STOCHASTIC MODELING OF LANGUAGE VIA SENTENCE SPACE PARTITIONING

Alex Martelli
IBM Rome Scientific Center
via Giorgione 159, ROME (Italy)

ABSTRACT

In some computer applications of linguistics (such as maximum-likelihood decoding of speech or handwriting), the purpose of the language-handling component (*Language Model*) is to estimate the linguistic (a priori) *probability* of arbitrary natural-language sentences. This paper discusses theoretical and practical issues regarding an approach to building such a language model based on any equivalence criterion defined on incomplete sentences, and experimental results and measurements performed on such a model of the Italian language, which is a part of the prototype for the recognition of spoken Italian built at the IBM Rome Scintific Center.

STOCHASTIC MODELS OF LANGUAGE

In some computer applications, it is necessary to have a way to estimate the probability of any arbitrary natural-language sentence. A prominent example is maximum-likelihood speech recognition (as discussed in [1], [4], [7]), whose underlying mathematical approach can be generalized to recognition of natural language "encoded" in any medium (e.g. handwriting). The subsystem which estimates this probability can be called a *stochastic model* of the target language.

If the sentence is to be recognized while it is being produced (as necessary for a *real-time* application), the computation of its probability should proceed "left-to-right," i.e. word by word from the beginning towards the end of the sentence, allowing application of fast tree-search algorithms such as *stack decoding*[5] . Left-to-right computation of the probability of any word string is made possible by a formal manipulation based on the definition of conditional probability: if W_i is the i-th word in the sequence $\overline{W}$ of length N, then:

$$P(\overline{W}) = \prod_{i=1}^{N} P(W_i | W_{i-1}, W_{i-2}, \dots , W_1)$$

In other terms, the probability of a sequence of words is the product of the conditional probability of each word, given all of the previous ones. As a formal step, this holds for full sentences as well as for any subsequence within a sentence, and also for multi-sentence pieces of text, as long as sentence boundaries are explicitly accounted for (typically by introducing a pseudo-word as sentence boundary marker). We shall apply this equation only to subsequences occurring at the *start* of sentences (i.e. "incomplete" sentences); thus, the unconditional probability $P(W_1)$ can meaningfully be read as the probability that the particular word W_1, rather than any other word, will be the one starting a sentence.

The language model will thus consist essentially of a way to compute the conditional probability of any (*target*) word given all of the words that precede it in the sentence. For brevity, we shall call this (possibly empty) subsequence of the sentence to the left of the target word its **prefix**, using this term interchangeably with **incomplete sentence**, and we shall refer to the operation of conditional probability estimation given an incomplete sentence as *predicting the next word* in the sentence. A stochastic language model in this form may be said to be in *predictive normal form* [2].

The predictive power of two language models in predictive normal form can always be compared on an empirical basis, no matter how different their internal structures may be, by using the *perplexity* statistic introduced in [6]; the perplexity, computed by applying a language model in predictive normal form to an arbitrary body of text, can be interpreted as the average number of words among which the model is "in doubt" at every context along the text (this can be made rigorous along the lines of the argument in [13]).

TRAINING THE MODEL

A naive statistical approach to the estimation of the conditional probabilities of words given prefixes, to build a language model in predictive normal form, would simply collect occurrences of each prefix in a large corpus, using the relative frequencies of following words as estimates of probability. This is clearly unfeasible: no matter how large the available corpus, the possible prefixes will be yet more numerous; thus, most of them will not be observed in the corpus, and those which are observed will only be seen followed by a very limited and unrepresentative subset of the words that can come after them.

This problem stems directly from the fact that the number of elements in the set ("space") of different possible (incomplete) sentences is too high; thus, it can be met head-on by simply reducing the number of incomplete sentences *which are deemed to differ* **significantly** *for prediction purposes*, i.e. by passing to the quotient space of the sentence space on a suitable equivalence relation; in other words, by using as, ***contexts*** of the language model, the equivalence classes in a partition of the set of all prefixes, rather than the prefixes themselves. The equivalence classification of prefixes can be based on any kind of linguistical knowledge, as long as it can be applied to two prefixes to judge if they can be deemed "similar enough" to allow us to expect that they should lead to the same prediction regarding the next word to be expected in the sentence. Indeed, the knowledge embodied in the equivalence classification need not be of the kind that would be commonly labeled "linguistical"; the equivalence criterion between two sentence prefixes need not be any more than the purely pragmatical "they behave similarly in predicting the next following word."

Let us assume that we already had a stochastic language model, in predictive normal form, somehow trained to our satisfaction. To each string of words, considered as a sentence prefix, there would be attached a probability distribution over all words in the dictionary, corresponding to the conditional probability that the word should follow this prefix. We could now apply sentence-space partitioning as follows: define a distance measure between probability distributions over the dictionary; apply any clustering algorithm to obtain the desired number of classes (or, cluster iteratively until further clustering would require merging of equivalence classes which are at a distance above some threshold). By this hypothetical process, we would be extracting linguistical knowledge (namely, which sequences of words can be deemed equivalent as regards the word which can be expected to follow them) from the model itself (thus, presumably, from the data it was trained upon). Since we **don't** have such a well-trained model to begin with, we will actually have to reverse the process: *start* by injecting some knowledge in the form of equivalence criteria, *obtain* from this a way to practically train the model.

One way to obtain the initial sentence-space partition could be from a parser able to work left-to-right on natural language sentences; each class in the partition would be the set of all sentence prefixes that take the parser's state to a given string of non-terminals (or rather, given the possibility of ambiguous parses, to a given *set* of such strings). We have not attempted this. What we *have* attempted is obtaining the equivalence relation on string of words from an equivalence relation on single words, which is far simpler to define (although, being a further approximation, it can be expected to give poorer results). Thus, if we define the equivalences:

```
Michele  ==  Giuseppe
pensa    ==  dice
```

we will have that "Michele dice" is equivalent to "Giuseppe pensa," and so on. One big advantage is that such equivalence classes on single words are relatively easy to obtain automatically (by clustering over any appropriate distance measure, as outlined in the hypothetical example above - the difference being that we **can** train single words adequately, without having to resort to a previous classification), thus leading to an automatical (although far from optimal) sentence-space partitioning on which the model's training can be based.

It should be noted at this point that this approach suffers from the "synonym problem": since equivalence relationships enjoy the transitive property, we risk deeming "equivalent" two items A and B which are actually quite different, by virtue of the fact that they both "resemble" a third item C. This problem depends on the "all or nothing" nature of equivalence relationships, and could be bypassed by a mathematically more general approach, based on the theory of Markov Sources (as outlined in [3], [8]). The latter can be said to stem from a generalization of sentence-space partitions to "fuzzy partitions" (probabilistic covers), i.e. from usage of a nondeterministic equivalence relation. However, as argued in [10], the greater generality, although aesthetically appealing, and no doubt useful against the "synonym problem," does not necessarily add enough power to the language model to offset the added computational burden; in many cases, Markov-source models can be practically reduced to sentence-space partitioning models.

One further generalization is the identification of equivalence relationships between word strings of different length. For example, verb forms such as "dice" or "pensa" could be deemed equivalent to themselves prefixed by the word "non," finally leading to equivalence between, say, "Mario dice" and "Giuseppe non pensa." Such equivalences could also, in principle, be tested automatically on statistical grounds. Finally, equivalence criteria thus obtained via statistical means are by no means ends in themselves, but can be integrated with other linguistical knowledge expressed as a partition of the sentence space, to build a stronger model. Indeed, the set of language models built on sentence space partitions inherits mathematical lattice properties from the set of partitions itself, through their natural correspondence, allowing simple but useful operation on language models to yield new language models. For example, the "least upper bound" operation on two language models gives the model based on the equivalence criterion which requires *both* equivalence criteria from the original models to be satisfied. Thus, for example, we could start from an equivalence criterion G defined on purely grammatical grounds (for example, by using a parser, such as suggested above), and another equivalence criterion S defined on statistical grounds (such as we have built as outlined above), and *merge* them into a new criterion SG, the laxer one which is still stronger than either, to obtain a finer partition (and thus, presumably, a better performing stochastical language model, assuming a reasonably large corpus is available to train it on).

APPLICATION AND RESULTS

Given a suitable equivalence criterion over prefixes, and a large corpus, the language model can now in principle be built by purely statistical means, by collecting the multiset of words following each equivalence class (context), and using relative frequencies as estimators of conditional probabilities.. However, this would require that the equivalence criterion be so lax (i.e., that it have so few contexts) that each of its contexts can be *guaranteed* to occur in the corpus *followed* by *all* different words that can possibly follow it, despite possible statistical fluctuations. This is an overly severe restriction that, even for a quite large corpus, would in practice constrain the model builder to use very weak equivalence classifications (i.e. ones of little discriminatory power).

A generalization of the *backing-off* methodology first proposed in [9] can be used to overcome this limitation. Rather than a single sentence-space partition, the model will need a *chain* of such partitions, progressively weaker, and ending with the weakest possible "partition" - the one which considers any prefix equivalent to any other (the maximal element in the above-mentioned lattice). "Elementary" models will be built, with the above statistical procedure, over each partition of the chain.

When using the model (now built as a chain of elementary models) in predictive form, if a prediction cannot be reliably obtained from the strongest model in the chain, the algorithm will then *back-off* to the next weakest model, and proceed recursively along the chain of elementary models until it finds one that can give a reliable prediction (the existence in the chain of the weakest conceivable model ensures termination).

The method requires that, along with its predictions, an elementary model deliver, for any given context, a measure of its own reliability. This can be quantified as follows: in any context, an elementary model must estimate the probability that the next word will *not* be in the set actually observed for that model in that context (i.e., the set of words it is able to predict). Thus, each step of backing-off will be performed in two cases: unconditionally, if an elementary model has no observations at all for prefixes equivalent to the target one; conditionally, if that context was indeed observed, but the target word was not observed in it (and in this latter case, the self-estimate of reliability of the elementary model will come into play).

For the estimation of the global probability of unobserved words in a context ("new" observations), there could be used the general approaches, based on Turing's heuristic, discussed in [11] and [12], which lead, in practice, to estimating the probability of "new" observations as the ratio of words observed once to total observations. We have found it more reliable to use a simpler approach (the "First-Time" heuristic), which directly estimates the probability of new observations as the ratio of different words observed to total observations.

This idea leads to strictly more pessimistic estimates of reliability of elementary models (in particular, it treats any word observed only once in a context as if never observed at all) and, judging from experimental results, seems to better model actual linguistic behavior. As expected, it proves particularly valuable when judging predictive power over poorly-trained material, specifically Italian sentences in a domain of discourse different from that of the training corpus. Using training data from the "Il Mondo" weekly magazine, the perplexity (with an 8000-word vocabulary) over other test sentences from the same magazine came to 113, and over news flashes from the Ansa agency to 174, using Turing's heuristic; while using the First-Time heuristic under the same experimental conditions gave values of 111 and 150 respectively.

Particularly with this heuristic, cross-domain behavior of such models appears quite acceptable. Our main training corpus was a set of articles and news flashes on economy and finance, from the "Il Mondo" weekly magazine and the "Ansa" new agency, for a total of about 6 million words; addition of just 50,000 words of inter-office memoranda made the perplexity of another test set of such memoranda (on a 3000-word vocabulary) decrease from 149 to 115, while naturally perplexity on test material homogeneous to the main body of the training corpus remained fixed (at 76).

REFERENCES

[1] L.R. Bahl, F. Jelinek, R.L. Mercer, **A maximum likelihood approach to continuous speech recognition,** *IEEE Trans. PAMI,* March 1983.

[2] R. Campo, L. Fissore, A. Martelli, G. Micca, G. Volpi, **Probabilistic Models of the Italian Language for Speech Recognition,** *Proc. Int. Work. Authomatic Speech Recognition,* Roma, Italy, May 1986.

[3] A.M. Derouault, B. Merialdo, **Language modeling at the syntactic level,** *Proc. Seventh Int. Conf. Pattern Recognition, Montreal, Canada,* July 30-August 2, 1984.

[4] P. D'Orta, M. Ferretti, A. Martelli, S. Melecrinis, S. Scarci, G. Volpi, **Il prototipo IBM per il riconoscimento del parlato,** *Note di Informatica, n. 13,* September 1986.

[5] F. Jelinek, **A fast sequential decoding algorithm using a stack,** *IBM Journal of Research and Development,* November 1969.

[6] F. Jelinek, R.L. Mercer, L.R. Bahl, J.K. Baker, **Perplexity - a measure of difficulty of speech recognition tasks,** *94th Meeting Acoustical Society of America, Miami Beach, FL,* December 15, 1977.

[7] F. Jelinek, **The development of an experimental discrete dictation recognizer,** *Proceedings of IEEE,* November 1985.

[8] F. Jelinek, **Self-Organized Language Modeling for Speech Recognition,** *IBM internal memo,* February 1986.

[9] S. Katz, **Recursive M-gram Language Model via a Smoothing of Turing's Formula,** *IBM Technical Disclosure Bulletin,* 1985.

[10] A. Martelli, **Modelli probabilistici della lingua italiana,** *Note di Informatica, n. 13,* September 1986.

[11] A. Nadas, **Estimation of probabilities in the language model of the IBM speech recognition system,** *IEEE Trans. on Acoustic, Speech and Signal Processing,* August 1984.

[12] A. Nadas, **On Turing's Formula for Word Probabilities,** *IEEE Trans. on Acoustic, Speech and Signal Processing,* December 1985.

[13] C.E. Shannon, **Prediction and entropy of printed English,** *Bell. Syst. Tech. Journal,* 1951.