Pascal Challenge
The Evaluation of Machine Learning for Information Extraction

Neil Ireson & Fabio Ciravegna
University of Sheffield, Department of Computer Science
{n.ireson, f.ciravegna}@dcs.shef.ac.uk

1 Introduction

If the Semantic Web is to utilise the vast number of documents available on the WWW it requires an effective way to automatically annotate those documents, enabling the extraction of relevant information. The Pascal Challenge on the Evaluation of Machine Learning for Information Extraction provided a common basis on which it assess the relative performance of multifarious machine learning systems. This paper describes the challenge and presents an initial analysis of the results.

2 Data

A corpus of 1100 documents, collected from various sources, comprises of 850 Workshop CFP and 250 Conference CFP. The majority of the documents come from the field of Computer Science, due to the readily available archives, although some other fields (e.g. biomedicine, linguistics) are also represented. The corpus is divided into three sections:

- Training Corpus (400 Workshop CFP): The documents in the training corpus are randomly divided into 4 sets of 100 documents. Each of these sets is further randomly divided into 10 subsets of 10 documents. Each document relates to a workshop held between 1993 and 2000.
- Test Corpus (200 Workshop CFP): The documents in the training corpus relate to workshops held between 2000 and 2005.
- Enrich Corpus (250 Workshop CFP & 250 Conference CFP): The documents in the enrich corpus relate to workshops held between 2000 and 2005 & conferences held between 1997 and 2005.

Thus there is a small temporal overlap between the training and test data. Whilst the enrich data offers documents taken from the same timeframe as the test corpus.

2.1 Annotation

The documents in the training and test corpora were annotated with 11 tags:

	Corpus Frequency	
Annotation Type	Training	Test
workshopname	543	245
workshopacronym	566	243
workshopdate	586	326
workshophomepage	367	215
workshoplocation	457	224
workshoppapersubmissiondate	590	316
workshopnotificationofacceptancedate	391	190
workshopcamerareadycopydate	355	163
conferencename	204	90
conferenceacronym	420	187
conferencehomepage	104	75
Total	4583	2274

2.2 Preprocessor

The data was preprocessed using GATE[1], which provides tokenisation, orthology, POS tagging and named-entity recognition text features.

3 Tasks

For each task participants were encouraged to submit results not only for testing on the test corpus but also for a four-fold cross-validation experiment on the training corpus, with a 300 training, 100 testing document split using the partitions provided.

Task1: Given all the available training documents, learn the textual patterns necessary to extract the annotated information.

Task2a: (Learning Curve) Examine the effect of limited training resources on the learning process by incrementally adding the provided subsets to the training data. Thus there are 9 experiments; for the four-fold cross-validation experiment the training data has 30, 60, 90, 120, 150. 180, 210, 240 and 270 documents and for the test data experiment the training data has 40, 80, 120, 160, 200, 240, 280, 320 and 360 documents.

Task2b: (Active Learning) Examine the effect of selecting which documents to annotate and add to the training data. Given each of the training data subsets used in Task2a, select the next subset to add from the remaining training documents. Thus a comparison of the Task2b and Task2a performance will show the advantage of the active learning strategy.

Task3a: (Enrich Data) To perform either of the above tasks but using the addition 500 unannotated documents. In practice only one participant attempted this task and only to enhance Task1 on the test corpus.

Task3b: (Enrich WWW data) To perform either of the above tasks but using any other (unannotated) documents, such as those found on the WWW. In practice only one participant attempted this task and only to enhance Task1 on the test corpus.

4 Systems

The challenge attracted 11 participants, who submitted 23 systems in total. The following table shows the number of systems which submitted for each task.

	Tasks				
Data	1	2a	2b	3a	3b
4-fold	15	8	4	0	0
Test	20	10	5	1	1

In the following sub-sections each of the participant's systems are briefly described. It can be assumed that, unless otherwise stated, the system utilise all the GATE features.

4.1 Amilcare

The system uses the LP^2 algorithm to induce general rules in two steps: The tagging phase identifies tags using two types of rules; firstly rules consider a left-right context of 5 tokens and secondly "contextual rules" which also consider the presence of other tags. The correction phase learns rules to shift misplaced tags from the mistakes made in tagging the training corpus. The contextual rules are applied in a loop until no new tags are inserted, thus some contextual rules can match tags inserted by other contextual rules. The validation step resolves any tag conflict and ensures coupling between tags. The Active Learning system classifies the documents to be selected then selects the documents which have the number of tags furthest from the expected number. The Enrich WWW system searches the web for documents containing the workshop names found in the training data. Other potential names are then extracted from the documents and included in the training data.

4.2 Bechet

This system is made of two components: Named entity tagger (based on a probabilistic Markov model) that takes a stream of text and associates a label to each word; either one of the entity tags or an empty tag. Low frequency words are replaced by their GATE features. Apply a text classifier (a boosting algorithm of weak classifiers) for each potential entity detected by the tagger, using a context of up to 10 tokens.

4.3 Canisius

The system performs the extraction task using a two-stage approach: Identify relevant sentences (i.e. likely to contain slots) using a BoW representation with an SVM classifier optimized for high recall. Feature selection removes low frequency features and selects the 500 features with the highest information-gain. A token-level classifier, based on the Memory-based tagger (Mbt), is applied to the relevant sentences. This classifier creates separate sub-classifiers for known and unknown words. In addition to the GATE features and those features automatically generated by Mbt, word bi-grams and tri-grams were also used.

4.4 Finn

The system uses an SVM based two-level approach for the learning algorithm, each slot was learnt independently and then combined. A feature window of 4 and a L2 lookahead/lookback of 10 was used, the 5000 most informative features (according to information-gain) were selected. The negative instances were randomly undersampled by 50% to cope with the significantly imbalanced datasets.

4.5 Hachey

The system implements a relatively simple query-by-committee approach using KL-divergence to calculate the disagreement between two maximum entropy classifiers: A conditional Markov model is used to train two different models by applying a feature split into word features (word tokens and word shapes) and non-word features (such as POS and GATE entity types). KL-divergence has been used for active learning to quantify the disagreement of classifiers over the probability distribution of output labels.

4.6 ITC-IRST

The system uses an SVM classifier, each tag was learnt independently and then combined. The system mainly evaluates a technique to filters out uninformative (very frequent) words from texts. The submitted systems vary in the fraction of uninformative words removed, including the case where all the dataset is used.

4.7 Kerloch

The submitted extraction systems are all based on HMMs (with one HMM per slot to be extracted), the most probable state sequence is computed via Viterbi. Each HMM is a 3 state model, plus one junk state. The three states identify the information and its left-right context of 15 tokens; the junk state identifies non-relevant information. system1 uses words only, while system2 uses all the token features. System3 is identical to system2, except that each state is duplicated.

4.8 Sigletos

The system performed voting (using probability estimates) on the predictions of three other IE systems: LP^2, Boosted Wrapper Induction (with a lookahead-L parameter of 3), and a locally developed system. This submission utilised other preprocessing tools so is not directly comparable with the other systems.

4.9 Stanford

Our Task1 system uses a Conditional Random Fields model with features defined across cliques of maximal size 2, trained using limited memory Quasi-Newton optimization. We use the Viterbi algorithm to find the best label sequence given a test document and the trained model. The Task3 submission uses a two-stage labelling process. First we train a Maximum-Entropy Markov Model with a window of size 4 to generate candidate labellings by sampling each token's labelling from the marginal distribution of possible labels given the labelling of the previous tokens. These labellings are translated into scored templates, where a template consists of the map from each target field to a string, and a template's score corresponds to the percentage of samplings that generated it. Each template is then given a top-down score in a second stage, which evaluates the feasibility of this template according to the a date model (which determines if the dates are ordered, temporally, in a way consistent with training data: workshop dates come after submission dates, etc.) and an acronym model (which judges if workshop and conference acronyms correspond to their respective names and URLs). Combining these two scores determines a final score for each template.

4.10 TRex

The system uses an SVM classifier (svmlight), each tag was learnt independently and then combined. The data model uses a left-right context of 6 tokens and only considers the token string, POS and orthograpy features. Before classification information-gain was used to select the 25% of features with the highest information value.

4.11 Yaoyung

The difference between system2 and system3 was that system2 uses an SVM with uneven margins, while system3 uses a Perceptron with uneven margins, system1 is a combination of system2 and system3. The classifiers use a left-right context of 10 tokens (the features exclude POS) to identify tags which are combined using the classifier scores to resolve tag conflict and to ensure tag coupling. The results of system1 were obtained by combining the tags from system2 and system3 and adopting the results of system2 wherever there was any conflict. For the active learning (Task2b) the Gram-Schmidt orthogonalisation algorithm was used to determine a subset of examples which were furthest from each other and were also furthest from another pre-defined subset (if we have one) in the feature space.

5 Results

The following section summarises the results from the Pascal Challenge, the results are available in full at http://tyne.shef.ac.uk/Pascal/results.php.

5.1 Task1

Table 1 shows that systems which performed well on the test corpus had decreases in performance from the cross-validation experiment lower than was generally observed; showing that these systems generalised well. However, the itc-irst systems, whilst performing well, suffer a considerable fall in recall indicating a degree of over-fitting. The yaoyong systems exhibit the same decrease but to a lesser degree, the combined system1 providing the most robust performance.

There is a considerable variation in the ability of all the systems to identify certain slots. The CFP "important dates" being relatively easy, whilst workshop name and location and conference name and homepage are poorly identified. Future improvements will have to concentrate on identifying these "difficult" slots. Unfortunately no statistics were kept for inter-annotator agreement. In retrospect this would have been very useful in comparing the relative performance of the machine learning algorithms against the annotators for each of the slot.

Table 2 shows the best performing systems on individual slots. As can be seen Amilcare achieves the maximum f-measure for most slots, however it performs poorly for workshoplocation and especially workshopname. By examining the documents it can be seen that these slots tend not to be specified by the surrounding text but are determined by their content and position in the document. Further analysis is need to determine why Amilcare fails to provide good performance for these two slots, especially as other systems which perform well overall provide more consistent performance over all the slots.

5.2 Task2a: Learning Curve

The Amilcare systems provided the best overall (FMeasure) performance for all the subsets on both test sets. Amilcare system2, which was only submitted for the 4-fold cross-validation experiment, had the best performance on the smallest 5 subsets which indicates increasing recall is important when there is a low amount of training data. The performance on individual slots for this task has not yet been analysed.

5.3 Task2b: Active Learning

The Amilcare and Hachey systems provided the most significant improvements using active learning. The Amilcare approach tended to provide the best performance for low amounts of training data, however the reasons for the improvements were unclear as they were sometimes due to increased recall at the expense of precision and at other times the opposite. The Hachey active learning system provided reasonably consistent improvements to both recall and precision.

5.4 Task3

Unfortunately only two systems were submitted from the Task3 experiments. The stanford system3 performed significantly worse than their system1. The Amilcare system improved recall and f-measure on the workshopname slot, although overall performance was not changed.

6 Conclusion

From the results of the challenge the Amilcare system is shown to have comparatively high performance for most slots. One major difference between this and the other systems (excluding the finn system) is the use of "contextual rules" which consider the presence of other tags when identifying tags. This strategy might be successfully adopted by the other systems.

It is interesting to note that the systems which provide the best overall performance on the test corpus use different learning algorithms. Future work will examine more fully the various features of the systems to determine the degree to which they influence the ability to successfully extract information.

The (Task2a) learning curve experiment indicates that the balance between precision and recall needs to be considered given the amount of training data available. The good performance of the hachey system shows that the disagreement of classifiers which use different features warrants further investigation for (Task2b) active learning. The enrich (Task3) area of the challenge, the use of unannotated data to aid information extraction, remains largely unexplored.

Table 1: Task1 results for all the systems

for the 4-fold cross-validation and test data experiments

		4-fold X-validation			Test data			% Change		
Participant	System	PRE	REC	FME	PRE	REC	FME	PRE	REC	FME
amilcare	system1	0.843	0.703	0.767	0.829	0.658	0.734	-1.7	-6.4	-4.3
yaoyong	system1	0.702	0.717	0.709	0.708	0.633	0.668	0.9	-11.7	-5.8
stanford	system1	N/A	N/A	N/A	0.731	0.589	0.653	N/A	N/A	N/A
yaoyong	system2	0.741	0.646	0.690	0.780	0.547	0.643	5.1	-15.2	-6.8
yaoyong	system3	0.699	0.683	0.691	0.714	0.533	0.611	2.2	-21.9	-11.6
itc-irst	system2	0.755	0.652	0.700	0.821	0.467	0.595	8.6	-28.4	-15.0
itc-irst	system1	0.748	0.654	0.698	0.812	0.462	0.589	8.5	-29.3	-15.6
itc-irst	system3	0.738	0.643	0.687	0.797	0.440	0.567	8.1	-31.6	-17.5
trex	system2	N/A	N/A	N/A	0.584	0.471	0.521	N/A	N/A	N/A
sigletos	system3	N/A	N/A	N/A	0.631	0.433	0.513	N/A	N/A	N/A
sigletos	system2	N/A	N/A	N/A	0.643	0.426	0.513	N/A	N/A	N/A
sigletos	system1	0.603	0.513	0.555	0.629	0.423	0.506	4.2	-17.5	-8.8
canisius	system1	0.657	0.434	0.523	0.665	0.409	0.506	1.1	-5.8	-3.2
trex	system1	N/A	N/A	N/A	0.588	0.442	0.505	N/A	N/A	N/A
bechet	system2	0.690	0.580	0.630	0.553	0.373	0.446	-19.9	-35.6	-29.3
kerloch	system3	N/A	N/A	N/A	0.373	0.544	0.443	N/A	N/A	N/A
bechet	system1	0.625	0.639	0.632	0.474	0.396	0.431	-24.2	-38.1	-31.7
finn	system1	0.688	0.626	0.656	0.716	0.304	0.427	4.1	-51.4	-34.9
kerloch	system2	0.312	0.659	0.424	0.293	0.578	0.389	-6.1	-12.2	-8.2
kerloch	system1	0.407	0.627	0.494	0.245	0.339	0.285	-39.8	-46.0	-42.4
amilcare	system2	0.768	0.757	0.762	N/A	N/A	N/A	N/A	N/A	N/A

Table 2: Task1 results for individual slots on the test data experiment

Only those systems which provided the highest F-Measure for atleast one slot are shown

			Workshop								Conference		
Participant	System	Score	name	acro	date	home	loca	pape	noti	came	name	acro	home
amilcare	system1	PRE	0.656	**0.887**	0.769	**0.864**	0.621	**0.876**	0.889	0.876	0.792	**0.922**	0.656
		REC	0.241	**0.844**	0.632	0.619	0.402	**0.851**	**0.889**	**0.865**	0.422	**0.888**	0.280
		FME	0.352	**0.865**	0.694	0.721	0.488	**0.864**	**0.889**	**0.870**	**0.551**	**0.905**	**0.393**
yaoyong	system1	PRE	0.629	0.738	0.810	0.656	0.611	0.719	0.867	0.764	0.649	0.619	0.368
		REC	0.539	0.523	0.666	0.870	**0.674**	0.763	0.821	0.736	0.411	0.348	0.093
		FME	0.580	0.612	0.731	**0.748**	0.641	0.740	0.843	0.750	0.503	0.445	0.149
stanford	system1	PRE	0.618	0.806	0.822	0.678	0.737	0.747	0.870	0.777	0.643	0.576	0.389
		REC	**0.576**	0.358	0.693	0.665	0.576	0.680	0.774	0.791	0.400	0.428	0.093
		FME	0.596	0.496	**0.752**	0.671	0.647	0.712	0.819	0.784	0.493	0.491	0.151
yaoyong	system2	PRE	0.713	0.796	0.838	0.734	0.717	0.767	**0.943**	0.845	0.775	0.634	0.455
		REC	0.437	0.481	0.586	0.679	0.612	0.636	0.784	0.669	0.344	0.278	0.067
		FME	0.542	0.600	0.690	0.705	**0.660**	0.696	0.856	0.747	0.477	0.387	0.116
itc-irst	system2	PRE	**0.852**	0.733	0.850	0.672	0.812	0.841	0.921	0.911	0.795	0.667	0.556
		REC	0.539	0.259	0.451	0.419	0.406	0.617	0.795	0.687	0.344	0.235	0.067
		FME	**0.660**	0.383	0.589	0.516	0.542	0.712	0.853	0.783	0.481	0.348	0.119

References

[1] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.