EVALUATING HIERARCHICAL HYBRID STATISTICAL LANGUAGE MODELS

Lucian Galescu, James Allen

University of Rochester, USA
{galescu,james}@cs.rochester.edu

ABSTRACT

We introduce in this paper a hierarchical hybrid statistical language model, represented as a collection of local models plus a general model that binds together the local ones. The model provides a unified framework for modelling language both above and below the word level, and we exemplify with models of both kinds for a large vocabulary task domain. To our knowledge this is the first paper to report an extensive evaluation of the improvements achieved from the use of local models within a hierarchical framework in comparison with a conventional word-based trigram model.

1. INTRODUCTION

In recent years the idea that statistical language models for speech recognition would benefit from taking into account some of the linguistic structure of speech has started to catch on [1, 4, 5, 10, 17]. As full-coverage probabilistic context-free grammars (PCFGs), which are inherently structural, are hard to obtain and inefficient to use, we would like to still use finite-state models, but whenever possible, to use local models of higher complexity. The rationale is that different parts of a sentence could be described more finely by such local models than by a conventional n-gram model trained on a large corpus. For further benefits of this approach, including portability of local models, we refer to the companion paper in these proceedings, [3].

The models we are concerned with, described in more detail in Section 2, can be thought of as a generalization of both class-based models ([12, 18]) and phrase-based models ([12]). The overall language model is composed of a collection of sub-models, including a sentence model and several local models that describe classes of words or phrases. The units of the sentence model are either words, or class tags that stand for the local models. In most cases the classes are domain-specific, but there are a few examples of hierarchical models using more general syntactically-derived classes ([8, 9, 17]).

Currently, all the similar models reported have a two-level overall structure. It may be useful that local models' units also include class tags, allowing the model to develop into a true hierarchical model. Also, the framework makes it possible to integrate models for words, which can be structured in terms of smaller units ([7, 14]). We exemplify in this paper with a model for numbers (for text), but the same principle can be applied to other entities. One direction we would like to pursue is modelling of proper names at syllable level. General-purpose language models based on sub-word units have not proved as good as the word-based ones, so we would like to retain the rich statistical information present in the frequent words, and for the infrequent ones to include components based on smaller units. This would have the effect of increasing the coverage of the overall language model with just a small size increase, and alleviating the data sparseness problem.

We would expect the extra linguistic knowledge encoded in the local models to improve the quality of the overall model. Most of the similar models report some improvements over word-based models. However, to our knowledge, none of them has been evaluated at a level of detail that would show where the improvements come from. We set ourselves to do it here for a couple of models, one that is a sub-word model, and two that are phrase models. The evaluation is carried on a large vocabulary domain (newspaper text), in contrast to the other similar models reported in the literature, which were all designed for small vocabulary applications with fairly constrained syntax. In Section 3 we detail the evaluation measures and in Section 4 we describe our experiments and the results obtained.

2. THE MODEL

The model proposed here, which we call a *hierarchical hybrid statistical model*, can be thought of as a generalization of class-based and phrase-based n-gram models. Word classes and lexicalized phrases provide rudimentary linguistic models ([14]), but they have already proved useful and are used in most state-of-the-art models. We believe richer structure would provide additional benefits. Due to lack of space, we only highlight some important aspects that we want to improve on. For further details on the model, we refer to [3].

First, the class model can assume a more sophisticated architecture than just a list of words. For example, it can be a PFSA, or even a PCFG if it is rather constrained ([5, 17]). Or else, they could be n-gram models, themselves. We call the model *hybrid* because we don't require all components of a model to be based on the same architecture, but rather that each have the most appropriate structure given the sublanguage that it tries to model and the amount of training data available for it. For practical reasons, though, we ask that all sub-models return a probability for every word or word sequence they might cover, so that the overall model remains probabilistic.

Second, as the top model includes class sub-models, these in their turn may include other sub-models, lending the overall model a *hierarchical structure*.

Third, the modelling doesn't have to be at the level of the word. There may be constructs for which better modelling can be done at sub-word level. In our model we integrate a character-based model, thus expanding the vocabulary to an infinite size. In such cases, it is impossible to gather good statistical information about the full set of words from any finite amount of training data. Also, as the test data is also going to be finite, some mechanism needs to be provided to adjust the probability model so that there is no loss of probability mass. We will explain our approach in Section 4.1.

Although the models analysed in this paper are rather simple, they serve to illustrate some of the important issues in language modelling with hybrid hierarchical statistical models:

- How to design sub-models at the sub-word level;
- In case such sub-word models have a very wide coverage (in our example it is infinite), what is an effective probability model for it; and
- How to assess the quality of each component, whether a sub-word model, or a phrase model, within the whole language model.

3. EVALUATION MEASURES

The only widely accepted measure for the quality of a statistical language model is *test-set perplexity* (*PP*) [13]. As this measure is based on averaging the probability of the words over a whole test set, it won't tell much about the gains in improving the model of just a part of the overall model. For this we will look at the perplexity values computed on a subset of the test set, that on which the probabilities are calculated using the part of the language model that is modified. Occasionally, the perplexity results may not be comparable across models (e.g., because they have different vocabularies). In such cases, it may be useful to look not at the absolute value of the perplexity, but at how much of the total perplexity is caused by that particular subset of the test set.

The *log-total probability* (*LTP*) of a subset T_1 of the test set $T=w_1..w_n$ with respect to the language model L is defined by:

$$LTP(T_1) = \sum_{\substack{t=1..n \\ w_t \in T_1}} \log P_L(w_t \mid w_1^{t-1})$$

The *fraction of LTP* caused by the subset T_1 of the test set T is defined as:

$$fLTP(T_1;T) = 100 * LTP(T_1) / LTP(T)$$

This measure was introduced in [16] for the purpose of analysing the "weaknesses" of a language model. A language model L can be improved by lowering $fLTP(T_1;T)$ for the part of the model used to compute the probabilities of the events in T_1.

As mentioned above, when the vocabularies of two models are different, we can't compare the models by means of perplexity. For such cases we will use the *adjusted perplexity* (*APP*) measure introduced in [15], which adjusts the value of *PP* by a quantity dependent on the number of unknown words in the test set, and the number of their occurrences:

$$APP(T;L) = \left(2^{LTP(T;L) - s*\log(r)}\right)^{-1/n},$$

where the test set T of length n contains s occurrences of r different unknown words.

4. EXPERIMENTAL RESULTS

We built models for newspaper text (WSJ) that combined a trigram sentence model with regular expression (RE) models for numbers (grapheme-based) and for two types of temporal phrases (word-based). These are representative for a number of classes of constructs that have an easily identifiable structure, and for which modelling either as single tokens or as mere sequences of tokens at the same level with the rest of the words in a sentence constitutes a particularly poor model. For these classes of constructs we should be able to extrapolate the statistics from a small corpus to the whole class (which may even be infinite) and not need to limit the vocabulary to just what was found in the training data. In the case of longer phrases, we want to treat them as single tokens in order to allow for meaningful longer-distance dependencies. Both number and date phrase constructs appear relatively often in the WSJ corpus, and are responsible for a comparatively large amount of both perplexity and out-of-vocabulary words (in the case of numbers).

For both numbers and date phrases, REs seem a good, intuitive model, able to represent succinctly a large, possibly infinite number of language constructs. Using such sub-models would have the advantage of both shrinking the size of the language model, and expanding its coverage far beyond the limits of the finite-state formalisms currently in use.

For our experiments, we selected for training the first 200k sentences (about 4.5M words) from the WSJ87 corpus and for testing the last 20k sentences (about 460k words) from the same corpus. The baseline model is a word trigram with a vocabulary comprising the most frequent 20,000 words. The out-of-vocabulary (OOV) rate is 4.8%. However, for the class of numbers the OOV rate is 24.5%. The OOV rate for the temporal phrases is insignificant.

The baseline model is a word-based trigram model, with Witten-Bell discounting, built with the CMU-SLM Toolkit ([2]).

For building the hierarchical model, we first tokenized the training corpus with the appropriate RE, replacing each token with a class tag. A trigram model, of the same type as the baseline model, trained on this tagged corpus provided the top-layer model. Its vocabulary was restricted to that of the baseline, plus the class tag(s). A second layer was constituted by the local PFSA model(s). We performed experiments separately with the model for decimal numbers (the **Dec** model) and with two models for temporal phrases (the **Temp** model).

4.1. Sub-Word Models: The Case of Decimal Numbers

We identified decimal numbers using the following RE:

```
((0-9)+(,[0-9][0-9][0-9])*)(.[0-9]*)?
```

for which we built an equivalent 7-state deterministic FSA. This automaton was turned into a PFSA; the probability model has six parameters, and was trained on the numbers found in the training data.

Figure 1 shows the *PP* and *APP* values for two **Dec** models, compared to the baseline model. OOV words are excluded from the computation of *PP*.

The **Dec-20k** model was trained and tested only on the same data available for the baseline model. However, the real value of this type of model is in increasing the size of the vocabulary, and thus providing additional coverage, at a minimal cost. Thus, we show results for the **Dec-20k+** model, which benefitted from the additional numbers available in the training data, and included all numbers found in the test data.

	baseline	Dec-20k		Dec-20k+	
		Type I	Type II	Type I	Type II
PP	121.87	126.99	121.52	135.4	124.15
APP	188.52	196.44	187.98	196.96	180.60

Figure 1: Overall *PP* (low) and *APP* (high) results for the **Dec** model, compared to the baseline model.

The two sets of results for the **Dec** models correspond to two modes of assigning probabilities to words by the PFSA model. The **Type I** models assign to every number the probability given by the PFSA by multiplying the probabilities along the arcs traversed in accepting that number. With such a model, a large mass of probability is assigned to an infinity of numbers that cannot be ever seen in the test set, which is finite. As a consequence, we notice that the *PP* and *APP* values are higher for these models than for the baseline.

The **Type II** models assign to all numbers an estimate of what the probability of a random number should be; the estimate is normalized on the training data. Thus, this model takes into account the fact that for a finite amount of test data only a certain proportion of words will be numbers. Note that the probability depends only on the fact that the word seen in the test data is a number, and not on which number it is; in particular, numbers not encountered in the training data will receive the same probability as the ones encountered[1]. Expectedly, the **Dec-20k** model performs just as well as the baseline (the small improvement is not significant). However, the **Dec-20k+** model, although it recognizes a larger vocabulary (practically infinite), has a 4.2% lower *APP* than the baseline model. The *PP* for this model is not comparable to the *PP* for the baseline because of the difference in the non-OOV test data and vocabularies.

Although numbers have a high OOV rate, in the corpus we worked with they account for a relatively small proportion of all words. That's why looking at overall perplexity figures is not telling. We would have to look at how much better can numbers be predicted, and also how much the new models improve the contexts that include numbers.

We did so by looking at the perplexity values for number words only. Moreover, we computed the same on the first and second words following those events, in order to assess the effect on context modelling. In all cases we also computed the fraction of total probability due to that particular subset of the test data. More formally, if the test set is $T=w_1..w_n$, the subsets of interest to us are:

- $\mathbf{T_0} = \{w_i \mid 1 \le i \le n \ \&\ w_i \text{ is a number}\}$
- $\mathbf{T_1} = \{w_i \mid 1 \le i \le n \ \&\ w_{i-1} \text{ is a number}\}$
- $\mathbf{T_2} = \{w_i \mid 1 \le i \le n \ \&\ w_{i-2} \text{ is a number}\}$
- $\mathbf{T_3} = \mathbf{T_0} \cup \mathbf{T_1} \cup \mathbf{T_2}$

The results are depicted in Table 1 and show improvements ranging from small to two orders of magnitude. The largest perplexity improvement is in predicting the numbers ($\mathbf{T}_0$), where a very large reduction in *fLTP* is also obtained. Although there

[1] This is justified intuitively by the fact that numbers are usually impossible to predict from just the word history, without world knowledge.

		baseline	Dec-20k	Dec-20k+
$\mathbf{T_0}$	*PP*	1616.6	5.96	4.83
	fLTP	2.732	0.674	0.596
$\mathbf{T_1}$	*PP*	23.24	21.73	22.36
	fLTP	1.140	1.139	1.152
$\mathbf{T_2}$	*PP*	59.16	46.04	45.80
	fLTP	1.452	1.392	1.393
$\mathbf{T_3}$	*PP*	128.56	18.25	17.15
	fLTP	5.195	3.173	3.112

Table 1: Perplexity (*PP*) and fraction of total log-probability (*fLTP*) figures for the **Type II Dec** models.

was a significant improvement in predicting the second next word after a number, there was almost no improvement in predicting the first word after a number. We believe this happened because the events in class $\mathbf{T}_1$ are already well modelled in the baseline (they have quite low PP); indeed, most words in that class have very high frequency: ",", ".", "to", etc. Overall, these results show that, indeed, although the new model has a larger vocabulary, it has better predictive power, both because of the better context modelling, and because of the probabilistic model chosen for the local model. The contribution of the class of numbers to the log-total probability was reduced by about 40%.

4.2. Phrase Models: The Case of Temporal Phrases

We built the **Temp** model using a procedure similar to the one described in the previous section. We tokenized the training data using the following two REs:

```
<month>(,? <year>)?
<month> <day> (, <year>)?
```

They define month and day phrases, respectively. `<month>` is any of the twelve months, plus their usual abbreviations, `<day>` is any number between 1 and 31, possibly with adjectival suffixes, and `<year>` is any number between 1700 and 2199.

The overall *PP* of the **Temp** model is 119.95, compared to 121.87 for the baseline model. We expect the reduction would have been much larger had we applied a probabilistic model similar to the **Type II** model for decimals, but in the current experiment we only used a uniform distribution over the class of expressions covered by the above REs.

Again, we looked at the perplexity values around the occurrences of temporal phrases in the test data, as we expect to see a marked improvement due to the longer and more meaningful context available in the HSLM. We considered the following subsets of the test set:

- $\mathbf{T_1} = \{w_i \mid 1 \le i \le n \ \&\ w_{i-1}$ is the last word in a temporal phrase$\}$
- $\mathbf{T_2} = \{w_i \mid 1 \le i \le n \ \&\ w_{i-2}$ is the last word in a temporal phrase$\}$
- $\mathbf{T_3} = \mathbf{T_1} \cup \mathbf{T_2} \cup \{w_i \mid 1 \le i \le n \ \&\ w_i$ belongs to a temporal phrase$\}$

		Month		Day		Month+ Day	
		base	**Temp**	**base**	**Temp**	**base**	**Temp**
$\mathbf{T_1}$	*PP*	99.61	25.31	32.73	10.14	62.48	17.25
	fLTP	0.160	0.113	0.088	0.059	0.248	0.172
$\mathbf{T_2}$	*PP*	82.60	57.74	49.39	31.94	66.59	45.05
	fLTP	0.154	0.142	0.098	0.088	0.252	0.230
$\mathbf{T_3}$	*PP*	220.68	35.36	109.01	17.35	154.99	26.19
	fLTP	0.593	0.369	0.517	0.215	1.110	0.584

Table 2: Perplexity (*PP*) and fraction of total log-probability (*fLTP*) figures for the **Temp** model.

The results are displayed in Table 2, separately for each type of phrase (the **Month** and **Day** columns), and together. Although the *fLTP* values seem low, in fact there are just about 900 temporal phrases in the test set; thus, the class of temporal phrases is responsible for a relatively large contribution to the overall *PP*. We again obtain very large *PP* reductions by using phrase class models, and an *fLTP* reduction of almost 50%.

Since the amount of temporal phrase data was small, we wanted to know whether the above results were reliable. We plotted the evolution of *PP* in time for each case, and found that in the cases marked in gray in Table 2 the data was insufficient for *PP* to converge, but enough to give us a reasonable estimate for the order of the relative *PP* reductions.

5. CONCLUSION

We introduced a hierarchical statistical language model that generalizes over most of the previous variations of n-gram models. We exemplified with two local models, one that describes an infinite class of words using sub-word units (graphemes), and another one that describes a large class of phrases with simple regular expressions. Both use a trigram model as the general model. We then showed that these models compare favorably with a conventional word-based trigram.

We plan to continue this work by building a model with more local models, and with more than two layers. For other types of constructs that can be easily tokenized with REs, see [6]. We would like to also try some automatic clustering techniques to define the classes to be modelled ([10, 12]). For the near future we also plan speech recognition experiments.

6. ACKNOWLEDGEMENTS

This work was supported by the ONR research grant no. N00014-95-1-1088, DARPA research grant no. F30602-98-2-0133, US. Air Force/Rome Labs research contract no. F30602-95-1-0025, and NSF research grant no. IRI-9623665.

7. REFERENCES

1. Brugnara, F., and M. Federico. “Dynamic language models for interactive speech applications”. In Proc. EUROSPEECH, pp. 2751–2754, 1997.

2. Clarkson, P., and R. Rosenfeld, "Statistical Language modelling using CMU-Cambridge Toolkit," Proc. EUROSPEECH, pp. 2707–2710, 1997.

3. Galescu, L., and J. Allen. “Hierarchical Statistical Language Models: Experiments on In-Domain Adaptation”. In these Proc., ICSLP, 2000.

4. Giachin, E.P. “Automatic training of stochastic finite-state language models for speech understanding”. In Proc. ICASSP, pp. 173–176, 1992.

5. Gillett, J. and W. Ward. “A Language Model Combining Trigrams and Stochastic Context-free Grammars”. In Proc. ICSLP, pp. 2319–2322, 1998.

6. Grefenstette, G., and P. Tapanainen. “What is a word, what is a sentence? Problems of tokenization”. In Proc. 3rd Int. Conf. Comp. Lexicography, pp. 79–87, 1994.

7. Guyon, I., and F. Pereira. “Design of a Linguistic Postprocessor using Variable Memory Length Markov Models”. In Proc. 3rd Int. Conf. Document Anal. and Recognition, pp. 454–457, 1995.

8. Meteer, M., and J.R. Rohlicek. “Statistical Language Modeling Combining N-gram and Context-free Grammars”. In Proc. ICASSP, vol. II, pp. 37–40, 1993.

9. Moore, R.C., *et al*. “Combining Linguistic and Statistical Knowledge Sources in Natural-Language Processing for ATIS”. In Proc. of the Spoken Language Systems Technology Workshop, pp. 261–264. Morgan Kaufmann, 1995.

10. Nasr, A., *et al*. “A Language Model Combining N-grams and Stochastic Finite State Automata”. In Proc. EUROSPEECH, pp. 2175–2178, 1999.

11. G. Riccardi, R. Pieraccini, and E. Bocchieri. “Stochastic Automata for Language Modeling”. *Computer Speech & Language*, 10(4):265–293, 1996.

12. Ries, K., F.D. Buø, and A. Waibel. “Class Phrase Models for Language Modeling”. In Proc. ICSLP, pp. 398–401, 1996.

13. Roucos, S. “Measuring perplexity of language models used in speech recognizers”. Technical report, BBN Laboratories, 1987.

14. Seneff, S. “The Use of Linguistic Hierarchies in Speech Understanding”. In Proc. ICSLP, pp. 3321–3330, 1998.

15. Ueberla, J. “Analyzing and Improving Statistical Language Models for Speech Recognition”. PhD thesis, Simon Fraser University, Vancouver, Canada, 1994.

16. Ueberla, J. “Analyzing Weaknesses of Language Models for Speech Recognition”. In Proc. ICASSP, pp. 205–208, 1995.

17. Wang, Y.-Y., M. Mahajan, and X. Huang. “A Unified Context-Free Grammar and N-Gram Model for Spoken Language Processing”. In Proc. ICASSP, 2000.

18. Ward, W., and S. Issar. “A Class Based Language Model For Speech Recognition”. In Proc. ICASSP, pp. 416–419, 1996.