AUGMENTING WORDS WITH LINGUISTIC INFORMATION FOR N-GRAM LANGUAGE MODELS

Lucian Galescu
Department of Computer Science
University of Rochester
http://www.cs.rochester.edu
galescu@cs.rochester.edu

Eric K. Ringger
NLP Group
Microsoft Research
http://research.microsoft.com
ringger@microsoft.com

ABSTRACT

The main goal of the present work is to explore the use of rich lexical information in language modelling. We reformulated the task of a language model from predicting the next word given its history to predicting simultaneously both the word and a tag encoding various types of lexical information. Using part-of-speech tags and syntactic/semantic feature tags obtained with a set of NLP tools developed at Microsoft Research, we obtained a reduction in perplexity compared to the baseline phrase trigram model in a set of preliminary tests performed on part of the WSJ corpus.

Keywords: speech recognition, statistical language modelling, *n*-gram models, phrase models, augmented-word models, POS tags, semantic/syntactic tags, NLPWin, WSJ corpus.

1. INTRODUCTION

Current approaches to language modelling consider words as being equivalent to their surface form. In many cases, the different meanings of lexically ambiguous words correlate with different syntactic categories and different pronunciations. A classical example is provided by the common noun/verb distinction in[1]:

OBJECT/N	/AA1 B JH EH0 K T/
OBJECT/V	/AH0 B JH EH1 K T/

Due to their different syntactic properties, such words tend to appear in different contexts, and thus statistical language models that don't make the distinction between different senses of syntactically ambiguous words tend to model them poorly.

Recently Heeman [7] presented an alternative formulation of the speech recognition problem, in which part-of speech (POS) tags are viewed as part of the output of the speech recognizer, rather than intermediate objects, as in class-based approaches [11]. The model we propose here is similar to Heeman's. We will outline the differences in section 3, where we formally define the model. In section 4 we describe experiments on the Wall Street Journal corpus with models in which words are augmented with an unprecedented rich set of syntactic and/or semantic features.

[1] The phonetic lexicon is that of the CMU pronunciation dictionary. The numbers attached to vowels indicate lexical stress.

2. BACKGROUND

Formally, the task of an automatic speech recognition system is to take an acoustic input and to derive from it the most likely string of words given the acoustic evidence. That is, it should decide in favor of the word string W^* satisfying

$$W^* = \arg\max_W P(W \mid A) = \arg\max_W P(W)P(A \mid W)$$

where A is the acoustic evidence and W is any string of words. The language model is the mechanism that computes the prior probability of any word sequence $W = w_1, w_2, \ldots, w_N = w_{1,N}$[2]:

$$P(w_{1,N}) = \prod_{i=1}^{N} P(w_i \mid w_{1,i-1})$$

Due to data sparseness, one cannot reliably estimate the probability distribution for contexts of arbitrary length. N-gram models provide the solution of restricting the contexts $w_{1,i-1}$ for predicting the next word, w_i, to the last n-1 words [11]. One can also mix in lower order models when there is not enough data to suport larger contexts, by using either interpolation [12], or a back-off approach [13,4].

Class-based models take a further step in trying to deal with the data sparseness by first grouping words into classes, and then using these classes as the basis for computing n-gram probabilities [11]. The probability of a word sequence becomes:

$$P(w_{1,N}) = \sum_{c_1,c_2,\ldots,c_N} \prod_{i=1}^{N} P(c_t \mid c_{1,i-1})P(w_i \mid c_i)$$

where c_i is the class associated with the word w_i, for all i. Classes can be determined either by automatic clustering (e.g., [3]), or they can be domain-specific semantic categories (e.g., [10]), or syntactic categories (POS) [11,15]. Although the latter approach has the advantage of capturing some linguistic information in the language

[2] We also assume the existence of a start symbol, w_0.

model, using POS classes in the above formulation has a major drawback: the POS tags remove too much of the lexical information needed for predicting the next word. POS-based language models can bring some improvement only by interpolation with word-based models [11].

Furthermore, significant improvements have been obtained by grouping words into multi-word units, commonly called *phrases*, and including them in the base vocabulary (i.e., *lexicalizing* them) [6]. The grouping may be done either manually, or automatically [6,8].

The most common measure of the effectiveness of a language model is the *test set perplexity* [1], which is an estimate of how well the language model is able to predict the next word in the test corpus, in terms of the number of alternatives that need to be considered at each point. The perplexity of a test set $W = w_{1,N}$ relative to a language model P is defined as

$$\mathrm{PP}(P; w_{1,N}) = [P(w_{1,N})]^{-1/N}$$

3. MODEL DESCRIPTION

3.1. Problem Formulation

Heeman [7] presented an alternative formulation of the speech recognition problem, in which POS tags are viewed as part of the output of the speech recognizer. The task of the speech recognizer becomes determining

$$W^*, T^* = \arg\max_{W,T} P(W,T \mid A) = \arg\max_{W,T} P(W,T)P(A \mid W,T)$$

where T is the sequence of POS tags associated with the word sequence W. The acoustic model, $P(A|W,T)$, is approximated as $P(A|W)$. The language model, $P(W,T)$, accounts for both the sequence of words and the assignment of POS tags to those words, and can be rewritten as

$$P(w_{1,N}, t_{1,N}) = \prod_{i=1}^{N} P(w_i, t_i \mid w_{1,i-1}, t_{1,i-1})$$

3.1.1. Heeman's Model

Heeman goes on to factorize this model into two probability distributions:

$$P(w_i, t_i \mid w_{1,i-1}, t_{1,i-1}) = P(w_i \mid w_{1,i-1}, t_{1,i})P(t_i \mid w_{1,i-1}, t_{1,i-1})$$

Note that this approach generalizes the class-based model: generalizes the class-based model can be thought of as making the assumptions:

$$P(w_i \mid w_{1,i-1}, t_{1,i}) = P(w_i \mid t_i)$$

$$P(t_i \mid w_{1,i-1}, t_{1,i-1}) = P(t_i \mid t_{1,i-1})$$

3.1.2. Our Model

In contrast with Heeman, we don't see the words and their tags as being produced by separate processes. Instead, we augment the words with tags encoding lexical information. We do not restrict the types of lexical information to syntactic categories only. In this formulation, we want to estimate directly

$$P(w_i, t_i \mid w_{1,i-1}, t_{1,i-1}) = P(\langle w,t\rangle_i \mid \langle w,t\rangle_{1,i-1})$$

3.2. Estimating the Probabilities

To estimate the probability distribution, we need to first annotate the training corpus. For example, syntactic categories can be obtained using one of the several POS taggers that achieve over 95% accuracy [14,2]. Using a natural language understanding system, the training corpus can also be annotated with semantic information.

As for the word-based models, a maximum likelihood estimation of the probability distribution can be obtained from the annotated corpus *n*-gram statistics [11].

3.3. Evaluation

In order to compute the word perplexity of a test set relative to the augmented-word language model, we should take care that the model is not penalized for incorrect tags. Since the probability of the word sequence is not estimated by the language model, we obtain it by summing over all possible tag assignments:

$$P(W) = \sum_T P(W,T)$$

$P(W)$ can be estimated efficiently with a variant of the *forward* algorithm [16]. In the following we present the algorithm for the case of bigrams; it should be clear how it generalizes for higher order *n*-grams. Suppose the set of available tags is $(t^k)_{k=1..M}$ and the sequence of observations is $W=w_{1,N}$. Define the *forward probability*

$$\alpha_i(k) = P(w_{1,i-1}, \langle w_i, t^k \rangle)$$

which is the probability that at step i the sequence of words $w_{1,i-1}$ has been recognized, and the ith prediction is the word/tag pair $\langle w_i,t^k\rangle$. Then,

$$P(w_{1,N}) = \sum_{k=1}^{M} \alpha_N(k)$$

As usual, the forward probability can be computed recursively,

$$\alpha_i(k) = \sum_{j=1}^{M} \alpha_{i-1}(j) P(\langle w_i, t^k \rangle \mid \langle w_{i-1}, t^j \rangle)$$

4. EXPERIMENTAL RESULTS

4.1. Experimental Setup

To demonstrate our model, we performed a series of experiments on the Wall Street Journal (WSJ) '87-89 corpus. The corpus contains about 1.6M sentences and about 34M words (markers excluded). Due to the large time requirements for repeatedly processing the corpus while various tools involved were still developed, we limited ourselves to a subset of the corpus, consisting of

the first 200,000 sentences[3], containing about 5M words. The test data comprised the last 20,000 sentences of the corpus (about 500K words). The punctuation is not verbalized. There are two pseudo-words, ⟨S⟩, and ⟨/S⟩, designating the start and the end of a sentence, respectively.

4.1.1. NLP Tools

The annotation was done fully automatically, using the Microsoft Natural Language Processing system (NLPWin), which includes a broad-coverage lexicon, morphology, and parser developed at Microsoft Research [9]. NLPWin aims to process any text and to yield linguistic analyses for use by arbitrary applications. The analysis system consists of a lexical processing layer, a syntactic processing layer, a semantic processing layer, and a discourse layer. As part of the lexical processing, word sequences may be grouped into phrases of two kinds: *factoids* and *captoids*. Factoid rules group word sequences such as times, dates, places, URLs, weights, measures, proper names, etc. Captoid rules group word sequences that can be interpreted as names of entities (titles of books, movies etc.), using first-letter capitalization as the primary cue. We used this mechanism to lexicalize phrases in order to build phrase models. Note that, as opposed to other automatic approaches to phrase detection, this mechanism doesn't group word sequences towards reducing the perplexity of the model. However, it has the advantage of using linguistically sound criteria, and it can capture even rare phrases.

NLPWin computes for every word (individual and/or lexicalized) POS tags (a slightly modified version of the Brown corpus tag-set) and *bit* tags encoding additional lexical-level syntactic and semantic features from a set of over 1000 (e.g., Mass, Count, Definite, Indefinite).

Using these tools, we obtained a lexicalized version of the corpus, in which all phrases are replaced by underscore-adjoined words (e.g., "New York City" is replaced by "New_York_City"). This is going to be our training corpus.

4.1.2 Language Models

Our baseline model is an open vocabulary back-off trigram model with Witten-Bell discounting and 0-1 cutoffs[4]. We also built three models with the same characteristics based on augmented words: one that uses only POS tags (the *POS-augmented* model), one that uses semantic and syntactic features (the *bit-augmented* model), and one that uses both POS and bit tags (the *POS&bit-augmented* model).

[3] Sentence boundaries were determined automatically and are not always correct. We didn't attempt to correct this problem, which means that some sentences are going to be ungrammatical.

[4] We used, among others, the CMU-Cambridge SLM Toolkit to build the language models. See [5] for details and terminology.

	PP	PP reduction [%]
Baseline	147.63	-
Bit-augmented	141.36	4.24
POS-augmented	140.10	5.10
POS&bit-augmented	139.54	5.47

Table 1. Perplexity results for the baseline model and for the augmented-word models. The perplexity reductions are relative.

The vocabulary of the baseline model comprises the 20,000 most frequent words in the training corpus. In order to be able to compare the augmented models to the baseline model, the vocabularies of the augmented models should be computed by including all the word/tag pairs corresponding to the words in the baseline vocabulary.

4.2. Results

The results reported here are still preliminary. One problem we encountered was that vocabularies expanded significantly when words were augmented as described in the previous section. Unfortunately, many augmented words had very low frequency in the training corpus, and often this was not an inherent property, but the result of occasional inconsistent results of the tagging procedure (e.g., inconsistent ordering of the features, features missing, etc.). As a consequence, we simplified our evaluation conditions, as described here.

The current evaluation was conducted with vocabularies having the same size, as opposed to having the same content. In all cases, we selected the vocabularies comprising the 20,000 most frequent augmented words. We will remedy this deficiency in time for the conference and will report additional results at that time.

The results are shown in *Table 1*. Note that the POS-augmented model performed slightly better than the bit-augmented model. The reason is that the tagging inconsistencies we mentioned above happened at the level of bit tags, and this reduced the quality of the statistics collected for estimating the bit-augmented model. We are aware of more potential in the semantic features, and we plan to further explore these models.

Similar results were obtained when we varied the size of the training corpus from 100,000 sentences to 300,000 sentences.

5. DISCUSSION

One challenge with considering joint word/tag events is increased data sparseness. On the 5M-word subcorpus of WSJ0 that we worked on, the number of lexicalized and augmented words is more than three times as large as the number of the bare words. Traditional smoothing procedures proved effective, though. For example,

imposing a cutoff on the rare trigrams helped reduce considerably the number of events to be modeled. This is equivalent to eliminating infrequent senses of common words. Of considerable interest are classes based on the semantic features. We found that a small number of semantic word classes (person names, dates, money, geographical names, etc) cover a lot of the training data, but many words in these categories are very infrequent. Using a class-based model for smoothing the probabilities of the word/tag models is likely to provide additional improvement. Work on this issue is in progress as is work involving interpolated models.

Since the corpus we worked on contains many agrammatical sentences that are the result of the coarse automatic procedure used for sentence segmentation, we tried to use a robust parser to filter them out. The criteria used, based on the scores of the best parses, proved to be too harsh, and less than 40% of the data passed. We evaluated the models estimated from the selected data, but found that parsability could not make up for the loss in bigram and trigram counts caused by the considerable reduction of the average sentence length in the selected sub-corpus compared to the full corpus.

6. CONCLUSION

We presented here a language model that can use rich lexical information attached to the word and included in the vocabulary. Tested on a portion of the WSJ corpus, it produced some improvement over a baseline trigram model, although the results are still preliminary. We also showed how general purpose NLP tools can be used for phrase detection and corpus annotation as a basis for building augmented-word language models.

Although the augmented-word models would benefit from better lexicalization and annotation, and also from further smoothing, at this point we have already noticed an improvement over baseline phrase trigram models. Other advantages of using the type of models advanced here are: better acoustic expectations, decoder output that is closer to understanding, and longer-distance predictive ability.

Acknowledgements

Most of the work reported here was done while the first author was a Microsoft Research intern.

7. REFERENCES

[1] L. Bahl, J. Baker, F. Jelinek and R. Mercer (1977), Perplexity — a measure of the difficulty of speech recognition tasks. *Proceedings of the Meeting of the Acoustical Society of America.*

[2] Brill, E. (1992), A Simple Rule-Based Part of Speech Tagger. *Proceedings of the 3rd Conference on Applied Natural Language Processing*, Trento, Italy, pp. 152-155.

[3] Brown P.F., V.J. Della Pietra, P.V. deSouza, J.C. Lai and R.L. Mercer (1992), Class-Based n-gram Models of Natural Language. *Computational Linguistics* **18** (4) pp. 467-479.

[4] Chen, S.F. and J.T. Goodman (1996), An Empirical Study of Smoothing Techniques for Language Modeling. *Proceedings of the 34th Meeting of the Association for Computational Linguistics (ACL '96)*, Santa Cruz, CA.

[5] Clarkson, P. and R. Rosenfeld (1997), Statistical Language Modeling Using the CMU-Cambridge Toolkit. *Proceedings Eurospeech'97*, Rhodes, Greece, pp. 2707-2710.

[6] Giachin E.P., P. Baggia and G. Micca (1994), Language modelling for spontaneous speech recognition: a bootstrap method for learning phrase bigrams. *Proceedings ICSLP '94*.

[7] Heeman, P.A. (1998), POS Tagging versus Classes in Language Modeling. *Proceedings of the 6th Workshop on Very Large Corpora*, Montreal, Canada.

[8] Heeman, P.A. and G. Damnati (1997), Deriving Phrase-based Language Models. *Proceedings IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA.

[9] Heidorn, G. (1999), Intelligent Writing Assistance. In: R. Dale, H. Moisl, & H. Somers (eds.) *A Handbook of Natural Language Processing Techniques*, Marcel Dekker. To appear.

[10] Issar, S. (1996), Estimation of Language Models for New Spoken Language Applications. *Proceedings ICSLP '96*, Philadelphia, PA, pp. 869-872.

[11] Jelinek, F. (1990), Self-Organized Language Modeling for Speech Recognition. In: A. Waibel, & K.-F. Lee (eds.), *Readings in Speech Recognition*, Morgan Kaufmann Publishers, San Mateo, CA, pp. 450-506.

[12] Jelinek, F and R. Mercer (1980), Interpolated estimation of Markov source parameters from sparse data. In: E.S. Gelsema, & L.N. Kanal (eds.), *Pattern Recognition in Practice*, North Holland Publishing Co., Amsterdam, pp. 381-397.

[13] Katz, S.M. (1987), Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Trans. Acoustics, Speech and Signal Processing*, 35(3):400-401.

[14] Kupiec, J. (1992), Robust Part-of-Speech Tagging Using a Hidden Markov Model. *Computer Speech and Language*, 6(3), pp. 225-242.

[15] Niesler, T. and P. Woodland (1996), Combination of Word-based and Category-based Language Models. *Proceedings ICSLP '96*, Philadelphia, PA, 220-223.

[16] Rabiner, L.R. (1989), A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2) pp. 257-285.