Template Sampling for Leveraging Domain Knowledge in Information Extraction

Christopher Cox, Jamie Nicolson, Jenny Rose Finkel, Christopher Manning, and Pat Langley
Computer Science Department
Stanford University
Stanford, CA 94305
{ditka, nicolson, jrfinkel, mannning}@cs.stanford.edu
langley@csli.stanford.edu

Abstract

We initially describe a feature-rich discriminative Conditional Random Field (*CRF*) model for Information Extraction in the workshop announcements domain, which offers good baseline performance in the PASCAL shared task. We then propose a method for leveraging domain knowledge in Information Extraction tasks, scoring candidate document labellings as one-value-per-field templates according to domain feasibility after generating sample labellings from a trained sequence classifier. Our relational models evaluate these templates according to our intuitions about agreement in the domain: workshop acronyms should resemble their names, workshop dates occur after paper submission dates. These methods see a 5% f-score improvement in fields retrieved when sampling labellings from a Maximum-Entropy Markov Model, however we do not observe improvement over a CRF model. We discuss reasons for this, including the problem of recovering all field instances from a best template, and propose future work in adapting such a model to the CRF, a better standalone system.

1 Introduction

The task of Information Extraction can be conceived as the filling of predefined template slots with likely values in text. To preserve tractability, machine learning IE systems generally label a text represented just as a sequence of tokens and use local context models to govern the selection of good values. Though such models can be effective, we recognize that informative relationships exists between target values that aren't captured in "flat" sequence classifiers. We want to leverage our knowledge of interaction between slot-fillers in a domain to filter or augment the decisions of the local models. In the domain of workshop announcements, we know that relevant dates are generally ordered in a certain way, that exclusive relationships exist between names and acronyms of conferences and their subsidiary workshops, and that names and acronyms of the same thing should "agree". We choose to sample candidate assignments from the sequence model and filter candidate templates according to these intuitions.

2 The Basic Sequence Model

We present as a baseline two "flat" sequence classifiers trained using identical features. The first, and better standalone system, uses as its basic algorithm a Conditional Random Field model (Lafferty et al., 2001) 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. Our features at a token consist of the word, POS tag, and shape of a token as indicated by the GATE preprocessing, the entity type of tokens (Person, Location, Date, etc.) and GATE rules that generated them as indicated by XML bracketing in the data, the order/position of the token in the document, the token's membership in a URL, and conjunctions of these features and those of nearby tokens. We submitted this system for Task 1. The second system uses a conditional Markov model sequence tagger, similar but not identical in features to the one described in (Klein et al., 2003). Token features for this model are exactly as described above, except that they see the previous four class labels instead of just one: a token labelling decision in the CMM therfore considers a significantly larger window of token labellings behind it than the CRF.

3 Sampling

After training a sequence classifier, we walk forward through a document and generate candidate fillers by sampling each token's labelling from the marginal distribution of possible labels given the labelling of the previous tokens. In this way we get a distribution over locally consistent labellings for the whole document. We generate 100 of these. From these we create distributions over two classes of templates: date templates and name templates. Each date template holds one filler value (or "none") for each of the four target dates. Each name template holds one value for each of the six target "name" fields: name, acronym, and webpage for both conference and workshop. In the case where a labelling chooses more than one filler for a particular field, the labels do not correspond to a single template, so we split the labelling into multiple templates that each get partial points that sum to 1. The sum of the scores for templates over all 100 samples represents a distribution as well.

4 Template Scoring

With a set of templates with probabilities given by the local model, we want to generate a *relational score* for each. Each score is meant to correspond to a probability given the domain model (though they are not well-founded).

4.1 Date Model

Each date template is scored according its feasibility in terms of present/absent fields and date ordering.

To begin, each template's score is the probability that its fields are present given a joint distribution over present/absent date fields according to the training data. For example, a template with all 4 fields present receives a higher score (.74) than one with only an acceptance date (.02). Each date field is then mapped to a normalized date using specialized regular expressions. If a tagged sequence is unrecognizable as a date, we discard the template.

As observed in the training data, workshop dates and camera ready copy dates follow submission and acceptance dates, and acceptance date follows submission date. Also, we don't observe that different date fields have the same filler. Accordingly, we penalize out of order dates and matching dates in different fields. We multiply the present/absent prior and the penalties to get our relational score for the date model. The product of this *relational score* and the *local score* for a template is the final score. We choose the date template with the highest final score.

4.2 Acronym and URL Model

We use an acronym model that gives a likelihood score to pairs of names and acronyms (Chang et al., 2002). It uses a variety of heuristics, whose weights are trained by logistic regression on a small hand-built acronym corpus. These measure how well the acronyms fit the names they are meant to abbreviate. For every candidate template, we apply the model to the pairs (*workshop name*, *workshop acronym*) and (*conference name*, *conference acronym*) to derive the likelihood of the template's assignment to those fields. The scores from the model are multiplied into the template probabilities output by the sampler.

Acronym/URL likelihood is then multiplied into the score, instantiating as a factor the probability that the acronym appears in the corresponding URL in the template using empirical probabilities from the training set.

If a template has a blank entry for a name or acronym or URL, the model cannot be applied to it. Since the acronym model score is always less than 1, a template that escapes acronym model scoring would be at an unfair advantage. As with the date model, we calculate a joint distribution over combinations of missing fields, and apply a corresponding prior probability to each template. In the training set, most fields are filled, so templates with missing fields generally have a lower prior probability. This penalizes timid templates that attempt to avoid making a wrong guess by not making any guess at all. As with the date model, we take the product of the *local* and *relational* score for each name template and pick the highest final score.

5 Template to Markup

Once we've chosen our templates, we need to translate template decisions into markup of the initial document. In the case of dates, we mark up any GATE annotated `DATE` entities whose normalized date (according to our regular expressions) matches the template value chosen.

In the case of workshop/conference names, we mark up all text subsequences labelled as candidate

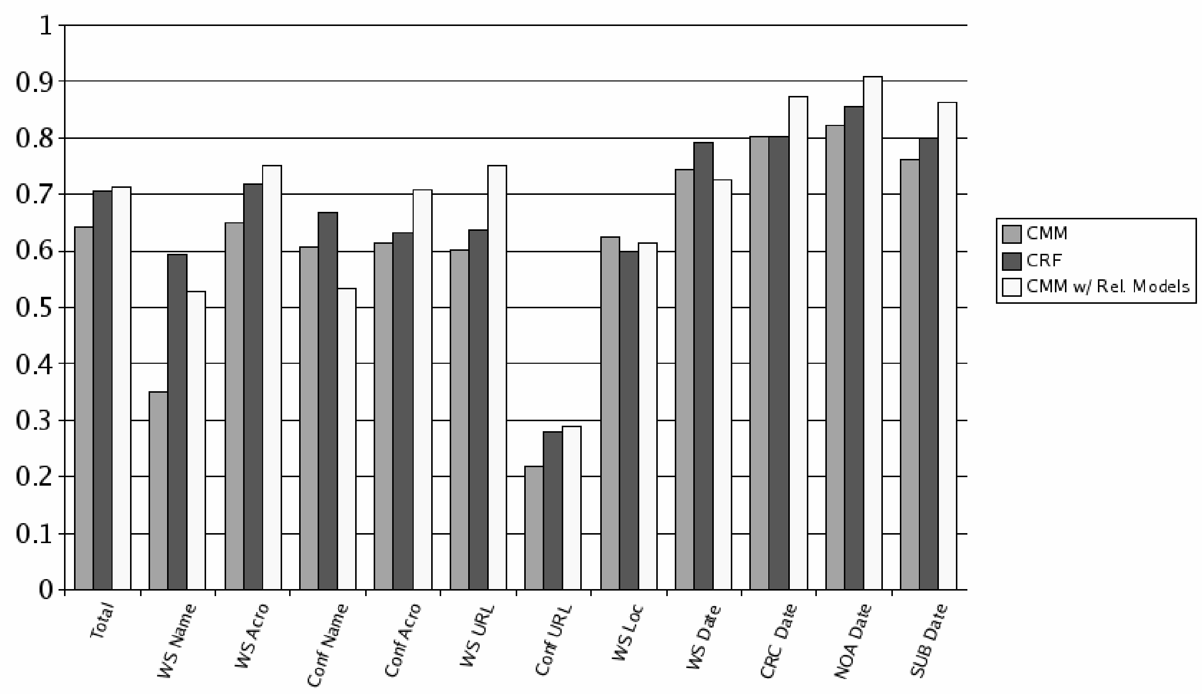

Figure 1: F-Score for Document Subsequences Recovered on Held Out Test Set

workshop/conference names at sampling time that match the chosen template field, where 'matching' here is defined as having exact string similarity minus a tabu list of uninformative name tokens, like "annual" and "international". Accordingly, when a template chooses "The First Annual Conference on Widgets", as a conference name, we'll mark up occurences of "First International Conference on Widgets".

To recover acronyms, we find all acronyms in the document that are candidates in the local model and had the same acronym 'stem' as the template. So if we decided that "ADBIS '99" was the best template filler for *workshop acronym*, we'd tag "ADBIS 1999" and "adbis" if they were found in the text and selected by the local model (at some point) as candidates. Homepages are marked up by finding exact matches of template instances.

6 Results

For the purposes of this report, we train on a set of 300 documents from the original *PASCAL* training set, and test on the remaining 100. We report in Figure 1 the f-score of fully and correctly tagged document subseqences on the held-out set (not the competition test set, as we were unable to compare all 3 systems on it).

We observe that sampling from the CMM and adding relational models marks a substantial improvement over the CMM alone (5% f-score), and insignificant improvement over the CRF alone. We observe (but do not report) substantially poorer performance in sampling from the CRF: the CRF alone performs better when choosing the Viterbi best sequence for labelling. We explain this with the observation that the CRF's low window-size is seriously preventative, and forces a less-informed distribution from which the forward samples are drawn. Sampling forward in the CMM allows for a larger lookback window when drawing a sample, giving us a better local model.

On the *PASCAL* test set, using the official scorer, we observe the CRF performing substantially better then the CMM w/ relational models in overall f-score (.653 vs. .609), where the CMM w/ relational models outperform the CRF per-field f-scores in dates retrieved, but not in other fields. Though we're not entirely clear on why the difference in system performance was substantially larger in the *PASCAL* test set vs. the held-out set, we set about understanding the disparity in performance in general.

7 Discussion

Generating sample templates before relational scoring is helpful in that it generates single filler templates which can be compared and scored in a straightforward way. Though templates are great models for documents with one consistent value per field, they introduce the problem of template to markup conversion, and a series of possibly detrimental assumptions about well-formedness and global consistency of data.

This task highlights an interesting problem: that of mapping best candidate fillers to variant and valid occurences in text. Figure 1 shows that the CRF outperforms the CMM with relational models in three fields: *workshop name*, *conference name*, and *workshop date*. We understand this to be a result of these fields having the most variation inside a document.

As an example, we observe a document where: "Internet Banking and Financial Services", "minitrack on Internet Banking and Financial Services", and "Internet Banking and Financial Services minitrack" are all tagged *workshop name* in the goldstandard. Here the relational model correctly identifies the last of these as a template filler, but does not recover the other two because they don't match. The CRF alone labels all 3 correctly.

These small but crucial deviations represent a larger problem we find regularly in recovering workshop and conference names from a correct template filler, and explain the substantially lower recall (though higher precision) scores of the CMM w/ relational models vs. CRF in name fields. The problem exists with recovery of acronyms as well.

The *workshop date* field is unique in this task in that it can be specified as a range of dates, and as a result is often expressed in different ways within a document: "August 17 - 21, 1998" and "Aug 17, 98" don't refer to the same date, formally speaking, yet they do both represent the workshop date. Again, the relational model doesn't handle this case where the CRF does, and again we see that the problem is mapping back to the document from a correct template filler which does not precisely agree with all correct subsequences.

8 Future Work

In the future, we would like to sample and score templates jointly, as opposed to sampling followed by scoring. One way to do this would be with a random walk along the sequence model, where each step would also incorporate the score from the relational model. If we view the relational model as another sequence model, then we can combine both models into a factored model and use Gibbs sampling, a form of Markov Chain Monte Carlo, to do the random walk. This sort of sampling suits the CRF's structure more closely, in that the marginal distribution over possible labellings at a token includes information about the labellings ofprevious and following tokens. Random (as opposed to forward) walking allows the CRF to generate samples whose local consistency is more constrained.

Constucting such a relational model would make use of many of the intuitions presented here, but would need to take a different form. We could no longer assume one filler value per template, and the model would need to handle a wider distribution of such templates when assigning a score. Despite this, it is certainly conceivable, and we still see good prospects for to further work in joint classification that leverages non-trivial long-distance relationships.

Acknowledgements

This work was carried out under the Edinburgh-Stanford Link programme, funded by Scottish Enterprise.

References

J. Chang, H. Schtze, and R. Altman. 2002. Creating an online dictionary of abbreviations from medline.

D. Klein, J. Smarr, H. Nguyen, and C. D. Manning. 2003. Named entity recognition with character-level models. In *Proceedings of the 7th CoNLL*, pages 180–183.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th ICML*, pages 282–289. Morgan Kaufmann, San Francisco, CA.