Proposal for a mutual-information based language model

Uwe Jost and Eric Atwell
Centre for Computer Analysis of Language And Speech (CCALAS),
A.I. Division, School of Computer Studies,
University of Leeds, LS2 9JT, United Kingdom
email : uwe@scs.leeds.ac.uk eric@scs.leeds.ac.uk
phone : (0532) 335761 FAX: (0532) 335468

Abstract

We propose a probabilistic language model that is intended to overcome some of the limitations of the well-known n-gram models, namely the strong dependence of the parameter values of the model on the discourse domain and the constant size of word context taken into account. The new model is based on the mutual information (MI) measurement for the correlation of events and derives a hierarchy of categories from unlabelled training text. It has close analogies to the bi-gram model and is therefore explained by comparing it with this model.

1 Introduction

Language models (LMs) are used to capture regularities in languages and in this way to provide information about the possibility or likelihood of certain language constructs. For large-vocabulary speech and handwriting recognition, the acoustic or graphemic evidence gained by the input device may not be sufficient to decide on the word spoken or written with a reasonable amount of certainty. Such devices therefore usually output a set of candidates for each word, possibly labelled with a certainty score or else sorted on decreasing likelihood.

This problem of uncertainty is not only a problem of the "imperfect" computer but humans also often rely on contextual clues to find a preferred interpretation of an utterance. The decision between different alternatives is rarely made with a one hundred percent certainty and if the cost of a misunderstanding is large compared with the difference in certainty, humans ask for clarification.

Having a model of the target language allows the computer to make more intelligent guesses about the likelihood that a particular sequence of words has been the sequence that the speaker actually intended to utter in a specific situation.

The next chapter discusses basic types of models to specify the place of our language model in the system of models, chapter three gives a short summary on bi-gram models and chapter four introduces the new model.

2 Modelling

Models are commonly regarded as consisting of a set of categories that are defined in terms of a set of attributes, and a set of relationships between those categories. We will use this termonology throughout the remaining text.

2.1 Prototype versus representation

There is a very basic difference between modelling a natural system and designing a new artificial system. For example, designing a new car is certainly not the same task as deriving a model of horses. It is not self evident that the means used for the definition of a new system are the most appropriate ones for the modelling of existing ones, even if both systems have certain properties in common.

The difference is obvious if we look at models of formal languages (usually grammars) and models of natural languages (see for instance [Sam92] for a discussion of this topic). The former actually define the members of the language. The grammar of a programming language actually exists before the first program is written in this language. Nobody has the power to define which natural language constructs actually belong to English. In this case the task is to observe the language actually spoken (or written) and to abstract (generalize) from those observations to find an appropriate model of the language.

The criteria for the success of a LM are quite different for formal languages and natural ones. In the first case, the expressive power of the language to be designed is a major concern, computational costs need to be taken into consideration and of course, nobody would really want to design an ambiguous programming language.

Natural language modeling in contrast is judged by the degree to which the model reflects the relevant properties of the existing language. One may come up with a very efficient, very expressive, easy to remember grammar for a natural language, in which rules are always valid and there can never be two interpretations for the same construct. But this grammar will certainly not be a grammar of everyday English and it wouldn't be the first unsuccessful attempt to introduce a manually designed artificial language to replace a natural one.

For most real-life applications, modelling of a natural language has to deal with such problems as ambiguity, varying degrees of grammaticality and the fact that there is no ultimate authority that could decide wheter a certain sentence or the interpretation of a certain sentence is correct at a certain point in time. Different experts (native speakers or even linguists) sometimes fail to reach an agreement about such questions and a very simplistic notion that only allows to distinguish between "correct" and "incorrect" seems problematic.

On the other hand, in many countries efforts have been made to somehow simplify the language and to make it more regular. The whole system of language teaching has a systematizing effect on language. Hence natural languages are not purely a product of evolution but a considerable but varying degree of design is involved. However, even in the cases of generally accepted (grammar) rules for language generation, the problem still exists that computers are seldom supposed to be language teachers and that the user expects an appropriate response and not a lecture in grammar.

For speech and handwriting recognition, a model that could (only) adjucate between legal and illegal English sentences (possibly using many knowledge sources) would not even be of much use. In most real-life situations, there are a number of sentences that are "legal" and can be mistaken, especially when "legal" is defined to cover most of the language constructs produced by native speakers. We do not need a long unsorted list of possible legal interpretations of an utterance (if such a list were very long it would be of about as much use as no list at all) but a likelihood-sorted list of candidates and some kind of certainty score to serve as a decision criterion for further processing.

2.2 Classification of language models

There are essentially two ways in which language models differ; they can be distinguished by the way they generalize (i.e., which information they discard) and by the role humans play in the model-building process. In natural-language processing (NLP), generalisation is necessary for two reasons. Limited resources in terms of storage, computation and training data require simplification. Furthermore, generalisation is necessary to capture a potentially infinite number of different language constructs in a finite model. Some examples of generalisations are:

- recursion/iteration to express the fact that sequences of 2, 3, ... items of the same kind may appear

- classifying all sentences above a certain threshold of the degree of commonness of the sentence as grammatical and all other sentences as ungrammatical
- regarding all contexts in which a language construct may occur as equal if the previous n and/or following m items are the same
- ignoring certain properties of items (e.g. syntactic classes of words)

It should be noted that generalisation with respect to natural languages almost always implies a loss of information. In a programming language, all names of variables of the same type have exactly the same syntactic properties but it is hard to find any two English words that can be treated in exactly the same way without losing some information. A sequence of 10000 modules is as grammatical as a sequence of two modules in a programming language but this is certainly not true of sequences of noun phrases in natural languages. Regarding all programs with less than a fixed number of modules as grammatical and all other ones as ungrammatical is perfectly acceptable for formal systems but certainly a much too coarse distinction in NLP.

Mapping certain attribute values to real numbers allows arbitrarily fine distinctions to be made without the need to deal (directly) with an extremely high number of categories, provided it is possible to reason with degrees of such attributes. Fuzzy-approaches and probabilistic models make use of this idea.

The second important dimension for distinguishing language models is the role humans play in the process of model building. As explained in the previous sub-chapter, the origin of a LM for natural languages is always the observation of actually written or spoken sentences. The difference lies in the role of the human in the learning process. Some models are designed completely by hand, i.e. human experts have learned the language and somehow make their knowledge explicit in the language model. Corpora and computers may be used to test hypotheses or to derive statistics. On the other end of the scale there are models that contain only a minimal amount of human linguistic expertise and learning is almost completely performed using machine learning techniques. There are of course pros and cons to both extremes. An approach that could combine already existing and easy to formalise rule-based human expertise with efficient learning techniques would arguably be most promising.

The model we are proposing performes generalisation in a hierarchical fashion using a uniform statistical meassure and is rather near the“learning” end of our scale.

3 Bi-gram models

The Bi-gram model has been used widely and successfully in statistical language modelling. It allows the calculation of the (approximate) probability of any string of words using the fact, that the probability of a string of length $l+1$ is the probability of the string of length l times the probability of the word at position $l+1$ appearing after the string of length l. (see [Jel90])

$$P([w_1, w_2, ..., w_i]) = \prod_{i=1}^{n} P(w_i | w_1, ..., w_{i-1})$$

For instance, the probability of the sequence of three words $[a, b, c]$ could be calculated using:

$$P([a, b, c]) = P(a) \cdot P(b|a) \cdot P(c|[a, b])$$

The histories $w_1, ..., w_{i-1}$ are mapped to a number of equivalence classes by assuming all histories that end in the same word to be equivalent. This can be seen as modelling the language in terms of a high number of categories of two kinds: each word being a category and each history ending in a certain word also being a category. Between each pair of categories of different kinds one relationship is modelled; the probability that a word occurs given that the history was observed.

For our example string the calculation would be performed using:

$$P([a, b, c]) = P(a) \cdot P(b|A) \cdot P(c|B)$$

where capital letters stand for equivalence classes in which the "histories" were unified that end in the words denoted with the respective lower case letters .

This approach has some shortcomings. One is the high number of categories. It seems reasonable not to use equivalence classes only for histories but also for words. The n-POS model overcomes this problem by using (usually manually designed) part-of-speech classifications. [Jel90, p.490] suggests the use of a mutual information (MI)-based criterion to derive word equivalence classes automatically for n-POS modelling.

Another problem is the dependence of the probabilities of language constructs on the domain. A bi-gram model may perform well on test data not used in the training process but taken from the same corpus, but quite poorly with test data taken from a different source. This is essentially because n-gram models attempt to compare the total probabilities of sentences and these probabilities may differ considerably between different sources. We feel that it is worth examining whether the mutual information between words is less prone to differ between different text sources.

The uniform treatment of all words, histories and relationships between them may also not be very efficient. It is clear that the reduction of all histories to one word is very coarse. N-gram models with n larger than 2 are more precise; but the amount of training data and memory increases dramatically and even a 4-gram model will miss out certain significant differences between histories while modelling many insignificant ones. A suggestion from R.L. Mercer to use MI for defining a vocabulary that consists of larger units of words for n-gram modelling was mentioned in [Jel90, p. 461].

Other related papers include:

[JLTW93] extends the concept of a bigram to the most informative (rather than the immediate) previous word

[Atw83] [Atw87] describes a bi-pos model augmented with tri-grams for some empirically specified cases

[BdM$^+$92] combines a 3-gram model with a 3-POS model in which the word classes are derived using MI-based statistical methods

4 The new model

4.1 Intuitions

The number of possible different modelling approaches for natural language is huge (probably infinite) and a systematic search through the space of language models is hard to imagine. In building models humans rely on their knowledge about the system to be modelled and on their intuition. The intuitions behind the design of our model are the following:

- When humans reason about their language they use a hierarchy of syntactic units and describe relationships between them.

- A considerable proportion of human language competence is often expressed in judgements like “It sounds correct.”. We think that the modelling of collocational patterns (at different syntactic levels) can simulate this competence to a certain degree.

- The strength of associative relationships probably differs less between discourse domains than the frequency of certain constructs.

As an example for the last point consider the collocation “strong coffee”. It may be found much more often in fiction than in scientific texts. But it is also not very likely that the word “coffee” occurres in a scientific text very often and if it occurres, it seems more likely that the preceeding word is “strong” than that it is any (specific) other word. The strength of this attraction between “strong” and “coffee” certainly also differs between types of texts, but it seems likely that its deviation is smaller than the deviation of the pure probability of the construct “strong coffee”.

It should be noted that as a result of the simplification used in the bi-gram model, this model would also store the relative probability of “coffee” given that “strong” was the last word. However, the reverse relation is not modelled and the score calculated for a sentence is the unconditioned probability.

4.2 Association ratio

In [CH90], a mutual-information based “association ratio” (AR) meassure was introduced as a “objective meassure based on the information theoretic notion of mutual information, for estimating word association norms from computer readable corpora.” Our model is based on a generalised AR meassure that can be applied to more than two words. It is the quotient of the probability of the sequence of words and the product of the probabilities of the words. [1]

$$AR([w_1, w_2, ..., w_s]) = \frac{P([w_1, w_2, ..., w_s])}{P(w_1)P(w_2)...P(w_s)}$$

In this formula, $[w_1, w_2, ..., w_s]$ denotes a string consisting of s words. The AR is a meassure for the strength of the associative relationship between a number of words. If there were no relationship between the words $w_1, w_2, ..., w_s$, then we should expect the string $[w_1, w_2, ..., w_s]$ to occur with a probability equal to the product of the probabilities of the (in this case independent) words; $P([w_1, w_2, ..., w_s]) = P(w_1)P(w_2)...P(w_s)$ and the AR would be one. If certain words tend to occur together, then the AR between them should be larger than one and if they rather not co-occur then the AR would be smaller than one.

Other mutual information based measures has been used in various ways in natural language modelling. One was used in [BdM$^+$92] for automaticly deriving meaningful hierarchical word classifications from unrestricted English text. Those classifications were then used for n-POS modelling. In [MM90] a generalized mutual information meassure was used to detect boundaries of syntactical units recursively as the points of minimal mutual information between adjacent constituents.

The basic formula of our language model allows the calculation of the association ratio of a sentence in a hierarchical fashion. It calculates the AR between the leaves of a tree as the product of the association ratio of the leaves of the sub-trees and the association ration between the sub-trees. For binary trees this means:

$$AR([w_1..w_s]) = AR([w_1..w_a])AR([[w_1..w_a][w_{a+1}..w_s]])AR([w_{a+1}..w_s])$$

[1] We do not use a logarithm here to make explainations simpler.

In those formulae, $[w_1, w_2, ..., w_x]$ denotes the x leaves (words) of a (sub-) tree, $1 < a < s$ and $AR([w]) = 1$.

The meaning of our AR formula may become clearer by looking at the following derivation of the formula for three-word sequences:

$$\begin{aligned} AR([[w_1, w_2], w_3])AR([w_1, w_2]) &= \frac{P([w_1, w_2, w_3])}{P([w_1, w_2])P(w_3)} \frac{P([w_1, w_2])}{P(w_1)P(w_2)} \\ &= \frac{P([w_1, w_2, w_3])}{P(w_1)P(w_2)P(w_3)} \\ &= AR([w_1, w_2, w_3]) \end{aligned}$$

4.3 Equivalence classes and training

Similar to bi-gram models, without simplification this formula would require to effectively store all (sub-) trees that can be built from the training corpus. The bi-gram model unites all histories that end in the same words in equivalence classes. Similarly, we might define equivalence classes for all trees that have the same left (right) sub-trees and whose AR lies in a certain interval.

Using our formula, two sub-tries are joined to a new tree and the AR of all leafes of this new tree is calculated. In analogy to the bi-gram model we might call the left sub-tree to be joined a “history” and consequently the right sub-tree a “future”. It seems sensible not to simplify at the point where the sub-trees touch each other since the leftmost leaf (word) of the right sub-tree immediately follows the rightmost leaf (word) in the left sub-tree in the sentence. Hence we need to distinguish between equivalence classes for trees that are potential left sub-trees and ones that will become new right sub-trees. We will experiment with different ways of building equivalence classes to find an efficient method.

The model building algorithm could proceed as follows:

```
• [ unite words in equivalence classes ]
• repeat
    – select all pairs of elements (words/symbols) with an AR
      above a certain threshold
    – unite elements in equivalence classes
    – replace all occurrences of the newly derived equivalence
      classes in the training corpus by a new symbol
• endrepeat
```

In the initial (optional) step, all words that fulfil certain criteria (e.g., have (roughly) the same AR to the same preceding and/or following words) are put in equivalence classes. Then, a certain number of word-pairs with a high AR is selected and stored, they are united in a number of equivalence classes and each class is assigned to a new (meta-) symbol. The new symbols are now treated exactly as words (we call both such symbols and words elements) and the process is repeated.

4.4 Recognition

Our trained model will have certain analogies to probabilistic context free grammars (PCFGs). The new symbols generated in the learning process can be seen as meta-symbols in grammars and the AR information is a score for the quality of the substring "parse". Recognition can be performed using parsing techniques as known from PCFGs (e.g. [Wri90]). The resulting score for a parse is not its probability; but the AR value is sufficient to compare different alternatives. It will be interesting to observe whether the trees found in the recognition process have any meaning to humans.

An apparent disadvantage of the proposed model is the inclusion of "right" context in the decision about words. On the other hand, although an n-gram model can "guess" the next word by only taking into account previous words, to find the optimal path through a lattice, it also needs to compare complete paths. To deal with the problem of sequential input in speech and handwriting recognition, n-gram models often make intermediate guesses and allow for later corrections. We might use similar methods, starting with small sub-trees and a dynamic AR threshold for accepting intermediate hypotheses.

The following table summarizes some analogies and differences between the bi-gram model and the proposed model:

	bi-gram	proposed model
score calculated for each sentence or part of sentence to be scored	probability	association ratio between the words
way in which calculation proceeds	left to right using the equation: $P([w_1, w_2, ..., w_i])$ $= \prod_{i=1}^{n} P(w_i \| w_1, ..., w_{i-1})$	hierarchical tree joining using: $AR([w_1..w_s])$ $= AR([w_1..w_a])$ $\cdot AR([[w_1..w_a][w_{a+1}..w_s]])$ $\cdot AR([w_{a+1}..w_s])$
simplification (generalisation)	equivalence classes for all strings that end in the same word ("histories")	equivalence classes for left and right sub-trees

References

[Atw83] Eric Steven Atwell. Constituent-likelihood grammar. *ICAME Journal*, 7:34–66, 1983.

[Atw87] Eric Steven Atwell. Constituent-likelihood grammar. In R. Garside, G.N. Leech, and G.R. Sampson, editors, *The Computational Analysis of English : A Corpus-Based Approach*, pages 57–65. Longman, London, 1987.

[BdM+92] P. F. Brown, P.V. deSouza, R. L. Mercer, V.J.D. Pietra, and J.C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[CH90] K.W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–9, March 1990.

[Jel90] Fred Jelinek. Self-organized language modelling for speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*, chapter 8.1, pages 450–506. Morgan Kaufmann, San Mateo, California, 1990.

[JLTW93] G.J.F. Jones, H. Lloyd-Thomas, and J.H. Wright. Adaptive statistical and grammar models of language for application to speech recognition. In *IEE Colloquium on 'Grammatical Inference : Theory, Applications and Alternatives' (Digest No.092)*, pages 25/1–8. Centre for Commun. Res., Bristol Univ., UK, 1993.

[MM90] D.M. Magerman and M.P. Marcus. Parsing a natural language using mutual information statistics. In *AAAI-90 Proceedings. Eighth National Conference on Artificial Intelligence*, volume 2, pages 984–9. CIS Dept., Pennsylvania Univ., Philadelphia, PA, USA, 1990.

[Sam92] G. Sampson. Probabilistic parsing. In Jan Svartvik, editor, *Directions in Corpus Linguistics*, pages 419–447. Mouton de Gruyter, 1992.

[Wri90] J.H. Wright. LR parsing of probabilistic grammars with input uncertainty for speech recognition. *Computer Speech and Language*, 4(4):297–323, Oct. 1990.