TR-IT-0334

Using Detailed Contextual Information To Build Language Models Of Part-Of-Speech Tagging And Language Models Of Speech Recognition By The Maximum Entropy Approach

Ruiqiang Zhang Ezra Black

Andrew Finch Yoshinori Sagisaka

February 14, 2000

This report is about us the latest results of part-of-speech tagging and language modeling of speech recognition. Detailed information including local N-gram, long distance constraints and information from sentence structure provided by ATR Parser are integrated in language models by maximum entropy approach. The experimental results prove our models are effective to improve pos tagging accuracy and reduce word error rate of ATR speech recognition system— ATRSPREC.

Contents

1 Overview

For years applying statistical methods to deal with language problems have been widely adopted in speech and language communities. Problems with regard to language eventually lead to that of building a specific language model. The well-known examples can be statistical language modeling of part-of-speech tagging and statistical language modeling of speech recognition.

Recently two new trends have appeared in language modeling research. One is to use multiple sources of information to complement local contextual information. Local contextual information such as n-gram has been used successfully for a long time. But it is not enough to use only n-gram to discriminate hypotheses in complicated tasks. People's interests have been extended to use more complement, riched information, including long distance information and higher knowledge from semantic and syntactic tag sequence and sentence structure. These information can be easily gleaned by an language expert.

The other is to apply maximum entropy to build language model. The main concern to people is how to collect information from training data and how to use it. An answer to this problem is to use mutual information rules to choose information and to use maximum entropy approach to integrate these information. MI and ME are binded together as to collecting information and building language model. Benefits of this kinds of usages have been reported in some papers [14, 18, 19, 17].

Two issues are highlighted in the present paper. One is part-of-speech tagging, that is, to assign the words in the sentence with a grammatical tag from a defined tagset. The other is to build an improved language model over n-gram model for ATR English speech recognition. The language model of tagging and the language model of speech recognition described here were both created by using the maximum entropy approach. The information sources used in our work are as follows:

(1) Local constraints/triggers

(2) Long distance POS constraints/triggers

(3) Long distance word constraints/triggers

(4) Linguistic question constraints/triggers

Triggers [14] are synonymous with constraints but used to formalize constraints. The first triggers embody the information from local context, which are the most important and used in most applications. The second and third triggers contain information from long distance context. The first three triggers have been discussed in other researchers' work [14, 12, 11]. But applying them in our work is much more complicated and extended. The last one is a new type of triggers introduced by us. Linguistic questions were written by a grammarian. These questions query information from parse structure or the tags and words to the left of predicting word. These questions are chosen to trigger the tag or word which we want to predict.

In what follows, Section 2 introduces the basic mathematical formulas. Section 3 describes the work of POS tagging, comparing the new tagger with the conventional N-gram tagger. Section 4 describes the language modeling experiments performed. I will give an account of the results in WSJ domain and hotel-reservation domain respectively. Section 5 discusses the overall research, the conclusions and future research.

2 Mathematical Fundamentals

In this section two mathematical formulas, maximum entropy and mutual information, are described because they are crucial and used throughout our research work. It is best to introduce them before proceeding.

2.1 Maximum Entropy Modeling

We consider a random process that produces an output symbol y, a member of a finite set Y. In generating y, the process may be influenced by some contextual information x. Our task is to estimate the conditional probability $p(y|x)$.

For the example of building word language model of speech recognition, y is a member of the vocabulary Y. x is all the words occurring before y in the history.

For the example of tagging, y is a member of the tagset. x is the word contexts surrounding y and tag contexts before y occurring in the tagged text.

If given a training data, we can collect a large number of samples $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, each sample (x, y) consists of a predicted output symbol y and $y's$ contextual information x.

Sometimes we want to choose some interesting 'triggers' from the training data. A trigger pair is formulated as (s, t), s is part of contexts of x, t is synonymous with y.

Suppose we select a large number of triggers $(s_1, t_1), (s_2, t_2), \ldots, (s_M, t_M)$ we define a trigger function as follows:

$$f_i(x, y) = \begin{cases} 1 & \text{if } s_i \text{ occurs in } x \text{ and } t_i \text{ is } y \\ 0 & \text{otherwise} \end{cases}$$

The above equation means the trigger function $f_{s,t}(x, y)$ is a binary-valued function. If and only if the trigger (s, t) occurs in the training sample (x, y), the value of the trigger's corresponding trigger function equals to 1.

Below is an example to explain the above concepts discussed. If given a sentence,

hello i 'd like to make a reservation for a room

In this sentence we want to estimate the probability of the last word 'room'. In this example we use one trigger pair *(reservation, room)*. The arguments listed above with regard to this sentence is:

x = *hello i 'd like to make a reservation for a*

y = *room*

s = *reservation*

t = *room*

$f(x, y)$=*1*

Now Let's enter the maximum entropy modeling.

Our task is to estimate the probability $p(y|x)$ if given training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ and triggers $(s_1, t_1), (s_2, t_2), \ldots, (s_M, t_M)$

The presence of triggers constraints the probability distributions $p(y|x)$ as follows:

$$\Sigma_{x,y} p(x, y) f_k(x, y) = \Sigma_{x,y} \tilde{p}(x, y) f_k(x, y) \tag{1}$$

where:

$$p(x, y) \approx \tilde{p}(x) p(y|x)$$

$\tilde{p}(x)$ and $\tilde{p}(x, y)$ are empirical probability distributions, defined by

$$\tilde{p}(x) = \frac{\#(x)}{N}, \tilde{p}(x, y) = \frac{\#(x, y)}{N} \tag{2}$$

$\#(.)$ means number of times that $(.)$ occurs in the sample.

Then the maximum entropy solution satisfying the constraint equations 1 and 2 is as follows:

$$p(y|x) = \frac{1}{Z(x)} exp(\Sigma_i \lambda_i f_i(x,y)) \tag{3}$$

where $Z(x)$ is a normalizing constant determined by the requirement that $\Sigma_y p(y|x) = 1$ for all y:

$$Z(x) = \Sigma_y exp(\Sigma_i \lambda_i f_i(x,y)) \tag{4}$$

If given an initial model $p_0(y|x)$, we add another constraint

$$p = argmin D(p||p_0) \tag{5}$$

where: D is the Kullback-Leibler distance.

Then the the maximum entropy solution satisfying the constraint equations 1, 2 and 5 is as follows:

$$p(y|x) = \frac{1}{Z(x)} exp(\Sigma_i \lambda_i f_i(x,y)) p_0(y|x) \tag{6}$$

Clearly if we choose the initial distribution as uniform distribution, Equation 3 is a special case of equation 6.

In equation 6 λ_i is a weight of trigger f_i. An improved iterative scaling algorithm is used to train model 6 to obtain λ_i.

For detailed discussion of maximum entropy methods, please refer to [14] [13] [1].

2.2 Mutual Information

When considering a particular trigger pair (s,t), we are interested in the correlation between s and t. We can assess the significance of the correlation between s and t by measuring their mutual information. We use the same formula as [14] to calculate mutual information.

$$\begin{aligned} MI(s,t) &= P(s,t) \log \frac{P(t|s)}{P(t)} \\ &+ P(s,\bar{t}) \log \frac{P(\bar{t}|s)}{P(\bar{t})} \\ &+ P(\bar{s},t) \log \frac{P(t|\bar{s})}{P(t)} \\ &+ P(\bar{s},\bar{t}) \log \frac{P(\bar{t}|\bar{s})}{P(\bar{t})} \end{aligned} \tag{7}$$

2.3 Perplexity and Evaluation

Perplexity is a measure of the average number of possible choices there are for a random variable. The perplexity of a random variable y given context x is defined as :

$$PP = 2^{H(Y|X)} \tag{8}$$

$H(Y|X)$ is the conditional entropy of X and Y, which is defined as:

$$H(Y|X) = \Sigma_{x,y} p(x,y) log p(y|x) \tag{9}$$

In speech recognition experiments, we use WER (word error rate) to measure recognition accuracy. It is defined as:

$$\text{WER} = \frac{\text{I+D+S}}{\text{N}}$$

where:

- I is number of insertions
- D is number of deletions
- S is number of substitions
- N is word number of answer

3 Part-of-speech Tagging

Part-of-speech tagging has been an important issue in natural language processing for many years. Many researchers have contributed to this problem by using a wide array of techniques, including: N-gram models([12]), decision trees([5]), transformations([9]) and maximum entropy approach([2]).

Now we re-consider this problem. But our research is distinguished from the previous by adopting some new features in pos tagging. The features are as follows:

(1) We use a much more detailed tagsets(semantically and syntactically). There are over 3,000 tags in ATR Tagset, far more than the rudimentary, 45-tag UPenn Tagset. The ATR English Tagset is unrestricted in its coverage, and particularly detailed and comprehensive, vis-a-vis other existing tagsets.

(2)The information we used to build the tagging model is extremely riched, vis-a-vis other taggers. In our tagger, we integrated into the tagger model local word and tag information, long history tag information and extrasentencial word and tag information conveyed by linguistic-questions, as opposed to other taggers where only one type of information of those mentioned above was used.

In what follows, subsection 3.1 introduces a conventional n-gram tagger, using it as a basis for evaluating other advanced taggers. Subsection 3.2 describes a maximum entropy tagger integrating detailed local information, that is more detailed than that used in [12] and we presented contributions of each type of triggers to tagging accuracy respectively. Subsection 3.3 describes a more advanced tagger in which we use a maximum entropy model to integrate local information and long distance tag triggers and linguistic-question triggers. Subsection 3.4 ends this chapter with conclusions.

3.1 Conventional N-gram Tagger

N-gram part of speech tagger is perhaps the most widely used of tagging algorithms. The basic idea is to maximize $p(T|W)$ given a word sequence in order to find its tag sequence. By using Bayes rule, this can be done to maximize $p(T) * p(W|T)$. $p(T)$ is the language model of tag sequence. $p(W|T)$ is the unigram model. In this experiment we use trigram to model $p(T)$. Both $p(T)$ and $p(W|T)$ were smoothed by Back-off methods [10]. We only write out the Backoff formula $p(W|T)$ due to the well-known backoff formula $p(T)$. It is of the following form.

$$p(w|t) = \begin{cases} \bar{p}(w|t) & \text{if } \bar{p}(w|t) \neq 0 \\ \beta(t)\bar{p}(w) & \text{otherwise} \end{cases} \tag{10}$$

where:

- $\bar{p}(w|t)$ and $\bar{p}(w)$ are discounting probabilities of $p(w|t)$ and $p(w)$, calculated by back-off discounting algorithm. The discount thresholds of $p(w|t)$ and $p(w)$ in present experiment were 12 and 1 respectively. A new word 'UNK' was added to the vocabulary, whose probability $\bar{p}(w)$ represents that of all the unseen words.

- $\beta(t)$ is a normalizing value to ensure $\Sigma_w p(w|t) = 1$.

We used the treebank data described in [5, 4]. It contains one million words and achieves a high degree of document variation. We separated this data into two parts:

- a set of 900,000 words, the training data, which was used to build the models
- a set of 35,000 words, the test data, which was used to test the quality of the models.

A tag and a word dictionary was built of listing all of tags and words that occur in the training data. It has 1877 tags and 41356 words including 'UNK'.

We use the beam-search method to tag a sentence. This method will be described in next subsection 3.2.

The tagging accuracy was 78.2% when using the N-gram tagger described above to tag the test data.

3.2 Maximum Entropy Tagger Using Detailed Local Context Information

In section 3.1 we built a n-gram tagger in which only information from unigram $p(w|t)$ and tag trigram was used. Of course there are other types of local context information such as local word constraints.

In this section we first listed all the types of local constraints we are interested and then built a tagger using every type of constraint by maximum entropy approach and finally integrated all these constraints into one tagger. The experimental results showed the contributions of each type of constraint to tagging accuracy and the joint tagging accuracy of the whole model.

Our tagging model is a maximum entropy(ME) model of the following form (a copied version of model 6):

$$p(t|h) = \gamma \prod_{k=0}^{K} \alpha_k^{f_k(h,t)} p_0 \tag{11}$$

where:

- t is tag we are predicting;
- h is the history (all prior words and tags) of t;
- γ is a normalization coefficient that ensures: $\Sigma_{t=0}^{L} \gamma \prod_{k=0}^{K} \alpha_k^{f_k(h,t)} p_0 = 1$;
- L is the number of tags in our tag set;
- α_k is the weight of trigger f_k;
- f_k are trigger functions and $f_k \epsilon \{0, 1\}$;
- p_0 is the default tagging model (in our case, the uniform distribution, since all of the information in the model is specified using ME constraints).

The model we use is similar to that of [2]. But the trigger types used in our experiments are richer than that of [2]. Our trigger types are shown in Table 1.

In Table 1:

- w is word whose tag we are predicting;
- t is tag we are predicting;
- t_{-1} is tag to the left of tag t;
- t_{-2} is tag to the left of tag t_{-1};
- w_{-1} is word to the left of word w;
- w_{-2} is word to the left of word w_{-1};
- w_1 is word to the right of word w;
- w_2 is word to the right of word w_1;

#	triggering word or tag	triggered tag
1	w	t
2	$w_{-2}w_{-1}w$	t
3	$w_{-1}ww_1$	t
4	ww_1ww_2	t
5	$w_{-1}w$	t
6	ww_1	t
7	t_{-1}	t
8	t_{-1}	t
9	$t_{-2}t_{-1}$	t
10	$t_{-1}w_1$	t
11	$t_{-1}ww_1$	t
12	$w_{-1}w_1$	t
13	w_{-1}	t
14	w_1	t
15	$t_{-1}w$	t
16	$t_{-2}t_{-1}w$	t
17	$w_{-2}w_{-1}$	t
18	w_1w_2	t

Table 1: Local Trigger Types

In the Table 1, we listed 18 types of triggers. These trigger types were sorted in the order of its importance to tagging intuitively. The most important triggers are at the top of the table.

In the experiments following, we used the same training data and test data as N-gram tagger in section 3.1.

Figure 1 shows the search algorithm employed. We select a beam width of $M = 5$ because choosing values higher than this yielded no significant improvement in tagging accuracy. The search algorithm is the same as that used in [2], except that we do not constrain the search by only generating tags, in the case of a known word, which have been assigned to that word in the training corpus.

The experimental results are shown in Table 2. We presented both the results of using single type trigger and of using all the trigger types together. In Table 2, "test PP" is the perplexity of the test data. "Accuracy" is the tagging accuracy.

Some conclusions could be obtained from the experiments as follows:

- Maximum entropy approach is a powerful method to integrate multiple information sources. When we combined all the triggers into the model, the results are much better than only one single type trigger is used.

- If we only use the information contained in triggers $(w,t) + (t_{-1},t) + (t_{-2}t_{-1},t)$, the results of N-gram tagger , 78.2%, is much better than the ME tagger, 76.2%. This explains why the N-gram tagger is well performed and welcomed in the tagging community. But if you want to integrate much more information, ME is a good selection. The final results we achieved are better than n-gram tagger, while it is only a small improvement.

- On average, word-trigger-tag is better than tag-trigger-tag. For example, the trigger type (w_{-1},t) is better than (t_{-1},t).

- Simple triggers like (w_{-1},t) are better than complex triggers like $(w_{-2}w_{-1}w,t)$ because

FOREACH *word* $\in$ *sentence*

FOREACH *beam* $\in \{beam_1, \cdots, beam_M\}$

Find the M highest probability tags (according to: $P(tag|h_{beam})$)

Extend *beam* using these M tags

Sort the extended beams to find M highest probability tag sequences

Set $beam_1, \cdots, beam_M$ to be these sequences

RETURN highest probability sequence $\in \{beam_1, \cdots, beam_M\}$

where:

h_{beam} is the history in *beam*.

M is the beam width.

Figure 1: The beam-search algorithm

Trigger Type	number of triggers	test PP	Accuracy(%)
(w,t)	73162	3.59	75.06
$(w,t)+(w_{-2}w_{-1}w,t)$	73162+15957	3.56	75.30
$(w,t)+(w_{-1}ww_1,t)$	73162+16667	3.54	75.90
$(w,t)+(ww_1w2,t)$	73162+16345	3.54	75.60
$(w,t)+(w_{-1}w,t)$	73162+14708	3.51	76.12
$(w,t)+(ww_1,t)$	73162+15789	3.47	76.52
$(w,t)+(t_{-1},t)$	73162+18520	3.15	76.14
$(w,t)+(t_{-1},t)+(t_{-2}t_{-1},t)$	**73162+18520+15660**	**3.11**	**76.24**
$(w,t)+(t_{-1}w_1,t)$	73162+12302	3.40	76.26
$(w,t)+(t_{-1}ww_1,t)$	73162+21564	3.51	76.12
$(w,t)+(w_{-1}w_1,t)$	73162+12496	3.47	76.14
$(w,t)+(w_{-1},t)$	73162+28415	3.33	76.90
$(w,t)+(w_1,t)$	73162+27380	3.34	76.78
$(w,t)+(t_{-1}w,t)$	73162+14212	3.44	75.78
$(w,t)+(t_{-2}t_{-1}w,t)$	73162+18699	3.47	75.40
$(w,t)+(w_{-2}w_{-1},t)$	73162+9811	3.53	75.92
$(w,t)+(w_1w_2,t)$	73162+9733	3.52	76.01
ALL		**3.07**	**78.80**

Table 2: Experimental Results of Tagging Using Detailed Local Constraints

#	Triggering Tag	Triggered Tag	I.e. Words Like:	Trigger Words Like:
1	NP1LOCNM	NP1STATENM	Hill, County, Bay	Utah, Maine, Alaska
2	JJSYSTEM	NP1ORG	national, federal	Party, Council
3	VVDINCHOATIVE	VVDPROCESSIVE	caused, died, made	began, happened
4	IIDESPITE	CFYET	despite	yet (conjunction)
5	DD	PPHO2	any, some, certain	them
6	PN1PERSON	LEBUT22	everyone, one	(not) only, (not) just
7	...	MPRICE	..., ..., ...	$452,983,000, $10,000
8	IIATSTANDIN	MPHONE22	at (sent.–final)	913-3434
9	IIFROMSTANDIN	MZIP	from (sent.–final)	22314-1698 (zip)
10	NNUNUM	NN1MONEY	25%, 12", 9.4m3	profit, price, cost

Table 3: Selected Tag Trigger–Pairs, ATR General–English Treebank

the influence of spareness affects complex triggers more than simple triggers, i.e., simply triggers occurring in the training data also easily occur in the test data.

- Number of triggers is related to the accuracy. The results of using more triggers are better than that of using less triggers.

3.3 Maximum Entropy Tagger Using Extrasentential Context

In last section 3.2 we discussed the effectiveness of local context constraints to part-of-speech tagging. In this section we will consider the information from long distance context and extrasentential context for part-of-speech tagging.

(3.3.1) Introduction

It appears intuitively that information from earlier sentences in a document ought to help reduce uncertainty as to a word's correct part–of–speech tag. This is especially so for a large semantic and syntactic tagset such as the roughly–3000–tag ATR General English Tagset [4, 5]. And in fact, [6] demonstrate a significant "tag trigger–pair" effect. That is, given that certain "triggering" tags have already occurred in a document, the probability of occurrence of specific "triggered" tags is raised significantly—with respect to the unigram tag probability model. Table 3, taken from [6], provides examples of the tag trigger–pair effect.

Yet, it is one thing to show that extrasentential context yields a gain in information with respect to a unigram tag probability model. But it is another thing to demonstrate that extrasentential context supports an improvement in perplexity vis–a–vis a part–of–speech tagging model which employs state–of–the–art techniques: such as, for instance, the tagging model of a maximum entropy tag–n–gram–based tagger.

The work of this section undertakes just such a demonstration. Both the model underlying a standard tag-n-gram-based tagger, and the same model augmented with extrasentential contextual information, are trained on the 850,000–word ATR General English Treebank [4], and then tested on the accompanying 53,000–word test treebank. Performance differences are measured, with the result that semantic information from previous sentences within a document is shown to help significantly in improving the perplexity of tagging with the indicated tagset.

In what follows, first, we provides a basic overview of the tagging approach used (a maximum entropy tagging model employing constraints equivalent to those of the standard hidden Markov model). Second, we discusses and offers examples of the sorts of extrasententially–based semantic constraints that were added to the basic tagging model. Then, we describes

the experiments we performed. And then again, we details our experimental results. Finally we glances at projected future research, and concludes.

(3.3.2) Tagging Model

(1)ME Model

The model used in this section is the same as model 11.But in this experiment the baseline model shares the following features with this tagging model; we will call this set of features the basic n-gram tagger constraints:

1. $w = X \ \& \ t = T$
2. $t_{-1} = X \ \& \ t = T$
3. $t_{-2}t_{-1} = XY \ \& \ t = T$

Our model exploits the same kind of tag–n–gram information that forms the core of many successful tagging models, for example, [12], [2]. We refer to this type of tagger as a tag–n–gram tagger.

(2)Trigger selection

We use mutual information(equation 7) to select the most useful trigger pairs.

where:

- t is the tag we are predicting;
- s can be any kind of triggering feature.

For each of our trigger predictors, s is defined below:

Bigram and trigram triggers : s is the presence of a particular tag as the first tag in the bigram pair, or the presence of two particular tags (in a particular order) as the first two tags of a trigram triple. In this case, t is the presence of a particular tag in the final position in the n-gram.

Extrasentential tag triggers : s is the presence of a particular tag in the extrasentential history.

Question triggers : s is the boolean answer to a question.

(3.3.3) The Constraints

To understand what extrasentential semantic constraints were added to the base tagging model in the current experiments, one needs some familiarity with the ATR General English Tagset. For detailed presentations, see [5, 4]. An apercu can be gained, however, from Figure 1, which shows two sample sentences from the ATR Treebank (and originally from a Chinese take–out food flier), tagged with respect to the ATR General English Tagset. Each verb, noun, adjective and adverb in the ATR tagset includes a semantic label, chosen from 42 noun/adjective/adverb categories and 29 verb/verbal categories, some overlap existing between these category sets. Proper nouns, plus certain adjectives and certain numerical expressions, are further categorized via an additional 35 "proper–noun" categories. These semantic categories are intended for any "Standard–American–English" text, in any domain. Sample categories include: "physical.attribute" (nouns/adjectives/adverbs), "alter" (verbs/verbals), "interpersonal.act" (nouns/adjectives/adverbs/verbs/verbals), "orgname" (proper nouns), and "zipcode" (numericals). They were developed by the ATR grammarian and then proven and refined via day–in–day–out tagging for six months at ATR by two human "treebankers",

```
(_( Please_RRCONCESSIVE Mention_VVIVERBAL-ACT this_DD1 coupon_NN1DOCUMENT
when_CSWHEN ordering_VVGINTER-ACT

OR_CCOR ONE_MC1WORD FREE_JJMONEY FANTAIL_NN1ANIMAL SHRIMPS_NN1FOOD
```

Figure 2: Two ATR Treebank Sentences from Chinese Take–Out Food Flier (Tagged Only – i.e. Parses Not Displayed)

then via four months of tagset–testing–only work at Lancaster University (UK) by five treebankers, with daily interactions among treebankers, and between the treebankers and the ATR grammarian. The semantic categorization is, of course, in addition to an extensive syntactic classification, involving some 165 basic syntactic tags.

Starting with a basic tag–n–gram tagger trained to tag raw text with respect to the ATR General English Tagset, then, we added constraints defined in terms of "tag families". A tag family is the set of all tags sharing a given semantic category. For instance, the tag family "MONEY" contains common nouns, proper nouns, adjectives, and adverbs, the semantic component of whose tags within the ATR General English Tagset, is "money": 500–stock, Deposit, TOLL–FREE, inexpensively, etc.

One class of constraints consisted of the presence, within the 6 sentences (from the same document)[1] preceding the current sentence, of one or more instances of a given tag family. This type of constraint came in two varieties: either including, or excluding, the words within the sentence of the word being tagged. Where these intrasentential words were included, they consisted of the set of words preceding the word being tagged, within its sentence.

A second class of constraints added to the requirements of the first class the representation, within the past 6 sentences, of related tag families. Boolean combinations of such events defined this group of constraints. An example is as follows: (a) an instance either of the tag family "person" or of the tag family "personal attribute"(or both) occurs within the 6 sentences preceding the current one; or else (b) an instance of the tag family "person" occurs in the current sentence, to the left of the word being tagged; or, finally, both (a) and (b) occur.

A third class of constraints had to do with the specific word being tagged. In particular, the word being classified is required to belong to a set of words which have been tagged at least once, in the training treebank, with some tag from a particular tag family; and which, further, always shared the same basic syntax in the training data. For instance, consider the words "currency" and "options". Not only have they both been tagged at least once in the training set with some member of the tag family "MONEY" (as well, it happens, as with tags from other tag families); but in addition they both occur in the training set only as nouns. Therefore these two words would occur on a list named "MONEY nouns", and when an instance of either of these words is being tagged, the constraint "MONEY nouns" is satisfied.

A fourth and final class of constraints combines the first or the second class, above, with the third class. E.g. it is both the case that some avatar of the tag family "MONEY" has occurred within the last 6 sentences to the left; and that the word being tagged satisfies the constraint "MONEY nouns". The advantage of this sort of composite constraint is that it is focused, and likely to be helpful when it does occur. The disadvantage is that it is unlikely to occur extremely often. On the other hand, constraints of the first, second, and third classes, above, are more likely to occur, but less focused and therefore less obviously helpful.

[1] [6] determined a 6-sentence window to be optimal for this task.

(3.3.4) The Four Models

To evaluate the utility of long-range semantic context we performed four separate experiments. All of the models in the experiments include the basic ME tag–n–gram tagger constraints listed in section (3.3.2). The models used in our experiments are as follows:

(1) The first model is a model consisting ONLY of these basic ME tag–n–gram tagger constraints. This model represents the baseline model.

(2) The second model consists of the baseline model together with constraints representing extrasentential tag triggers. This experiment measures the effect of employing the triggers specified in [6] —i.e. the presence (or absence) in the previous 6 sentences of each tag in the tagset, in turn— to assist a real tagger, as opposed to simply measuring their mutual information. In other words, we are measuring the contribution of this long-range information over and above a model which uses local tag–n–grams as context, rather than measuring the gain over a naive model which does not take context into account, as was the case with the mutual information experiments in [6].

(3) The third model consists of the baseline model together with the four classes of more sophisticated question-based triggers defined in the previous section.

(4) The fourth model consists of the baseline model together with both the long-range tag trigger constraints and the question-based trigger constraints.

We chose the model underlying a standard tag–n–gram tagger as the baseline because it represents a respectable tagging model which most readers will be familiar with. The ME framework was used to build the models since it provides a principled manner in which to integrate the diverse sources of information needed for these experiments.

(3.3.5) Experimental Procedure

The performance of each the tagging models is measured on a 53,000–word test treebank hand-labelled to an accuracy of over 97% [4, 5]. We measure the model performance in terms of the perplexity of the tag being predicted. This measurement gives an indication of how useful the features we supply could be to an n–gram tagger when it consults its model to obtain a probability distribution over the tagset for a particular word. Since our intention is to gauge the usefulness of long-range context, we measure the performance improvement with respect to correctly (very accurately) labeled context. We chose to do this to isolate the effect of the correct markup of the history on tagging performance (i.e. to measure the performance gain in the absence of noise from the tagging process itself). Earlier experiments using predicted tags in the history showed that at current levels of tagging accuracy for this tagset, these predicted tags yielded very little benefit to a tagging model. However, removing the noise from these tags showed clearly that improvement was possible from this information. As a consequence, we chose to investigate in the absence of noise, so that we could see the utility of exploiting the history when labelled with syntactic/semantic tags.

The resulting measure is an idealization of a component of a real tagging process, and is a measure of the usefulness of knowing the tags in the history. In order to make the comparisons between models fair, we use correctly-labelled history in the n-gram components of our models as well as for the long-range triggers. As a consequence of this, no search is nescessary.

The number of possible triggers is obviously very large and needs to be limited for reasons of practicability. The number of triggers used for these experiments is shown in Table 4. Using these limits we were able to build each model in around one week on a 600MHz DEC-alpha. The constraints were selected by mutual information. Thus, as an example, the 82425 question trigger constraints shown in Table 4 represent the 82425 question trigger constraints with the highest mutual information.

Description	Number
Tag set size	1837
Word vocabulary size	38138
Bigram trigger number	18520
Trigram trigger number	15660
Long history trigger number	15751
Question trigger number	82425

Table 4: Vocabulary sizes and number of triggers used

#	Question Description	MI (bits)
1	Person or personal attribute word in full history	0.024410
2	Word being tagged has taken NN1PERSON in training set	0.024355
3	Person or personal attribute word in remote history	0.024294
4	Person or personal attribute or other related tags in full history	0.020777
5	Person or personal attribute or other related tags in remote history	0.020156

Table 5: The 5 triggers for tag NN1PERSON with the highest MI

The improved iterative scaling technique [13] was used to train the parameters in the ME model.

(3.3.6) The Results

Table 6 shows the perplexity of each of the four models on the testset.

The maximum entropy framework adopted for these experiments virtually guarantees that models which utilize more information will perform as well as or better than models which do not include this extra information. Therefore, it comes as no surprise that all models improve upon the baseline model, since every model effectively includes the baseline model as a component.

However, despite promising results when measuring mutual information gain [6], the baseline model combined only with extrasentential tag triggers reduced perplexity by just a modest 7.6% . The explanation for this is that the information these triggers provide is already present to some degree in the n–grams of the tagger and is therefore redundant.

In spite of this, when long-range information is captured using more sophisticated, linguistically meaningful questions generated by an expert grammarian (as in experiment 3), the perplexity reduction is a more substantial 19.4%. The explanation for this lies in the fact that these question-based triggers are much more specific. The simple tag-based triggers will

#	Model	Perplexity	Perplexity Reduction
1	Baseline n-gram model	2.99	0.0%
2	Baseline + long-range tag triggers	2.76	7.6%
3	Baseline + question-based triggers	2.41	19.4%
4	Baseline + all triggers	2.35	21.4%

Table 6: Perplexity of the four models

be active much more frequently and often inappropriately. The more sophisticated question-based triggers are less of a blunt instrument. As an example, constraints from the fourth class (described in the constraints section of this paper) are likely to only be active for words able to take the particular tag the constraint was designed to apply to. In effect, tuning the ME constraints has recovered much ground lost to the n–grams in the model.

The final experiment shows that using all the triggers reduces perplexity by 21.4%. This is a modest improvement over the results obtained in experiment 3. This suggests that even though this long-range trigger information is less useful, it is still providing some additional information to the more sophisticated question–based triggers.

Table 5 shows the five constraints with the highest mutual information for the tag NN1PERSON (singular common noun of person, e.g. lawyer, friend, niece). All five of these constraints happen to fall within the twenty-five constraints of any type with the highest mutual information with their predicted tags. Within Table 5, "full history" refers to the previous 6 sentences as well as the previous words in the current sentence, while "remote history" indicates only the previous 6 sentences. A "person word" is any word in the tag family "person", hence adjectives, adverbs, and both common and proper nouns of person. Similarly, a "personal attribute word" is any word in the tag family "personal attribute", e.g. left–wing, liberty, courageously.

(3.3.7) Conclusion

Our main concern has been to show that extrasentential information can provide significant assistance to a real tagger. There has been almost no research done in this area, possibly due to the fact that, for small syntax–only tagsets, very accurate performance can be obtained labelling the Wall Street Journal corpus using only local context. In the experiments presented, we have used a much more detailed, semantic and syntactic tagset, on which the performance is much lower. Extrasentential semantic information is needed to disambiguate these tags. We have observed that the simple approach of only using the occurrence of tags in the history as features did not significantly improve performance. However, when more sophisticated questions are employed to mine this long-range contextual information, a more significant contribution to performance is made. This motivates further research toward finding more predictive features. Clearly, the work here has only scratched the surface in terms of the kinds of questions that it is possible to ask of the history. The maximum entropy approach that we have adopted is extremely accommodating in this respect. It is possible to go much further in the direction of querying the historical tag structure. For example, we can, in effect, exploit grammatical relations within previous sentences with an eye to predicting the tags of similarly related words in the current sentence. It is also possible to go even further and exploit the structure of full parses in the history.

3.4 Conclusions of Language Models of Tagging

In this section information from local context, long distance context and linguistic questions were applied and hierarchically implemented in tagging models from N-gram tagger of section 3.1 to the ME model using detailed local triggers of section 3.2 to the combined model using question triggers of section 3.3. With regard to the tagging accuracy, the model of section 3.2 and the model of 3.3 are higher than the N-gram tagger, nevertheless, the N-gram model has been welcomed due to its simplicity and effectiveness. The approaches proposed here are proved effective and a bigger space of research is left to achieve significant results in the future via finding better predictive triggers.

4 Language Modeling of ATRSPREC

In section 3 we applied ME to build a tag model on part-of-speech tagging. In this section we will use ME to build a word language model to estimate the probability of a recognized word.

4.1 Introduction

It appears intuitively that information from earlier sentences in a document ought to help reduce uncertainty as to the identity of the next word at a given point in the document. [14] and [11] demonstrate a significant "word/word trigger–pair" effect. That is, given that certain "triggering" words have already occurred in a document, the probability of occurrence of specific "triggered" words is raised significantly.

The work reported here undertakes to demonstrate that semantic/syntactic part–of–speech tags, and parse structure of *previous* sentences of the document being processed, can add trigger information to a standard n–gram language model, over and above the improvement delivered by word/word triggering along the lines of the work by Rosenfeld and Lau et al.[2] We formulate "linguistic–question" triggers which query either: (a) the tags of the words to the left of, and in the same sentence as, the word being predicted; or (b) parse structure and/or tags within any or all of the previous sentences of the document to which the word belongs that is being predicted; or both of (a) and (b) together. Each of these questions then triggers a particular word in the vocabulary, i.e. raises the probability of that word's being the next word of the document.

In section 4.3 of what follows, we use a 181,000–word subset of the approximately–1–million–word ATR General English Treebank [4]. This treebank subset consists exclusively of text drawn from Associated Press newswire and Wall Street Journal articles. The 181,000 words are partitioned into a training set of 167,000 words and a test set of 14,000 words. We utilize this portion only of the treebank, as opposed to the entire corpus, in order to match the text type of the raw data set used to train our baseline n–gram language model, which is AP and WSJ text in roughly the same proportions as in our treebank, and of course not including any portion of our training or test text.

We train (i) a baseline 200–million–word n–gram language model; (ii) a model combining this baseline plus a word/word trigger model trained on a 10–million–word subset of the larger training corpus; and finally (iii) a model combining both (i) and (ii) with linguistic–question triggers trained as just indicated. Performance differences of (i/ii/iii) are measured, with the result that model (iii) is shown to yield a significant perplexity reduction vis-a-vis models (i) and (ii).

In section 4.4 of what follows, we extends the work of section 2. The model was realized in a speech recognition system and word error rate (WER) was used to measure model performance. But the application domain was the hotel reservation task (HRT), differing from WSJ domain.

We used a linguistic-question-trigger-and-word-trigger-based language model to rescore the N-best candidates output by the ATRSPREC speech recognizer. The questions were asked of a history labelled using a trigram syntactic and semantic tagger.

4.2 ME Model and MI Trigger Selection

Our language model is a maximum entropy (ME) model of the following form:

[2] [8] explore the problem of utilizing the parse structure of the sentence in which the word to be predicted occurs. The current work can be viewed as complementary to the line of research of Chelba and Jelinek, in that we ignore, to a fair extent, the syntactic structure of the sentence in which the word occurs that is being predicted, and we focus instead on the syntactic and semantic information contained in the sentences prior to the one featuring the word being predicted.

$$P(w|h) = \gamma \prod_{k=0}^{K} \alpha_k^{f_k(h,w)} P_b(w|h) \tag{12}$$

where:

- w is the word we are predicting;
- h is the history of w;
- γ is a normalization coefficient;
- K is the number of triggers;
- $\alpha_k (k = 0, 1, \cdots, K)$ is the weight of trigger f_k;
- $f_k (i = 0, 1, \cdots, K)$ are trigger functions. $f_k \in \{0, 1\}$;
- $P_b(w|h)$ is the base language model.

In our experiments we use as base language models both a conventional trigram model and the extension of this model with long history word triggers.

The linguistic–question information is embodied in our model in the form of "triggers". A trigger pair $qw = (q, w)$ consists of a triggering question q together with a triggered word w. The number of possible triggers is the product of the number of questions with the number of words in the vocabulary. This gives rise to too many features from which to build an ME model in a reasonable time. We therefore select only those trigger pairs which can be expected to provide the most benefit to the model. We use mutual information (MI) to select the most useful trigger pairs.

4.3 WSJ Experiments

(4.3.1) Linguistic Information

The experiments reported here consist in adding "linguistic–question constraints"[3] to a baseline n–gram language model. To understand the linguistic questions used, one needs some familiarity with the ATR General English Treebank and the the ATR General English Grammar and Tagset. For detailed presentations, see [5, 3, 4]. You can also look back to section (3.3.3) for a simpler explain about ATR Grammar.

The ATR English Grammar is unrestricted in its coverage, and particularly detailed and comprehensive, vis-a-vis other existing grammars. For instance, complete syntactic and semantic analysis is performed on all nominal compounds. Further, the full range of attachment sites is available within the Grammar for sentential and phrasal modifiers, so that differences in meaning can be accurately reflected in parses. Again, see the above–cited references for details.

One can get a feel for the type of linguistic–question triggers we defined via Tables 1 and 2, which show four triggers with high mutual information with the word "Mrs.", and four for the word "added". The trigger with the highest mutual information with the word "Mrs." among all linguistic–question triggers does not ask either about tags or parse structure, but simply makes good use, over raw text, of our "Question Language", the flexible language for formulating grammar–based and lexically–based questions about Treebank text, which we normally use to compose contextual questions about text which we are parsing with our probabilistic parser.[4] Specifically, the question, defined over raw text, determines whether any reference has been made to a female, within the last 12 sentences of the current document.

[3] as well as "word/word triggers"

[4] For details, see [3].

#	Question Description	MI (bits)
1	Any reference to a female within the last 12 sents of doc	0.001210
2	Many nouns, adj or adv of verbal action (e.g. statement) within last 100 sents	0.000803
3	Many nouns, adj or adv of helping (e.g. assistance) within last 100 sents	0.000737
4	Many occurrences of female first names within last 100 sents	0.000586

Table 7: Selected triggers from top–20–highest–MI linguistic–question triggers for the word "Mrs."

#	Question Description	MI (bits)
1	Any subject pronoun to the immediate left	0.000579
2	Subject of current sentence is a person and verb is likely	0.000407
3	Many recent sents had person subjects and "saying" main verbs AND Subject of current sentence is a person and verb is likely	0.000314
4	Many verbs of saying in last 3 sentences of document	0.000204

Table 8: Selected triggers from top–20–highest–MI linguistic–question triggers for the word "added"

A question which asks about tags is the second question of Table 1. It queries the semantic portion of tags within the entire history of the document, and determines whether tags have frequently occurred which label nouns, adjectives or adverbs of saying, writing, objecting, or other verbal activities. A "yes" answer to this question turns out to raise the probability of the word "Mrs." as the next word of a document.

Finally, a question which queries the complex parse structure of previous sentences of the document, is the third question of Table 2. The question tests whether frequently in the history of the document, sentences occurred with a human subject and a main verb of verbal activity, e.g. "Mr. Smith stated..." In addition, it tests the current sentence to see whether a human subject has just been received, and a verb now appears to be likely to occur. The expectation, thus, is that a verb of saying will now occur. This expectation turns out to be realized for the verb "added", as there is a relatively high correlation between a "yes" for this question and the occurrence of the word "added".

(4.3.2) Experimental Procedure

We used the well–known trigram LM as the base LM for our experiments. This model was selected because it represents a respectable language model which most readers will be familiar with. The ME framework was used to build the derivative models since it provides a principled manner in which to integrate the diverse sources of information needed for these experiments.

In all models built for these experiments we use a word vocabulary of 20001 (the 20000 most frequent words plus a token for words not in the vocabulary). We used a corpus of newspaper text drawn from 1987–1996 Wall Street Journal and Associated Press Newswire in equal proportion. Certain types of words were mapped to generic tokens representing the class of word. These were: words representing time of day (e.g. 12:21), dates (e.g. 11/02/64), price expressions (e.g. $100) and year expressions (e.g. 1970–1999). The mapping was done using simple regular-expression pattern matching. The substitutions were implemented to assist the trigram model, which is unable to ask questions about the internal structure of words and cannot be expected to form useful n-grams from this class of words. The linguistic questions, however, being able to query the word's internal structure, were more effective

Model	Tri20M.k1	Tri20M.k2	Tri20M.k4
unigram	20001	20001	20001
bigram	930091	623054	395663
trigram	2055346	1064808	527782

Table 9: Model size varying cutoff

Model	Tri20M.k1	Tri20M.k2	Tri20M.k4
Base	133.3	140.9	153
Base + Q's	126	132.6	142.7
Change(%)	5.5	5.9	6.7

Table 10: PP varying cutoff

on the raw words themselves and were used in that way. The vocabulary, and therefore the words being predicted, was constructed from data in which these tokens had been mapped.

The training set used to train the linguistic question–based triggers for all experiments was approximately 167,000 words of hand–labelled and –parsed ATR treebank, drawn from Wall Street Journal and Associated Press texts. The test set consisted of 14,000 words of hand–labelled and –parsed ATR treebank, again drawn in the same proportion from Wall Street Journal and Associated Press.

For the main experiments a large question set of approximately 6,000 questions was used to generate trigger candidates. In addition, a large 200 million word corpus was used to train the baseline LM. For the sake of speed, we reduced the LM training set to 20 million words and/or the number of questions to 300, for the experiments designed to investigate the effect of cutoff, dataset size and number of triggers used.

We measure the test set perplexity (PP) to gauge the quality of the models produced.

(4.3.3) Effect of Cutoff

A trigram model was trained on 20 million words of data. N-grams which occurred fewer then a fixed cutoff value were excluded from the model. Tables 9 and 10 show the effect of varying the cutoff value. We used cutoff values of 1, 2 and 4. In Table 10, the number of question–based triggers used was 33,000 and the question set size from which the triggers were produced was 300.

In Table 10, "Base" is the perplexity of the base trigram model before any ME training. "Base + Q's" is the perplexity of the full ME model after training. "Change" is the perplexity reduction resulting from using our question triggers.

(4.3.4) Effect of Dataset Size

In this experiment we used base trigram models of three differing sizes. The three models: Tri20M.k4, Tri100M.k4 and Tri200M.k8 were built from 20M, 100M and 200M words of training data, respectively. Table 11 shows the number of n–grams we used in our models. Table 12 shows the reduction in perplexity. We used 33000 question–based triggers and the question set size from which the triggers were produced was 300.

Notice that increasing the quality of the underlying trigram LM has little effect on the change in perplexity resulting from adding the information from linguistic questions. This indicates that the additional information will be useful to any trigram LM and that simply

Model	Tri20M.k4	Tri100M.k4	Tri200M.k8
unigram	20001	20001	20001
bigram	395663	1230040	1204727
trigram	527782	2724346	2492309

Table 11: Trigram model size varying dataset size

Model	Base PP	Base+Q's	Change(%)
Tri20M.k4	153.0	142.7	6.7
Tri100M.k4	117.8	110.0	6.6
Tri200M.k8	108.0	101.0	6.5

Table 12: Effect of varying dataset size

improving the LM by adding more data is no substitute for this information.

(4.3.5) Effect of Adding Word Triggers

In this experiment we measure the effect of using long–range word triggers on our corpus together with the effect of combining these with our question–based triggers. 39367 long history word triggers are chosen by mutual information from 200 million words of data. Due to the prohibitively long training times needed to train models using word triggers we restricted the training set for the ME training to 10M words. The base language model was trained on the full 200M word corpus. We then used the ME model built by adding word–triggers to the base model as the base model for a second ME model which incorporated our question–based triggers. We found this approach effective in dealing with the large number of triggers involved. The number of question–based triggers used was 110,000 and the question set size from which the triggers were produced was 6,000. The results are shown in Table 13.

(4.3.6) Effect of the Number of Triggers

In this experiment we measure the effect of varying the number of question triggers used by the ME model. We used a 20M word trigram language model together with either 17,700 or 33,000 question–based triggers. The question set size from which the triggers were produced was 300. The results are shown in Table 14.

Model	PP	Change (%)
Base (Tri200M.k8)	108.0	–
Base + WTModel	94.4	12.6
Base + Q's	95.8	11.3
Base + WTModel + Q's	84.6	21.7

Table 13: The effect of combining the models

Model	PP	Change (%)
Base (Tri20M.k4)	153	–
Base + 17700	144	5.88
Base + 33000	143	6.70

Table 14: Varying the number of triggers used

(4.3.7) Discussion

The maximum entropy framework adopted for these experiments virtually guarantees that models which utilize more information will perform as well as or better than models which do not include this extra information. Therefore, it comes as no surprise that all models improve upon the baseline model, since every model effectively includes the baseline model as a component. The experiments presented here have focused on showing that that we can glean useful information from the parse structure and part–of–speech tags in the history of the word being predicted. Our main result is that this information is useful, and is of similar magnitude to that provided by the long–range word triggers used by [14]. Moreover, when these triggers are used in conjunction with a model incorporating long–range word triggers, almost all of the perplexity gain is inherited by the new model. This indicates that the information we are providing is largely new and complementary. This is in line with our intuition, given the nature of the questions we ask. Furthermore, we obtained this gain from a very small 167,000 word training corpus (as opposed to the 10 million word corpus used to train the long–range word triggers). It is reasonable to expect significant improvement on domains where more data is available to train from.

4.4 Hotel Reservation Experiments

(4.4.1) Introduction

In section 4.3 we demonstrated that our language model, integrated information from linguistic-question triggers, was able to reduce perplexity 90% of the gained by using long history word triggers. But the improvement over perplexity could not demonstrate the reduction of word error rate of a real speech recognition system. In this section we will demonstrate this point by realizing our model in a real speech recognition system—ATRSPREC and use word error rate (WER) to measure our model. But the application domain was the hotel reservation task (HRT), differing from WSJ domain.

The questions used in the experiments are divided into two classes. Questions of the first class query the words and tags of the words to the left of the word being predicted. The answers to these questions can be obtained without reference to the full parse of the sentence, and this makes it possible to use questions of this type in speech recognition, predicting words left to right, as they are received. The second class of questions query parse structure and/or tags of one or more of the previous sentences of the document to which the word belongs that is being predicted. These questions depend on the availability of the full parse for previous sentences that are queried. Therefore, questions of this second class are answered only after the recognition process has finished for all sentences to the left of the current one, and, for practicability, are used only in our perplexity experiments. Some examples of these questions are shown in Table 15. The first two questions ask about semantic features of tags to the left of the word being predicted, and within the same sentence. The last two questions query sentence structure of sentences to the left.

#	Question Description
1	v sem help to left
2	n sem money within two words to left
3	current or recent node is sprime coorded
4	current or recent node is sd with n subject n sem person verb v sem verbal act

Table 15: Examples of Questions

(4.4.2) The Trigram Tagger

In the following experiments, the history was labelled using a syntactic/semantic tagger. The tagset used is very rich, containing around 3000 possible tags. After evaluating the numerous tagging methods (see [7]), the trigram tagger was selected. The model is the same as that used by Merialdo [12]), except that the Katz backoff method [10] was used(See section 3.1. The training data size was about 376,000 words drawn from the domain of general travel.

The tagger was tested using about 2000 words of unseen travel data. The accuracy rate was 87% for an exact match against the treebank tag. However, with this tagset, often several tags are acceptable. Testing against a test set containing all acceptable tags would yield a higher and more realistic accuracy figure (for example, on the full ATR General English treebank, where this type of testing data is available, this tagger is 79% accurate at matching exact treebank tag, and 85% accurate at matching a list of correct tags).

(4.4.3) Perplexity Evaluation

The well–established trigram LM was used as the base LM for our experiments. This model was selected because it represents a respectable language model which most readers will be familiar with. The ME framework was used to build the derivative models since it provides a principled manner in which to integrate the diverse sources of information needed for these experiments.

The training set used to train the linguistic question–based triggers for all experiments was approximately 45,000 words of hand–labelled ATR treebank drawn from the 'hotel reservation task' domain. The training set used to train the baseline trigram LM and long history word trigger model was 80,000 words of hotel reservation data. The number of constraints used in all the models is shown in Table 16

Perplexity evaluation data was 2,000 words from hand–labelled hotel reservation treebank. Full parse of these sentences was made beforehand, therefore, the parse structure was queried in this experiment. The perplexity of all the models is shown in Table 17.

In Table 17, "trigram" is the perplexity of the base trigram model before any ME training. "trigram+WTModel" is the perplexity of the ME model which combines long history word triggers with the base trigram model. "trigram+Q's" is the perplexity of the ME model combining question triggers with the base trigram model. "trigram+WTModel+Q's" is the perplexity of the full ME model (using all the features) after training. The perplexity reduction for this task is similar to that attained on the WSJ domain by Zhang et al. [18].

(4.4.4) Recognition Error Rate

To evaluate our technique we used the ATRSPREC speech recognition system, designed by ATR for speaker-independent English recognition. Our language model was used to rescore the best N hypotheses output by the speech recognition system, yielding a new, reordered N-best list of hypotheses.

trigram models	1800(uni)	6020(bi)	9240(tri)
word triggers	10312		
question triggers	10796		
question numbers	6455		

Table 16: Number of constraints in the models

Model	PP	Change (%)
trigram	39.0	–
trigram + WTModel	34.7	11.0
trigram + Q's	35.2	9.7
trigram + WTModel + Q's	31.5	19.2

Table 17: The effect of combining the models

In order to guarantee the experiments are speaker-independent, the training speech data and test speech data are taken from different speakers. None of the data used for training the acoustic and language models occurred in the test set.

The acoustic model was trained using 42 hotel reservation conversations spoken by 11 speakers. The language models were trained on 80,000 words of hotel reservation data. The test data consisted of 344 utterances spoken by 11 different speakers. Arbitrarily, we set the number of hypotheses output from the speech recognition system N at 200. If the best hypothesis is chosen by an oracle, from these 200 hypotheses, the test set word error rate of the combined system is 24.4%. This represents the upper-bound in WER for these experiments.

The rescoring process proceeds as follows and is shown algorithmically in Figure 3. The tag for each word in each hypothesis is predicted using our tagger. This tag sequence is then combined with the predicted tags from previous sentences to form a history for this hypothesis. Linguistic questions are asked of the history and the answers are combined with the base LM (Equation 1) to obtain probabilities for each word in the hypothesis. Finally, the chain rule is used to combine these, to produce the language model's probability of the hypothesis. This probability is linearly interpolated with the probability from the acoustic model to yield the probability used to rank the N-best list of hypotheses.

Querying the full parse of the sentences would be desirable, and although possible, the parsing process is much slower than tagging, and possibly too inaccurate to be useful (the parses will be on top of already errorful predicted tags). Thus for the purposes of these experiments we restricted ourselves to questions over a tagged history.

All the LMs used in this experiment for rescoring were the same as those in the perplexity experiment. The experimental results are shown in Table 18.

Model	WER (%)
trigram	27.80
trigram + WTModel	27.45
trigram + Q's	27.12
trigram + WTModel + Q's	26.98

Table 18: The recognition results

FOREACH hypothesis $hp_0, hp_1, \ldots, hp_N$ (N=200)

(The hypothesis hp_i is a word sequence $w_0 w_1 \ldots w_L$ where L is the number of words in this hypothesis)

TAG this word sequence, to obtain tag sequence $t_0 t_1 \ldots t_L$

FOREACH $j \in \{0, 1, \ldots, L\}$

Create history $h_j = w_0 w_1 \ldots w_{j-1} + t_0 t_1 \ldots t_{j-1}$ + history derived from previous sentences

Find active word triggers and question triggers based on h_j and use Equation 1 to get $P(w_j|h_j)$

Calculate $P_{LM}(hp_i) = \prod_{j=0}^{L} P(w_j|h_j)$

Calculate $P_{interp}(hp_i) \doteq \lambda_1 P_{LM}(hp_i) + \lambda_2 P_{AM}(hp_i)$

OUTPUT the best hypothesis: $hp_{best} = \max_{i=0,N} P_{interp}(hp_i)$

ADD the word sequence and tag sequence of hp_{best} to the history derived from previous sentences

where:

- $P_{LM}(hp)$,$P_{AM}(hp)$ are the language/acoustic model probabilities of hypothesis hp;
- h_j is the word and tag history preceeding word w_j;
- Interpolation weights λ_1 and λ_2 were chosen empirically.

Figure 3: The rescoring process

(4.4.5) Discussion

The experiments presented here have focused on showing that we can glean useful information from the linguistic analysis of syntactic and semantic tags in the history of the word being predicted. The experiments demonstrated that this information gives a reduction in perplexity, of 88% (90% on WSJ [18]) of that provided by the long–range word triggers used by [14]. Moreover, when these triggers are used in conjunction with a model incorporating long–range word triggers, 93% (91% on WSJ [18]) of the perplexity gain from both sources is inherited by the new model. When the technique (restricted to questions about tagged text in the history) is integrated into a speech recognition system, the error rate of the system is reduced. The gain in error rate using linguistic questions is approximately double that of the long history word triggers. When both sets of features are used, most of the error rate improvement from both techniques is passed on. This indicates that the information we are providing is new and complementary. This is in line with our intuition, given the nature of the questions we ask.

The work presented here together with [18, 15, 16] represents a first step in exploiting tags and parse structure in the extrasentential history to assist a language model. Even at this early stage, it is clear from the results of these experiments that there is a significant amount of useful information for a language model in the parses and tags in the history. We feel that further improvements can be made by developing the language we are using to ask these questions and thereby improving their expressive power. Although impracticable at present, it would also be desirable to extend the technique to query the parse structure of the history. Finally, we feel that large gains will follow improvements in the base speech recognition system accuracy. A 0.8% improvement in WER was achieved over a trigram model. It is a modest amount but this result is arrived at given only a 3.4% possible improvement. Our rescoring removed 24.4% of the removable error. However, most of the time the system makes an error, the true utterance is not present in the N-best list and therefore the rescoring is unable to help. For those cases which rescoring is able to help, these additional features make a big difference.

4.5 Conclusions of Language Models of Speech Recognition

In this section we described an advanced language model of speech recognition. Our main concern has been to show that linguistic structure information, contributed by an English parser can be applied and combined with local N-gram information to enhance the performance of a common N-gram model. A new type of triggers, linguistic question triggers, are proposed to formalize and convey our linguistic structure. The maximum entropy approach was used to integrate all the information we collected such as local n-gram, long distance word triggers and question triggers. In the experiments presented we showed the contributions of each information to recognition respectively and we tested our model both in WSJ and hotel-reservation-task. Although in the experiments the improvement over N-gram model contributed by linguistic questions was not significant, nevertheless, the approaches of using linguistic question triggers and applying maximum entropy methods were proved effective. We hope we could achieve a better result in the future continuing work.

5 Concluding Remarks on Language Models of Tagging and Language Models of Speech Recognition

In this report we described our methods in dealing with part-of-speech tagging and language modeling of speech recognition. Both the problems were considered as building related language models. For the case of pos tagging it is to build a tagging language model of assigning a tag with a higher probability to a word from many selectable tags, for the case of speech

recognition to build a word language model of discriminating many hypotheses from a speech recognizer. In considering the two problems we have employed the same motivations of using riched information to improve the traditional baseline system which is a n-gram pos tagger and a n-gram language model in speech recognition. The information sources used in our work are as follows:

- Neighboring constraints/triggers (Conventional N-gram)
- Long distance POS constraints/triggers
- Long distance word constraints/triggers
- Linguistic questions constraints/triggers

Among these constraints linguistic question triggers are a new type of triggers proposed by us to convey the information from linguistic structure and detailed syntactic and semintac tags. The maximum entropy approach that we have adopted is extremely accommodating in the research reported here. This approach provides a unified framework to combine multiple sources of information. The experimental results showed that local information, the most useful, contributed to 97% of the improvement given by language model and 3% of the improvement was devoted by higher linguistic knowledge. With regard to a lower and first goal we are satisfied with the experimental results which proved our initial motivation though the improvement over baseline system was not significant. With regard to a high goal a big challenge was put forward that in the future research we want to improve our model by finding more predictive linguistic question triggers. We feel that we can realize considerable gain by further developing the language we are using to ask these questions, and thereby improving their expressive power.

6 Acknowledgments

The present report is on my last two years research activities and outcomes of ATR ITL. There are many people who contributed to this process, both technically and personally.

First, I would like to express my gratitude to my supervisors, Dr. Ezra Black and Dr. Andrew Finch, who contributed a great number of ideas to my research. None of the work presented here could have ever been accomplished without the technical contributions of their two.

I would also like to thank Dr. John Lafferty, professor of CMU, who enthusiastically answered any of my questions whenever I asked. I owe nearly all that I know about maximum entropy modeling to him.

In approaching this research work I have received a number of assistance from many people of ITL to whom I want to say thank you very very much. They are as follows: Dr. Shuwu Zhang, Dr. Harald Singer, Mr. Hirofumi Yamamoto, Dr. Hideki Kashioka, Dr. Tomoko Matsui, Mr. Jeremy Bateman and Mr. Tal Shalif.

Finally, I wish to thank Dr. Seiichi Yamamoto, president of ITL, and, Dr. Yoshinori Sagisaka, head of Dept. 1, for their enthusiastic support of this research work described here.

References

[1] A.Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–67, 1996.

[2] A.Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, 1996.

[3] E. Black, S. Eubank, and H. Kashioka. Probabilistic parsing of unrestricted english text, with a highly-detailed grammar. In *Proceedings, Fifth Workshop on Very Large Corpora*, Beijing/Hong Kong, 1997.

[4] E. Black, S. Eubank, H. Kashioka, R. Garside, G. Leech, and D. Magerman. Beyond skeleton parsing: producing a comprehensive large–scale general–english treebank with full grammatical analysis. In *Proceedings of the 16th Annual Conference on Computational Linguistics*, pages 107–112, 1996.

[5] E. Black, S. Eubank, H. Kashioka, and J. Saia. Reinventing part-of-speech tagging. *Journal of Natural Language Processing (Japan)*, 5(1), 1998.

[6] E. Black, A. Finch, and H. Kashioka. Trigger-pair predictors in parsing and tagging. In *Proceedings, 36th Annual Meeting of the Association for Computational Linguistics, 17th Annual Conference on Computational Linguistics*, pages 131–137, Montreal, 1998.

[7] E. Black, A. Finch, and R. Zhang. Applying extrasentential context to maximum entropy based tagging with a large semantic and syntactic tagset. In *Proceedings, Seventh Workshop on Very Large Corpora*, 1999.

[8] C. Chelba and F. Jelinek. Exploiting syntactic structure for language modelling. In *Proceedings, 36th Annual Meeting of the Association for Computational Linguistics, 17th Annual Conference on Computational Linguistics*, pages 225–231, Montreal, 1998.

[9] E.Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 1995.

[10] S. Katz. Estimation of probabilities for sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics Speech and Signal Processing*, 35:400–401, 1987.

[11] R. Lau, R. Rosenfeld, and S. Roukos. Trigger-based language models: a maximum entropy approach. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, pages II:45–48, 1993.

[12] B. Merialdo. Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–172, 1994.

[13] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[14] R. Rosenfeld. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech and Language*, 10:187–228, 1996.

[15] R.Zhang, A.Finch, E.Black, and Y.Sagisaka. Linguistic question-based language modeling built by me. In *Proceedings of ASJ Conference*, pages 45–46, March 1999.

[16] R.Zhang, A.Finch, E.Black, and Y.Sagisaka. The maximum entropy approach to language modeling of atrsprec. In *Proceedings of ASJ Conference*, pages 83–84, September 1999.

[17] R.Zhang, A.Finch, E.Black, and Y.Sagisaka. Applying contextual syntactic and semantic tags to maximum entropy based language modeling. In *Proceedings of ASJ Conference*, March 2000.

[18] R. Zhang, E. Black, and A. Finch. Using detailed linguistic structure in language modelling. In *Eurospeech'99*, pages 1815–1817, Budapest, 1999.

[19] R. Zhang, E. Black, A. Finch, and Y. Sagisaka. Integrating detailed information into a language model. In *ICASSP'2000*, Istanbul, 2000.