Beyond Tag Trigrams: New Local Features for Tagging

Andrew Finch, Ezra Black and Ringo Wathelet

ATR Spoken Language Translation Laboratories
2-2 Hikaridai Seika-cho, Soraku-gun,
Kyoto, Japan 619-02

finch,black@slt.atr.co.jp watheletau@yahoo.com.au

Abstract

The set of features used by any predictive model is of pivotal importance to its performance. In this paper we show the utility and quantify the effect of adding features consisting of arrangements of words and tags (selected by an expert grammarian) in the local context of a trigram tagger. We look in detail at the effect, on tagging with a large syntactic and semantic tagset, of adding these features. We show that the addition of a set of such features improves the the error rate of a trigram tagger by approximately 11%.

1. Introduction

One of the main areas of research in tagging has always been the nature of the model used to predict the tags for the words in a sentence. In this paper we look at the problem from a slightly different, and we feel equally important point of view. We investigate an often overlooked aspect of tagger development, the features used by the model to perform the tagging.

1.1. Tagging Model

Although a large number of varied techniques have been applied to the task of tagging, perhaps the most widely used (and effective) set of features found in the literature for tagging are the features used by "n-gram" taggers. These taggers use the following features from the context of the word being tagged:

- The identity of the word being tagged
- The identity of the tags of the two words prior to the word being tagged

To build our tagging models we selected the Maximum Entropy framework as our basic paradigm. Within this framework, exponential models of the following form are constructed.

$$p(t|h) = \gamma \prod_{k=0}^{K} \alpha_k^{f_k(h,t)} p_0 \qquad (1)$$

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 feature 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 (Ratnaparkhi,1996). Our 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. $t_{-2}t_{-1} = XY$ & $t = T$

where:

- 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};

Our baseline model differs from Ratnaparkhi's in that it does not use any information about the occurrence of words in the context or their properties (other than in constraint 1). Our model exploits the same kind of tag–n–gram information that forms the core of many successful tagging models, for example, (Kupiec,1992), (Merialdo,1994), (Ratnaparkhi,1996). We refer to this type of tagger as a tag–n–gram tagger.

This model has many attractive characteristics for these experiments. One major advantage is that this method allows for the possibility of using a wide variety of diverse features in the same model. Moreover, existing models can simply be 'enhanced' by the addition of new features, allowing the possibility of comparison between models built from two feature sets, one being a subset of the other. Finally, it is possible to build a model that uses exactly the set of "n-gram" features described above.

2. Experimental Methodology

2.1. Training and Testing Data

All models were trained on the tags from 850,000–word ATR General English Treebank. This full treebank consists of data drawn from a very wide spectrum of American English text sources, containing data from sources ranging from articles drawn from the Wall Street Journal to fliers written to promote Chinese restaurants. All the data we use has been split by hand into sentences and the textual data has also been "tokenized", that is, regualarized into a stream of tokens drawn from a common token vocabulary. During the tagging of real data, these two tasks are of course done automatically. These experiments assume that the sentence splitting and tokenization has been done for all data.

All taggers were tested on the accompanying 53,000–word test treebank which contains data unseen by any of the models during training and also unseen by the expert grammarian who selected the features to be used by the models.

2.1.1. The Tagset

To understand what 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 (Black et al.,1998; Black et al.,1996). 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", 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. To illustrate the level of detail expressed by our tagset, consider the word-tag "gas_NN1SUBSTANCE" in the first sentence. The NN part of the tag indicates the word is a common noun. The "1" part of the tag means that the noun is singular, and the "SUBSTANCE" part of the tag represents the semantics of the word.

2.2. Predictive Features

In this paper we use the term "feature" to mean an input feature to a predictive model, rather than, for example grammatical features. The choice of which features to use in any model is of critical importance. If the features are chosen poorly, the model, no matter how sophisticated, may be missing the information it needs to solve the problem. Clearly the number of possible features available in the context of the word being tagged is enormous. For these experiments we will restrict ourselves to simple extensions of the "n-gram" features described earlier. However now will will allow features to be composed of any combination of the identity of words and/or tags in the history *and* the words in the future of the word being tagged. That is; when generating a feature, we may choose to look at the previous three words and tags together with the next three words, and also the identity of the word being tagged itself.

The features themselves were expressed as templates for feature generation and these templates were written by an expert grammarian.

As an example, these are the templates used to generate the standard "n-gram" feature set used in the ME n-gram tagger: $\{(w,t),\ (t_{-1},t),\ (t_{-2}t_{-1},t)\}$. Here w's refer to words and t's refer to tags. The subscript indicates the relative position of the word/tag to the word being tagged. For example t_1 is the identity of the previous word's tag, w is the identity of the word being tagged and w_1 is the identity of the word after the word being tagged.

We chose to select simple local features as the feature set for these experiments because, intuitively at least, one might expect to receive the largest amount of information about the identity of a word's tag to come from the words and tags in the immediate neighbourhood. However, other feature sets are possible and Black et al (Black et al.,1998) have investigated the effects of adding into tagging models features relating to extrasentential context, and demonstrated that such information is also useful to a tagger.

2.2.1. Triggers

The features used as input for a maximum entropy model are often referred to as "triggers". Triggers are features that embody the following notion: when certain "triggering" events have occurred in a document, the probability of occurrence of specific "triggered" tags is adjusted to reflect this.

2.2.2. Feature selection

Although the feature templates were chosen by a human expert, the total number of features it is possible to generate from these templates is large. This is due to the large size of the tag set, and (more importantly for the features that include word identities), the size of the word vocabulary. Therefore, a rapid way to reduce the number of features used by the model is needed. One way is to add the features one at a time and evaluate their usefulnesss in the model (Della Pietra et al.,1997). For the purpose of these experiments we chose to use mutual information to gauge the effectiveness of candidate features. While less pricipled, this technique is nonetheless effective, is often used (Rosenfeld,1996), and has a considerable advantage

```
It_PPH1 has_VHZ meant_VVNMEAN great_JJDEGREE savings_NN2MONEY ,_,
both_RRCONCESSIVE in_IIIN  time_NN1TIME &_CCAMP gas_NN1SUBSTANCE  !_! "_"R

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

OR_CCOR ONE_MC1WORD FREE_JJSTATUS  FANTAIL_NN1ANIMAL SHRIMPS_NN1FOOD
```

Figure 1: Three ATR/Lancaster English Treebank Tagged Sentences: One from a Credit Union (Bank) Brochure, and Two (Non–Sequential) from a Flier Advertising a Restaurant Offering Chinese Food

in terms of the computational time required.

That is, we use the following formula to gauge a feature's usefulness to the model:

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

where:

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

Only those features with a high mutual information with t were used in the models.

2.3. Tagging Models

As stated earlier, the taggers we used for these experiments use exponential models to provide the tag probabilities, given the context. However, we are not seeking to evaluate the taggers on their performance on invividual words, but rather their performance on sequences of words. Therefore, we also need a way to search for the best (highest–probability) sequence of tags for the sentence.

We use the beam search algorithm shown in Figure 2 to find this optimal sequence.

2.3.1. Experiment 1: The Standard n-gram Model

A tagger was built using only the standard n-gram features: $\{(w,t),\ (t_{-1},t),\ (t_{-2}t_{-1},t)\}$. The accuracy results for this tagger are shown in the last line of Table 1.

2.3.2. Experiment 2: The Standard n-gram Model with Additional Local Features

In this experiment all features used in the previous experiement were also used. The features used in this augmented tagger included in addition:

$\{(w_{-2}w_{-1}w,t),\ (w_{-1}ww_1,t),\ (ww_1w_2,t),\ (w_{-1}w,t),\ (ww_1,t),\ (t_{-2},t),\ (t_{-1}w_1,t),\ (t_{-1}ww_1,t),\ (w_{-1}w_1,t),\ (w_{-1},t),\ (w_1,t),\ (t_{-1}w,t),\ (t_{-2}t_{-1}w,t),\ (w_{-2}w_{-1},t),\ (w_1w_2,t)\}$.

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

3. Discussion

Table 1 lists the effect of each of the features when introduced separately into a base model containing only the (w,t) features. The most significant features in the model were the identity of the previous (76.9%) and the next (76.9%) word. Surprisingly, these features improved the model more than the commonly used tag-trigram features (highlighted in bold). The n-gram only tagger gave an accuracy of 76.24%, whereas the enhanced model gave an accuracy of 78.8%, an improvement in error rate of around 11%.

We feel the results here demonstrate that there is a lot to be gained by exploring the form of features that are used in tagging models. Clearly a simple tag n-gram model, although effective and easy to implement is missing a lot of information that is readily available in the local tag and word context of the word being tagged.

Although not used for the experiments reported here, we have developed a more accurate metric for evaluating tagsets of this kind (where many possible tags are valid for a word in context). This measure, evalutates the output of the tagger against a list of valid tags for the word, as opposed to a single tag. When this metric is used to test the tagger, considerably (approximately 7%) higher (and more representative) performance figures are achieved.

```
[1] Calculate the probability of each tag in the tagset for the first word
[2] Create a partial hypothesis for the tag sequence using this tag and its
    probability
[3] Sort the hypothesis list according to probability
[4] FOREACH hypothesis in the top BEAMWIDTH
    - Calculate the probability of each tag in the tagset for the next word,
      given the context of this hypothesis
    - Extend all hypotheses by each new tag and add to the list of hypotheses
[5] IF at last word THEN
       GOTO [6]
    ELSE
       GOTO [3]
[6] Output the highest probability hypothsis in the hypothesis list
```

Figure 2: The beam search algorithm used to find the best tag sequence.

Trigger Type	Number of triggers	Test set 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
$(w,t)+(ww_1w_2,t)$	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 1: Experimental Results of Tagging Using Detailed Local Constraints

4. References

E. Black, A. Finch, H. Kashioka. 1998. 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.

E. Black, S. Eubank, H. Kashioka, J. Saia. 1998. Reinventing Part-of-Speech Tagging. *Journal of Natural Language Processing (Japan)*, 5:1.

E. Black, S. Eubank, H. Kashioka, R. Garside, G. Leech, and D. Magerman. 1996. 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, Copenhagen.

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

J. Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. In *Computer Speech and Language*, 6:225–242.

B. Merialdo 1994. Tagging English text with a probabilistic model. In *Computational Linguistics*, 20(2):155–172..

A. Ratnaparkhi. 1996. A Maximum Entropy Part-Of-Speech Tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, University of Pennsylvania.

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