Pacific Symposium on Biocomputing 12:257-268(2007)

A STACKED GRAPHICAL MODEL FOR ASSOCIATING SUB-IMAGES WITH SUB-CAPTIONS

ZHENZHEN KOU, WILLIAM W. COHEN, AND ROBERT F. MURPHY

Machine Learning Department, Carnegie Mellon University
5000 Forbes Avenue,
Pittsburgh, PA 15213, USA
E-mail: zkou@andrew.cmu.edu, wcohen@cs.cmu.edu, murphy@cmu.edu

There is extensive interest in mining data from full text. We have built a system called SLIF (for Subcellular Location Image Finder), which extracts information on one particular aspect of biology from a combination of text and images in journal articles. Associating the information from the text and image requires matching sub-figures with the sentences in the text. We introduce a stacked graphical model, a meta-learning scheme to augment a base learner by expanding features based on related instances, to match the labels of sub-figures with labels of sentences. The experimental results show a significant improvement in the matching accuracy of the stacked graphical model (81.3%) as compared with a relational dependency network (70.8%) or the current algorithm in SLIF (64.3%).

1. Introduction

The vast size of the biological literature and the knowledge contained therein makes it essential to organize and summarize pertinent scientific results. Biological literature mining has been increasingly studied to extract information from huge amounts of biological articles [1−3]. Most of the existing IE systems are limited to extracting information only from text. Recently there has been great interest in mining from both text and image. Yu and Lee[4] designed BioEx that analyses abstract sentences to retrieve the image in an article. Rafkind et al[5] explored the classification of general bioscience images into generic categories based on features from both text (image caption) and image. Shatkay et al[6] described a method to obtain features from images to categorize biomedical documents. We have built a system called SLIF[7−8] (for Subcellular Location Image Finder) that extracts information about protein subcellular locations from both text and images. SLIF analyzes *figures* in biological papers, which include both images and captions. In SLIF, a large corpus of articles is fully analyzed

Pacific Symposium on Biocomputing 12:257-268(2007)

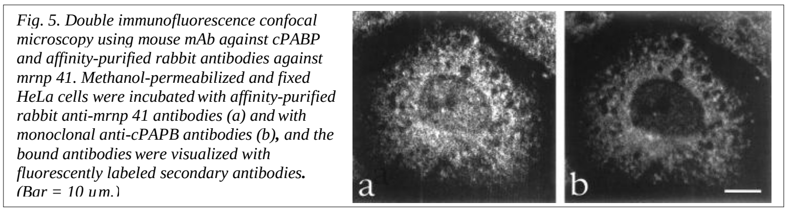


Figure 1. A figure caption pair reproduced from the biomedical literature.

and the results of analysis steps are stored in an SQL database as traceable assertions. An interface to the database (http://slif.cbi.cmu.edu) has been designed such that images and text of interest can be retrieved and presented to users[7].

In a system mining both text and images, associating the information from the text and the image is very challenging since usually there are multiple sub-figures in a figure and we must match sub-figures with the sentences in the text. In the initial version of SLIF, we extracted the labels for the sub-figures and sentences separately and matched them by finding equal-value pairs. This naive matching approach ignores much context information, i.e., the labels for sub-figures are usually a sequence of letters and people assign labels in a particular order rather than randomly, and could only achieve a matching accuracy of 64.3%. To obtain a satisfactory matching accuracy the naive approach requires high-accuracy image analysis and text analysis to get the labels. However, extracting labels from image is non-trivial. Inferring the label sequences and improving image processing allowed us to increase the F1 for panel label extraction to 78%[9]. In this paper, we introduce a stacked graphical model to match the labels of sub-figures with labels of sentences. The stacked model can take advantage of the context information and achieves an 81.3% accuracy.

In the following, we give a brief review of SLIF in Section 2. Section 3 describes the stacked model used for the matching. Section 4 summarizes the experimental results and Section 5 concludes the paper.

2. SLIF Overview

SLIF applies both image analysis and text interpretation to figures. Figure 1[a] is a typical figure that SLIF can analyse.

[a]This figure is reproduced from the article "mRNA binding protein mrnp 41 localizes to both nucleus and cytoplasm", by Doris Kraemer and Günter Blobel, Cell Biology Vol.

Pacific Symposium on Biocomputing 12:257-268(2007)

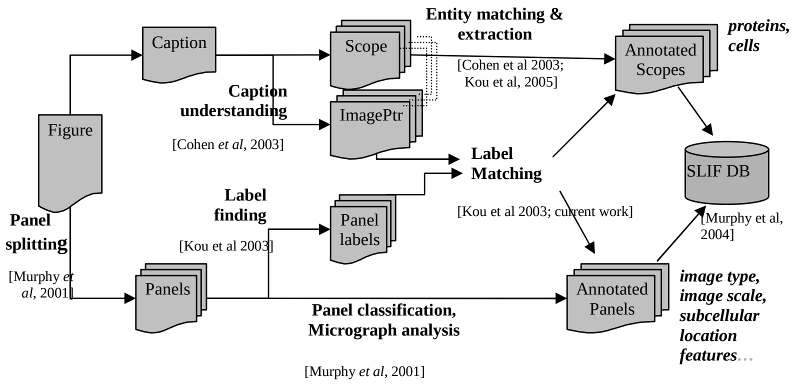


Figure 2. Overview of the image and text processing steps in SLIF.

Figure 2 shows an overview of the steps in the SLIF system with references to publications in which they are described in more details.

Image processing includes several steps: **Decomposing images into panels.** For images containing multiple panels, the individual panels are recovered from the image. **Identifying fluorescence microscope images.** Panels are classified as to whether they are fluorescence microscope images, so that appropriate image processing steps can be performed. **Image preprocessing and feature computations.** Firstly the annotations such as labels, arrows and indicators of scale contained within the image are detected, analyzed, and then removed from the image. In this step, *panel labels* are recognized by Optical Character Recognition (OCR). Panel labels are textual labels which appear as annotations to images, for example, "a" and "b" printed in panels in Figure 1. Recognizing panel labels is very challenging. Even after careful image pre-processing and enhancement the F1 accuracy is only about 75%. The OCR results are used as *candidate panel labels* and after filtering candidates an F1 accuracy of 78% is obtained[9]. Secondly, the scale bar is extracted, and finally subcellular location features (SLFs) are produced and the localization pattern of each cell is determined.

Caption Processing is done as follows. **Entity name extraction.** In the current version of SLIF we use an extractor trained on conditional random fields[10] and an extractor trained on Dictionary-HMMs[11] to extract the protein name. The cell name is extracted using hand-coded rules. **Image**

94, pp. 9119-9124, August 1997.

pointer extraction. The linkage between the panels and the text of captions is usually based on textual labels which appear as annotations to the images (i.e., panel labels), and which are also interspersed with the caption text. We call these textual labels appearing in text *image pointers*, for example, "(a)" and "(b)" in the caption in Figure 1. In our analysis, image pointers are classified into four categories according to their linguistic function: *Bullet-style image pointers*, *NP-style image pointers*, *Citation-style image pointers*, and *other*[12]. The image-pointer extraction and classification steps are done via a machine learning method[12]. **Entity to image pointer alignment**. The *scope* of an image pointer is the section of text (sub-caption) that should be associated with it. The scope is determined by the class assigned to an image pointer.[12]

3. A Stacked Model to Map Panel Labels to Image Pointers

3.1. *Stacked Graphical Models for Classification*

Stacked graphical models are a meta-learning scheme to do collective classification[13], in which a base learner is augmented by expanding one instance's features with predictions on other related instances. Stacked graphical models work well on predicting labels for relational data with graphical structures (Kou and Cohen, in preparation). The inference converges much faster than the traditional Gibbs sampling method and it has been shown empirically that one iteration of stacking is able to achieve good performance on many tasks. The disadvantage of stacking is that it requires more training time to achieve faster testing inference.

Figure 3 shows the inference and learning methods for stacked graphical models. In a stacked graphical model, the *relational template* C finds the related instances. For instance x_i, $C(x_i)$ retrieves the indices $i_1, ..., i_L$ of instances $x_{i_1}, ..., x_{i_L}$ that are related to x_i. Given predictions $\hat{\mathbf{y}}$ for a set of instances $\mathbf{x}$, $C(x_i, \hat{\mathbf{y}})$ returns the predictions on the related instances, i.e., $\hat{y}_{i_1}, ..., \hat{y}_{i_L}$.

The idea of stacking is to take advantage of the dependencies among instances, or the relevance between inter-related tasks. In our application in this paper, we conjecture that panel label extraction and image pointer extraction are inter-related, and design a stacked model that combines them.

3.2. *A Stacked Model for Mapping*

In the previous version of SLIF, we map panel labels to image pointers by finding the equal-value pair. Below we apply the idea of stacked graphical

- Parameters: a relational template C and a cross-validation parameter J.
- Learning algorithm: Given a training set $D = \{(\mathbf{x}, \mathbf{y})\}$ and a base learner A:
  - Learn the local model, i.e., when $k = 0$:
    Return $f^0 = A(D^0)$. Please note that $D^0 = D, \mathbf{x}^0 = \mathbf{x}, \mathbf{y}^0 = \mathbf{y}$.
  - Learn the stacked models, for $k = 1...K$:
    (1) Construct cross-validated predictions $\hat{\mathbf{y}}^{k-1}$ for $\mathbf{x} \in D$ as follows:
    (a) Split D into J equal-sized disjoint subsets $D_1...D_J$.
    (b) For $j = 1...J$, let $f_j^{k-1} = A(D^{k-1} - D_j^{k-1})$.
    (c) For $\mathbf{x} \in D_j, \hat{\mathbf{y}}^{k-1} = f_j^{k-1}(\mathbf{x}^{k-1})$.
    (2) Construct an extended dataset $D^k = (\mathbf{x}^k, \mathbf{y})$ by converting each instance x_i to x_i^k as follows: $x_i^k = (x_i, C(x_i, \hat{\mathbf{y}}^{k-1}))$, where $C(x_i, \hat{\mathbf{y}}^{k-1})$ will return the predictions for examples related to x_i such that $x_i^k = (x_i, \hat{y}_{i_1}^{k-1}, ..., \hat{y}_{i_L}^{k-1})$.
    (3) Return $f^k = A(D^k)$.
- Inference algorithm: given $\mathbf{x}$:
  (1) $\hat{\mathbf{y}}^0 = f^0(\mathbf{x})$.
  For $k = 1...K$,
  (2) Carry out Step 2 above to produce $\mathbf{x}^k$.
  (3) $\mathbf{y}^k = f^k(\mathbf{x}^k)$.
  Return $\mathbf{y}^K$.

Figure 3. Stacked Graphical Learning and Inference

models to map the panel labels and image pointers.

In SLIF the image pointer finding was done as follows. Most image pointers are parenthesized, and relatively short. We thus hand-coded an extractor that finds all parenthesized expressions that are (a) less than 15 characters long and (b) do not contain a nested parenthesized expression, and replaces X-Y constructs with the equivalent complete sequence. (E.g., constructs like "B-D" are replaced with "B,C,D".) We call the image pointers extracted by this hand-coded approach *candidate image pointers*. The hand-coded extractor has high recall but only moderate precision. Using a classifier trained with machine learning approaches, we then classify the candidate image pointers as bullet-style, citation-style, NP-style, or other. Image pointers classified as "other" are discarded, which compensates for the relatively low precision of the hand-coded extractor.[12]

In SLIF the panel label extraction was done as follows. Image processing techniques and OCR techniques are applied to find the labels printed within the panel. That is, firstly *candidate text regions* are computed via image processing techniques, and OCR is run on these candidate regions to get *candidate panel labels*. This approach has a relatively high precision yet low recall. We call the panel labels recognized by image processing and OCR *candidate panel labels*. A strategy based on *grid analysis* (a procedure which analyzes how many panels there are in a figure and finds out how the panels are ranged) is applied to the candidate panel labels to get a better accuracy.[9]

The match between panels labels and image pointers can be formulated as a classification problem. We construct a set of pairs $< o_i, p_j >$ for all candidate panel labels o_i's and candidate image pointers p_j's from the same figure. That is, for a panel with l_i representing the real label, o_i representing the panel label recognized by OCR, and p_j's representing the image pointers in the same figure, we construct a set of pairs $< o_i, p_j >$. We label the pair $< o_i, p_j >$ as positive only if $l_i = p_j$, otherwise negative. For example, in Figure 1, the real label l_i for panel a is "a". If OCR recognizes o_i where $o_i =$"a", image pointers for the figure are "a" and "b", we construct two pairs, $< a, a >$ labelled as positive and $< a, b >$ labeled as negative. Note that the pair is labelled according to the real label and the image pointers. If unfortunately, OCR recognizes o_i incorrectly for panel a in Figure 1, for example $o_i =$"o", we have two pairs, $< o, a >$ labelled as positive and $< o, b >$ labeled as negative.

We design features based on o_i's and p_j's. For a base feature set, there are 3 binary features: one boolean value indicating whether $o_i = p_j$, one boolean value indicating whether $o_{i\_left} = p_j - 1$ or $o_{i\_upper} = p_j - 1$, and another boolean value indicating whether $o_{i\_right} = p_j + 1$ or $o_{i\_down} p_j + 1$, where i_left is the index of the left panel of panel i in the same row, i_upper is the index of the upper panel of panel i in the same column, $p_j + 1$ is the successive letter of p_j and $p_j - 1$ is the previous letter of p_j. This feature set takes advantage of the context information by comparing o_{i_left} to $p_j - 1$ and so on. The second and third features capture the first-order dependency. That is, if the *neighboring panel* (an adjacent panel in the same row or the same column) is recognized as the corresponding "adjacent" letter, there is

a	b	c
d	e	f
g	h	i

Figure 4. Second-order dependency.

a higher chance that o_i is equal to p_j.

In the inference step for the base learner in the stacked model, if a pair $< o_i, p_j >$ is predicted as positive, we set the value of o_i to be p_j since empirically the image pointer extraction has a higher accuracy than the panel label recognition. That is, the predicted value $\hat{o}_i$ is p_j for a positive pair and $\hat{o}_i$ remains as o_i for a negative pair. After obtaining $\hat{o}_i$, we recalculate the features via comparing $\hat{o}_i$'s and p_j's. We call the procedure of predicting $< o_i, p_j >$, updating $\hat{o}_i$, and re-calculating features "stacking". We choose MaxEnt as the base learner to classify $< o_i, p_j >$ and in our experiments we implement one iteration of stacking.

Besides the basic features, we also include another feature that captures the "second-order context", i.e., consider the spatial dependency among all the "sibling" panels, even though they are not adjacent. In general the arrangement of labels might be complex: labels may appear outside panels, or several panels may share one label. However, in the majority of cases, panels are grouped into grids, each panel has its own label, and labels are assigned to panels either in column-major or row-major order. The "panels" shown in Figure 4 are typical of this case. For such cases, we analyze the locations of the panels in the figure and reconstruct this grid, i.e., the number of total columns and rows, and also determine the row and column position of each panel. We compute the second-order feature as follows: for a panel located at row r and column c with label o, as long as there is a panel located at row $r^{'}$ and column $c^{'}$ with label $o^{'}$ ($r^{'} \neq r$ and $c^{'} \neq c$) and according to either row-major order or column-major order the label assigned to panel ($r^{'}$, $c^{'}$) is $o^{'}$ given the label for panel (r, c) is o, we assign 1 to the second-order feature. For example, in Figure 4, recognizing the panel label "a" at row 1, column 1 would help to recognize "e" at row 2, column 2 and "h" at row 3, column 2.

With the first order-features and second-order features, it increases the chance of a missing or mis-recognized label to be matched to an image pointer.

4. Experiments

4.1. *Dataset*

To evaluate the stacked model for panel label and image pointer matching, we collected a dataset of 200 figures which includes 1070 sub-figures. This is a random subsample of a larger set of papers from the Proceedings of the National Academy of Sciences. Our current approach can only analyse labels contained within panels (internal labels) due to the limitations on the image processing stage therefore in our dataset we only collected figures with internal labels. Though our dataset does not cover all the cases, panels with internal labels are the vast majority in our corpus.

We hand-labeled all the image pointers in the caption and the label for each panel. The match between image pointers and panels is also assigned manually.

4.2. *Baseline algorithms*

The approaches to find the candidate image pointers and panel labels have been described in Section 3.2. In this paper, we take the hand-code approach and machine learning approach[12] as the baseline algorithms for image pointer extraction. The OCR-based approach and grid analysis approach[9] are baseline algorithms for panel label extraction.

We also compare the stacked model to relational dependency networks (RDNs).[14] RDNs are an undirected graphical model for relational data. Given a set of entities and the links between them, a RDN defines a full joint probability distribution over the attributes of the entities. Attributes of an object can depend probabilistically on other attributes of the object, as well as on attributes of objects in its relational neighborhood. We build an RDN model as shown in Figure 5.

In the RDN model there are two types of entities, image pointer and panel label. For an image pointer, the attribute p_j is the value of the candidate image pointer and o_i is the candidate panel label. $p_{\_tru}$ and $o_{\_tru}$ are the true values to be predicted. The linkage *L_pre* and *L_next* capture the dependency among the sequence of image pointers: *L_pre* points to the previous letter and *L_next* points to the successive letter. *P_left*, *P_right*, *P_upper*, and *P_down* point to the panels to the left, right, upper and down direction respectively. The RDN model takes the candidate image pointers

Pacific Symposium on Biocomputing 12:257-268(2007)

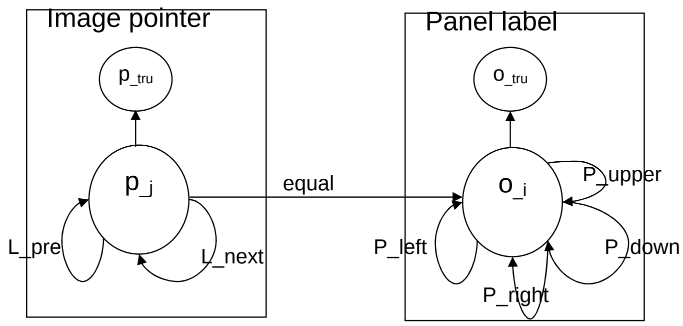


Figure 5. An RDN model

and panel labels as input and predicts their true values. The match between the panel label and the image pointer is done via finding the equal-value pair.

4.3. *Experimental Results*

We used 5-fold cross validation to evaluate the performance of the stacked graphical model for image pointer to panel label matching. The evaluation was reported in two ways; the performance on the matching and the performance on image pointer and panel label extraction. To determine the matching is the "real" problem, i.e., what we really care about are the matches, not getting the labels correctly. Evaluation on the image pointer and panel label extraction is a secondary check on the learning technique.

Table 1 shows the accuracy of image pointer to panel label matching. For the baseline algorithms, the match was done by finding the equal-value pair. Baseline algorithm 1 was done by comparing the candidate image pointers to the candidate panel labels. Baseline algorithm 2 was done by comparing the image pointers extracted by the learning approach to the panel labels obtained after grid analysis. The stacked graphical model takes the same input as Baseline algorithm 2, i.e., the candidate image pointers extracted by the hand-coded algorithm and the candidate panel labels obtained by OCR. We observe that the stacked graphical model improves the accuracy of matching. Both the first-order dependency and second-order dependency help to achieve a better performance. RDN also achieved a better performance than the two baseline algorithms. Our stacked model achieves a better performance than RDN, because in stacking the dependency is captured and indicated "strongly" by the way we design features.

Table 1. Accuracy of image pointer to panel label matching.

	Image pointer to panel label matching
Baseline algorithm 1	48.7%
Baseline algorithm 2 (current algorithm in SLIF)	64.3%
RDN	70.8%
Stacked model (first-order)	75.1%
Stacked model (second-order)	81.3%

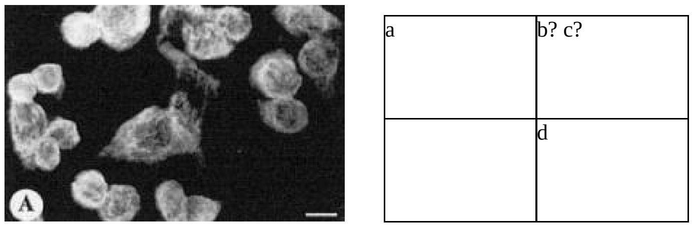
Table 2. Performance on image pointer extraction and panel label extraction.

	Image pointer extraction	Panel label extraction
Baseline algorithm 1	60.9%	52.3%
Baseline algorithm 2	89.7%	65.7%
RDN	85.2%	73.6%
Stacked model with first order dependency	-	77.8%
Stacked model with second order dependency	-	83.1%

That is, the stacked model can model the matching as a binary classification of $< o_i, p_j >$ and capture the first-order dependency and second-order dependency directly according to our feature definition. However, in RDNs, the data must be formulated as types of entities described with attributes and the dependency is modeled with links among attributes. Though RDNs can model the dependency among data, the matching problem is decomposed to a multi-class classification problem and a matching procedure. Besides that, the second-order dependency can not be modeled explicitly in the RDN.

Table 2 shows the performance on the sub-task of image pointer extraction and panel label extraction. The results are reported with F1-measurement. Since during the stacked model we update the value of o_i and set it to be p_j when finding a match, the stacking also improves the accuracy of panel label extraction. The accuracy for image pointer extraction remains the same since we do not update the value of p_j. Baseline algorithm 1 is the approach of finding candidate image pointers or candidate panel labels. Baseline algorithm 2 for image pointer extraction is the learning approach, and the grid analysis strategy for panel label extraction. The inputs for the stacked graphical model are candidate image pointers and candidate panel labels. We observe that by updating the value of o_i,

Pacific Symposium on Biocomputing 12:257-268(2007)



(a) A hard case for OCR

(b) A hard case for the stacked model

Figure 6. Cases where current algorithms fail

we can achieve a better performance of panel label extraction, i.e., provide more “accurate” features for stacking. RDN also helps to improve the performance yet the best performance is obtained via stacking.

4.4. *Error Analysis*

As mentioned in Section 2, OCR on panel labels is very challenging and we suffer a low recall of baseline algorithm 1. Most errors occur when there are not enough o_i recognized from the baseline algorithm to obtain information of the first-order and second-order dependency. Figure 6(a) shows a case where the current OCR fails. Figure 6(b) shows a case where there is not enough contextual information to determine the label for the upper-left panel.

5. Conclusions

In this paper we briefly reviewed the SLIF system, which extracts information on one particular aspect of biology from a combination of text and images in journal articles. In such a system, associating the information from the text and image requires matching sub-figures in a figure with the sentences in the text. We used a stacked graphical model to match the labels of sub-figures with labels of sentences. The experimental results show that the stacked graphical model can take advantage of the context information and achieve a significant improvement in the matching accuracy of the stacked graphical model as compared with a relational dependency network or the current algorithm in SLIF. In addition to accomplish the matching at a higher accuracy, the stacked model helps to improve the performance of finding labels for sub-figures as well.

The idea of stacking is to take advantage of the context information, or the relevance between inter-related tasks. Future work will focus on applying stacked models to more tasks in SLIF, such as protein name extraction.

Acknowledgments

The work was supported by research grant 017396 from the Commonwealth of Pennsylvania Department of Health, NIH grants K25 DA017357 and R01 GM078622, and grants from the Information Processing Technology Office (IPTO) of the Defense Advanced Research Projects Agency (DARPA).

References

1. B. de Bruijn and J. Martin, *Getting to the (c)ore of knowledge: mining biomedical literature.* Int. J. Med. Inf., **67**(2002), 7-18.
2. M. Krallinger and A. Valencia *Text-mining and information-retrieval services for molecular biology.* Genome Biology 2005, **6**:224.
3. L. Hunter and K. B. Cohen, *Biomedical language processing: what's beyond PubMed?* Molecular Cell **21**(2006), 589-594.
4. H. Yu and M. Lee. *Accessing Bioscience Images from Abstract Sentences.* Bioinformatics 2006, 22(14), 547-556.
5. B. Rafkind, M. Lee, SF Chang, and H. Yu. *Exploring text and image features to classify images in bioscience literature.* Proceedings of BioNLP 2006, 73-80.
6. H. Shatkay, N. Chen, and D. Blostein. *Integrating Image Data into Biomedical Text Categorization.* Bioinformatics 2006, 22(14), 446-453.
7. R. F. Murphy, Z. Kou, J. Hua, M. Joffe, and W. W. Cohen, *Extracting and Structuring Subcellular Location Information from On-line Journal Articles: The Subcellular Location Image Finder.* Proceedings of KSCE 2004, 109-114.
8. R.F. Murphy, M. Velliste, J. Yao, and G. Porreca,*Searching Online Journals for Fluorescence Microscope Images Depicting Protein Subcellular Locations.* Proceedings of BIBE 2001, 119-128.
9. Z. Kou, W. W. Cohen, and R. F. Murphy, *Extracting Information from Text and Images for Location Proteomics.* Proceedings of BIOKDD 2003, 2-9.
10. M. Ryan and P. Fernando, *Identifying Gene and Protein Mentions in Text Using Conditional Random Field.* BMC Bioinformatics, 6(Suppl 1):S6, May 2005.
11. Z. Kou, W. W. Cohen, and R. F. Murphy, *High-Recall Protein Entity Recognition Using a Dictionary.* Bioinformatics 2005, 21(Suppl 1), 266-273.
12. W. W. Cohen, R. Wang and R. F. Murphy, *Understanding Captions in Biomedical Publications.* Proceedings of KDD 2003, 499-504.
13. B. Taskar, P. Abbeel and D. Koller, *Discriminative probabilistic models for relational data.* Proceedings of UAI 2002, 485-492.
14. D. Jensen and J. Neville, *Dependency Networks for Relational Data.* Proceedings of ICDM 2004, 170-177.