Semantic Role Labeling with Boosting, SVMs, Maximum Entropy, SNOW, and Decision Lists

Grace NGAI[†1], **Dekai WU**[‡2]
Marine CARPUAT[‡], **Chi-Shing WANG**[†], **Chi-Yung WANG**[†]

[†] Dept. of Computing
HK Polytechnic University
Hong Kong

[‡] HKUST, Dept of Computer Science
Human Language Technology Center
Hong Kong

csgngai@polyu.edu.hk, dekai@cs.ust.hk
marine@cs.ust.hk, wcsshing@netvigator.com, cscywang@comp.polyu.edu.hk

Abstract

This paper describes the HKPolyU-HKUST systems which were entered into the Semantic Role Labeling task in Senseval-3. Results show that these systems, which are based upon common machine learning algorithms, all manage to achieve good performances on the non-restricted Semantic Role Labeling task.

1 Introduction

This paper describes the HKPolyU-HKUST systems which participated in the Senseval-3 Semantic Role Labeling task. The systems represent a diverse array of machine learning algorithms, from decision lists to SVMs to Winnow-type networks.

Semantic Role Labeling (SRL) is a task that has recently received a lot of attention in the NLP community. The SRL task in Senseval-3 used the Framenet (Baker et al., 1998) corpus: given a sentence instance from the corpus, a system's job would be to identify the phrase constituents and their corresponding role.

The Senseval-3 task was divided into restricted and non-restricted subtasks. In the non-restricted subtask, any and all of the gold standard annotations contained in the FrameNet corpus could be used. Since this includes information on the boundaries of the parse constituents which correspond to some frame element, this effectively maps the SRL task to that of a role-labeling classification task: given a constituent parse, identify the frame element that it belongs to.

Due to the lack of time and resources, we chose to participate only in the non-restricted subtask. This enabled our systems to take the classification approach mentioned in the previous paragraph.

[1]The author would like to thank the Hong Kong Polytechnic University for supporting this research in part through research grants A-PE37 and 4-Z03S.

[2]The author would like to thank the Hong Kong Research Grants Council (RGC) for supporting this research in part through research grants RGC6083/99E, RGC6256/00E, and DAG03/04.EG09.

2 Experimental Features

This section describes the features that were used for the SRL task. Since the non-restricted SRL task is essentially a classification task, each parse constituent that was known to correspond to a frame element was considered to be a sample.

The features that we used for each sample have been previously shown to be helpful for the SRL task (Gildea and Jurafsky, 2002). Some of these features can be obtained directly from the Framenet annotations:

- The name of the frame.
- The lexical unit of the sentence — i.e. the lexical identity of the target word in the sentence.
- The general part-of-speech tag of the target word.
- The "phrase type" of the constituent — i.e. the syntactic category (e.g. NP, VP) that the constituent falls into.
- The "grammatical function" (e.g. subject, object, modifier, etc) of the constituent, with respect to the target word.
- The position (e.g. before, after) of the constituent, with respect to the target word.

In addition to the above features, we also extracted a set of features which required the use of some statistical NLP tools:

- Transitivity and voice of the target word — The sentence was first part-of-speech tagged and chunked with the fnTBL transformation-based learning tools (Ngai and Florian, 2001). Simple heuristics were then used to deduce the transitivity voice of the target word.
- Head word (and its part-of-speech tag) of the constituent — After POS tagging, a syntactic parser (Collins, 1997) was then used to obtain the parse tree for the sentence. The head word (and the POS tag of the head word) of

the syntactic parse constituent whose span corresponded most closely to the candidate constituent was then assumed to be the head word of the candidate constituent.

The resulting training data set consisted of 51,366 constituent samples with a total of 151 frame element types. These ranged from “Descriptor” (3520 constituents) to “Baggage” and “Carrier” (1 constituent each). This training data was randomly partitioned into a 80/20 “development training” and “validation” set.

3 Methodology

The previous section described the features that were extracted for each constituent. This section will describe the experiment methodology as well as the learning systems used to construct the models.

Our systems had originally been trained on the entire development training (devtrain) set, generating one global model per system. However, on closer examination of the task, it quickly became evident that distinguishing between 151 possible outcomes was a difficult task for any system. It was also not clear that there was going to be a lot of information that could be generalized across frame types. We therefore partitioned the data by frame, so that one model would be trained for each frame. (This was also the approach taken by (Gildea and Jurafsky, 2002).) Some of our individual systems tried both approaches; the results are compared in the following subsections. For comparison purposes, a baseline model was constructed by simply classifying all constituents with the most frequently-seen (in the training set) frame element for the frame.

In total, five individual systems were trained for the SRL task, and four ensemble models were generated by using various combinations of the individual systems. With one exception, all of the individual systems were constructed using off-the-shelf machine learning software. The following subsections describe each system; however, it should be noted that some of the individual systems were not officially entered as competing systems; therefore, their scores are not listed in the final rankings.

3.1 Boosting

The most successful of our individual systems is based on boosting, a powerful machine learning algorithm which has been shown to achieve good results on NLP problems in the past. Our system was constructed around the Boostexter software (Schapire and Singer, 2000), which implements boosting on top of decision stumps (decision trees of one level), and was originally designed for text classification.

Model	Prec.	Recall	Attempted
Single Model	0.891	0.795	89.2%
Frame Separated	0.894	0.798	89.2%
Baseline	0.444	0.396	89.2%

Table 1: Boosting Models: Validation Set Results

Table 1 compares the results of training one single overall boosting model (Single) versus training separate models for each frame (Frame). It can be seen that training frame-specific models produces a small improvement over the single model. The frame-specific model was used in all of the ensemble systems, and was also entered into the competition as an individual system (hkpust-boost).

3.2 Support Vector Machines

The second of our individual systems was based on support vector machines, and implemented using the TinySVM software package (Boser et al., 1992).

Since SVMs are binary classifiers, we used a *one-against-all* method to reduce the SRL task to a binary classification problem. One model is constructed for each possible frame element and the task of the model is to decide, for a given constituent, whether it should be classified with that frame element. Since it is possible for all the binary classifiers to decide on “NOT-<*element*>”, the model is effectively allowed to pass on samples that it is not confident about. This results in a very precise model, but unfortunately at a significant hit to recall.

A number of kernel parameter settings were investigated, and the best performance was achieved with a polynomial kernel of degree 4. The rest of the parameters were left at the default values. Table 2 shows the results of the best SVM model on the validation set. This model participated in the all of the ensemble systems, and was also entered into the competition as an individual system.

System	Prec.	Recall	Attempted
SVM	0.945	0.669	70.8%
Baseline	0.444	0.396	89.2%

Table 2: SVM Models: Validation Set Results

3.3 Maximum Entropy

The third of our individual systems was based on the maximum entropy model, and implemented on top of the YASMET package (Och, 2002).

Our maximum entropy models can be classified into two main approaches. Both approaches used the frame-partitioned data. The more conventional approach (*"multi"*) then trained one model per frame; that model would be responsible for classifying a constituent belonging to that frame with one of several possible frame elements. The second approach (*binary*) used the same approach as the SVM models, and trained one binary *one-against-all* classifier for each frame type-frame element combination. (Unlike the boosting models, a single maximum entropy model could not be trained for all possible frame types and elements, since YASMET crashed on the sheer size of the feature space.)

System	Prec.	Recall	Attempted
multi	0.856	0.764	89.2%
binary	0.956	0.539	56.4%
Baseline	0.444	0.396	89.2%

Table 3: Maximum Entropy Models: Validation Set Results

Table 3 shows the results for the maximum entropy models. As would have been expected, the binary model achieves very high levels of precision, but at considerable expense of recall. Both systems were eventually used in the some of the ensemble models but were not submitted as individual contestants.

3.4 SNOW

The fourth of our individual systems is based on SNOW — Sparse Network Of Winnows (Muńoz et al., 1999).

The development approach for the SNOW models was similar to that of the boosting models. Two main model types were generated: one which generated a single overall model for all the possible frame elements, and one which generated one model per frame type. Due to a bug in the coding which was not discovered until the last minute, however, the results for the frame-separated model were invalidated. The single model system was eventually used in some of the ensemble systems, but not entered as an official contestant. Table 4 shows the results.

System	Prec.	Recall	Attempted
Single Model	0.764	0.682	89.2%
Baseline	0.444	0.396	89.2%

Table 4: SNOW Models: Validation Set Results

3.5 Decision Lists

The final individual system was a decision list implementation contributed from the Swarthmore College team (Wicentowski et al., 2004), which participated in some of the lexical sample tasks.

The Swarthmore team followed the frame-separated approach in building the decision list models. Table 5 shows the result on the validation set. This system participated in some of the final ensemble systems as well as being an official participant (hkpust-swat-dl).

System	Prec.	Recall	Attempted
DL	0.837	0.747	89.2%
Baseline	0.444	0.396	89.2%

Table 5: Decision List Models: Validation Set Results

3.6 Ensemble Systems

Classifier combination, where the results of different models are combined in some way to make a new model, has been well studied in the literature. A successful combined classifier can result in the combined model outperforming the best base models, as the advantages of one model make up for the shortcomings of another.

Classifier combination is most successful when the base models are biased differently. That condition applies to our set of base models, and it was reasonable to make an attempt at combining them.

Since the performances of our systems spanned a large range, we did not want to use a simple majority vote in creating the combined system. Rather, we used a set of heuristics which trusted the most precise systems (the SVM and the binary maximum entropy) when they made a prediction, or a combination of the others when they did not.

Table 6 shows the results of the top-scoring combined systems which were entered as official contestants. As expected, the best of our combined systems outperformed the best base model.

Base Models	Prec.	Recall	Attempted
svm, boosting, maxent (bin)	0.901	0.803	89.2%
svm, boosting, maxent (bin), snow	0.938	0.8	85.2%
svm, boosting, maxent (bin), DL	0.926	0.783	84.6%
svm, boosting, maxent (multi), DL, snow	0.935	0.797	85.2%
Baseline	0.444	0.396	89.2%

Table 6: Combined Models: Validation Set Results

Model	Prec.	Recall	Attempted
svm, boosting, maxent (binary) (hkpolyust-all(a))	0.874	0.867	99.2%
boosting (hkpolyust-boost)	0.859	0.852	0.846%
svm, boosting, maxent (binary), DL (hkpolyust-swat(a))	0.902	0.849	94.1%
svm, boosting, maxent (binary), DL, snow (hkpolyust-swat(b))	0.908	0.846	93.2%
svm, boosting, maxent (multi), DL, snow (hkpolyust-all(b))	0.905	0.846	93.5%
decision list (hkpolyust-swat-dl)	0.819	0.812	99.2%
maxent (multi)	0.827	0.735	88.8%
svm (hkpolyust-svm)	0.926	0.725	76.1%
snow	0.713	0.499	70.0%
maxent (binary)	0.935	0.454	48.6%
Baseline	0.438	0.388	88.6%

Table 7: Test set results for all our official systems, as well as the base models used in the ensemble system.

4 Test Set Results

Table 7 shows the test set results for all systems which participated in some way in the official competition, either as part of a combined system or as an individual contestant.

The top-performing system is the combined system that uses the SVM, boosting and the binary implementation of maximum entropy. Of the individual systems, boosting performs the best, even outperforming 3 of the combined systems. The SVM suffers from its high-precision approach, as does the binary implementation of maximum entropy. The rest of the systems fall somewhere in between.

5 Conclusion

This paper presented the HKPolyU-HKUST systems for the non-restricted Semantic Role Labeling task for Senseval-3. We mapped the task to that of a simple classification task, and used features and systems which were easily extracted and constructed. Our systems achieved good performance on the SRL task, easily beating the baseline.

6 Acknowledgments

The “hkpolyust-swat-*” systems are the result of joint work between our team and Richard Wicentowski’s team at Swarthmore College. The authors would like to express their immense gratitude to the Swarthmore team for providing their decision list system as one of our models.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 86–90, San Francisco, California. Morgan Kaufmann Publishers.

Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the ACL (jointly with the 8th Conference of the EACL)*, Madrid.

Daniel Gildea and Dan Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):256–288.

Marcia Muńoz, Vasin Punyakanok, Dan Roth, and Dav Zimak. 1999. A learning approach to shallow parsing. In *Proceedings of EMNLP-WVLC’99*, pages 168–178, College Park. Association for Computational Linguistics.

G. Ngai and R. Florian. 2001. Transformation-based learning in the fast lane. In *Proceedings of the 39th Conference of the Association for Comp utational Linguistics*, Pittsburgh, PA.

Franz Josef Och. 2002. Yet Another Small Maxent Toolkit: Yasmet. http://www-i6.informatik.rwth-aachen.de/Colleagues/och.

Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Richard Wicentowski, Emily Thomforde, and Adrian Packel. 2004. The Swarthmore College Senseval-3 system. In *Proceedings of Senseval-3*, Barcelona.