

Confusion Matrices approach to Transformation-Based Learner with Fine Grain Transitivity Tag-set.

Ting Law

Department of Computation

University of Manchester Institution Science and Technology

UK

T.Law@postgrad.umist.ac.uk

Abstract

This paper describes a fully unsupervised Transformation-Based Learning (TBL) Tagger that learns a sophisticated tag-set, with a minimal manually tagged dictionary. No tagger, is perfect, most POS taggers can score highly at around 95% to 97% of accuracy. The tagging accuracy decreases as the tag-set gets more sophisticated. The confusion matrices contain information about likely alternatives for the verbs. This approach can bring up the tagging accuracy with a minimal process, even when the tagger already achieved 95% to 97%.

1. Introduction

Parsers parse sentences with the provided POS (Part of Speech) information. In many cases the parser fails to parse due to lack of POS information. Manually tagged POS dictionaries can be very accurate, but they are hard to produce on a large scale. Therefore we propose to use a Transformation-Based Learner to learn a rule set and confusion matrices to assist and lighten the work load of the parser.

The Brill Tagger (Brill 1995) is a well-known automatic POS tagger.

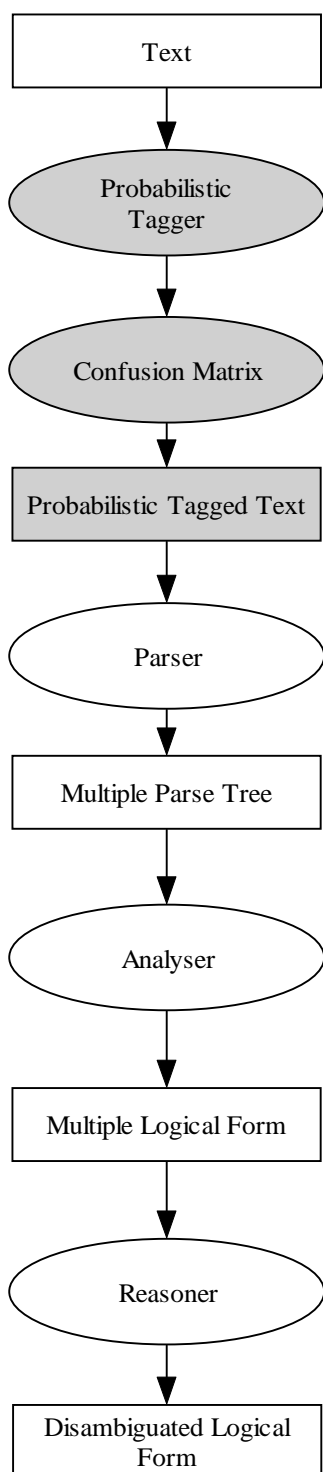
Mutbl_lite (Lager 1999) turns the big TBL system into a piece of small and flexible code. The Mutbl_lite is modified for my needs together with the Brill Templates (Brill 1995) and I embedded the learned rules together with the confusion matrixes into an existing parser, Parasite (Ramsay, 2002),

“*Figure 1*” shows the flows of a Natural Language (NL) system that is embedded with a probabilistic tagger and confusion matrices.

2. Previous taggers

Merialdo (1995) used the Baum-Welch algorithm with a dictionary built for the learning article that is manually tagged which achieved the peak accuracy at 86.6%. Elworthy did a similar experiment in 1994, he achieved the peak accuracy of 92% with LOB corpus and 83,6% with the Penn Treebank corpus. Brill (1995) used transformation-based learning in his tagger and achieved a very high peak result at 96.5% with the Penn Treebank corpus, which is similar to our result with the BNC corpus. Hence non of the accuracy above score with transitivity tag-set

Figure 1:



3 Tag-set

The accuracy of a tagger is very dependent on the quality of the tagger, but the sophisticated level of the tag-set takes a

more important role. It will be very easy to score 100% accuracy for any tagger when there are only two tags, e.g. "punctuation" and "non punctuation". The tagging accuracy would have decreased as the level of the sophistication of the tag-set increased. Mutbl_lite can only score around 90%, in our case, with the transitivity tag-set. "Graph 1" shows the tagging result in different sizes of corpora.

The following is my transitivity tag-set:

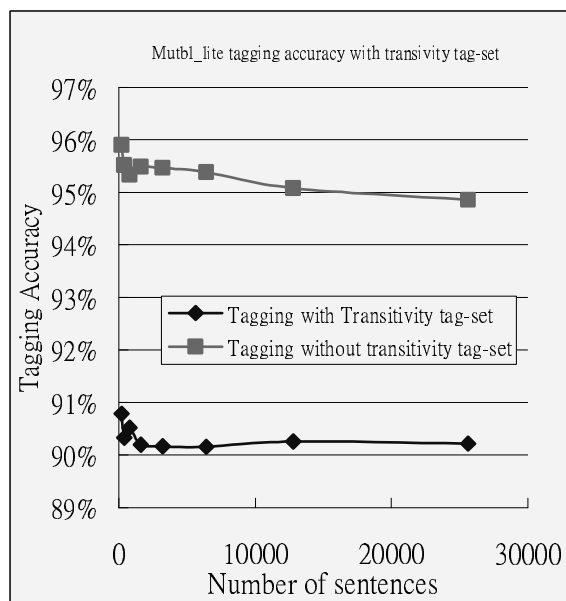
Verb1 = intransitive verb.

Verb2 = transitive verb.

Verb3 = ditransitive verb.

Scomp = Sentence composition.

Graph 1:



The transitivity approach can pre-disambiguate sentences and prevent miss-parse in some cases, where sentences can get away with a parse without the transitivity tags. The following example, "Figure 2" and "Figure 3" illustrated the

need of the transitivity sub-categorisation tags:

Figure 2:

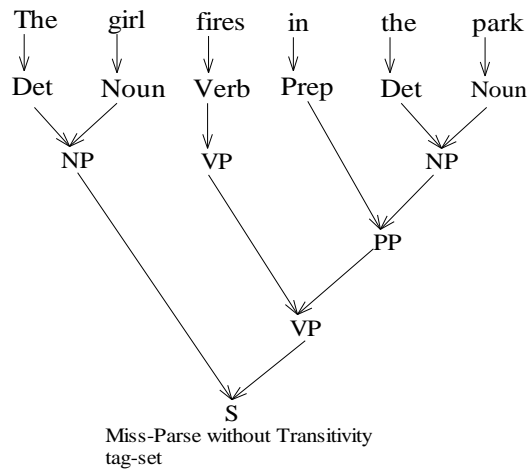


Figure 3:

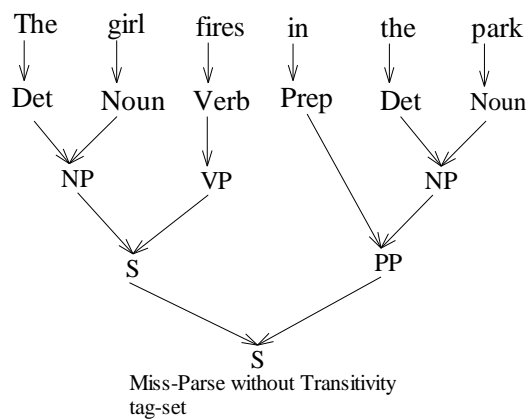
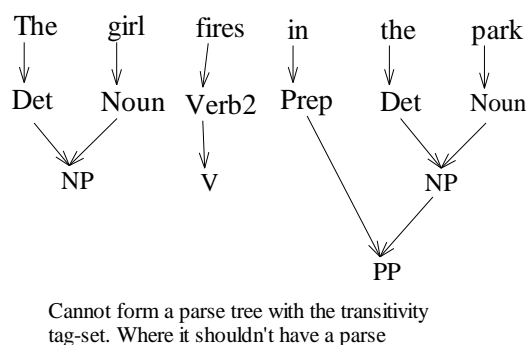


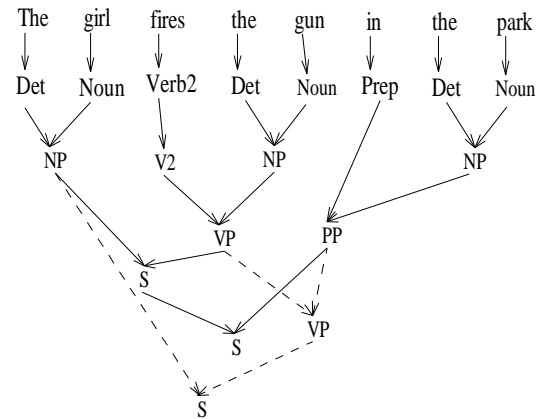
Figure 4:



The verb “fires” is a transitive verb which takes two arguments and cannot form a VP on its own, “Figure 4”. The above example contains only one argument, “girl”, for the

verb “fires” on the left. The argument on the right is missing, and so it cannot form a VP, but it has been treated as an intransitive verb in the above case so it gets a parse.

Figure 5:



The dotted line = second parse.

In the above case, “Figure 5”, if the “Fires” had not classified as “Verb2” the parser will generate many parses and most of them are miss-parse. Unless I tell the parser in advance, then the parser will only generates the above 2 parses. This shows the importance of the transitivity tags

4. Resource

We have the British National Corpus (BNC) as our resource. This version of the BNC comes with the C5 (Claws 5) tag-set, which is fairly fine grained but along the wrong dimensions for our purposes. I need a tag-set with sub-categorisation information rather than tense information for our syntactic parser, Parasite, and that is what the parser takes. We convert the BNC's words together with POS tags into my format. In particular, I convert all verbs in the BNC into “vtype_unknownv”, which

says that the item is a verb but we do not know its transitivity class.

Example 1:

"zip" \$\$ immediate verb(X)
delayed(vtype(X,unknownv)).

"rain" \$\$ X imminiate basic_noun delay
ntype(X, simple).

The source of training data is the set of sentences in the BNC for which Parasite produces an unambiguous analysis (approximately 37000 sentences). The training data has to be unambiguous otherwise the rules learned will be ambiguous. Parasite produces parses with actual tags and a list of potential tags, including the actual tag, of each word as in "Example 2".

Example 2:

[[{0,A,[det],det},{1,burden,[bNoun,scomp,verb1,verb2,verb3],bNoun},{2,had,[aux1_verb,verb2],aux1_verb},{3,been,[aux1_verb,bNoun,copula_verb,scomp,thereCop,unknown,verb1,verb2,verb3],aux1_verb},{4,taken,[scomp,verb1,verb2,verb3],verb2},{5,off,[adv,bNoun,m_comp_prep,scomp,verb1,verb2,verb3],m_comp_prep},{6,her,[c_pro_ref,det],c_pro_ref},{7,'',[stop],stop}]].

"has" here, for instance, may be either an auxiliary or a transitive verb ("verb2"), "been" has none possible tags, ...

5. Learning

We transformed the output of the parser into "Mutbl_lite format".

Mutbl_lite format:

'wd(P,W)' is true if the word 'W' is at position 'P' in the corpus.

'tag(P,A)' is true if the word at position 'P' in the corpus is tagged as 'A'

'tag(A,B,P)' is true if the word at 'P' is tagged as 'A' and the correct tag for the word at 'P' is 'B'

In "Example 3" the word "cast" is stored at position "10", and the current tag is "unknownv". The actual tag for the verb "cast" is "verb2". The tag is correct if it is stored as in "Example 4".

Example 3:

wd(10,cast).
tag(10,unknownv).
tag(unknownv,verb2,10).

Example 4:

wd(10,cast).
tag(10,verb2).
tag(verb2,verb2,10).

We acquired rule sets from retagging verbs whose transitivity class was unknown. "Example 5" is a rule that tells us to replace the tag "verb2" with the tag "verb1" when one of the next three "words" is a full stop.

Example 5:

tag:verb2>verb1<-tag:stop@[1,2,3].

Mutbl_lite produces a set of sequential rules and corrects the initial errors. In order to cope with cases where the tagging rules do not assign a label which leads to a successful final parse, we created a set of confusion matrices, example "*Confusion Matrix 1*". For instance, when it says the recall is "*Verb1*" there are 96.24% is a "*Verb1*", 2.51% to be a "*Verb2*", 0.12% to be "*Verb3*", 1.13% to be "*Scomp*" and 0.04% to be "*unknown*"

6. Confusion matrix.

If the tagger fails to make the right tag for the current verb, then the system can select the most likely alternative tag from the confusion matrix. According to the "*confusion_matrix_1*" if the current tag is "*Verb1*" and it is wrong, "*Example 6*", it is most likely to be a "*Verb2*", because the "*Verb2*" has the second highest score in the row of "*verb1*".

Example 6:

wd(10,cast).

tag(10,verb1).

tag(verb1,verb2,10).

Confusion Matrix 1:

		Precision				
		Verb1	Verb2	Verb3	Scomp	Unknown
Recall	Verb1	96.24%	2.51%	0.12	1.13%	.004%
	Verb2	7.40%	91.12%	0.39	1.09%	0.01%
	Verb3	3.50%	32.23%	61.4	2.87%	0.22%
	Scomp	19.76%	10.93%	0.84	68.47%	0.06%

7. Test + Result

We tested the tagger on an unseen corpus

from the rest of the BNC. We applied the rule set learned from the corpus with 25600 sentences, approx 204800 words, to our tagger.

The initial naive tagging accuracy for verbs is 0%, because all tagged as "*unknownvs*". The result of applying the tagger scored an average of 90.33% on verb tagging. The tagger with the confusion matrices approach scored an average of 97.07% in verb tagging. In fact it fixed 69.7% of errors that the tagger made initially. "*Table 2*" shows the tagging accuracy on verbs with different sizes of corpora and "*Graph 2*" clearly display the effect of the confusion matrix approach.

Graph 2:

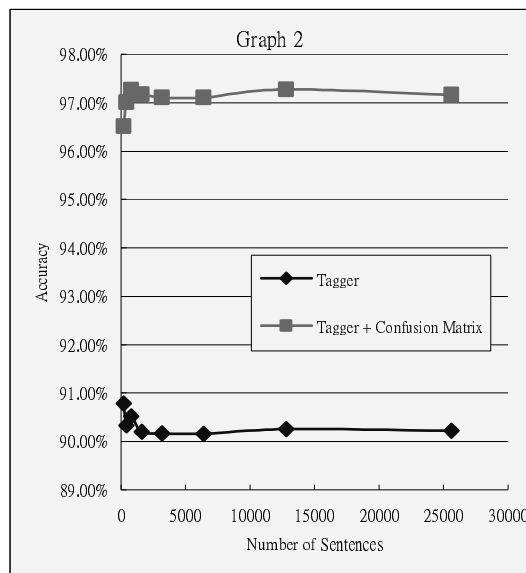


Table 2

Sentence	Tagger	Tagger+confusion matrices
200	90.79%	96.51%
400	90.33%	97.01%
800	90.52%	97.26%

1600	90.20%	97.17%
3200	90.17%	97.11%
6400	90.16%	97.11%
12800	90.26%	97.27%
25600	90.22%	97.16%

8. Conclusions

The confusion matrix approach clearly brings the tagging accuracy up to certain extent. It is feasible enough to move one step back on to the *Confusion Matrix* when the tagger has gone wrong. This is the most inexpensive and effective way to improve the parser performance. The implementation of the transitivity tag-set reduces the tagging accuracy but it helps to pre-disambiguate some structural ambiguity which will be both expensive for the parser and the reasoner to perform.

The confusion matrix approach can pick up the reduced accuracy of the tagging caused by the transitivity tag-set.

These two inexpensive approaches can be intergraded into syntactic parsers in order to improve the preferment and lighten the workload and perform some pre-disambiguations, which a parser cannot perform.

9. References

Brill, Eric (1993). A Corpus-Based Approach to Language Learning, Department of Computer and Information Science, University of Pennsylvania.

Brill, Eric (1994). Transformation-Based Error driven Learning and Natural Language Processing: A case study in Part of Speech Tagging, The Johns Hopkins University.

Dale Robert (Editor), Moisl Hermann (Editor), Somers Harold (Editor),(2000). Hand Book of Natural Language processing (Chemical industries Series).

Lager, Torbjorn (1999a) The μ -TBL system: Logic Programming Tool for Transformation Based Learning Computational Natural Language Learning (CoNLL-99), Bergen.

Lager, Torbjorn(1999b). The μ -TBL Lite: A Small, Extendible Transformation-based Learner. In (EACL'99), Bergen, Jun 8-12.

Lager, Torbjorn(1999b). The μ -TBL Mutbl user manual, http://www.ling.gu.se/~lager/Mutbl/new_manual.html

Oxford university, (2002) British National Corpus (BNC) <Http://www.hcu.oc.ac.uk/BNC/>

Ramsay, 1999 Parsing with discontinuous phrases, Natural Language Engineering.