching virtually one for one of the classes and virtually zero for all others. This of course cannot be the true posterior, since the average classification error rate is typically much greater than zero.

Shannon-style experiments: Claude Shannon pioneered the technique of eliciting human knowledge of language by asking human subjects to predict the next element of text [9, 10]. Shannon used this technique to bound the entropy of English. [11] formulated a gambling setup and used it to derive its own estimate of the entropy of English. In the 1980s, the speech and language research group at IBM performed ‘Shannon-style’ experiments, in which potential sources for language modeling improvement were identified by observing and analyzing the performance of human subjects in predicting or correcting text. Since then, Shannon-style experiments have been performed by several other researchers. For example, [12] performed experiments aimed at establishing the potential for language modeling improvements in specific linguistic areas. A common observation during all these experiments is that people improve on the performance of a language model easily, routinely and substantially. They apparently do so by using reasoning at the linguistic, common sense, and domain levels.

3. SURVEY OF MAJOR SLM TECHNIQUES

This section briefly reviews major established SLM techniques. For a more detailed technical treatment, see [13].

Almost all language models to date decompose the probability of a sentence into a product of conditional probabilities:

$$\Pr(s) \overset{\text{def}}{=} \Pr(w_1 \ldots w_n) = \prod_{i=1}^{n} \Pr(w_i|h_i) \qquad (6)$$

where w_i is the ith word in the sentence, and $h_i \overset{\text{def}}{=} \{w_1, w_2, \ldots w_{i-1}\}$ is called the *history*.

3.1. n-grams

n-grams are the staple of current speech recognition technology. Virtually all commercial speech recognition products use some form of an n-gram. An n-gram reduces the dimensionality of the estimation problem by modeling language as a Markov source of order n-1:

$$P(w_i|h_i) \approx P(w_i|w_{i-n+1}, \ldots, w_{i-1}) \qquad (7)$$

The value of n trades off the stability of the estimate (i.e. its variance) against its appropriateness (i.e. bias). A trigram (n=3) is a common choice with large training corpora (millions of words), whereas a bigram (n=2) is often used with smaller ones.

Deriving trigram and even bigram probabilities is still a sparse estimation problem, even with very large corpora. For example, after observing all trigrams (i.e., consecutive word triplets) in 38 million words’ worth of newspaper articles, a full third of trigrams in new articles from the same source are novel [8, p. 8]. Furthermore, even among the observed trigrams, the vast majority occurred only once, and the majority of the rest had similarly low counts. Therefore, straightforward maximum likelihood (ML) estimation of n-gram probabilities from counts is not advisable. Instead, various smoothing techniques have been developed. These include discounting the ML estimates [14, 15], recursively backing off to lower order n-grams [16, 17, 18], and linearly interpolating n-grams of different order [19]. Other approaches include variable-length n-gram [20, 21, 22, 23, 24] as well as a lattice approach [25]. Much work has been done to compare and perfect smoothing techniques under various conditions. A good recent analysis can be found in [26]. In addition, toolkits implementing the various techniques have been disseminated [27, 28, 29, 30].

Yet another way to battle sparseness is via vocabulary clustering. Let C_i be the class word w_i was assigned to. Then any of several model structures could be used. For example, for a trigram:

$$\begin{aligned}
\Pr(w_3|w_1, w_2) &= \Pr(w_3|C_3) \cdot \Pr(C_3|w_1, w_2) && (8)\\
\Pr(w_3|w_1, w_2) &= \Pr(w_3|C_3) \cdot \Pr(C_3|w_1, C_2) && (9)\\
\Pr(w_3|w_1, w_2) &= \Pr(w_3|C_3) \cdot \Pr(C_3|C_1, C_2) && (10)\\
\Pr(w_3|w_1, w_2) &= \Pr(w_3|C_1, C_2) && (11)
\end{aligned}$$

The quality of the resulting model depends of course on the clustering $C()$. In narrow discourse domains (e.g. ATIS, [31]), good results are often achieved by manual clustering of semantic categories (e.g. [32]). But in less constrained domains, manual clustering by linguistic categories (e.g. parts of speech) does not usually improve on the word-based model. Automatic, iterative clustering using information theoretic criteria [33, 34] applied to large corpora can sometimes reduce perplexity by 10% or so, but only after the model is interpolated with its word-based counterpart.

3.2. Decision tree models

Decision trees and CART-style [35] algorithms were first applied to language modeling by [36]. A decision tree can arbitrarily partition the space of histories by asking arbitrary binary questions about the history h at each of the internal nodes. The training data at each leaf is then used to construct a probability distribution $\Pr(w|h)$ over the next word. To reduce the variance of the estimate, this leaf distribution is interpolated with internal-node distributions found along the path to the root.

As usual, trees are grown by greedily selecting, at each node, the most informative question (as judged by reduction in entropy). Pruning and cross validation are also used.

Applying CART technology to language modeling is quite a challenge: The space of histories is very large (10^{100} for a 20 word sequence over a 100,000 word vocabulary), and the space of possible questions is even larger ($2^{10^{100}}$). Even if questions are restricted to individual words in the history, there are still $20 \cdot 2^{10^5}$ such questions. Very strong bias must be introduced, by restricting the class of questions to be considered and using greedy search algorithms. To support optimal single-word questions at a given node, algorithms were developed for rapid optimal binary partitioning of the vocabulary (e.g. [37]).

The first attempt at CART-style LM [36] used a history window of 20 words and restricted questions to individual words, though it allowed more complicated questions consisting of composites of simple questions. It took many months to train, and the result fell short of expectations: a 4% reduction in perplexity over the baseline trigram, and a further 9% reduction when interpolated with the latter. In the second attempt [38], much stronger bias was introduced: first, the vocabulary was clustered into a binary hierarchy as in [33], and each word was assigned a bit-string representing the path leading to it from the root. Then, tree questions were restricted to the identity of the most significant as-yet-unknown bit in each word in the history. This reduced the candidate set to a handful of questions at each node. Unfortunately, results here were also disappointing, and the approach was largely abandoned.

Theoretically, decision trees represent the ultimate in partition based models. It is likely that trees exist which significantly outperform ngrams. But finding them seems difficult, for both computational and data sparseness reasons.

3.3. Linguistically motivated models

While all SLMs get some inspiration from an intuitive view of language, in most models actual linguistic content is quite negligible. Several SLM techniques, however, are directly derived from grammars commonly uses by linguists.

Context free grammar (CFG) is a crude yet well understood model of natural language. A CFG is defined by a vocabulary, a set of non-terminal symbols and a set of production or transition rules. Sentences are generated, starting with an initial non-terminal, by repeated application of the transition rules, each transforming a non-terminal into a sequence of terminals (i.e. words) and non-terminals, until a terminals-only sequence is achieved. Specific CFGs have been created based on parsed and annotated corpora such as [39], with good, though still incomplete, coverage of new data.

A probabilistic (or stochastic) context free grammar puts a probability distribution on the transitions emanating from each non-terminal, thereby inducing a distribution over the set of all sentences. These transition probabilities can be estimated from annotated corpora using the Inside-Outside algorithm [40], an Estimation-Maximization (EM) algorithm (see [41]). However, the likelihood surfaces of these models tend to contain many local maxima, and the locally maximal likelihood points found by the algorithm usually fall short of the global maximum. Furthermore, even if global ML estimation were feasible, it is generally believed that context sensitive transition probabilities are needed to adequately account for actual behavior of language. Unfortunately, no efficient training algorithm is known for this situation.

In spite of this, [42] successfully incorporated CFG knowledge sources into a SLM to achieve a 15% reduction in a speech recognition error rate in the ATIS domain. They did so by parsing the utterances with a CFG to produce a sequence of grammatical fragments of various types, then constructing a trigram of fragment types to supplant the standard ngram.

Link grammar is a lexicalized grammar proposed by [43]. Each word is associated with one or more ordered sets of typed links; each such link must be connected to a similarly typed link of another word in the sentence. A legal parse consists of satisfying all links in the sentence via a planar graph. Link grammar has the same expressive power as a CFG, but arguably conforms better to human linguistic intuition. A link grammar for English has been constructed manually with good coverage. Probabilistic forms of link grammar have also been attempted [44]. Link grammar is related to dependency grammar, which will be discussed in section 4.

3.4. Exponential models

All models discussed so far suffer from data fragmentation, in that more detailed modeling necessarily results in each new parameter being estimated with less and less data. This is very apparent in decision trees, where, as the tree grows, leaves contain fewer and fewer data points.

Fragmentation can be avoided by using an exponential model of the form:

$$P(w|h) = \frac{1}{Z(h)} \cdot \exp[\sum_i \lambda_i f_i(h,w)] \qquad (12)$$

where λ_i are the parameters, $Z(h)$ is a normalizing term, and the features $f_i(h,w)$ are arbitrary functions of the word-history pair. Given a training corpus, the ML estimate can be shown to satisfy the constraints:

$$\sum_h \tilde{P}(h) \cdot \sum_w P(w|h) \cdot f_i(h,w) \;=\; E_{\tilde{P}}\; f_i(h,w) \qquad (13)$$

where $\tilde{P}$ is the empirical distribution of the training corpus.

The ML estimate can also be shown to coincide with the Maximum Entropy (ME) distribution [45], namely the one with highest entropy among all distributions satisfying equation 13. This unique ML/ME solution can be found by an iterative procedure [46, 47].

The ME paradigm, and the more general MDI framework, were first suggested for language modeling by [48], and have since seen considerable success (e.g. [49, 50, 8]). Its strength lies in principly incorporating arbitrary knowledge sources while avoiding fragmentation. For example, in [8], conventional ngrams, distance-2 ngrams, and long distance word pairs (“triggers”) were encoded as features, and resulted in up to 39% perplexity reduction and up to 14% speech recognition word error rate reduction over the trigram baseline.

While ME modeling is elegant and general, it is not without its weaknesses. Training a ME model is computationally challenging, and sometimes altogether infeasible. Using a ME model is also CPU intensive, because of the need for explicit normalization. Unnormalized ME modeling is attempted in [51]. ME smoothing is analyzed in [52].

The relative success of ME modeling focused attention on the remaining problem of feature induction, namely, selection of useful features to be included in the model. An automatic iterative procedure for selecting features from a given candidate set is described in [47]. An interactive procedure for eliciting candidate sets is described in [53].

ME language modeling remains the subject of intensive research; see for example [54, 55, 56, 57, 58].

3.5. Adaptive models

So far we have treated language as a homogeneous source. But in fact natural language is highly heterogeneous, with varying topics, genres and styles.

In **cross-domain adaptation**, test data comes from a source to which the language model has not been exposed during training. The only useful adaptation information is in the current document itself. A common and quite effective technique for exploiting this information is the cache: the (continuously developing) history is used to create, at runtime, a dynamic n-gram $P_{\text{cache}}(w|h)$, which in turn is interpolated with the static model:

$$P_{\text{adaptive}}(w|h) = \lambda P_{\text{static}}(w|h) + (1-\lambda) P_{\text{cache}}(w|h) \qquad (14)$$

with the weight λ optimized on held-out data. Cache LMs were first introduced by [59] and [60]. [61, 62] report reduction in perplexity, and [63] also reports reduction in recognition error rate. [64] introduced yet another adaptation scheme.

In **within-domain adaptation**, test data comes from the same source as the training data, but the latter is heterogeneous, consisting of many subsets with varying topics, styles, or both. Adaptation then proceeds in the following steps:

1. Clustering the training corpus along the dimension of variability, say, topic (e.g. [65]).

2. At runtime, identifying the topic or set of topics ([66, 67]) of the test data.

3. Locating appropriate subsets of the training corpus, and using them to build a specific model.

4. Combining the specific model with a corpus-wide model (in statistical terminology, shrinking the specific model towards the general one, to trade off the former’s variance against the latter’s bias). This is usually done via linear interpolation, at either the word probability level or the sentence probability level [65].

A special (and very common) case is when one has only small amounts of data in the target domain and large amounts in other domains. In this case, the only relevant step is the last one: combining models from the two domains. The outcome here is often disappointing, though: training data outside the domain has surprisingly little benefit. For example, when modeling the Switchboard domain (conversational speech, [68]), the 40 million words of the WSJ corpus (newspaper articles, [69]) and even the 140 million words of the BN corpus (broadcast news transcriptions, [70]) improve by only a few percentage points the application performance of the in-domain model trained on a paltry 2.5

million words. Although this is a significant improvement on such a difficult corpus, it is nonetheless disappointing considering the amount of data involved. By some estimates [71], another 1 million words of Switchboard data would help the model more than 30 million words of out-of-domain data. This suggest that our adaptation techniques are too crude.

4. PROMISING CURRENT DIRECTIONS

This section discusses current research directions that, in this author's subjective opinion, show significant promise.

4.1. Dependency models

Dependency grammars (DG) describe sentences in terms of asymmetric pairwise relationships among words. With a single exception, each word in the sentence is *dependent* upon one other word, called its *head* or *parent*. The single exception is the *root*, which serves as the head of the entire sentence. For more about DGs, see [72]. Probabilistic DGs have also been developed, together with algorithms for learning them from corpora (e.g. [73]).

Probabilistic dependency grammars are particularly suited to n-gram style modeling, where each word is predicted based on a small number of other words. The main difference is that in a conventional n-gram, the structure of the model is predetermined: each word is predicted from a few words that immediately preceded it. In DG, which words serve as predictors depends on the dependency graph, which is a hidden variable. A typical implementation will parse a sentence s to generate the most likely dependency graphs G_i (with attendant probabilities $P(G_i)$), compute for each of them a generation probability $P(s|G_i)$ (either n-gram style or perhaps as an ME model), and finally estimate the complete sentence probability as $P(s) \approx \sum_i P(G_i) \cdot P(s|G_i)$ (this is only approximate because the $P(G_i)$ themselves were derived from the sentence s.) Sometime $P(s)$ is further approximated as $P(s|G^*)$, where G^* is the single best scoring parse.

An example of such a model is [74], which uses the parser of [75] to generate the candidate parses, and trains the parameters using maximum entropy. The probabilistic link grammar [44] mentioned in section 3.3 also falls roughly in this category. Most recently, [76] employed a parser with probabilistic parameterization of a pushdown automata, and used an EM-type algorithm for training, with encouraging results (1% recognition word error rate reduction on the notoriously difficult Switchboard corpus). In all, this method of combining hidden linguistic structure with chain-rule parameterization can yield a linguistically grounded yet computationally tractable model.

4.2. Dimensionality reduction

One of the reasons language is so hard to model statistically is that it is ostensibly categorical, with an extremely large number of categories, or dimensions. A prime example is the vocabulary. To most language models, the vocabulary is but a very large set of unrelated entries. BANK is no closer to LOAN or to BANKS than it is to, say, BRAZIL. This results in a large number of parameters. Yet our linguistic intuition is that there is a great deal of structure in the relationship among words. We feel that the "true" dimension of the vocabulary is actually quite lower.

Similarly, for other phenomena in language, the underlying space may be of moderate or even low dimensionality. Consider topic adaptation. As the topic changes, the probabilities of almost all words in the vocabulary change. Since no two documents are exactly about the same thing, a straightforward approach would require an inordinate number of parameters. Yet the underlying topic space can be reasonably modeled in much fewer dimensions.

This is the motivation behind [77], which uses the technique of Latent Semantic Analysis ([78]) to simultaneously reduce the dimensionality of the vocabulary and that of the topic space. First, the occurrence of each vocabulary word in each document is tabulated. This very large matrix is then reduced via Singular Value Decomposition to a much lower dimension (typically 100–150). The new, smaller matrix captures the most salient correlations between specific combinations of words on one hand and clusters of documents on the other. The decomposition also yields matrices that project from document-space and word-space into the new, combined space. Consequently, any new document can be projected into the combined space, effectively being classified as a combination of the fundamental underlying topics, and adapted to accordingly. In [77], this type of adaptation is combined with an n-gram, and a perplexity reduction of 30% over a trigram baseline is reported. In [79], the technique is further developed and is found to also reduce recognition errors by 16% over a trigram baseline.

4.3. Whole sentence models

All language models described so far use the chain rule to decompose the probability of a sentence into a product of conditional probabilities of the type $\Pr(w|h)$. Historically, this has been done to facilitate estimation by relative counts. The decomposition is ostensibly harmless: after all, it is not an approximation but an exact equality. However, as a result, language modeling by and large has been reduced to modeling the distribution of a single word. This in turn may be a significant hindrance to modeling linguistic structure: some linguistic phenomena are impossible or at best awkward to think about, let alone encode, in a conditional framework. These include sentence-level features such as person and number agreement, semantic coherence, parsability, and even length. Furthermore, external influences on the sentence (e.g. previous sentences, topic) must be factored into the prediction of every word, which can cause small biases to compound.

To address these issues, [53] proposed a whole sentence exponential model:

$$P(s) = \frac{1}{Z} \cdot P_0(s) \cdot \exp\,[\,\sum_i \lambda_i f_i(s)\,] \qquad (15)$$

Compared with the conditional exponential model of equation 12, Z is now a true constant, which eliminates the serious burden of normalization. Most importantly, the features $f_i(s)$ can capture arbitrary properties of the entire sentence.

Training this model requires sampling from an exponential distribution, a non-trivial task. The use of Monte Carlo Markov Chain and other sampling methods for language is studied in [80]. Sampling efficiency is crucial. Consequently, the bottleneck in this model is not the number of features or amount of data, but rather how rare the features are, and how accurately they need to be modeled. Interestingly, it has been shown [81] that most of the benefit is likely to come from the more common features.

Parse-based features have been tried in [81], and semantic features are discussed in [53]. An interactive methodology for feature induction was also proposed in [53]. This methodology leads to a formulation of the training problem as logistic regression, with significant practical benefits over ML training.

5. CHALLENGES

Perhaps the most frustrating aspect of statistical language modeling is the contrast between our intuition as speakers of natural language and the over-simplistic nature of our most successful models.

As native speakers, we feel strongly that language has a deep structure. Yet we are not sure how to articulate that structure, let alone encode it, in a probabilistic framework. Established linguistic theories have been of surprisingly little help here, probably because their goal is to draw a line between what is properly in the language and what isn't, whereas SLM's goals are quite different.

As an example, consider the problem of clustering the vocabulary words which was discussed in section 3.1. As mentioned there, several automatic iterative methods have been proposed (e.g. [33, 34]). Table 1 lists example word classes derived by such a method [82]. While most words' placement appear satisfactory, a few of the words seem out of place. Not surprisingly, these are often words whose count in the corpus was insufficient for reliable assignment. Ironically, it is exactly these words which stood to benefit the most from clustering. In general, the more reliably a word can be assigned to a class, the less it will benefit from that assignment. How then is vocabulary clustering to become effective?

Table 1: Data driven word classes.

COMMITTEE COMMISSION PANEL SUBCOMMITTEE WONK
THEMSELVES MYSELF YOURSELF UNBECOMING ...
ATTORNEY SURGEON RUKEYSER CONSUL RICKEY ...
ACTION ACTIVITY INTERVENTION ATTACHE WARFARE ...
CENTER ASSOCIATION FACETED INSTITUTE GUILD ...
PARTICULAR YEAR'S NIGHT'S MORNING'S FATEFUL ...

I believe that the solution to this problem, and others like it, is to inject human knowledge of language into the process. This can take the following forms:

Interactive modeling. Data-driven optimization and human knowledge and decision making can play complementary roles in an intertwined iterative process. For the vocabulary clustering problem, this means that a human is put in the loop, to arbitrate some borderline decisions and override others. For example, a human can decide that 'TUESDAY' belongs in the same cluster as 'MONDAY', 'WEDNESDAY', 'THURSDAY' and 'FRIDAY', even if it did not occur enough times to be placed there automatically, and even if it did not occur at all. Another example of this approach is the interactive feature induction methodology described in [53].

Encoding knowledge as priors. One of the perils of using human knowledge is that it is often overstated, and sometimes wrong. Thus a better solution might be to encode such knowledge as a prior in a Bayesian updating scheme. After training, whatever phenomena are not sufficiently represented in the training corpus will continue to be captured thanks to the prior. Whenever enough data exist, however, they will override the prior. For the vocabulary clustering problem, experts' beliefs about the relationships between vocabulary entries must be suitably encoded, and the clustering paradigm must be changed to optimize an appropriate posterior measure. Thus, in the example above, enough data may exist to separate out 'FRIDAY' because of its use in phrases like "Thank God It's Friday".

Encoding linguistic knowledge as a prior is an exciting challenge which has yet to be seriously attempted. This will likely include defining a distance metric over words and phrases, and a stochastic version of structured word ontologies like WordNet [83]. At the syntactic level, it could include Bayesian versions of manually created lexicalized grammars. In practice, the Bayesian framework and the interactive process may be combined, taking advantage of the superior theoretical foundation of the former and the computational advantages of the latter.

Acknowledgements

I am grateful to Stanly Chen, Sanjeev Khudanpur, John Lafferty and Bob Moore for helpful comments.

6. REFERENCES

[1] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June 1990.

[2] Ralf brown and Robert & Frederking. Applying statistical English language modeling to symbolic machine translation. In *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'95)*, pages 221–239, July 1995.

[3] J. Ponte and W. Bruce. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st international conference on research and development in information retrieval (SIGIR'98)*, pages 275–281, 1998.

[4] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual conference on research and development in information retrieval (SIGIR'99)*, pages 222–229, 1999.

[5] Fred Jelinek. The 1995 language modeling summer workshop at Johns Hopkins University. Closing remarks.

[6] Lalit R. Bahl, Jim K. Baker, Frederick Jelinek, and Robert L. Mercer. Perplexity - a measure of the difficulty of speech recognition tasks. *Program of the 94th Meeting of the Acoustical Society of America J. Acoust. Soc. Am.*, 62:S63, 1977. Suppl. no. 1.

[7] Stanley F. Chen, Douglas Beeferman, and Ronald Rosenfeld. Evaluation metrics for language models. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 275–280, 1998.

[8] Ronald Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228, 1996. longer version published as "Adaptive Statistical Language Modeling: A Maximum Entropy Approach," Ph.D. thesis, Computer Science Department, Carnegie Mellon University, TR CMU-CS-94-138, April 1994.

[9] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423,623–656, 1948.

[10] C.E. Shannon. Prediction and entropy of printed English. *Bell Systems Technical Journal*, 30:50–64, January 1951.

[11] T.M. Cover and R.C. King. A convergent gambling estimate of the entropy of English. *IEEE Transactions on Information Theory*, 24(4):413–421, 1978.

[12] E. Brill, R. Florian, C. Henderson, and L. Mangu. Beyond n-grams: Can linguistic sophistication improve language modeling? In *Proceedings of the 36th Annual Meeting of the ACL*, 1998.

[13] Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, Massachusetts, 1997.

[14] I.J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3 and 4):237–264, 1953.

[15] Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, July 1991.

[16] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, March 1987.

[17] Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependences in stochastic language modeling. *Computer Speech and Language*, 8:1–38, 1994.

[18] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184, Detroit, Michigan, May 1995.

[19] Frederick Jelinek and Robert L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands: North-Holland, May 1980.

[20] D. Ron, Y. Singer, and N. Tishby. The power of amnesia. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 176–183. Morgam Kaufmann, San Mateo, CA, 1994.

[21] I. Guyon and F. Pereira. Design of a linguistic postprocessor using variable memory length Markov models. In *Proceedings of the 3rd ICDAR*, pages 454–457, 1995.

[22] Reinhard Kneser. Statistical language modeling using a variable context length. In *Proceedings of ICSLP*, volume 1, pages 494–497, Philadelphia, October 1996.

[23] Thomas Niesler and Philip Woodland. Variable-length category n-gram language models. *Computer Speech and Language*, 21:1–26, 1999.

[24] Man-Hung Siu and Mari Ostendorf. Variable n-gram and extensions for conversational speech language modeling. *IEEE Transactions on Speech and Audio Processing*, 8(1):63–75, 2000.

[25] Pierre Dupont and Ronald Rosenfeld. Lattice based language models. Technical Report CMU-CS-97-173, Carnegie Mellon University, Department of Computer Science, September 1997.

[26] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the ACL*, pages 310–318, Santa Cruz, California, June 1996.

[27] Ronald Rosenfeld. The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation. In *Proceedings of the Spoken Language Systems Technology Workshop*, pages 47–50, Austin, Texas, January 1995.

[28] Philip Clarkson and Ronald Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, 1997.

[29] Andreas Stolcke. SRILM—the SRI language modeling toolkit. http://www.speech.sri.com/projects/srilm/, 1999.

[30] Stanley F. Chen. Language model tools (v0.1) user's guide. http://www.cs.cmu.edu/ sfc/manuals/h015c.ps, December 1998.

[31] Patti J. Price. Evaluation of spoken language systems: the atis domain. In *Proceedings of the DARPA Speech and Natural Language Workshop*, June 1990.

[32] Wayne H. Ward. The cmu air travel information service: understanding spontaneous speech. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 127–129, June 1990.

[33] Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December 1992.

[34] Reinhard Kneser and Hermann Ney. Improved clustering techniques for class-based statistical language modeling. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, 1993.

[35] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees.* Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, California, 1984.

[36] Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:1001–1008, July 1989.

[37] Arthur Nádas, David Nahamoo, Michael A. Picheny, and Jeffrey Powell. An iterative "flip-flop" approximation of the most informative split in the construction of decision trees. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, May 1991.

[38] Peter F. Brown, Steven A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer, and Philip S. Resnik. Language modeling using decision trees. research report, I.B.M. Research, Yorktown Heights, NY, 1991.

[39] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treeback. *Computational Linguistics*, 19(2), 1993.

[40] James K. Baker. Trainable grammars for speech recognition. In *Proceedings of the Spring Conference of the Acoustical Society of America*, pages 547–550, Boston, MA, June 1979.

[41] Frederick Jelinek, John D. Lafferty, and Robert L. Mercer. Basic methods of probabilistic context-free grammars. In P. Laface and R. De Mori, editors, *Speech Recognition and Understanding: Recent Advances, Trends, and Applications*, volume 75 of *F: Computer and Systems Sciences*, pages 345–360. Springer Verlag, 1992.

[42] R. Moore, D. Appelt, J. Dowding, J. M. Gawron, and D. Moran. Combining linguistic and statistical knowledge sources in natural-language processing for atis. In *Spoken Language Systems Technology Workshop*, pages 261–264, Austin, Texas, February 1995. Morgan Kaufmann Publishers, Inc.

[43] Danny Sleator and Davy Temperley. Parsing English with a link grammar. Technical Report CMU-CS-91-196, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, October 1991.

[44] John D. Lafferty, Danny Sleator, and Davy Temperley. Grammatical trigrams: a probabilistic model of link grammar. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural language*, Cambridge, MA, October 1992.

[45] E.T. Jaynes. Information theory and statistical mechanics. *Physics Reviews*, 106:620–630, 1957.

[46] J.N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43:1470–1480, 1972.

[47] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April 1997.

[48] S. Della Pietra, V. Della Pietra, R.L. Mercer, and S. Roukos. Adaptive language modeling using minimum discriminant estimation. In *Proceedings of the Speech and Natural Language DARPA Workshop*, February 1992.

[49] Raymond Lau, Ronald Rosenfeld, and Salim Roukos. Trigger-based language models: A maximum entropy approach. In *Proceedings of ICASSP-93*, pages II–45 – II–48, April 1993.

[50] Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[51] Stanley F. Chen, Kristie Seymore, and Ronald Rosenfeld. Topic adaptation for language modeling using unnormalized exponential models. In *ICASSP-98*, Seattle, Washington, 1998.

[52] Stan F. Chen and Ronald Rosenfeld. A survey of smoothing techniques for me models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50, 2000.

[53] Ronald Rosenfeld, Larry Wasserman, Can Cai, and Xiaojin Zhu. Interactive feature induction and logistic regression for whole sentence exponential language models. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Keystone, CO, December 1999.

[54] Doug Beeferman, Adam Berger, and John Lafferty. A model of lexical attraction and repulsion. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 373–380, Madrid, Spain, 1997.

[55] John D. Lafferty and Bernard Suhm. Cluster expansions and iterative scaling for maximum entropy language models. In K. Hanson and R. Silver, editors, *Maximum Entropy and Bayesian Methods*, pages 195–202. Kluwer Academic Publishers, 1995.

[56] Jochen Peters and Dietrich Klakow. Compact maximum entropy language models. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Keystone, CO, December 1999.

[57] Sanjeev Khudanpur and Jun Wu. A maximum entropy language model integrating n-grams and topic dependencies for conversational speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, AZ, 1999.

[58] Jun Wu and Sanjeev Khudanpur. Combining nonlocal, syntactic and n-gram dependencies in language modeling. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, Budapest, Hungary, 1999.

[59] Roland Kuhn. Speech recognition and the frequency of recently used words: A modified markov model for natural language. In *12th International Conference on Computational Linguistics*, pages 348–350, Budapest, August 1988.

[60] Julian Kupiec. Probabilistic models of short and long distance word dependencies in running text. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, pages 290–295, February 1989.

[61] Roland Kuhn and Renato De Mori. A cache-based natural language model for speech reproduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(6):570–583, 1990.

[62] Roland Kuhn and Renato De Mori. Correction to: A cache-based natural language model for speech reproduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-14(6):691–692, June 1992.

[63] Fred Jelinek, Salim Roukos Bernard Merialdo, and M. Strauss. A dynamic language model for speech recognition. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, pages 293–295, February 1991.

[64] Reinhard Kneser and Volker Steinbiss. On the dynamic adaptation of stochastic language models. In *Proceedings of the IEEE conference on acoustics, speech and signal processing*, pages 586–589, Minneapolis, MN, 1993. volume II.

[65] Rukmini Iyer and Mari Ostendorf. Modeling long distance dependence in language: Topic mixture vs. dynamic cache models. *IEEE Transactions on Speech and Audio Processing IEEE-SAP*, 7:30–39, 1999.

[66] Kristie Seymore and Ronald Rosenfeld. Using story topics for language model adaptation. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, 1997.

[67] Kristie Seymore, Stanley Chen, and Ronald Rosenfeld. Nonlinear interpolation of topic models for language model adaptation. In *Proceedings of ICSLP-98*, 1998.

[68] J.J. Godfrey, E.C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 517–520, March 1992.

[69] Douglas B. Paul and Janet M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 357–362, February 1992.

[70] David Graff. The 1996 broadcast news speech and language model corpus. In *Proceedings of the DARPA Workshop on Spoken Language technology*, pages 11–14, 1997.

[71] Ronald Rosenfeld, Rajeev Agarwal, Bill Byrne, Rukmini Iyer, Mark Liberman, Elizabeth Shriberg, Jack Unverferth, Dimitra Vergyri, and Enrique Vidal. Error analysis and disfluency modeling in the switchbboard domain. In *Proceedings of the International Conference on Speech and Language Processing*, 1996.

[72] http://ufal.mff.cuni.cz/dg-bib2.html.

[73] Glenn Carrol and Eugene Charniak. Two experiments on learning probabilistic dependency grammars from corpora. Technical Report TR 92-16, Computer Science Department, Brown University, 1992.

[74] Ciprian Chelba, David Engle, frederick Jelinek, Victor Jimenaz, Sanjeev Khudanpur, Lidia Mangu, Harry Printz, Eric Ristad, Ronald Rosenfeld, Andreas Stolcke, and Dekai Wu. Structure and performance of a dependency language model. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, pages 2775–2778, 1997. volume 5.

[75] Michael Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th annual meeting of the association for Computational Linguistics*, pages 184–191, May 1996.

[76] Ciprian Chelba and Fred Jelinek. Recognition performance of a structured language model. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, pages 1567–1570, 1999. volume 4.

[77] Jerome R. Bellegarda. A multi-span language modeling framework for large vocabulary speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6:456–467, 1998.

[78] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. Am. Soc. Inform. Science*, 41:391–407, 1990.

[79] Jerome R. Bellegarda. Large vocabulary speech recognition with multi-span statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1):76–84, 2000.

[80] Stanley F. Chen and Ronald Rosenfeld. Efficient sampling and feature selection in whole sentence maximum entropy language models. In *ICASSP-99*, Phoenix, Arizona, 1999.

[81] Xiaojin Zhu, Stanley F. Chen, and Ronald Rosenfeld. Linguistic features for whole sentence maximum entropy language models. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, Budapest, Hungary, 1999.

[82] Stanley F. Chen. Unpublished work. 1998.

[83] Christiane FellBaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. MIT Press, 1998.