Clause Detection using HMM

Antonio Molina and Ferran Pla
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València (Spain)
{*amolina,fpla*}*@dsic.upv.es*

1 Introduction

In this work, we apply a specialized HMM approach to the shared task: clause identification (Tjong Kim Sang and Déjean, 2001). The HMM formalism (Rabiner and Juang, 1986) has widely been used to solve other NLP problems, such as POS tagging, chunking, partial parsing, etc. A similar technique (Lexicalized HMM), that takes into account certain words to lexicalize the contextual language model, was previously applied for solving POS tagging and chunking problems (Pla et al., 2000b). Usually, for these tasks, specialized HMMs perform better than non-specialized HMMs (Pla, 2000a).

We used specialized HMMs to solve the three parts of the task. Part 1 (clause start detection) and Part 2 (clause end detection) are performed as a tagging problem, that is, we assign the more probable tag to each token of the input. Part 3 (embedded clause detection) is a more complex task because the output must be correctly balanced. Therefore, it is carried out in two phases: first, we find the best sequence of tags (segmentation in clauses) for the input sentence; second, we correct some balancing inconsistencies observed in the output by applying some rules.

2 Clause detection as a tagging problem

We consider clause detection to be tagging problem. From the statistical point of view, tagging can be defined as a maximization problem.

Let $\mathcal{O}$ be a set of output tags and $\mathcal{V}$ the vocabulary of the application. Given an input sentence $I = i_1, \ldots, i_T$, where $i_j \in \mathcal{V} : \forall j$, the process consists of finding the sequence of states of maximum probability on the model. That is, the sequence of output tags, $O = o_1, \ldots, o_T$, where $o_i \in \mathcal{O} : \forall i$. This process can be formalized as follows:

$$
\begin{aligned}
\widehat{O} &= \arg\max_{O} P(O|I) \\
&= \arg\max_{O} \left(\frac{P(O) \cdot P(I|O)}{P(I)} \right); \; O \in \mathcal{O}^T
\end{aligned} \quad (1)
$$

Due to the fact that this maximization process is independent of the input sequence, and taking into account the Markov assumptions, the problem is reduced to solving the following equation (for a second–order HMM):

$$
\arg\max_{o_1 \ldots o_T} \left(\prod_{j:1 \ldots T} P(o_j|o_{j-1}, o_{j-2}) \cdot P(i_j|o_j) \right) \quad (2)
$$

The parameters of equation 2 can be represented as a second–order HMM whose states correspond to a tag pair. Contextual probabilities, $P(o_j|o_{j-1}, o_{j-2})$, represent the transition probabilities between states and $P(i_j|o_j)$ represents the output probabilities.

For this clause–splitting task, the available information consists of words, POS tags, chunk tags, start tags (Part 1), end tags (Part 2) and clause tags (Part 3). (See Figure 1).

In our approach, we must define the input vocabulary ($\mathcal{V}$) and the output vocabulary ($\mathcal{O}$) from the available information. We tested our system with different combinations of this information for each part of the task. Following, we present the criteria that yield the best performance on the development set.

For the three parts, we consider that input sentences are composed of the sequence of POS and Chunk tags associated to each word,

WORDS	POS	CHUNK	START	END	CLAUSE	SP START	SP END	SP CLAUSE
The	DT	B-NP	S	X	(S(S*	DT–S	DT–E	DT–(S1(S2*
carrier	NN	I-NP	X	X	*	NN–X	NN–X	NN–*2
has	VBZ	B-VP	X	X	*	VBZ–X	VBZ–X	VBZ–*2
valuable	JJ	B-NP	X	X	*	JJ–X	JJ–X	JJ–*2
trans-Pacific	JJ	I-NP	X	X	*	JJ–X	JJ–X	JJ–*2
and	CC	I-NP	X	X	*	CC–X	CC–X	CC–*2
Asian	JJ	I-NP	X	X	*	JJ–X	JJ–X	JJ–*2
routes	NNS	I-NP	X	E	*S)	NNS–X	NNS–E	NNS–*S2)
but	CC	O	X	X	*	CC–X	CC–X	CC–*1
it	PRP	B-NP	S	X	(S*	PRP–S	PRP–X	PRP–(S2*
remains	VBZ	B-VP	X	X	*	VBZ–X	VBZ–X	VBZ–*2
debt-laden	JJ	B-ADJP	X	X	*	JJ–X	JJ–X	JJ–*2
and	CC	O	X	X	*	CC–X	CC–X	CC–*2
poorly	RB	B-ADVP	X	X	*	RB–X	RB–X	RB–*2
managed	VBD	B-VP	X	E	*S)	VBD–X	VBD–E	VBD–*S2)
.	.	O	X	E	*S)	.–X	.–E	.–*S1)

Figure 1: Example of the result of applying the specialization on the training set for the different parts of the task.

$I = (p_1, ch_1), (p_2, ch_2), \ldots, (p_T, ch_T)$. Therefore, the vocabulary of the application, $\mathcal{V}$, is defined as tuples (POS tag, Chunk tag).

The output vocabulary is different for each part of the task. In Part 1, $\mathcal{O} = \{S, X\}$; in Part 2, $\mathcal{O} = \{E, X\}$; and in Part 3, we considered the clause boundaries whose depth level is lower than a certain value, that is, $\mathcal{O} = \{(S*, *S), *, (S * S), *S)S), \ldots\}$. In this case, this value corresponds to the maximum depth level observed in the training set.

Thus, given an input sentence I, the process of clause detection consists of finding the sequence of states (the sequence of clause tags) of maximum probability on the model. This process is carried out by Dynamic Programming Decoding using the Viterbi algorithm.

This basic model has two main drawbacks. The first one is that the output tag set is too generic to produce accurate models. Moreover, in Part 3, the model does not assure a correct balancing of the embedded clauses in the sentence. To solve these problems, we define a technique which specializes the models. This technique consists of enriching the language model by incorporating a set of features to the output vocabulary.

In Part 1 and Part 2, we have relabeled the output tag of an input word by adding the corresponding POS tag. In Part 3, we have also added the number corresponding to the depth level of the clause in the sentence in order to reduce the incorrectly balanced clauses.

An example of the result of applying this specialization criteria to a sentence from the training sets can be seen in Figure 1 (see columns with specialized –SP– tags).

We learnt the corresponding specialized HMM for each part of the task using this new training set. Each model was smoothed by applying a standard linear interpolation method.

Note that, when specialized HMMs are used, no change is needed both in the learning and the tagging processes. You simply have to map the sequence of specialized output tags to the original output tags. This substitution can be done in a direct way.

In Part 3, the smoothed model guarantees a complete coverage of the language, but does not assure the correct balancing of the output. Therefore, we have used some correcting rules to repair the inconsistencies in the output. We have applied the following rules:

1. If the clause segmentation presents more *start* than *end* boundaries, we add the *end* boundaries that are needed to the last word in the sentence (just before the dot).

2. If the clause segmentation presents more *end* that *start* boundaries, we add the *start* boundaries that are needed to the first word in the sentence.

3. If the sentence does not start with a *start* boundary or does not finish with an *end* boundary, we add these *start* and *end* tags.

3 Experimental Results

We considered second–order HMM (3–grams) which were specialized according to the criteria described above, taking as input the tuples of POS and chunk tags associated to each word. We also tested the system using other criteria. We chose different input vocabularies: only words, only POS, only chunks, etc. Moreover, we used different specialization criteria: specializing with chunk tags, partial specialization, etc. None of these criteria outperformed the results reported in Table 1.

Although Part 1 and Part 2 were tasks that seem to have a similar difficulty, the experimental results show that clause end detection is more difficult than clause start detection. We think this could be because the relation between POS and clause start is stronger than the relation between POS and clause end marks.

We performed two additional experiments. First, we combined[1] the output of Part 1 and Part 2 in order to obtain Part 3 output. The obtained results on the test set are lower than the results presented in Table 1 (precision=70.74%; recall= 58.62%; $F_{\beta=1}$= 64.11). Second, we derived the output for Part 1 and Part 2 from the output obtained in Part 3. In this case, we obtained worse results for Part 1 ($F_{\beta=1}$= 84.62), but better results for Part 2 ($F_{\beta=1}$= 84.24). However, these results are not correct for the shared task because we used the information of embedded clauses which is not available for solving the first two parts.

4 Conclusions

In this work, we have successfully applied a specialized HMM to a clause detection task. These models have been specialized using different criteria. The best results for Part 3 were obtained considering POS and clause depth level

[1]We have used the baseline script provided by the workshop organizers.

development	precision	recall	$F_{\beta=1}$
part 1	89.21%	87.72%	88.46
part 2	78.81%	78.54%	78.68
part 3	70.70%	71.35%	71.03

test	precision	recall	$F_{\beta=1}$
part 1	88.15%	84.88%	86.48
part 2	79.63%	77.17%	78.38
part 3	69.62%	64.17%	66.79

Table 1: Results obtained for the development and test data set for each part of the shared task.

($F_{\beta=1}$= 66.79). Future works would include testing other specialization criteria and studying the way to incorporate the words in the models (Lexicalized HMM).

Moreover, we think that a more detailed study of the problem is needed to assure the correct balancing of the output by including restrictions on the model; for example, modifying the smoothing methods or including linguistic restrictions.

5 Acknowledgments

This work has been supported by the Spanish research project CICYT TIC2000–0664–C02–01

References

Ferran Pla. 2000a. Etiquetado Léxico y Análisis Sintáctico Superficial basado en Modelos Estadísticos. Ph.D. Thesis. Departament de Sistemes Informàtics i Computació. Universitat Politècnica de València, Spain.

Ferran Pla, Antonio Molina and Natividad Prieto. 2000b. Improving Chunking by means of Lexical-Contextual Information in Statistical Language Models. In *Proceedings of 4th CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

L. R. Rabiner and B. H. Juang. 1986. An Introduction to Hidden Markov Models *IEEE ASSP MAGAZINE*.

Erik F. Tjong Kim Sang and Hervé Déjean. 2001. Introduction to the conll-2001 shared task: Clause identification. In *Proceedings of the CoNLL-2001*. Toulouse, France.