# Intra-document Structural Frequency Features for Semi-supervised Domain Adaptation

Andrew Arnold
Machine Learning Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
aarnold@cs.cmu.edu

William W. Cohen
Machine Learning Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
wcohen@cs.cmu.edu

## ABSTRACT

In this work we try to bridge the gap often encountered by researchers who find themselves with few or no labeled examples from their desired target domain, yet still have access to large amounts of labeled data from other related, but distinct source domains, and seemingly no way to transfer knowledge from one to the other. Experimentally, we focus on the problem of extracting protein mentions from academic publications in the field of biology, where the source domain data are abstracts labeled with protein mentions, and the target domain data are wholly unlabeled captions. We mine the large number of such full text articles freely available on the Internet in order to supplement the limited amount of annotated data available. By exploiting the explicit and implicit common structure of the different subsections of these documents, including the unlabeled full text, we are able to generate robust features that are insensitive to changes in marginal and conditional distributions of classes and data across domains. We supplement these domain-insensitive features with automatically obtained high-confidence positive and negative predictions on the target domain to learn extractors that generalize well from one section of a document to another. Finally, lacking labeled target testing data, we employ comparative user preference studies to evaluate the relative performance of the proposed methods with respect to existing baselines.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; J.3 [**Computer Applications**]: Life and Medical Sciences; M.4 [**Knowledge Management**]: Knowledge modeling

## General Terms

Algorithms


*CIKM'08,* October 26–30, 2008, Napa Valley, California, USA.


## Keywords

km_information_extraction, ir_content::structured, semi structured, meta data, social tagging, km_text_mining, km_statistical_techniques

## 1. INTRODUCTION

The desire to exploit information attained from previous effort, and not to start each new endeavor *de novo* is perhaps part of human nature, and certainly a maxim of the scientific method. Nevertheless, due to the difficulty of integrating knowledge from distinct, but related, experimental domains, it is common practice in most machine learning studies to focus on training and tuning a model to a single, particular domain, at the expense of all others. Often, once work has completed on one domain, the researcher begins afresh on the next, carrying over only the techniques and experience learned, but often not the data itself.

In this work we try to address this issue by providing methods to facilitate the adaptation of data from one domain (called the **source)** to problems defined on another related domain (called the **target)**. This type of problem is generally referred to as **domain adaptation** [13] in the literature and constitutes a subproblem in the broader field of **transfer learning,** which has been studied as such for at least the past ten years [26, 4].

For the rest of this paper we will focus on the specific problem of learning to extract protein names from articles published in biological journals. In the *named entity resolution* (NER) formalism, a document is segmented into a sequence of tokens, with each of these tokens then being classified as belonging to one of a set of possible label classes – in our case, the binary set {PROTEIN, NON_PROTEIN}. A standard technique for this kind of problem is to gather a corpus of documents drawn from the domain on which you will eventually be evaluated. These documents then need to be painstakingly hand-labeled by a domain expert in order to identify which tokens in the document represent proteins, and which do not. The 'expertise' of this domain specialist should not be underestimated, since such biological distinctions are subtle and often elude all but the most experienced annotators. The work is therefore slow, and the resulting annotated datasets are often relatively small and expensive.

We have access to such a corpus of protein-labeled abstracts from biological articles. Several techniques have been proposed for building protein-name extractors over these abstracts and their performances have been evaluated with re-

spect to extracting new proteins from other, previously unseen abstracts drawn from a similar distribution of articles [14]. In our work, however, we are interested in identifying proteins, not in abstracts, but in the captions of papers (we use this information to create a structured search engine of images and captions from biological articles [19]). To this end we have downloaded tens of thousands of open-access, full text articles from the Internet. Unfortunately, all of these documents are wholly unlabeled and we do not have the resources to label them ourselves. Thus, our problem is: given labeled abstracts (source training domain) and unlabeled captions and full text (source auxiliary training data), how can we train a model that will extract proteins well from unseen captions (target test domain). This is at once a semi-supervised learning problem (due to the unlabeled auxiliary training data) [28], and a domain adaptation problem (due to the difference in domains from which the source and target data are drawn).

In §1.1 we formally define the problem of domain adaptation and provide some context from related work in transfer and semi-supervised learning. Section 2.2 introduces a key insight into the structure of documents that allows us to link the source, target, and auxiliary domains. Given this perspective, our problem, stated generally as domain adaption from the abstract (source domain) to the captions (target domain) of a paper, can be viewed more specifically as learning to transfer information from one part of a structured document to another, allowing us to overcome the 'domain-brittleness' of the commonly used lexical features, described in §2.1.

Sections 2.3 and 2.4 introduce three new techniques that leverage this structure to produce models that are able to exploit the unlabeled auxiliary data while at the same time being robust to shifts between train and test domains. Section 3 explains the particulars of the data and experiments we used to validate these new techniques, while §4 offers a summary of the paper with views towards future work.

## 1.1 Domain adaptation

The standard discriminative statistical machine learning classification task is, given an example $x$ and a class label $y$, to assign a probability, $p(y|x)$, to $x$ of belonging to class $y$. In the supervised setting, the data is usually segmented into two disjoint subsets: the training set $(X_{train}, Y_{train}) = \{(x_1, y_1) \cdots, (x_N, y_N)\}$, which can be used for training, and the test set $X_{test} = (x_1, \cdots, x_M)$, for which labels are not available at training time. In the semi-supervised setting [28], the training data is supplemented with a set of **auxiliary** data, $X_{aux} = (x_1, \cdots, x_P)$, for which no corresponding labels are provided.

In the normal, non-transfer setting $X_{test}$ and $X_{train}$ are both assumed to have been drawn from the same distribution, $\mathcal{D}$. In the semi-supervised *domain adaptation* setting, however, we allow the distribution from which the test examples are drawn to differ from that of both the training and auxiliary examples. More formally, we propose two different, but related, distributions, $\mathcal{D}^{source}$ and $\mathcal{D}^{target}$, and posit that the training examples $(X_{train}^{source}, Y_{train}^{source})$ are drawn from $\mathcal{D}^{source}$, while the test examples, $X_{test}^{target}$, will be drawn from $\mathcal{D}^{target}$. We do not specify from which distribution the auxiliary data is drawn.

*Domain adaptation* is distinct from other forms of transfer learning (such as multitask learning [1, 9, 24, 27]) because we are assuming that the set of possible labels, $Y$, remains constant across the various domains, while allowing the distribution of $X$ and, most importantly, $Y|X$ to change. In our setting, the labels, $Y$, are members of the binary set {PROTEIN, NON_PROTEIN}, while the instances, $X$, are the tokens of the documents themselves.

In prior work, different researchers have made different assumptions about the relationship between the source and target domain, a defining characteristic of domain adaptation. In the supervised setting, one can directly compare both the marginal and conditional distributions of the data in both domains, looking for patterns of generalizability across domains [13, 17, 12], as well as examining the common structure of related problems [5, 22, 3, 6]. There is likewise work that tries to quantify these inter-domain relationships in the unsupervised [2], semi-supervised [16, 7], and transductive learning settings [25]. Similarly, in the biological domain, there has been work on using semi-supervised machine learning techniques to extract protein names by combining dictionaries with large, full-text corpora [23], but without the explicit modeling of differences between data domains that we attempt in this paper. In our work, we take advantage of the fact that the source and target domains are different sections of the same structured document and use this fact to develop features that are robust across those different domains.

# 2. METHODS

## 2.1 Lexical features

Most modern information extraction systems rely on some kind of representation, usually a set of **features**, that distills the document into a form the algorithm can interpret and manipulate. The exact form of these features is a vital component of the overall system, balancing the complexity of a rich representation with the parsimony of an insightful view of the domain and problem being solved. For named entity recognition, **lexical features**, which try to capture patterns of words within the text of a document, are one of the most common, and intuitive, types of these representations. Generally, a lexical feature is a function of a word and its context. The specific definition of this function may vary widely across domains and implementations. In our setting, each lexical feature is a boolean function over a token in a document representing the value and morphology of that token and its neighbors. For example, given the sentence fragment from a caption of a biological paper: 'Figure 4: Tyrosine phosphorylation...', some lexical features for the token 'Tyrosine' would look like:

| |
|---|
| CurrentToken.isWord.Tyrosine |
| CurrentToken.charPattern.Xx |
| CurrentToken.endsWith.ine |
| Right1Token.endsWith.ation |
| Left1Token.isWord.: |
| Left3Token.isWord.Figure |

**Table 1: Lexical features for token 'Tyrosine' in sample caption: 'Figure 4: *Tyrosine* phosphorylation...'.**

Notice that, although these features are defined with respect to a certain current token, 'Tyrosine', they also take

into account the context of that word in the document. In this example, if we knew that this occurrence of 'Tyrosine' was labeled as a protein, the fact that the token immediately to the left of the current token was a semi-colon (*Left1Token.isWord.:*) might be useful in predicting whether other, heretofore unseen tokens besides 'Tyrosine', that also happen to be preceded by a semi-colon, might also be proteins.

Since each word in one's vocabulary may constitute a feature (e.g., CurrentToken.isWord.A, CurrentToken.isWord.B, ...), it is not uncommon to have tens or even hundreds of thousands of such binary lexical features defined in one's feature space. The benefit of this is that such a large feature space can richly represent most any training set. The examples in Table 1 also include domain-specific features such as 'CurrentToken.endsWith.ine' (a common suffix for amino-acids). These custom features allow the researcher to bias his feature space towards specific features that he feels might be more informative with respect to his particular problem domain. While this specificity may be advantageous for an expert dealing with a limited domain, it can become a liability when that domain is uncertain, or even variable, as is the case in our transfer learning setting.

For instance, while the occurrence of a semi-colon or the word 'Figure' may be very informative in terms of identifying words as proteins in the captions of papers, if our extractor is trained only on abstracts it may never see those types of features. Indeed, since lexical features are merely functions of the specific sections of text seen during training, they are unable to capture information residing in other sections of the document which may prove useful. Even in the semi-supervised case where the learning algorithm has access to unlabeled target domain data, lexical features are unable to take advantage of this information since there is no way to relate the unlabeled tokens to the labeled ones.

Lexical features thus provide a valuable, but brittle, representation of the training data. Our work tries to augment these rich, though domain-specific, lexical features with other non-lexical features based on the internal structure of a document, contributing another view of the data that is more robust to changes in the domain. We hope to show that combining these types of domain-specific and domain-robust features produces a classifier that performs well across domains.

## 2.2 Document structure

We begin by highlighting the common observation that most documents are written with some kind of internal structure. For instance, the biological papers we studied in this experiment (like most academic papers) can be divided into roughly three sections:

- **Abstract**: summarizing, at a high level, the main points of the paper such as the problem, contribution, and results.

- **Caption**: summarizing the figure it is attached to. These are especially important in biological papers where most important results are represented graphically. Unlike computer science papers, which usually have brief captions, in our corpus the average caption was over 125 words long thus supporting our belief that they might contain useful information for our NER task.

- **Full text**: the main text of a paper, that is, everything else besides the abstract and captions.

An example of such a structured document is provided in Figure 1. In this figure we see the various ways a protein can be referred to throughout the sections of a document. Notice how the distribution of these types of occurrences varies across the structure of the document. For instance, full name references (red) do not appear in the caption, while non-protein parentheticals (brown) do not appear in the abstract. Here we see the importance of explicitly modeling the difference between the source and target domains: if one were to naïvely train a purely lexical feature based extractor on the abstracts and try to apply it to the captions, the extractor might be confused by the non-protein parentheticals, having never seen them in its training data. Likewise, it might waste significant probability mass on features representing the unabbreviated form of protein names which it might never see in its caption test data. It is important to note that in order to support this interpretation of the data we have to make the so-called *one-sense-per-discourse* assumption [15], namely, that tokens in one section of a document have the same meaning as identical tokens in other sections of the same document. In this way we are able to link references across the source and target domains.

Since we have no labeled target domain data, however, it is not obvious how we might amend or supplement our source domain training data so as to avoid these problems. The key insight is the fact that these domains, while distinct, are nevertheless related by the overarching structure of the documents in which they reside. For instance, while unabbreviated protein names never appear in the caption, and non-protein parentheticals never appear in the abstract, both of these occur in the full text of the paper. Thus, our goal is to find some class of features that can relate these different types of occurrences together across the differing subsections of a document's structure. We will achieve this by leveraging the one-sense-per-discourse assumption and our knowledge about our documents' structure to create two new types of features:

- **Structural frequency features**: Informative with respect to protein extraction, but make repeated occurrences of the same token in different sections look similar.

- **Snippets**: Pseudo-examples that push a learned classifier towards being consistent with the one-sense-per-discourse assumption.

## 2.3 Structural frequency features

Structural frequency features, like lexical features, are simply functions of tokens in context. Unlike purely lexical features, however, structural frequency features *are* able to leverage the occurrence of tokens across all sections of a document, including the unlabeled captions and full text. The idea is to leverage the fact that different types of tokens (e.g., unabbreviated protein names, non-protein parentheticals, etc.) occur with different frequencies in different sections of a document. In the example from Figure 1 in §2.2, we noticed that non-protein parentheticals occurred quite often in the caption, but not at all in the abstract. While this seems informative, in our setting, unfortunately, we do not have labels for the caption data. We are therefore unable to make a distinction between *protein* and *non-protein*

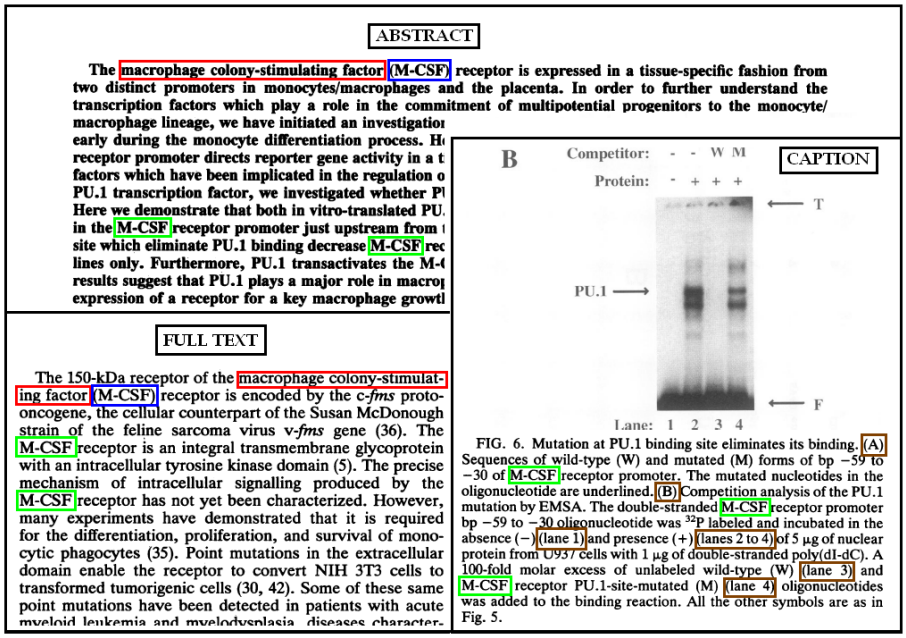


**Figure 1: Sample biology paper. Each black box represents a different subsection of the document's structure: abstract, caption and full text. Each highlighted box represents a different type of information: full protein name (red), abbreviated protein name (green), parenthetical abbreviated protein name (blue), non-protein parentheticals (brown).**

parentheticals in the caption section of the document. We can, however, make such a distinction in the *abstract* section of the same document, for which we do have labels. Thus, if we see a parenthesized token in a caption, and see the same token parenthesized in the abstract, we might be able to transfer that abstract token's label to the unlabeled caption occurrence. In this respect, these structural frequency features provide the links necessary to perform a kind of label propagation across the subsections of a document [29].

Given our previously stated one-sense-per-discourse assumption, we now have a means of transferring our labels across the different unlabeled sections of a document and may have a useful, non-transfer, semi-supervised learning model. Our ultimate goal, however, is semi-supervised domain adaptation, and these structural features, as described thus far, still lack a way of ensuring they will be robust across shifts in domain. The key to addressing that issue is to consider the occurrence of tokens not in isolation within each subsection of a document, but rather jointly across sections. For instance, in Figure 1 we see the token '(lane *)' occurs quite often in the caption, but never in the full text. In fact, there are many such non-proteins that only ever appear in the caption section of the document. In contrast, the token 'M-CSF' occurs with high frequency across all three sections of the document. Indeed, there are relatively few proteins that do *not* occur in the abstract of a paper. It seems we can use the relative distribution of tokens across the different sections of a document, in and of itself and without any lexical information, as a signal of that token's likelihood of being a protein. This makes sense, since authors are conveying different kinds of information, in different ways, across the various sections of a document and so are not equally likely to mention a protein, in the same particular way, across the entire document.

Specifically, for each unique word-type in a document, we counted the number of times it appeared in each of the different sections of that document (for example, the word-type 'M-CSF' occurs three times in the abstract, four times in the full text, and three times in the caption of the example in Figure 1). We then normalized these counts by the total number of tokens in a given section to come up with an empirical probability of a word-type occurring in a particular section. We also computed the conditional forms of these features, that is, we counted the number of times a token appeared in section x, given that it also appeared in section y, again normalizing to form an empirical probability distribution. Continuing our example, the token 'macrophage' never occurs in the caption and thus, although the token does occur in the abstract, p(word occurring in caption | word occurs in abstract) is still zero (see Table 2 for more examples). These conditional structural frequency features allow us to characterize the particular distribution patterns that different types of words have across the sections of a document. In particular, we might be interested in modeling things like p(word is a protein | word appears in caption but not in abstract). Figures 2 and 3 show the distribution of two such features across our training data.

Figure 2 shows a histogram of the number of times words labeled in the *abstract* as proteins (left) and non-proteins (right) occurred with a given log normalized probability in the document's full text, given that it also appeared (at least once) in the same document's abstract section. Since these probabilities are plotted on the log scale, any zero values (i.e., words that appear in abstracts but never in the full text), will be assigned to the bin at negative infinity. The lack of instances at negative infinity in the left plot is evidence that, if a protein is in an abstract, it is also always in the full text at least once. But this is not so for non-proteins

| | Times in: | | | Total tokens in : | | | | | Log prob. of occurring in: | | | Log conditional prob.: | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Word | A | C | F | A | C | F | C\|A | F\|A | A | C | F | P(in C\|in A) | P(in F\|in A) |
| 'M-CSF' | 3 | 3 | 4 | 206 | 121 | 4,971 | 47 | 53 | -1.84 | -1.61 | -3.10 | -1.20 | -1.12 |
| 'macrophage' | 2 | 0 | 1 | 206 | 121 | 4,971 | 47 | 53 | -2.01 | -Inf | -3.70 | -Inf | -1.72 |
| '(M-CSF)' | 1 | 0 | 1 | 206 | 121 | 4,971 | 47 | 53 | -2.30 | -Inf | -3.70 | -Inf | -1.72 |
| 'PU.1' | 5 | 2 | 0 | 206 | 121 | 4,971 | 47 | 53 | -1.61 | -1.78 | -Inf | -1.37 | -Inf |
| 'kDa' | 0 | 0 | 1 | 206 | 121 | 4,971 | 47 | 53 | -Inf | -Inf | -3.70 | Never in A | Never in A |

**Table 2: Sample structural frequency features for tokens in example paper from Figure 1, as distributed across the (A)bstract, (C)aptions and (F)ull text.**

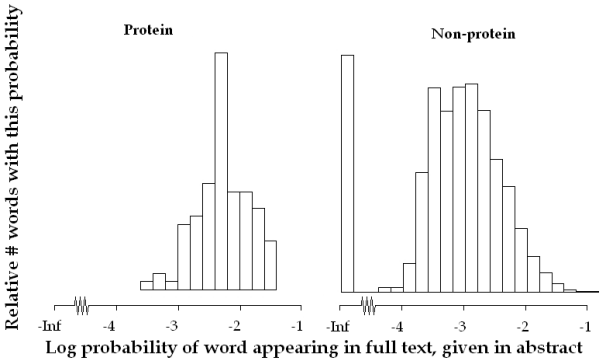


**Figure 2: Histogram of the number of occurrences of protein (left) and non-protein (right) words with the given log normalized probability of appearing in *full text*, given that they also appear in an article's *abstract*.**

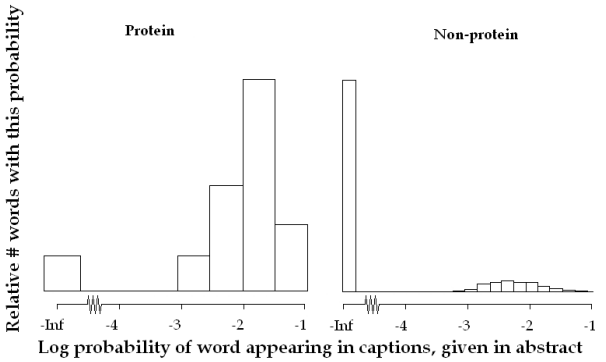


**Figure 3: Histogram of the number of occurrences of protein (left) and non-protein (right) words with the given log normalized probability of appearing in *captions*, given that they also appear in an article's *abstract*.**

– the large spike on the left side of the right plot shows a large number of non-proteins that appear in abstracts but never in the full text. Also notice the general right-shift of the entire distribution in the left plot, indicating an overall higher proportion of proteins occurring in full-text, given that they appear in an abstract, as compared to non-proteins.

Figure 3 shows a similar distribution, only this time the conditional structural frequency feature is measuring the likelihood of a word occurring in the *captions* of a paper, given that it appeared in the abstract. Notice, again, the left spike in the non-protein histogram on the right, indicating that a large number of non-proteins never appear in article's captions, despite appearing in its abstract. In contrast, the higher peaks to the right of the protein plot on the left show a much higher proportion of proteins appearing in captions, given they also appear in the abstract.

These plots clearly demonstrate a significant difference in the distribution of protein and non-protein tokens across the various subsections (abstract, captions, and full text) of a document's structure and suggest these structural frequency features may be informative with respect to identifying and extracting proteins. Thus, at training time, we compute these structural frequency features for each token in our labeled training abstracts. Since counting token occurrences across document sections, however, does not require labels itself, we can freely use all the unlabeled text from the papers we have to calculate the features. Likewise, by leveraging the one-sense-per-discourse assumption, we can attach the word-type's label (found in the abstract) to each of these features defined across the various sections of the document. In the end, we are left with a semi-supervised intra-document representation of the labeled abstract data that is, due to its cross structural nature robust to shifts across the various document section domains.

## 2.4 Snippets

Although structural frequency features provide domain-robust signals to our extractor, they do not directly ameliorate the domain-brittleness of the lexical features discussed in §2.1. To address this issue, we introduce a kind of pseudo-data we call **snippets**. Snippets are tokens or short phrases taken from one of the unlabeled sections of the document and added to the training data, having been automatically positively or negatively labeled by some high confidence method.

### *2.4.1 Positive snippets*

**Positive snippets** (i.e., snippets automatically labeled as positive examples) are an attempt to leverage the overlap between and across domains, by taking high confidence examples from one domain and transferring them to the other. In this sense, it is related to co-training [8]. Specifically, positive snippets leverage the one-meaning-per-discourse assumption (which we again rely upon due to our lack of labeled target data). The unlabeled target sections of a document are searched for tokens that match positively labeled tokens from the labeled source sections. Any matching instances are copied, along with a bit of neighboring context, into the training data, with the matching tokens labeled positive, and their context (where it does not match a pro-

tein name observed in the abstract) labeled negative, the idea being that this surrounding context will help inform the extractor of the differences in the distribution of lexical features in the target domain. Since our goal is to train an extractor that will be robust to shifts from source to target domain, we would like to introduce some examples of the target domain into the source domain training data to make it look more like the target domain. Since we don't have labels for the target domain, however, we have to rely on this high-confidence token matching heuristic.

#### 2.4.2 Negative snippets

Similarly, **negative snippets** (i.e., snippets automatically labeled as negative examples) provide examples of tokens which may appear to be proteins when viewed with respect to the source domain, but are in fact not proteins in the target domain. These must rely on some form of prior knowledge about the target domain for their high-confidence automatic labeling, perhaps some kind of extractor previously trained for the target domain. For example, a researcher may have previously trained an extractor to identify tokens in captions that refer to specific panel locations in the accompanying image (e.g., the token '(B)' in Figure 1's caption). We call these types of references *image pointers* [11]. Although this kind of token pattern may look like a parenthetical protein mention if seen in an abstract, since we have an existing extractor able to identify it as an image pointer in captions (and thus, by mutual exclusion, not a protein), we are able to add all occurrences in a paper's captions of similarly identified image pointers (labeled as negative) to that paper's labeled training data.

In this way, snippets allow us to use our unlabeled target data not just to add new inter-domain information (as with structural frequency features), but also, perhaps as importantly, to adjust and augment the distribution of existing source domain derived lexical features to make them more in accord with the target domain, ultimately producing extractors that are more robust to changes between training and test domains.

### 2.5 Conditional random fields

When it comes to actually training a model, we need a learning algorithm that can integrate and balance the variety of features and disparate sources of information we are trying to exploit. We used **conditional random fields** (CRF's) [18], a generalization of the common maximum entropy model from the i.i.d. case (where each token is classified in isolation), to the sequential case (where each token's classification influences the classification of its neighbors). This attribute is especially useful in a setting such as domain adaptation, where we would like to spread high-confidence predictions made on examples resembling the source domain to lower-confidence predictions of less familiar target domain instances. Similarly, like maximum entropy models, CRF's allow great flexibility with respect to the definition of the model's features, freeing us from worrying about the relative independence of specific features.

## 3. INVESTIGATION

### 3.1 Data

Our training data was drawn from two sources:

- GENIA: a corpus of Medline abstracts with each token annotated as to whether it is a protein names or not [21]
- PubMed Central (PMC): a free, on-line archive of biological publications [20]

Since our methods rely on having access to a document's labeled abstract along with the unlabeled captions and full text, and GENIA only provided labeled abstracts, we had to search PMC for the corresponding full text, where available. Of GENIA's 1,999 labeled abstracts, we were able to find the corresponding full article text (in PDF format) for 303 of them on PMC. These PDF's were (noisily) converted to text[1] and segmented into abstract, captions, and full text using automated tools. Figure 1 shows an example of one such segmented PDF.

Of these 303 papers, consisting of abstracts labeled with protein names along with corresponding unlabeled captions and full text, 218 (consisting of over 1.5 million tokens) were used for training, and 85 (almost 640,000 tokens) were used for testing.

### 3.2 Experiment

Experimentally, we used ablation studies to assess the amount of information our novel features:

- Structural frequency features (FREQ)
- Positive snippets (POS)
- Negative snippets (NEG)

each contribute to the task of protein name extraction, both in the non-transfer (abstract to abstract) and domain adaptation (abstract to caption) setting. In each case, we trained an extractor on a version of the training data constructed with the appropriate set of features. In all experiments we used the Minorthird toolkit to construct the lexical features and perform the CRF training [10].

### 3.3 Results

#### 3.3.1 Structural frequency features

Figure 4 compares the performance on held-out abstracts (in terms of precision and recall) of extractors training only on lexical features (**LEX** of §2.1), only on structural frequency features (**FREQ** of §2.2), and on a combination of both types of features (**LEX+FREQ**).

We can observe that, while the lexically trained model always outperforms the strictly structural frequency informed model (LEX dominates FREQ), the FREQ model nevertheless produces a competitive precision-recall curve despite having no access to any lexical information. This supports the intuition developed from observing the difference between protein and non-protein distributions in Figures 2 and 3.

[1]e-PDF PDF to Text Converter v2.1: http://www.e-pdfconverter.com

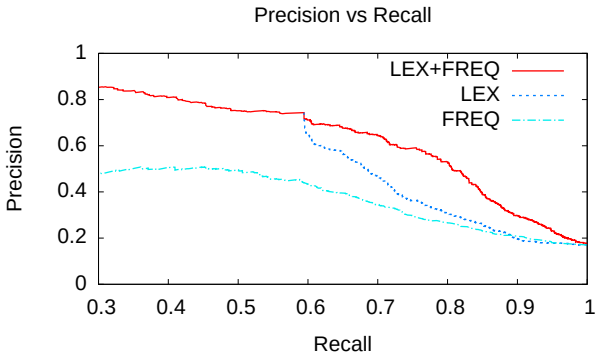


**Figure 4: Precision versus recall of extractors trained on only lexical features (LEX), only structural frequency features (FREQ), and both sets of features (LEX+FREQ).**

Similarly, the fact that the combined model LEX+FREQ dominates each constituent model (LEX and FREQ individually) demonstrates that each type of feature (lexical and structural) is contributing a share of unique information, not represented by the other. This supports the connection with co-training proposed in §2.4 by indicating that the feature sets are somewhat independent with respect to identifying protein names (the fact that their effect in the combined model is not completely additive suggests they are not wholly independent either).

### 3.3.2 *Non-transfer: abstract to abstract*

Table 3 shows the performance of seven different extractors (sorted by F1), each trained on a unique combination of our proposed features: positive snippets (POS), negative snippets (NEG), and structural frequency features (FREQ), all along with the standard lexical features (LEX). A check mark in a feature's column means that row's extractor was provided with that column's features at train-time. In this non-transfer experiment, our model labeled tokens of held-out *abstracts* as protein or not, and these predictions were automatically evaluated with respect to token-level precision, recall and F1 measure using the held-out GENIA labels for those abstracts.

| Model name | POS | NEG | FREQ | Prec | Rec | F1 |
|---|---|---|---|---|---|---|
| FULL | ✓ | ✓ | ✓ | .738 | .673 | .704 |
| FREQ | | | ✓ | .744 | .640 | .688 |
| POS_FREQ | ✓ | | ✓ | .727 | .637 | .679 |
| POS | ✓ | | | .760 | .555 | .641 |
| POS_NEG | ✓ | ✓ | | .760 | .547 | .636 |
| BASE | | | | .753 | .550 | .636 |
| NEG_FREQ | | ✓ | ✓ | .751 | .535 | .625 |

**Table 3: Summary of ablation study results for extractors trained on full papers and evaluated on *abstracts*.**

From this table we can notice a number of trends. With respect to the baseline model (BASE) trained only on lexical features, adding positive snippets (POS) doesn't seem to help precision or recall much, while adding structural frequency features (FREQ) improves recall (and thus F1) dramatically. This makes sense, since positive snippets were proposed as a method of increasing domain-robustness, and these results are for the non-transfer setting. On the other hand, structural frequency features were proposed as a general purpose method of using an article's internal structure to help extract useful information from the unsupervised sections of the document. In this respect, FREQ features might be expected to aid in even the non-transfer setting, as they do here. Interestingly, although in isolation, and even in combination, POS and NEG snippets themselves don't seem to improve on the baseline model in the non-transfer setting, when combined with FREQ features (FULL) they do seem to provide another boost to recall. This may be due to the fact the inter-domain information implicitly incorporated by the structural frequency features allows the model to better make use of the cross-domain snippets.

We should note that, although this non-transfer, abstract to abstract setting is convenient (since we can get precise evaluation numbers) and the results encouraging, it is unclear what they might indicate about performance in the transfer setting.

### 3.3.3 *Transfer: abstract to caption, full vs. baseline*

Finally, we present the results of a user study in the domain adaptation setting. We trained extractors on various combinations of features computed on the training data, and compared them to the full model trained on lexical, structural, positive and negative snippets, evaluating each with respect to the proteins they predicted in the held-out *captions*. Unlike the non-transfer setting, however, since we had no labels for any captions, we could not perform automatic evaluation. Instead, we employed human experts to manually compare the predictions made by variously constructed extractors and evaluate which they preferred. Using this method we found that our proposed model (FULL, the joint combination of all three new feature types: POS, NEG and FREQ) was preferred by users significantly more often ($p < .01$, see Table 4) than the baseline model trained only on lexical features.

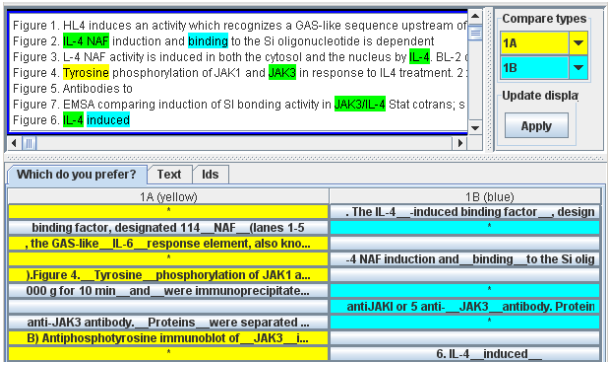


**Figure 5: Screenshot of application used to compare various protein extractors' performance on captions in the face of no labeled data.**

Figure 5 shows a screenshot of the tool we used to perform these evaluations. In the top-right, two extractors are being compared: 1A in yellow and 1B in blue (their names have been blinded from the evaluator). The top-left panel shows

the captions of a particular test article with each extractor's positive (protein) predictions highlighted in its color, with green highlights representing tokens on which both extractors predict positive. The bottom panel shows two columns of buttons: 1A's predictions are on the left, and 1B's on the right. Since we are evaluating user preference, only the predictions where the extractors disagree are shown. For each row (corresponding to a disagreement between extractors) the human expert clicks the cell of the prediction he prefers: clicking an empty cell in one column means the user believes the other column's extractor made a type I (false positive) error, while clicking a non-empty cell implies the other column's extractor made a type II (false negative) error. Each of these judgments can be viewed as the outcome of a paired trial, and by using a paired t-test, we can assess how the extractors differ along with which the user prefers, but can't exactly quantify by how much one has improved with respect to the other.

Evaluation is an important consideration in semi-supervised domain adaptation, since, by definition, no labeled test (target domain) data is available. The type of comparative evaluation we performed could be instrumented into various end-user applications (for example, click-through logs from protein name search engines such as SLIF[2]) to automatically extract the necessary user-preference information, thus obviating the need of a special evaluator.

### 3.3.4 *Transfer: abstract to caption, full vs. ablated*

Having established that a model based on a combination of our new features (incorporated in the FULL model) improved user preference over the baseline, purely lexical model, we then performed an ablation study to ascertain which of these new features (structural frequency (FREQ), positive snippets (POS), or negative snippets (NEG)) were responsible for the improvements observed. Table 4 summarizes these results for each ablation considered. In each such study comparing the full model to a degraded model, the full model was preferred significantly more often than the ablated model (one-sided paired t-test, $p<.01$), indicating that our proposed features are, in fact, useful for unsupervised domain adaptation.

| Model | Compared to | p-value | # user labels |
|---|---|---|---|
| **FULL** | BASE | 3.6 E-4 | 182 |
| **FULL** | NEG_FREQ | 9.9 E-9 | 78 |
| **FULL** | POS_NEG | 1.8 E-4 | 120 |
| **FULL** | POS_FREQ | 1.1 E-4 | 46 |

**Table 4: Summary of transfer results for extractors trained on full papers and evaluated on *captions*. The preferred model is in bold.**

From these results we can further observe that adding POS snippets seems to have a noticeable effect on user preference. This is a nice complement to the result from §3.3.2 which indicated that POS snippets are not as useful in the non-transfer setting. Indeed, it is the ability of POS snippets to shape the labeled training source data to look more like the target data that allows the extractors so trained to be robust across shifts in domains. Similar user preference is seen for the contribution of NEG snippets and FREQ features, indicating that they too aid in domain-adaptation, both by leveraging unlabeled training data and by helping to inform the training data with some target domain attributes.

[2]http://slif.cbi.cmu.edu/

## 4. CONCLUSIONS & FUTURE WORK

In this work we have shown how exploiting structure, in the form of frequency features and positive and negative snippets, can help in the problem of semi-supervised domain adaptation. We have defined a new set of features based on structural frequency statistics and demonstrated their utility in representing inter-domain information drawn from both supervised and unsupervised sources, in a manner somewhat orthogonal to the traditional lexically based feature sets. Similarly, we have defined a technique for introducing high-confidence positively and negatively labeled pseudo examples (snippets) from the target domain into the source domain, and shown that these too provide a convenient, and effective, method for producing an extractor that is robust to domain shifts between training and testing data sets.

Finally, through a comparative analysis of each new feature's contribution to same-domain and inter-domain information extraction performance, we have discovered an intriguing relationship between a feature's utility in the non-transfer and transfer settings. We hope to exploit this relationship to help more systematically assess the relative domain-specificity of certain classes and combinations of features.

More generally, we would like to further examine and characterize the particular relationships between features and models that facilitate good transfer learning, along with the more abstract quality of robustness. In particular, we would like to develop more automated techniques for finding features and representations that are generally robust to shifts in domain and feature spaces.

## 5. ACKNOWLEDGMENTS

The work described here was supported in part by National Institutes of Health grant R01 GM078622.

## 6. REFERENCES

[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. In *JMLR 6*, pages 1817 – 1853, 2005.

[2] A. Arnold, R. Nallapati, and W. W. Cohen. A comparative study of methods for transductive transfer learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM) 2007 Workshop on Mining and Management of Biological Data*, 2007.

[3] A. Arnold, R. Nallapati, and W. W. Cohen. Exploiting feature hierarchy for transfer learning in named entity recognition. In *ACL:HLT '08*, 2008.

[4] J. Baxter. A Bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28(1):7–39, 1997.

[5] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS 20*, Cambridge, MA, 2007. MIT Press.

[6] D. M. Blei, J. A. Bagnell, and A. McCallum. Learning with scope, with application to information extraction and classification. In *UAI*, pages 53–60, 2002.

[7] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *EMNLP*, Sydney, Australia, 2006.

[8] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, pages 92–100, 1998.

[9] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[10] W. W. Cohen. Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. http://minorthird.sourceforge.net, 2004.

[11] W. W. Cohen, R. Wang, and R. Murphy. Understanding captions in biomedical publications. In *KDD*, pages 499–504, 2003.

[12] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[13] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. In *Journal of Artificial Intelligence Research 26*, pages 101–126, 2006.

[14] K. Franzén, G. Eriksson, F. Olsson, L. Asker, P. Lidén, and J. Cöster. Protein names and how to find them. In *International Journal of Medical Informatics*, 2002.

[15] W. A. Gale, K. W. Church, and D. Yarowsky. One sense per discourse. In *HLT '91: Proceedings of the workshop on Speech and Natural Language*, pages 233–237, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[16] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *CAP*, Nice, France, 2005.

[17] J. Jiang and C. Zhai. Exploiting domain structure for named entity recognition. In *Human Language Technology Conference*, pages 74 – 81, 2006.

[18] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[19] 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. In *KSCE*, 2004.

[20] National Institues of Health. http://www.pubmedcentral.nih.gov/.

[21] T. Ohta, Y. Tateisi, H. Mima, and J. Tsujii. Genia corpus: an annotated research abstract corpus in molecular biology domain. In *HLT: Human Language Technology Conference*, pages 92–100, 2002.

[22] B. Schölkopf, F. Steinke, and V. Blanz. Object correspondence as a machine learning problem. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 776–783, New York, NY, USA, 2005. ACM.

[23] L. Shi and F. Campagne. Building a protein name dictionary from full text: a machine learning term extraction approach. *BMC Bioinformatics*, 6(88), 2005.

[24] C. Sutton and A. McCallum. Composition of conditional random fields for transfer learning. In *HLT/EMLNLP*, 2005.

[25] B. Taskar, M.-F. Wong, and D. Koller. Learning on the test data: Leveraging 'unseen' features. In *Proc. Twentieth International Conference on Machine Learning (ICML)*, 2003.

[26] S. Thrun. Is learning the $n$-th thing any easier than learning the first? In *NIPS*, volume 8, pages 640–646. MIT, 1996.

[27] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis, 2005.

[28] X. Zhu. Semi-supervised learning literature survey. In *Technical Report 1530*. University of Wisconsin, 2005.

[29] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.