퍼지 및 지능시스템학회 논문지 2004, Vol. 14, No. 3, pp. 357-362

Enhancement of a language model using two separate corpora of distinct characteristics

Sehyeong Cho and Tae-Sun Chung

MyongJi University, Department of Computer Science
San 38-2 Yong In, KyungGi, Korea

요 약

언어 모델은 음성 인식이나 필기체 문자 인식 등에서 다음 단어를 예측함으로써 인식률을 높이게 된다. 그러나 언어 모델은 그 도메인에 따라 모두 다르며 충분한 분량의 말뭉치를 수집하는 것이 거의 불가능하다. 본 논문에서는 N그램 방식의 언어모델을 구축함에 있어서 크기가 제한적인 말뭉치의 한계를 극복하기 위하여 두개의 말뭉치, 즉 소규모의 구어체 말뭉치와 대규모의 문어체 말뭉치의 통계를 이용하는 방법을 제시한다. 이 이론을 검증하기 위하여 수십만 단어 규모의 방송용 말뭉치에 수백만 이상의 신문 말뭉치를 결합하여 방송 스크립트에 대한 퍼플렉시티를 30% 향상시킨 결과를 획득하였다.

Abstract

Language models are essential in predicting the next word in a spoken sentence, thereby enhancing the speech recognition accuracy, among other things. However, spoken language domains are too numerous, and therefore developers suffer from the lack of corpora with sufficient sizes. This paper proposes a method of combining two n-gram language models, one constructed from a very small corpus of the right domain of interest, the other constructed from a large but less adequate corpus, resulting in a significantly enhanced language model. This method is based on the observation that a small corpus from the right domain has high quality n-grams but has serious sparseness problem, while a large corpus from a different domain has more n-gram statistics but incorrectly biased. With our approach, two n-gram statistics are combined by extending the idea of Katz's backoff and therefore is called a dual-source backoff. We ran experiments with 3-gram language models constructed from newspaper corpora of several million to tens of million words together with models from smaller broadcast news corpora. The target domain was broadcast news. We obtained significant improvement (30%) by incorporating a small corpus around one thirtieth size of the newspaper corpus.

Key words : language model, speech recognition, backoff, perplexity

1. Introduction

Languages have redundancy and therefore have regularity, due partly to languages themselves and partly to regularity or predictability in the reality that is described by the language. Once you heard "in terms" you are more likely to hear "of" than "off". This is an example of linguistic regularity. Once you heard "U.S. open" you are more likely to hear "Tiger Woods" than "Pablo Picasso." This is due to regularity in reality.

Language modeling is an attempt to capture the regularities and make predictions. One use of language modeling has been automatic speech recognition. Optical character recognition and spelling correction also make use of language modeling.

Recent attempts in language modeling are mostly based on statistical approaches. This is because statistics has a solid theoretical foundation for dealing with uncertainty. It is easier to integrate information from various sources to reach a conclusion. If we see a linguistic process as a stochastic process, speech recognition can be modeled statistically by using Bayes's law as in equation 1.

$$\arg\max_s P(s \mid a) = \arg\max_s P(a \mid s)P(s) \qquad (1)$$

In equation 1, a represents acoustic signal and s represents a sentence. Unless extra-sentential information is used, a statistical language model sees $P(s)$ as $\prod P(w_i \mid w_1 w_2 ... w_{i-1})$, where s is a sequence of words $w_1^{i=1,n} ... w_n$. Further, if dependency is assumed to be local to previous $n-1$ words, that is, $\prod P(w_i \mid w_1 w_2 ... w_{i-1}) = \prod P(w_i \mid w_{i-n+1} ... w_{i-2} w_{i-1})$, then we call it an n-gram language model. We restrict our attention only to n-gram based models, approximating

접수일자 : 2003년 12월 3일
완료일자 : 2004년 3월 9일
감사의 글 : This work was supported by grant R05-2003-000-11830-0 from the Basic Research Programs of the Korea Science a Engineering Foundation.

the probability of a sentence as $\prod P(w_i \mid w_{i-n+1}...w_{i-2}w_{i-1})$. In this paper, we use 3-gram[i] language model for practical reasons.

In order to use an *n*-gram language model, it is necessary to somehow estimate the probability of the form $P(w_i \mid w_{i-n+1}w_{i-n+2}...w_{i-1})$. This is usually obtained by maximum likelihood estimation, which is simply a relative number of occurrences in a large text, or corpus. The process of collecting the occurrences (hence probabilities) is called training.

The biggest obstacle for training by a limited text is sparseness. For instance, if we had ten million words in a corpus and 10,000 words in the vocabulary, The average number of occurrences of a 3-gram is a mere 0.00001, since there are 10^{12} possible 3-gram types. This means that by simple MLE, most *n*-grams will have zero probabilities, which is certainly not correct. Therefore we need a means of estimating probabilities for zero-occurrence *n*-grams.

Smoothing, or discounting is a way of giving non-zero probabilities to *n*-grams that have zero MLE probabilities. Good-Turing[1] and Witten-Bell[2] are two examples of smoothing. They are also called *discounting* because part of the probabilities of existing *n*-grams are taken away and given to non-existent 3-grams, or *zerotons*.

The discounted probability mass can either be distributed uniformly or based on some linguistic information. With Katz's backoff method [3], we distribute the residual probability mass proportional to the *n*-1-gram probabilities. [4] has a good summary on more recent approaches to discounting.

Perplexity[5] is used as a measure to judge the quality of statistical language models. Perplexity is 2 to power of cross-entropy, where cross entropy is defined as equation 2.

$$H(L,M) = -\lim_{n\to\infty}\frac{1}{n}\sum_{x_{1..n}} P_L(x_{1..n})\log P_M(x_{1..n}) \qquad (2)$$

L represents the language and M represents the model. Perplexity is preferred to cross-entropy, because it is more intuitive. Cross entropy (and hence perplexity) will be minimized if the estimated probabilities were equal to actual probabilities of occurrences.

2. Motivation: the lack of right corpus

What motivated this research is simply the lack of right corpus. By "right corpus," we mean sentences from the same domain. Further, the corpus should be big enough. For instance, to construct a reasonable 3-gram model, several million words are generally considered barely useable, though a billion words is considered to be a saturation point[6]. Some one hundred words will probably be considered to be unacceptable.

Table 1. Perplexities of models from various corpora. Test text from broadcast news.

Training corpus Size in 100,000 words	1	2	4	8	16	32
Broadcast corpus	582	567	485	420	331	248
Newspaper corpus	1170	1150	1051	926	793	631

As we see in Table 1, the perplexity of a language model constructed from a newspaper corpus is consistently greater than that of the language model constructed from the broadcast news corpus of the same size. This means newspaper language and broadcast news language are different. Therefore, no matter how large the corpus is, you cannot break the barrier of inherent perplexity. Unfortunately, it is extremely difficult to have corpora of sufficient size for each domain, like broadcast news, travel domain, dictation, and so on and so forth.

This granted, what we need then is a way of making use of existing information to help lower the perplexity of the language model. However, simply merging two corpora will not help much, as we shall see later in the next section.

3. Related Work

Linear combination is probably the simplest way of combining two language models as shown in equation 3.

$$P_{combined}(w \mid h) = \sum_{k=1..n} \lambda_k P_k(w \mid h) \qquad (3)$$

In this equation, *h* represents history, *w* represents a word, and *n* is the number of individual language models. The sum $\sum_{k=1..n} \lambda_k$ should be equal to unity, for the sake of consistency. Further, if there are only two information sources, equation 3 is simplified as equation 4.

$$P_{combined}(w \mid h) = \lambda_1 P_1(w \mid h) + \lambda_2 P_2(w \mid h) \qquad (4)$$

Rosenfeld[7] points out that the optimal coefficients can be found by Expectation-Maximization algorithm. If the information sources are only two, determining the

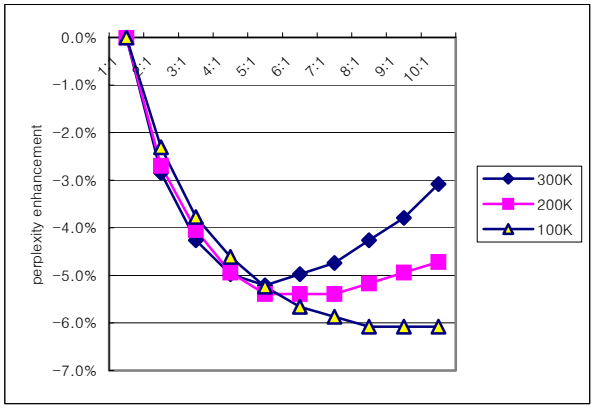

Fig. 1. Reduction in perplexity by linear interpolation. X axis is the ratio of $\lambda_1 : \lambda_2$ from 1:1 to 10. Improvements are depicted relative to 1:1 cases.

practical optimum is much easier: just trial and error will do practically.

Linear interpolation has the advantage of extreme simplicity. It is easy to implement, easy to compute. Linear combination is consistent as far as *n*-gram models are concerned. Fig.1 depicts the reduction in perplexity by linear interpolation, for varying $\lambda_1 : \lambda_2$ ratios.

Maximum entropy method[7] is another option. Maximum entropy method gives a consistent solution even when the event spaces are not the same. For instance, suppose we had an *n*-gram model probability and a trigger pair model probability: $P(bank \mid in, the)$ and $P(bank \mid loan \in history)$. When the two conditions are both satisfied, that is, the history contained the word 'loan' and previous two words were "in the", then maximum entropy method can find a solution without sacrificing the consistency, by imposing that the constraints are satisfied *on the average*. On the other hand, linearly combining the two will give out inconsistent probabilities.

However, if we had the same event space, then Maximum entropy method will result in trouble.

1. With maximum entropy method, the expectation,

$$E_{h \text{ ends in 'in the'}}[P_{combined}(bank \mid h)] = P_1(bank \mid in, the) \ .$$

2. Also, by the same token,

$$E_{h \text{ ends in 'in the'}}[P_{combined}(bank \mid h)] = P_2(bank \mid in, the)$$

Except by rare coincidence, $P_1(bank \mid in, the) \neq P_2(bank \mid in, the)$, which obviously is a contradiction. Therefore maximum entropy method is good only when we have different event spaces, but cannot be consistently used in our problem.

Akiba [8] proposed using selective backoff. Their approach is similar to ours in that they use backoff with two different models. One of the models is probabilistic model and the other is a grammar network. The aim of their combination is to delete probabilities of all unnecessary *n*-grams, that is, those that are not possible word sequences according to the simpler grammar-based transition network.

Adaptation([9], for example) is a dynamic switching of language models based on the present situation. Adaptation can further be divided into cross-domain adaptation and intra-domain adaptation. Cross-domain adaptation means switching the language model to a different one when the domain has changed. Intra-domain adaptation deals with the same domain, but even inside the same domain, topics or sub-topics may change, or speaker may change, and therefore the languages change. While adaptation focuses on dynamically detecting the shift among domains or topics, our problems deals with constructing a language model *per se* by using information from two models. We can create several models using the method proposed in this paper and in the process of speech recognition, one may change among models (i.e., adapt) depending on the current situation.

4. Combining two models

We start describing the proposed method by defining a few terms.

A *primary corpus* is a corpus from a domain of interest. A *secondary corpus* is a (relatively larger) corpus, from another domain. A *primary language model*, then, is a language model constructed from a primary corpus. A *secondary language model* is a language model constructed from a secondary corpus. C_1 is the primary corpus, and C_2 is the secondary corpus. P_1 denotes the probability obtained by maximum likelihood estimation from the primary corpus. $\overline{P}_1$ denotes a discounted primary probability. P_2 and $\overline{P}_2$ are likewise defined.

We prepared 3-gram models from corpora of various sizes. One set used broadcast news script, the other newspaper articles. The test data is from a separate text from broadcast news. It is not difficult to figure out, given the same size, the broadcast news corpora (primary corpus, hence primary language model) performed better (i.e., lower perplexity). What is interesting in the result is that given the same (or roughly the same) perplexity, the 3-gram hit ratio of the primary model is significantly lower. Conversely, with

similar 3-gram hit ratios, the secondary model has significantly higher perplexity.

The reason for lower 3-gram hit ratio is simple: the model is constructed from smaller corpus. Nevertheless, it performs better because of the quality of *n*-gram probability distribution.

Conversely, once again, the secondary model had higher 3-gram hit ratio because it was constructed from a bigger corpus, but poorer because the difference in the language made the probability estimate inadequately biased. Then what if we combined the two merits: quality *n*-gram statistics and higher hit ratio. That is the basic idea behind our approach.

Table 2. Perplexity and 3-gram his ratio(using Cambridge-CMU toolkit v.2, w/ Good-Turing discounting, range 1-7-7).

Test and training corpus from same domain (broadcast news)			Test corpus: broadcast news Training corpus: newspaper articles		
size	3-gram hit ratio	perplexity	size	3-gram hit ratio	perplexity
100K	14.2	582	100K	7.09	1170.47
200K	17.8	567.16	200K	10.06	1150.5
400K	22.67	485.1	400K	13.38	1051.69
800K	27.89	420.11	800K	17.44	926.91
1600K	34.53	331.07	1600K	23.19	793.34
3200K	42.15	248.19	3200K	29.15	631
5000K	47.47	200.96	5000K	33.48	543.27

From the observations, it follows that by using a 3-gram probability obtained from the corpus of the same domain we can obtain lower perplexity. Then what about a 2-gram primary model and a 3-gram secondary model? We observed that if the primary model and the secondary model used the same size, then the 2-gram primary model performed far better than the 3-gram secondary model.(Table 3) However, this does not mean that 3-gram probabilities in the secondary model is useless, since usually secondary model is constructed from a far bigger corpus. For instance, a secondary 2-gram model constructed from a 10,000K size corpus outperformed a primary 3-gram model from a 800K size corpus.

Table 3. Perplexity measures of 2.3-gram models, primary and secondary.

	Primary model		Secondary model	
	2-gram model	3-gram model	2-gram model	3-gram model
100 K	587.75	582	1175.66	1170.47
200 K	585.09	567.16	1171.96	1150.5
400 K	514.78	485.1	1083.05	1051.69
800 K	459.53	420.11	970.69	926.91
1600 K	386.03	331.07	838.08	793.34
3200 K	313.25	248.19	694.9	631
10,000K			526.42	433.58
20,000K			457.79	348.24

Therefore given appropriate sizes, we may be able to take advantage of *n*-gram probabilities in both models. We assumed that the secondary corpus is at least one order of magnitude larger than the primary corpus, based on the observation in Table 2 and Table 3. Then we may conclude that the relative qualities of *n*-grams are:

$$\text{3-gram(primary)} \succ \text{3-gram(secondary)} \succ \text{2-gram(primary)} \succ \text{2-gram(secondary)} \succ \text{1-gram(primary)} \succ \text{1-gram (secondary)},$$

where the $\succ$ stands for the (informal) relation "more important."

However, a straightforward solution will lead to inconsistency. In other words, the conditional probabilities do not sum up to unity (i.e., $\sum_{xyz\in C_1} P_1(z\mid x,y) + \sum_{\substack{xyz\notin C_1 \\ xyz\in C_2}} P_1(z\mid x,y) \neq 1$).

This is where Katz's idea comes into play. First, we note that the *n*-gram probabilities in the primary model is generally either overestimated (when the count is greater than zero) or underestimated (when the count is zero). Therefore we first discount the MLE probabilities of the non-zerotons. Let $\beta = 1 - \sum_{xyz\in C_1} \overline{P}_1(z\mid x,y)$. Then we redistribute the mass to zeroton 3-grams (i.e., the 3-gram *xyz*'s, such that $xyz \notin C_1$). The redistribution is not uniform, but proportional to either secondary 3-gram probability or primary 2-gram. Assuming that the secondary corpus is larger by at least one order of magnitude,

$$\overline{P}(z\mid xy) = \begin{cases} \overline{P}_1(z\mid xy) & \text{if } xyz \in C_1 \\ \alpha_{xy}\overline{P}_2(z\mid xy)\gamma_1 & \text{if } xyz \notin C_1, xyz \in C_2 \\ \alpha_{xy}\overline{P}(z\mid y)\gamma_2 & \text{otherwise} \end{cases} \quad (4)$$

γ_1 and γ_2 are coefficients that reflect the relative importance of secondary 3-gram and primary 2-gram. However, we experienced these values other than 1:1 yielded no significant improvement, and equation 4' will be used instead.

$$\overline{P}(z\mid xy) = \begin{cases} \overline{P}_1(z\mid xy) & \text{if } xyz \in C_1 \\ \alpha_{xy}\overline{P}_2(z\mid xy) & \text{if } xyz \notin C_1, xyz \in C_2 \\ \alpha_{xy}\overline{P}(z\mid y) & \text{otherwise} \end{cases} \quad (4')$$

In the above formula, α_{xy} is a normalizing constant such that $\sum_{xyz}\overline{P}(z\mid xy)=1$. Therefore

$$\alpha_{xy}=\frac{\beta}{\sum_{\substack{xyz\notin C_1\\ xyz\in C_2}}\overline{P}_2(z\mid xy)+\sum_{\substack{xyz\notin C_1\\ xyz\notin C_2}}\overline{P}(z\mid y)} \tag{5}$$

Unlike Katz's coefficients, there is no simple computation procedure for α_{xy}, and thus repeated summation is required, which took hours in a machine with two Xeon 2GHz processors. Fortunately, the calculation needs to be done only once and it need not be calculated in real-time.

The 2-gram probability $\overline{P}(z\mid y)$ is recursively defined in a similar manner.

$$\overline{P}(z\mid y)=\begin{cases}\overline{P}_1(z\mid y) & \text{if } yz\in C_1\\ \alpha_y\overline{P}_2(z\mid y)\delta_1 & \text{if } yz\notin C_1, yz\in C_2\\ \alpha_y\overline{P}(z)\delta_2 & \text{otherwise}\end{cases} \tag{6}$$

or by the same reason equation 4' replaced 4, we use equation 6.

$$\overline{P}(z\mid y)=\begin{cases}\overline{P}_1(z\mid y) & \text{if } yz\in C_1\\ \alpha_y\overline{P}_2(z\mid y) & \text{if } yz\notin C_1, yz\in C_2\\ \alpha_y\overline{P}(z) & \text{otherwise}\end{cases} \tag{6'}$$

Finally, 1-gram probability can also be defined in a similar fashion.

$$\overline{P}(z)=\begin{cases}\overline{P}_1(z) & \text{if } z\in C_1\\ \alpha_0\overline{P}_2(z) & \text{if } z\notin C_1, z\in C_2\\ \alpha_0' & \text{otherwise}\end{cases} \tag{7}$$

For practical purposes, equation 7 may be simplified to either 8 or 9.

$$\overline{P}(z)=\begin{cases}\lambda_1\overline{P}_1(z)+\lambda_2\overline{P}_2(z)\\ \alpha_0' \quad \text{otherwise}\end{cases} \tag{8}$$

$$\overline{P}(z)=\begin{cases}\overline{P}_1(z)\\ \alpha_0' \quad \text{otherwise}\end{cases} \tag{9}$$

5. Results

We used CMU-Cambridge toolkit to construct secondary models in ARPA-format from a newspaper corpus (Dong A Ilbo news) from 4 million to 8 million words. We also constructed 4 primary models from SBS broadcast news (100K to 400K words). Test corpus was a separate SBS broadcast news text of 10K size.

By simply mixing up primary and secondary models, we obtained 10 to 17 percent decrease in perplexity. With optimal mixing ratio by linear interpolation, additional 5 to 6 % decrease is seen (see Fig.1). The result of the dual-source experiment showed around 30% decrease in perplexity (see Table 4). Considering that 20% decrease in perplexity shows notable increase in the accuracy of the speech recognizer, this can be regarded a meaningful result.

Table 4. Resulting Perplexity of interpolated model and dual-source backoff model.

Mixture of primary and secondary	Linear Interpolation (1:1)	dual-source backoff
100K/4M	377	242
200K/5M	359	244
300K/6M	333	230
400K/8M	300	206

6. Conclusion and Future Work

The experiment clearly showed that there is improvement. However, it is not certain if this is indeed the optimal. As we discussed earlier the relative quality of the primary and the secondary *n*-grams depend on the corpora sizes. For instance, if the size of the primary corpus is very small compared to the secondary model, the secondary 2-gram probability may prove to be more reliable than the primary 3-gram.

Table 5. Enhancement in average log probabilities.

	Case 1	Case 2	Case 3	Case 4	Case 5	Etc.
Case 1	0	0	0	0	0	…
Case 2	0	0	-27.5	0	-118	…
Case 3	0	0	**18.5**	0	0	…
Case 4	0	0	0	0	-78.7	…
Case 5	0	0	0	0	**17.59**	…
Etc.	…	…	…	…	…	…

Table 5 shows how 3-gram log probabilities average in each case. Row and column headings represent case numbers, top heading for dual-source backoff and side heading for Katz's style original backoff. For instance, case 1 means the 3-gram exists in the primary corpus. Case 2 means the 3-gram does not appear in primary corpus but appears in secondary corpus, and so on. Therefore –27.5 in row 2 column 3 means the average log probabilities of next words (where Katz's method used 2-gram and the proposed method used 3-gram from secondary corpus) were enhanced by –27.5.

Negative numbers indicate average log probability decreased and positive numbers indicate the reverse. Even though as a whole the average decreased, in some cases it turned out to the opposite. This may indicate there are possibilities for further enhancement.

Lastly, the algorithm needs to be generalized to *n*-gram models of arbitrary n values. Theoretically, it seems possible. However, the real problem is in determining the order of applications. This is not merely a theoretical a problem, but a practical one, since it may well depend on the sizes of the corpora – relative or absolute – and also on the similarity among primary, secondary, and the test corpora.

References

[1] Good, I.J. "The Population frequencies of species and the Estimation of Population parameters," Biometrica, vol.40, parts3,4 pp.237-264

[2] Witten, I.H. and Bell, T.C. "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," in IEEE Transactions on Information Theory, vol. 37-4, pp.1085-1094. 1991

[3] Katz, S.M. "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-35, pp 400-401, March 1987

[4] Goodman, J.T. "A Bit of Progress in Language Modeling," Computer Speech and Language vol 15, pp.403-434, 2001

[5] Jelinek, F. et al, "Perplexity – A Measure of the difficulty of speech recognition tasks," Journal of the Acoustics Society of America, 62, S63. Supplement 1, 1977

[6] Jurafsky, D. and Martin, J.H., Speech and Language Processing, Prentice-Hall, 2000

[7] Rosenfeld, R. Adaptive Statistical Language Modeling: A Maximum Entropy Approach, Ph.D. dissertation, April 1994, Carnegie-Mellon University

[8] Akiba, T., Itou, K., Fujii, A. and Ishikawa, T. "Selective Backoff smoothing for incorporating grammatical constraints into the n-gram language model," in Proc. International Conference on Spoken Language Processing, pp. 881-884, Sept. 2002

[9] Chen, S. F. et al, "Topic Adaptation for Language Modeling Using Unnormalized Exponential Models," in Proc. ICASSP'98, Vol. 2, pp. 681-684, May 12-15, 1998

저 자 소 개

조세형(Sehyeong Cho)

1981년 : 서울대학교 섬유공학과 학사
1983년 : 서울대학교 계산통계학과 석사
1992년 : Pennsylvania State University 전산학 박사
1984-2000년 2월 : 한국전자통신연구원 책임연구원
2000년 3월-현재 : 명지대학교 컴퓨터소프트웨어학과 부교수

관심분야 : 언어 처리, 인공지능
Email : shcho@mju.ac.kr

정태선(Tae-Sun Chung)

1995년 2월 : KAIST 전산학과 학사
1997년 2월 : 서울대학교 전산과학과 석사
2002년 8월 : 서울대학교 전기컴퓨터공학부 박사
2002년 3월-2004년 2월 : 삼성전자 소프트웨어센터 책임연구원
2004년 3월-현재 명지대학교 컴퓨터소프트웨어학과 조교수

관심분야 : 지능형시스템, 데이터베이스, 객체지향 시스템
Email : tschung@mju.ac.kr