Maximum Entropy Markov Models for Semantic Role Labelling

Phil Blunsom
Department of Computer Science and Software Engineering
University of Melbourne, Vic 3010, Australia
email: pcbl@cs.mu.oz.au

Abstract

This paper investigates the application of Maximum Entropy Markov Models to semantic role labelling. Syntactic chunks are labelled according to the semantic role they fill for sentence verb predicates. The model is trained on the subset of Propbank data provided for the Conference on Computational Natural Language Learning 2004. Good precision is achieved, which is of key importance for information extraction from large corpora containing redundant data, and for generalising systems beyond task specific, hand coded template methods.

1 Introduction

In recent years, much progress has been made in the fields of information extraction and question answering. Research systems developed for conference competitions have graduated into the commercial world in such applications as air travel information and booking, call handling, and banking. Although these systems perform well on their chosen tasks, they are generally based on a frame–and–slot approach. This approach uses application–dependent frames defined for propositions and then attempts to fill slots from words surrounding the proposition that triggered the frame. For example, in a financial system we may be interested in extracting company mergers from newswires. We could define a frame for the verb stem *merge*, and slots in that frame for the companies involved in the merger. Although this could be effective for the chosen domain, each time we want to develop a system for a new domain we need to start from scratch. In order to build broad coverage systems capable of generalising, we need a way of defining and labelling propositions and their arguments that is not tied to a particular application.

Systems have been developed to address the task of semantic role labelling (SRL) using a variety of machine learning techniques and features ranging from simple lexical information to those derived from parse trees. These semantic roles identify arguments of verb predicates and can be general, such as *agent*, *theme*, or verb specific, such as *A0*, *A1*.

```
<!DOCTYPE frameset SYSTEM "frameset.dtd">
<frameset>
<predicate lemma="begin">
<roleset id="begin.01" name="start" vncls="55.1">
<roles>
  <role descr="Agent" n="0">
    <vnrole vncls="55.1" vntheta="Agent"/></role>
  <role descr="Theme(-Creation)" n="1">
    <vnrole vncls="55.1" vntheta="Theme"/></role>
  <role descr="Instrument" n="2"/>
</roles>
```

Figure 1: Propbank XML frame for *begin*

This paper presents a discriminative Markov model for the semantic role labelling task proposed for the Conference on Computational Natural Language Learning 2004 (CoNLL). Firstly, recent work on SRL is examined and the data available for this task is presented. A discriminative maximum entropy framework is defined and a SRL model described. This model trains maximum entropy classifiers for each state in the Markov model, representing the probability distribution for transitions from that state for a given observation feature vector. This approach combines the advantages of transition sequence based models with the ability of maximum entropy classifiers to handle a diverse range of overlapping features. The results obtained achieve precision comparable to the best performing support vector model, while requiring significantly less time to train. However, recall is not as high as other approaches and provides an area for further improvement.

2 Background

Many researchers have tackled semantic role labelling. Traditional parsing systems have performed tasks that incorporate a level of semantic labelling and many information extraction systems attempt to solve the labelling problem for a single,

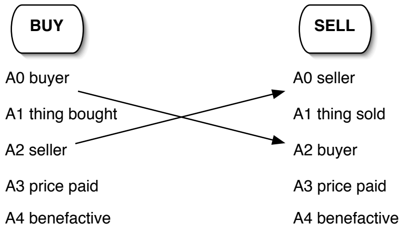

Figure 2: Propbank frames for two related predicates

or small group of propositions, e.g. the biomedical domain. Recently, the development of semantically labelled corpora has led to a number of statistical systems being developed.

Following the pattern of other areas of NLP, it was not until the development of projects to systematically label propositions and their arguments in a corpus, that researchers began to attack the problem of developing generalised statistical systems for SRL. FrameNet was one of the first such projects and aimed to create a hierarchy of semantic frames describing predicates and the roles they accept. Sentences from the British National Corpus (BNC) where annotated with roles derived from these frames.

Gildea and Jurafsky (2002) used FrameNet to train networks of statistical classifiers based on a variety of lexical and syntactic features. They used the statistical parser of Collins (1997) to create the parse trees upon which their syntactic features were based. As the BNC doesn't contain gold standard parse trees, they were unable to quantify the affect of an imperfect parse on the classifiers. However, this work served to highlight the two distinct tasks in SRL: segmenting argument constituents and identifying their semantic role. On the full task of segmenting and labelling arguments, Gildea and Jurafsky achieved a performance of 65% recall and 61% precision, but it must be noted that their system assumes knowledge of the proposition arguments being labelled. Fleischman et al. (2003) applied maximum entropy techniques to the problem defined by Gildea and Jurafsky and achieved a slight increase in performance. This work cast the labelling task as one of tagging, using a maximum entropy formulation over eleven feature sets, and the Viterbi algorithm to search for the best tag sequence. Thompson et al. (2003) applied a generative model to labelling semantic roles in FrameNet data which was somewhat similar to a hidden Markov model (HMM) approach. Their model calculated the probability that for a particular frame a role sequence would generate an observed constituent sequence and assumed that roles correspond to sentence constituents. On their described task they record approximately 70% accuracy, but it is not clear how their system could be applied to the case of general SRL given a raw sentence with no additional information.

The Proposition Bank is a project to add propositional annotations to the Penn Treebank corpus (Kingsbury et al., 2002). In comparison to FrameBank, Propbank labels roles with a generic set of tags: *A0*, *A1*, *A2* etc. Frame entries map these generic labels to proposition specific semantic roles, such as theme and agent (figure 1). Although the semantic roles are not strictly comparable across propositions, annotators were instructed to be consistent in naming and numbering semantic related verb roles (figure 2). The more statistically representative approach and increased coverage, as well as the availability of the treebank parse trees, has made Propbank the most popular resource for current SRL research. Pradhan et al. (2003) developed a support vector machine that chunked semantic arguments from sentences in the Propbank corpus. Their system used similar features as previous systems, including parse trees, and explored using both word-by-word and chunk-by-chunk instances for the classifier. Of interest in this work is an investigation into the effect of not using features derived from a parse tree and relying only on shallow syntactic information, they found that this reduced performance by around 20%.

In an attempt to motivate the development of SRL systems that have direct application to current problems in information extraction and question answering, CoNLL made SRL the topic of their 2004 shared task (Carreras and Marquez, 2004). The challenge was to come up with "machine learning strategies which address the SRL problem on the basis of only partial syntactic information, avoiding the use of full parsers and external lexico-semantic knowledge bases," thus recognising the importance of SRL techniques that are fast and domain independent. The data provided for the conference was based on Propbank and didn't contain parse trees. Of the systems developed for CoNLL, two used maximum entropy techniques. Both these systems used a single instance–by–instance maximum entropy classifier. Lim et al. (2004) achieved the highest performance of the two with an F score of 64.76 on the test data. In this paper we aim to improve precision over previous maximum entropy techniques by optimising the whole sentence tag sequence and thus reducing sequence errors such as repeated arguments and unlikely argument orderings.

3 Data & Evaluation

The CoNLL shared task supplied training, development and testing data created from Propbank annotations on six sections of the Wall Street Journal component of the Penn Treebank. The standard semantic labels from Propbank were used:

Verb specific arguments Arguments with a specific semantic meaning for a verb are labelled *A0-A5*. The semantics of the roles corresponding to these numbered arguments are defined in the Propbank frame for the predicate verb, but in general *A0* maps to agent and *A1* to patient or theme.

Adjunctive arguments General arguments that any verb may take. These include *AM-LOC*, for locative and *AM-TMP*, for temporal. For the complete list see (Carreras and Marquez, 2004).

Argument references Predicate arguments that reference other predicate arguments. Labelled *R-A?* with the argument referenced as the suffix, eg. *R-A1* is a reference to the *A1* defined elsewhere.

Predicate verb The predicate verb that defines the proposition being labelled is tagged *V*.

The data contained annotations of part-of-speech (PoS), base-phrase chunks, clause embedding and named entities. An example of the data is shown in figure 3. For each of the target verbs a column is provided with the argument labelling for that verb.

PoS and base–phrase chunks are annotated in IOB2(inside, outside, begin) format (Ramshaw and Marcus, 1994) and don't allow embedding. Clauses and arguments are annotated in begin, end format, with clauses allowing overlapping and arguments not. So *(A0** represents the start of the *A0* argument and **A0)* represents the end, while *(A0*A0)* labels an argument spanning a single word.

CoNLL also developed the evaluation script *srl-eval.pl* that ranks systems on the standard criteria of *precision*, *recall* and F1 score.

4 Maximum Entropy Markov Models

The system described in this paper is based on a discriminative Markov model, allowing both the optimisation of the tag sequence and the incorporation of multiple features over observations. A limitation of HMMs is that it is hard to extend them to allow multiple features of observations, rather than atomic observations themselves. An alternative to the HMM was proposed by McCallum et al. (2000) in which the transition and observation probability matrices are replaced by maximum entropy classifiers for each state. These classifiers encode the probability distribution $P_{s'}(s|o)$, the probability of making the transition to s from s' and observing o.

4.1 Conditional Exponential Transition Model

The maximum entropy framework, as presented by Berger et al. (1996), aims to "model all that is known and assume nothing about that which is unknown." This is achieved by choosing the model that fits all the constraints expressed by the training data and is the most uniform, i.e. the one with the highest entropy.

Many classification tasks are most naturally handled by representing the instance to be classified as a vector of features. By combining the state and observation transition functions into a single maximum entropy model for each state we can condition the tag sequence assigned to a sentence on such things as part–of–speech tags, phrasal tags, predicate verbs, etc.

For this work we represent features as binary functions of two arguments, $f_i(o, s)$, were o is the observation and s is the possible next state. In order to encode properties of instances which are not binary, such as part–of–speech, a binary feature function is defined for each possible value of the property, i.e. $f_0(o, s)$ is true *iff* o contains the tag *NN* and a transition to s is observed, $f_1(o, s)$ is true *iff* o contains the tag *VB* and a transition to s is observed, and so on.

Given our set of feature functions, and a set of labelled training instances, we can formulate the following constraint equation:

$$expected[f_i] = empirical[f_i] \tag{1}$$

i.e. we should aim to have the expected value for the feature function i in the predicted distribution equal to its average on the empirical training sequence. The maximum entropy distribution is a conditional exponential model of the form:

$$P_{s'}(s|o) = \frac{1}{Z(o, s')} \exp(\sum_i \lambda_i f_i(o, s)) \tag{2}$$

where λ_i are the feature weights that need to be estimated from the training data, and $Z(o, s')$ is a normalisation factor to ensure P is a probability distribution.

Accepted after full paper review

```
That             DT    B-NP      (S*         -               (A0*               *
settlement       NN    I-NP        *         -                  *A0)            *
represented      VBD   B-VP        *         represent        (V*V)             *
the              DT    B-NP        *         -               (A1*               *
first            JJ    I-NP        *         -                  *               *
time             NN    I-NP        *         -                  *               *
shareholders     NNS   B-NP      (S*         -                  *            (A2*A2)
were             VBD   B-VP        *         -                  *               *
granted          VBN   I-VP        *         grant              *             (V*V)
a                DT    B-NP        *         -                  *            (A1*
major            JJ    I-NP        *         -                  *               *
payment          NN    I-NP        *         -                  *               *A1)
in               IN    B-PP        *         -                  *        (AM-LOC*
a                DT    B-NP        *         -                  *               *
greenmail        NN    I-NP        *         -                  *               *
case             NN    I-NP        *S)       -                  *A1)            *AM-LOC)
.                .     O           *S)       -                  *               *
```

Figure 3: CoNLL data format

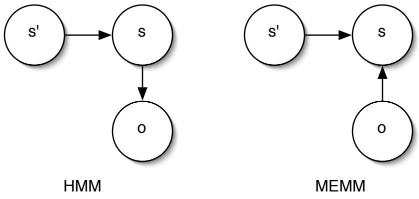

Figure 4: In a MEMM states are conditioned on the previous state and the observation.

4.2 Training and Evaluating the Model

In order to train the MEMM we first need to split the global training data into subsets which will be used to train each individual maximum entropy model. The list of training instances for each state represents the transitions and observations made from that state in the training data. Once all the training data has been processed into sub training lists, the Generalized Iterative Scaling (GIS) (Darroch and Ratcliff, 1972) algorithm is used to train each maximum entropy classifier.

A requirement of natural language systems, especially those based on tagging sentences, is that the sequence of classifications produced by a model should be coherent, this is what the HMM approach and the Viterbi dynamic programming algorithm facilitate. The recursive Viterbi step for the MEMM is defined as:

$$\delta_t(s) = \max_{0 \leq s \leq N} [\delta_{t-1}(s') \times P_{s'}(s|o_t)] \qquad (3)$$

where $\delta_t(s)$ is the probability of seeing the observation sequence up until time t and being in state s, having followed the most probable sequence of state transitions into s. Figure 4 illustrates that both the previous state and the observation determine the next state probability distribution.

4.3 Limitations

The MEMM approach has limitations that must be kept in mind for any implementation. By splitting the training data on the basis of state transitions we are removing the ability of the model to make global generalisations over particular semantic role properties. For example, the properties that mark the start of a particular role will be dispersed among all the states that transition into that role. Therefore the fact that a role is often realised as a prepositional phrase with its first word *on* will be diluted in the split training data. As an added problem, some roles will only appear very rarely in the training data and thus their state transition functions will need to be estimated from a very small amount of data, without the ability to represent general role properties represented in the global data. In order to partially address the problem of data sparseness, a gaussian prior is used to smooth the individual transition functions which have less than 10,000 [1] supporting training examples.

Lafferty et al. (2001) identified the label bias problem as a potential concern for MEMMs. If a particular state has a low–entropy next state distribution, with the extreme case being a single next state, then the observation that the transition is conditioned on will effectively be ignored. Thus the Viterbi path will be biased towards state transitions with low entropy that may be supported by very little training data, over other transitions which are much more supported in the training data. As the MEMM developed in this paper was trained assuming a fully connected initial structure, the transition functions were unlikely to contain a single next state with a nonzero probability. Thus, it was assumed that the label bias problem had little effect, but this could be verified experimentally in further work.

[1] Determined experimentally. A discussion of smoothing maximum entropy models can be found in (Chen and Rosenfeld, 1999).

5 A MEMM for SRL

Section 3 described the framing of SRL as a tagging task. As MEMMs allow the determination of an optimum tagging sequence for a sentence, and allow the modelling of the data as multiple features, it is of interest to investigate their performance on the CoNLL SRL labelling task.

The MEMM described in this paper is a phrase-by-phrase model that maps argument labels to states, and feature vectors over phrases to observations. The original data format displayed in figure 3 was mapped into a phrase based format as shown in figure 5 by converting role boundaries to IOB2 representation (Ramshaw and Marcus, 1994) and collapsing each phrase into a single instance represented by its head word.

Tagging phrase–by–phrase leads to the loss of some information as it is no longer possible to represent roles that have sub-phrasal boundaries. However, as such roles are rare, or easy to handle with post-processing, the loss is acceptable. The advantage of this representation is that it compresses the data and thus decreases the processing required by the model.

A further verb specific compression is performed on the data clauses. It was observed that phrases in clauses below the target verb clause participate completely in roles, therefore these clauses were collapsed with only the first phrase retained for lexical and syntactic information, as shown in figures 6 and 7.

The following features were calculated for each of the phrase/clause instances presented to the model as an observation vector:

Syntactic features Phrase type, head word, head PoS were used. Also, for context, the two preceding and following phrase types and head PoS.

Clause delta The difference of the clause depth of the instance phrase and the clause containing the target verb. If the phrase is lower in the clause tree than the target verb, this value is negative.

Predicate clause A boolean feature that is *true* if the instance clause depth is equal to the target verb phrase depth, *false* otherwise.

Position relative to the verb A feature to indicate whether the phrase instance is before, in, or after the target verb phrase.

Target verb stem The stemmed target verb, as supplied in the CoNLL input data.

Predicate verb suffix Regular expression suffix matching is performed on the target verb. Suffixes matched are: *ing*, *ogy*, *ed*, *s*, *ly*, *ion* and *ies*.

Most occured frame The frame of a verb is defined as the role sequence that it appears with in a sentence, excluding adjunctive roles, for example: *(A0, V, A1, A2)*. A frequency distribution of frames that a target verb appears in is created from the training data, and the most frequent frame for a particular target verb presented as a feature. This aims to represent a verbs preferred sentential structure.

Number of NPs from the target verb A count of the number of NP chunks between the instance phrase and the target verb phrase. If the instance is after the verb, this is a negative count.

Number of base-phrases to the target verb A count of the number of base-phrase chunks between the instance phrase and the target verb phrase. If the instance is after the verb, this value is negative.

Target verb voice A heuristic is used to estimate the voice of the verb as either *active* or *passive*. If the verb phrase contains a form of “to be” and the verb is not gerundive, it is labelled *passive*, otherwise it’s labelled *active*.

Prepositional head All noun phrases following a prepositional phrase used a feature encoding the prepositional phrase’s head word. This improved the handling of adjunctive arguments realised as prepositional phrases.

Clause As described previously, clauses below the verb are compressed to a single instance. If this instance phrase is a compressed clause, this feature is *true*.

Feature pairs The following features were paired to provide an indication of dependency: *verb stem + head*, *verb voice + position*, *verb stem + phrase type* and *verb stem + phrase path*.

6 Implementation

The model was implemented in the Python scripting language using the Natural Language Toolkit (NLTK) (Bird and Loper, 2004). The system was implemented as a pipeline of processes:

1. First the training data was tokenized and global statistics calculated, such as verb frames etc.

```
settlement      NN    NP      -            B-A0       O
represented     VBD   VP      represent    B-V        O
time            NN    NP      -            B-A1       O
shareholders    NNS   NP      -            I-A1       B-A2
granted         VBN   VP      grant        I-A1       B-V
payment         NN    NP      -            I-A1       B-A1
in              IN    PP      -            I-A1       B-AM-LOC
case            NN    NP      -            I-A1       I-AM-LOC
.               .     O       -            O          O
```

Figure 5: Phrase-by-phrase compressed data format

```
settlement      NN    NP      -            B-A0
represented     VBD   VP      represent    B-V
time            NN    NP      -            B-A1
shareholders    NNS   NP      -            I-A1   (a compressed clause)
.               .     O       -            O
```

Figure 6: Data representation with a clause compressed for the verb *represent*

2. The data was then separated into phrase/clause chunks and the features determined. It was then in the form of training instances, an instance corresponding to a sentence and target verb pair, that could be used as training data.

3. Individual state transition function training lists were then created as described in section 4.2

4. The GIS algorithm with a Gaussian prior was then used to train each state transition function

Testing instances were labelled by the model using the Viterbi algorithm described in section 4.2 to determine the optimal tagging. A post processing step was used to tag *AM-MOD* and *AM-NEG* adjuncts as these occur at a lexical level and thus can't be correctly classified by a phrase-by-phrase system. All words in the target verb phrase that had PoS tags *MD* (modal auxiliary) were tagged as *AM-MOD*, while occurrences of the words *n't* or *not* were tagged as *AM-NEG*.

Table 1 shows the output of the *srl-eval.pl* script when the model was used to tag the test data. The overall F_1 score achieved by the system was 59.09. This score corresponds to 24% of the sentences being perfectly labelled.

Due to the large amount of time required to tag the development and test sets using a full set of role tags, the tagging was performed with a reduced set of tags. The tags used were: *A0*, *A1*, *A2*, *A3*, *A4*, *A5*, *R-A0*, *R-A1*, *R-A2*, *C-A1*, *AM-TMP*, *R-AM-TMP*, *AM-ADV*, *AM-LOC*, *AM-MNR*, *AM-MOD*, *AM-DIR*, *AM-NEG*, *AM-DIS*, *AM-CAU*, *AM-EXT*, *AM-PNC*, and *V*. This accounts for some of the reduced performance on the test set as the the reduced tag set was chosen on the basis of the development data, and thus had less coverage on the test data.

7 Discussion

The motivation for using a MEMM approach was to produce more coherent output by finding the optimal Markov sequence, and this is borne out in the results. As is to be expected when not using a parse, prepositional phrase attachment and other syntactic ambiguity, such as propositions late in long sentences, contribute to a number of errors.

Figure 8 shows two sentences that were correctly tagged by the system, and three that contained errors. We can see from the correct sentence that the model is capable of recognising some of the more complex roles such as *discourse (DIS)* and *purpose (PNC)*. In the third sentence it is apparent that the system has not recognised *January* as temporal, instead labelling the preposition as *locative (LOC)*. This is a common error, *LOC* and *TMP* arguments are both frequent and occur in similar prepositional syntax and the model has not been able to capture the lexical distinctions between the two. Sentences four and five show that the system struggles with the semantic ambiguity of the *LOC* role. In four, the gold standard identifies *'in the Reagan administration'* as the location where the defense secretary served, whereas the model tags the preposition as part of the theme of *served*, a tagging which seems quite reasonable. The opposite occurs in sentence five where the model has tagged *'in real estate'* as a location. In this case the gold standard is more semantically consistent. Overall these errors support the argument that even when the model errs, the output is still often consistent.

The CoNLL data requires the labelling of predicate verbs that occur within noun phrases. The model developed generally fails to handle these predicates correctly. This is to be expected as these predicates have a completely different structure and behavior to the standard verb phrase predicates and

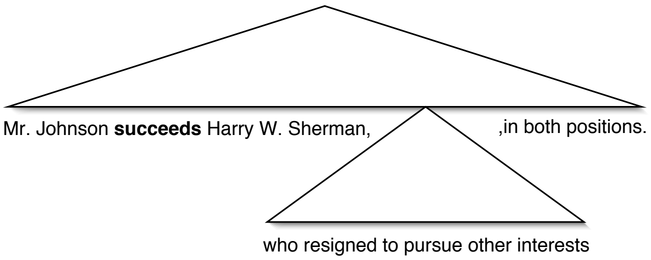
Mr. Johnson **succeeds** Harry W. Sherman,
,in both positions.
who resigned to pursue other interests

Figure 7: A clause tree showing a clause that would be collapsed to a single instance

1. [Of course]$_{DIS}$, [Mr. Wolf, 48 years old,]$_{A0}$ [has]$_{V}$ [some savings]$_{A1}$.
2. Mr Johnsom succeeds [Harry W. Sherman]$_{A0}$, [who]$_{R-A0}$ [resigned]$_{V}$ [to pursue other interests]$_{PNC}$, in both positions.
3. ([In January]$_{LOC}$)^{TMP}, [he]$_{A0}$ [accepted]$_{V}$ [the position ...]$_{A1}$.
4. [Mr. Carlucci, 59 years old,]$_{A0}$ [served]$_{V}$ as [defense secretary (in the Reagan administration)^{LOC}]$_{A1}$.
5. [Balcor]$_{A0}$, [which]$_{R-A0}$ [has]$_{V}$ ([interests]$_{A1}$[in real estate]$_{LOC}$)^{A1}, said the position is newly created.

Figure 8: Example model output. Square brackets indicate tags applied by the system, while round brackets and indicate the tagging from the gold standard

	Precision	Recall	$F_{\beta=1}$
Overall	71.29%	50.45%	59.09
A0	84.02%	66.11%	74.00
A1	65.94%	51.81%	58.03
A2	58.96%	42.92%	49.68
A3	65.15%	28.86%	40.00
A4	70.00%	42.00%	52.50
A5	0.00%	0.00%	0.00
AM-ADV	46.75%	11.76%	18.80
AM-CAU	66.67%	4.08%	7.69
AM-DIR	45.16%	28.00%	34.57
AM-DIS	66.99%	32.39%	43.67
AM-EXT	70.00%	50.00%	58.33
AM-LOC	37.41%	22.81%	28.34
AM-MNR	47.96%	18.43%	26.63
AM-MOD	87.57%	92.26%	89.86
AM-NEG	82.96%	88.19%	85.50
AM-PNC	35.29%	7.06%	11.76
AM-PRD	0.00%	0.00%	0.00
AM-TMP	61.23%	26.64%	37.13
R-A0	90.65%	61.01%	72.93
R-A1	73.91%	48.57%	58.62
R-A2	75.00%	33.33%	46.15
R-A3	0.00%	0.00%	0.00
R-AM-LOC	0.00%	0.00%	0.00
R-AM-MNR	0.00%	0.00%	0.00
R-AM-PNC	0.00%	0.00%	0.00
R-AM-TMP	100.00%	7.14%	13.33
V	96.63%	95.69%	96.16

Table 1: Model performance on the test set

are also in the minority in the training data. A separately trained model could solve this problem.

There are a number of possible improvements to the MEMM model. Prepositional phrase attachment could be handled explicitly in a subsystem. Each state transition function could induce the n most informative features and discard the rest, enabling the model to be more able to avoid over–training problems. Another obvious path to explore would be the application of conditional random fields (Lafferty et al., 2001) to the SRL task. These models have the advantage of solving the label bias problem, as well as allowing more flexibility in modelling, and by using an MEMM for initial weights they converge more quickly.

The advantage of the approach described lies in the increase in precision and the improved coherence of the sentence tag sequence. However, the nature of the SRL task as one that requires both local phrase level feature information and global sentence information, would indicate that both need to be modelled for maximal performance. Most current approaches, including the approach presented in this paper, use word or phrase level classifiers and then try to capture sentence level information through features such as phrase paths and counts.

The evaluation metrics chosen for the CoNLL shared task appear to be brittle, giving no recognition to systems that come close to predicting arguments but fail to find exact matches. The main motivation for SRL techniques is to improve information extraction systems, but it is possible that systems that are able to accurately label the head–words that fill roles of predicate verbs, but miss prepositions and additional noun-phrases, would be very useful also. This is less of a concern for systems employing parse trees as phrase attachment information is known.

8 Conclusion

A maximum entropy Markov model was developed for the SRL task defined at CoNLL 2004. The model managed good performance on the test data, achieving a precision of 71.29 and an F_1 score of 59.09. This result suggests that discriminative sequence models are worth further investigation for the semantic role labelling task. Many errors in tagging made by the model can be attributed to lack of information about the syntactic relationships among phrase chunks, providing an argument for the use of full parse trees when labelling semantic roles.

Acknowledgements

The author would like to thank his research supervisor Steven Bird for his comments and feedback on this work and also the reviewers for their helpful comments. The research undertaken for this paper was supported by an Australian Postgraduate Award scholarship.

References

A. Berger, S. Della Pietra, and V. Della Pietra 1996 A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

S. Bird and E. Loper 2004 NLTK: The Natural Language Toolkit *Proceedings of the ACL demonstration session*, Barcelona, Spain.

X. Carreras and L. Marquez. 2004. Introduction to the CoNLL–2004 Shared Task: Semantic Role Labeling *Proceedings of CoNLL–2004 Shared Task*.

S. Chen and R. Rosenfeld 1999 A Gaussian prior for smoothing maximum entropy models. Technical Report CMUCS-99-108, Carnegie Mellon University.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing *Proceedings of 35th Annual Meeting of the ACL*, Madrid, Spain.

J. Darroch and D. Ratcliff 1972 Generalized Iterative Scaling for Log-linear Models. *Annals of Mathematical Statistics*, no. 43, 1470–1480.

M. Fleischman, N. Kwon and E. Hovy. 2003. Maximum Entropy Models for FrameNet Classification *Proceedings of EMNLP–2003*.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles *Computational Linguistics*, 28(3):245–288.

P. Kingsbury, M. Palmer and M. Marcus. 2002. Adding semantic annotation to the Penn Treebank *Proceedings of the Human Language Technology Conference*, San Diego, CA.

J. Lafferty, A. McCallum, and F. Pereira. 2001 Conditional random fields: probabilistic models for segmenting and labeling sequence data. *Proceedings of the International Conference on Machine Learning*.

J. Lim, Y. Hwang, S. Park, and H. Rim. 2004 Semantic role labeling using maximum entropy model. *Proceedings of CoNLL-2004*.

A. McCallum, D. Freitag and F. Pereira 2000 Maximum entropy Markov models for information extraction and segmentation. *Proceedings of ICML 2000* , 591–598. Stanford, California.

S. Pradhan, K. Hacioglu, W. Ward, J. Martin and D. Jurafsky Oct 2003 TR-CSLR-2003-03: Shallow semantic parsing using support vector machines *Center for Spoken Language Research, University of Colorado*.

L. Ramshaw and M. Marcus 1994 Text Chunking Transformation Based Learning *Proceedings of the 3rd ACL Workshop on Very Large Corpora*.

C. Thompson, R. Levy and C. Manning. 2003. A Generative Model for Semantic Role Labeling *Proceedings of ECML 2003*.