Increasingly, the amount of biological data challenges our newly discovered sequences grows at a much slower speed. However, human effort in annotating protein sequences is deposited into UniProtKB, they are often associated with PubMed abstracts, and the abstracts can provide additional information to predict the protein subcellular localization. Our work focuses on extracting and predicting protein subcell labels from a query protein’s associated PubMed abstracts. We explore two categories of methods for this task: match-and-resolve and supervised classification. In the match-and-resolve approach, we first match the original PubMed abstracts as well as the recognized biomedical named entities with GeneOntology (GO) terms and synonyms; we then resolve the matched terms among the GO hierarchy to their corresponding subcell labels. In supervised classifications, we classify proteins based on features extracted from abstracts: bag-of-words and MeSH terms from abstracts, as well as GO terms extracted using the GOPubMed algorithm. In general, supervised classification outperforms the match-and-resolve approach. However, supervised classification is limited by the availability of training data as well as the size of the feature space, while match-and-resolve is always applicable to proteins annotated with PubMed abstracts.

Keywords
proteome annotation, subcellular localization prediction, biomedical text mining, associative classifier, support vector machine

1. INTRODUCTION

High throughput genome sequencing produces an explosion of biological data; however, human effort in annotating newly discovered sequences grows at a much slower speed. Increasingly, the amount of biological data challenges our ability to assimilate them; biologists therefore resort to computational methods for extracting useful information from data rich but information-poor protein sequences.

An important question in annotating novel protein is to predict a protein’s subcellular localization, the location where a protein functions within a living cell. Predicting a protein’s subcellular localization (subcell label) helps biologists elucidate protein functions and facilitates such biomedical applications as protein purification and drug target discovery. Many protein sequences in UniProtKB lack subcellular localization annotation but contain references to PubMed abstracts. It is likely that during the investigations for these proteins, biomedical researchers discover and express the knowledge about subcellular localization in the associated abstracts. However, protein annotators still need to go through all the related references to “rediscover” the query protein’s actual subcell label. Such task is tedious and time consuming; automating the process of assigning labels to novel proteins (followed by human verification) would greatly increase the efficiency of protein annotation for biomedical research.

Our work focuses on extracting and predicting subcell labels from a protein’s related PubMed entries. We tackle the problem with two different approaches: match-and-resolve and supervised classification. In both approaches, we use the GeneOntology (GO) extensively both as a biological thesaurus and a well-organized biomedical concept hierarchy. Given a query protein, we first retrieve its associated titles, abstracts and MeSH (Medical Subject Heading) terms from PubMed, and then extract various features from the retrieved data and classify proteins to subcell labels using machine-learned classifiers. We can also find the mentioned GO terms from the PubMed titles and abstracts using direct string matching, named entity recognition (NER)-based matching or the GO term extraction method [4]. After extracting the mentioned GO terms, we can map them to subcell labels by subcell label resolution in the GO hierarchy. Figure 1 presents a graphical summary of our approaches.

This paper is organized as follows. We first review the related work in Section 2. We then describe our methodologies in Section 3, and present the results for different approaches in Section 4. We discuss some interesting issues in Section 5, and finally we conclude our findings in Section 6.

2. RELATED WORK

Identifying the localization of proteins is key to understanding their function within cells. However, given the increasing number of discovered proteins, human annotation
alone is not sufficient. In the past two decades, biomedical researchers proposed many computational methods to predict protein subcellular labels based on primary sequence, amino acid composition and UniProt annotation texts [11]. With the increasing number of digitalized biomedical publications, many recent approaches were proposed to mine useful information about proteins (e.g. subcellular localization, GO terms) from biomedical text repository (e.g. PubMed) [13] [6].

A straightforward method for protein localization prediction is to first extract the gene and protein names related to the query protein, and then derive the localization from these extracted entities using an ontology knowledge base. A number of systems have been developed for this task — interested readers can consult [2] for a more detailed introduction.

Instead of direct matching, other approaches have been proposed to predict protein localization based on a labeled training set. Höglund et al. [7] predicted protein localization by Support Vector Machine (SVM) classifiers based on most distinguished terms extracted from PubMed abstracts. They achieved 72%-76% accuracy for their plant and animal datasets using only text features. Fyshe et al. [6] also predicted subcellular localization by binary SVM classifiers but based on bag-of-words features weighted by tf.idf. Fyshe et al. improved the classification results by adding GO synonyms and ancestral GO terms to the bag-of-words features (Synonym Resolution and Term Generalization). They achieved 94%-96% accuracy on the Proteome Analyst datasets. Fyshe et al. also proposed a PubMed abstract filtering strategy based on an abstract’s referencing protein’s subcellular localizations. For more details about abstract filtering see [6]. GOPubMed organizes PubMed entries using the GO hierarchy. Delfs et al. [4] extracted the mentioned GO terms from PubMed abstracts by first matching general GO terms and further refining and expanding the matching in the GO hierarchy.

A number of supervised classification methods have been employed for the protein localization prediction task, including SVM, Neural Networks and Bayesian Networks [11]. Particularly noteworthy is associative classification, a relatively novel classification method developed by the data mining community [5]. Associative classifiers are considered to be more transparent than SVM, and more accurate and efficient than decision trees. Vadhi and Zaâine used associative classifiers to identify extracellular plant proteins employing sequence-based classification features called the “partition-based subsequences”; they achieved 98.83% F-measure on their plant dataset [8].

Our approaches differ from the above approaches in the following aspects. First, in addition to bag-of-words, we also use extracted biomedical named entities and extract GO terms as classification features. Second, we explore the match-and-resolve approach by directly mining the GO terms and resolving subcell labels. Third, we apply associative classification to the problem of protein subcellular localization with the potential for high accuracy and transparent explanation for predictions. Fourth, unlike Fyshe et al. who use binary SVM to classify whether the query protein belongs to a single subcellular localization, we use multi-class SVM to classify the query protein as one of many possible subcellular localizations. Finally, we do not rely on feature expansion (e.g. Synonym Resolution or Term Generalization) for our bag-of-words features.
3. METHODS

In this section, we introduce the bioinformatics databases and datasets used in our experiments and present methods for feature extraction from text and supervised classification.

3.1 Databases

We use three bioinformatics databases extensively in our project:

- **UniProtKB** [14], a protein sequence and annotation database containing over 10 million novel protein sequences and 0.5 million high quality human curated annotations.
- **PubMed** [12], an online biomedical literature database, containing over 14 million newly published and legacy biomedical journal abstracts.
- **GeneOntology (GO)** [3] provides a controlled vocabulary and concept hierarchy of biological terms to reflect the working knowledge of current biomedical discoveries.

All protein annotations, including their associations with PubMed entries, were extracted from SwissProt (part of UniprotKB) version 51.3. PubMed titles, abstracts and MeSH terms were downloaded from the NCBI PubMed website. The GO hierarchy was constructed using the GeneOntology OBO flat-file version 1.2.

3.2 Feature Extraction

We extract four types of features from PubMed entries: bag-of-words, MeSH terms, biomedical named entities and mentioned GO terms.

**Bag-of-words**

Given a query protein, we concatenate titles and abstracts for all associated PubMed entries into a body of text. We remove standard English stop words and tokenize the text by white space. We then stem each text token using the Porter stemmer\(^1\). The stemmed tokens form the bag-of-words features for the query protein.

We weight each feature according to the following methods: for associative classification, features are weighted by their presence or absence, i.e. a feature receives weight 1 if it is present and weight 0 if it is absent; for SVM, features are weighted with presence/absence and their \(tf.idf\) (Term Frequency - Inverse Document Frequency) value. \(tf.idf\) is defined as:

\[
\text{tf.idf}_{i,j} = \frac{tf_{i,j}}{\sum_k tf_{k,j}} \log \frac{N}{df_i}
\]

where \(tf_{i,j}\) is the frequency of term \(i\) in the PubMed abstracts of protein \(j\), \(\sum_k tf_{k,j}\) is the total number of terms in all PubMed abstracts \(k\) referenced by protein \(j\), \(N\) is the total number of proteins in the dataset, and \(df_i\) is the number of proteins with term \(i\) in their referencing PubMed abstracts.

**Biomedical Named Entities**

Biomedical named entities are specialized terms used in the biomedical literature. They may be directly or indirectly related to a protein’s subcellular localization, and can be

\(^1\)http://tartarus.org/~martin/PorterStemmer

used to help with subcell label prediction. Similar to bag-of-words feature generation, we concatenate related titles and abstracts of each protein into a body of text, and then recognize biomedical named entities using the Named Entity Resolution toolkit in LingPipe\(^2\). In supervised classifications, named entities are weighted by their presence or absence.

**MeSH Terms**

Tokenized MeSH terms are simply extracted from PubMed entries without modification. MeSH term features are weighted by their presence or absence.

**Mentioned GO Terms**

Using the hierarchical structure of the GeneOntology, we can infer the localization labels of a given GO term by recursively examining the GO term’s ancestors in the ontology hierarchy. Finding related GO terms for each protein helps localization prediction. Mentioned GO terms are GO terms that match text segments in PubMed titles and abstracts. We implemented the algorithm proposed in GOPubMed [4] for extracting mentioned GO terms for both subcell label resolution and associative classification. In supervised classifications, GO terms are weighted by their presence or absence.

3.3 Supervised Classification

For supervised classification, we used two types of associative classifiers (CMAR [9] and CPAR [15]) and linear kernel SVM. Associative classification is a novel supervised machine learning method that combines frequent item set mining and association rule generation. An associative classifier finds the features that often co-occur with class labels, and generates classification rules mapping features to class labels. The resulting rules may be pruned to reduce the model size and to increase classification speed as well as accuracy. Associative classification has recently gained popularity thanks to its efficiency and prediction transparency. The technical details of associative classification are beyond the scope of this paper; interested readers are encouraged to consult the survey [5] for more details. SVM classifiers are also popular for classification tasks, and can handle large feature spaces. We include SVM classification results for comparison to associative classifications.

For our experiments, we adapted the source code of the CMAR (Classification based on Multiple Association Rules) and CPAR (Classification based on Predictive Association Rules) implementations created by Frans Coenen from the University of Liverpool\(^3\). We used the LIBSVM package by Chang et al.\(^4\) for SVM classification. All cross validation results in this paper were optimized with the screening of optimal parameter settings within our limit of computational power. For CMAR, we varied support from 0.1%, 1% and 5% to 95% with 5% increments and confidence from 10% to 90% with 10% increments. For CPAR, we used only the top 5 rules for classification (\(K = 5\)) and varied the minimum best gain from 0.1 to 0.7 with 0.1 increment and gain similarity ratio from 0.07 to 0.99 with 0.05 increments. For LIBSVM,

\(^2\)LingPipe: http://alias-i.com/lingpipe/

\(^3\)The CMAR and CPAR source codes are available at http://www.csc.liv.ac.uk/~frans/KDD/Software/.

\(^4\)LIBSVM http://www.csie.ntu.edu.tw/~cjlin/libsvm/
we tried linear, polynomial and radial basis function kernels and found that linear kernel was the most robust with the highest accuracy. Using linear kernel, we screened for optimal settings of the cost (C) and kernel (γ) parameter using LIBSVM’s parameter screening utility.

3.4 GO Term Matching

For extracting mentioned GO terms, we match text segments from PubMed abstracts with GO terms using three different methods: direct string matching, the GOPubMed algorithm, and matching based on named entity recognition (NER).

**Direct String Matching (baseline)**

Direct string matching is used as our baseline. It finds the GO terms by matching the protein’s associated PubMed abstracts with GO term descriptions and synonyms. Instead of performing a character-by-character or word-by-word matching, we build an inverted index to speed up the matching process. The first step is to build an inverted index over all words in PubMed abstracts. Then we go through the entire GO flat-file database and find all the proteins whose abstracts contain the GO terms (the actual matching process). Finally, we re-organize the matching results and retrieve all the GO terms for each protein.

**GOPubMed Term Matching**

GOPubMed term matching is an approach proposed by Delfs et al. [4] in the GOPubMed project. Actual texts seldom match the GO terms perfectly. For example, PubMed entry 1274796 contains text “cAMP-dependent kinase”, which corresponds to the GO-term “cAMP-dependent protein kinase activity”. In order to handle such cases, GOPubMed first matches the terms from the rightmost (the most general) words, and finds short GO terms as seeds. These seeds are expanded using regular expressions, and then used to locate more specific and longer GO terms. For “cAMP-dependent kinase”, we first find the GO term “kinase activity”, and expand it using regular expression pattern “cAMP-dependent.* kinase activity”, which matches the maximal GO-term “cAMP-dependent protein kinase activity”. We re-implemented the GOPubMed term matching algorithm for our experiments.

**NER-based Matching**

Most text segments are unrelated to GO terms, and may not be useful for matching GO terms. Based on this intuition, we explore the NER-based matching approach to extract GO terms. NER-based matching first extracts the biomedical named entities using a named entity resolution method, and then matches the extracted named entities against the GO terms. In our experiments, we used the LingPipe NER module, which had been trained on the GENIA biomedical corpus⁵.

3.5 Subcell Label Resolution

After matching GO terms, we resolve them among the GO hierarchy. More specifically, subcell label resolution means that given a GO term, we recursively examine its ancestors until we reach one or more of the target GO terms and predict the corresponding class label, or reach the root of the GO hierarchy and make no predictions. Table 1 lists the target GO terms and their corresponding class labels used in our subcell label resolution.

<table>
<thead>
<tr>
<th>Class Label</th>
<th>Target GO Terms</th>
<th>Organism Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>cytoplasm</td>
<td>GO:0005737</td>
<td>all</td>
</tr>
<tr>
<td>endoplasmic reticulum</td>
<td>GO:0005783</td>
<td>AN</td>
</tr>
<tr>
<td>extracellular</td>
<td>GO:0005576</td>
<td>AN</td>
</tr>
<tr>
<td>gootide</td>
<td>GO:0005794</td>
<td>AN, FU</td>
</tr>
<tr>
<td>lysosome</td>
<td>GO:0005726</td>
<td>AN</td>
</tr>
<tr>
<td>mitochondrion</td>
<td>GO:0005739</td>
<td>AN</td>
</tr>
<tr>
<td>nucleus</td>
<td>GO:0005644</td>
<td>AN, PL, FU</td>
</tr>
<tr>
<td>peroxisome</td>
<td>GO:0005777</td>
<td>AN, PL, FU</td>
</tr>
<tr>
<td>plasma membrane</td>
<td>GO:0005886</td>
<td>all</td>
</tr>
<tr>
<td>vacuole</td>
<td>GO:0005777</td>
<td>AN, PL, FU, GN</td>
</tr>
<tr>
<td>inner membrane</td>
<td>GO:0005886</td>
<td>GN</td>
</tr>
<tr>
<td>outer membrane</td>
<td>GO:0009867</td>
<td>GN</td>
</tr>
<tr>
<td>periplasm</td>
<td>GO:0042597</td>
<td>GN</td>
</tr>
</tbody>
</table>

3.6 Datasets

We evaluate our approaches using the Proteome Analyst [10] datasets created by Fyshe et al. (2008)⁶. Table 2 summaries the datasets, showing the number of classes, proteins, retrieved PubMed abstracts, and total abstract word counts for each organism type. In the rest of this paper, organism types are referenced by their corresponding abbreviations: Animal as AN, Plant as PL, Fungi as FU, Gram-positive Bacteria as GP, and Gram-negative Bacteria as GN.

<table>
<thead>
<tr>
<th>Organism Type</th>
<th>Class</th>
<th>Protein</th>
<th>PubMed</th>
<th>Word Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>9</td>
<td>12,261</td>
<td>53,549</td>
<td>11 million</td>
</tr>
<tr>
<td>PL</td>
<td>9</td>
<td>3,387</td>
<td>6,782</td>
<td>1.5 million</td>
</tr>
<tr>
<td>FU</td>
<td>9</td>
<td>2,651</td>
<td>15,752</td>
<td>3.2 million</td>
</tr>
<tr>
<td>GP</td>
<td>5</td>
<td>2,594</td>
<td>4,991</td>
<td>0.9 million</td>
</tr>
<tr>
<td>GN</td>
<td>5</td>
<td>6,440</td>
<td>15,598</td>
<td>3.4 million</td>
</tr>
</tbody>
</table>

3.7 Evaluation Metric

We evaluate the classification results of associative classifiers and SVM using stratified 10-fold cross validation. We evaluate the performance of match-and-resolve methods and supervised classifications using precision, recall, F-measure and percent accuracy as defined below:

\[
\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad \text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},
\]

Table 1: Target GO terms for subcell label resolution.

Table 2: Proteome Analyst dataset statistics.

---

⁵http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

⁶The datasets are available at http://webdocs.cs.ualberta.ca/~bioinfo/nlp/.
Table 3: Overall F-measure values for the match-and-resolve approach with different GO term matching methods. Best results for each organism type (excluding upper bounds) are shown in bold; we follow this convention throughout the paper.

<table>
<thead>
<tr>
<th>Organism Type</th>
<th>Direct Match. (baseline)</th>
<th>NER-based Matching</th>
<th>GOPubMed Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>0.212</td>
<td>0.183</td>
<td>0.298</td>
</tr>
<tr>
<td>PL</td>
<td>0.562</td>
<td>0.354</td>
<td>0.586</td>
</tr>
<tr>
<td>FU</td>
<td>0.273</td>
<td>0.202</td>
<td>0.387</td>
</tr>
<tr>
<td>GP</td>
<td>0.154</td>
<td>0.037</td>
<td>0.782</td>
</tr>
<tr>
<td>GN</td>
<td>0.274</td>
<td>0.205</td>
<td>0.161</td>
</tr>
<tr>
<td>Average</td>
<td>0.296 (±0.156)</td>
<td>0.196 (±0.112)</td>
<td>0.443 (±0.244)</td>
</tr>
</tbody>
</table>

Accuracy = \(\frac{TP + TN}{TP + FP + FN + TN}\),

where TP stands for True Positive, FP stands for False Positive, FN stands for False Negative, and TN stands for True Negative.

For evaluating whether the differences between alternative methods or feature sources are statistically significant, we apply two-sample t-test (95% confidence) with Welch correction. In the remainder of this paper, “significant” should be understood as “statistically significant”. We provide p-values for t-tests where the results are significant.

4. RESULTS

In this section, we present the results for both match-and-resolve and supervised classifications.

4.1 Match-and-resolve

Table 3 shows the overall F-measure values using various match-and-resolve approaches on different datasets. Baseline F-measure is very low for most organism types. A careful examination of the results reveals that average baseline precision is 0.466, while the average recall is only 0.224. The low recall is partly caused by the “no prediction rate”, which is the percentage of proteins without any predictions. The average “no prediction” rate is 62%. We found that the higher the “no prediction” rate, the lower the recall.

To our surprise, the result corresponding to NER-based matching is much worse than the baseline, which can be explained by two factors. First, the training corpus covers only part of GO terms, while the accuracy of the NER approach is not high by itself (state-of-the-art method achieves roughly 0.75 F-measure); therefore NER errors are propagated to our matching process. Second, extracted named entities do not have word contexts that may help with term matching. For example, for the GO term “protein amino acid phosphorylation”, the extracted named entity “amino acid” does not match it. This is confirmed by the high “no prediction” rate of 79.3%. As we expected, the GOPubMed algorithm performs better than the other two approaches for most organism types. The overall “no prediction” rate for GOPubMed algorithm is 52%, well below the rate of the other two approaches. However, the results corresponding to the GOPubMed method are not significantly better than either direct string matching or the NER-based matching.

4.2 Supervised Classification

Table 4 shows the overall F-measure values for associative classification with CMAR. The classification results with annotation GO terms (the GO terms extracted from the query protein’s UniProt annotation) represent the upper bound of performance, since annotation GO terms are just class labels in disguise. Among all the features we extracted from PubMed entries, MeSH terms achieve the best performance for AN, PL and FU, while bag-of-words features achieve the best performance for GP and GN. For GP, the difference between MeSH and bag-of-words is arguably small; for GN, bag-of-words is better than MeSH by 0.065 in terms of F-measure. We found that neither bag-of-words nor MeSH terms are significantly better than other feature sources. Neither bag-of-words features nor MeSH terms achieve significantly worse results than the upper bounds. On the other hand, both named entity (\(p = 0.022\)) and GOPubMed (\(p = 0.048\)) features achieve significantly worse results than the upper bounds.

Table 5 shows the overall F-measure values for associative classification with CPAR. For CPAR, bag-of-words features achieve the best performance for PL, GP and GN, while MeSH terms achieve the best performance for AN. Named entity features outperform other feature types in FU. Similar to the results with CMAR, we found that neither bag-of-words nor MeSH terms are significantly better than other features sources. Unlike classification results with CMAR, none of the four feature types are significantly worse than the upper bounds.

Table 6 shows the overall F-measure values for SVM classification results with various features. For bag-of-words, we have two different weighting schemas: binary and tf.idf as described in Section 3; other feature types are weighted using the binary schema. Bag-of-words features outperform all other feature types, but the differences in performance are not significant. Similar to the result with CPAR, none of the feature types are significantly worse than the upper bound.

4.3 Performance Comparison

Table 7 shows a comparison between the best performance of our subcell label extraction and predictions methods. We also compare our results to the best performance from Fyshe et al. (2008) with bag-of-words features plus feature expansions but without any abstract filtering. Figure 2 shows a graphical view of the comparison. We found that when using simple bag-of-words features without any feature expansion, multi-class SVM slightly outperformed the results from Fyshe et al. (2008) with sophisticated feature expansion in AN, PL and FU; for GP and GN, multi-class SVM is only slightly worse than the results of Fyshe et al. The differences in performance between multi-class SVM and the method of Fyshe et al. are not significant.

Comparing the performance of SVM and associative classifiers, we found that for all organism types, SVM classifiers outperformed associative classifiers, although for PL and GP, associative classifiers achieved very similar accuracy. The differences between CMAR, CPAR and multi-class SVM are not significant. In summary, associative classifiers are almost as good as SVM on certain datasets. Comparing the performance of supervised classifications and the match-and-resolve approach, we found that for all organism types, supervised classification methods outperformed match-and-resolve methods. All supervised classification methods we used in this paper significantly outperformed match-and-resolve methods (CMAR \(p = 0.048\), CPAR \(p = 0.031\), SVM \(p = 0.014\)).
5. DISCUSSION

In this section, we discuss the results of both supervised classification and match-and-resolve approaches.

5.1 Supervised Classification

In addition to employing GO terms present in the annotation or extracted from PubMed abstracts as classification features, we experimented with generalizing the current collection of GO terms by adding all ancestral terms (Term Generalization). Term Generalization was proposed by Fyshe et al. for expanding the feature set of classification instances; we use the technique to increase the likelihood of proteins with same subcell label sharing common GO terms. For example, protein P1 is associated with GO term “extracellular matrix” (GO:0031012), while protein P2 is associated with GO term “extracellular space” (GO:0043245). “Extracellular matrix” and “extracellular space” are sibling GO terms under a common parent “extracellular region part” (GO:0044421). We hypothesized that without Term Generalization, such terms would have no GO terms in common, thus preventing the associative classifiers from recognizing their association. To our surprise, we found that generalizing GO terms all the way to the GO root node actually decreases the F-measure of association classification with extracted GO terms by 0.01 – 0.07. We speculate that Term Generalization adds too many common GO terms to the feature set, which makes proteins with different labels share common GO terms, and therefore degrades the classification accuracy. In future work, we plan to verify this hypothesis by limiting the generalization through adding only the parent or “grand-parent” terms instead of all ancestral terms.

5.2 Feature Selection for Bag-of-words

The idea of association rule mining in associative classification was initially developed for market basket analysis in data mining. Associative classifiers were not designed to handle large feature space. Their classification performance depends on the discovery of strong associations between features and class labels, which in turn depends on the result of frequent item set mining. When the feature space contains thousands or millions of features, the frequent item set mining procedure becomes computationally expensive or even intractable. Therefore, associative classifier is not particularly suitable for feature types with large feature space such as features generated by the bag-of-words or named entity approaches. However, when the feature space is small enough, as in GP, associative classifier yields comparable performance with SVM.

In order to reduce the feature space for the bag-of-words approach, we use the following feature selection method. We first calculate the $tf.idf_{i,j}$ values for each term $i$ and for each referencing protein $j$ in the bag-of-words feature space. As shown in the definition, a term could have different $tf.idf$ values when referenced by different proteins. We sum the $tf.idf_{i,j}$ values for the same term $i$ under all referencing proteins $j$ to produce the term ranking score $tf.idf_i$:

$$tf.idf_i = \sum_j tf.idf_{i,j} = \left( \sum_j \frac{t_j f_{i,j}}{\sum_k t_j f_{k,j}} \right) \log \frac{N}{df_i}$$  

We then sort the terms (features) according to their ranking score in non-increasing order and select the topmost $x\%$ features with the highest $tf.idf_i$ ranking score. The intu-
Figure 3: The effect on overall F-measure values for different percentages of features retained using sum tf.idf feature selection on the plant organism type.

Table 6: Overall F-measure values for SVM Classification with various features

<table>
<thead>
<tr>
<th>SVM</th>
<th>Bag-of-words binary weighted</th>
<th>Bag-of-words tf.idf weighted</th>
<th>MeSH</th>
<th>Named Entities</th>
<th>GO terms</th>
<th>Annotation GO terms (upper bound)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>0.793</td>
<td>0.854</td>
<td>0.812</td>
<td>0.774</td>
<td>0.703</td>
<td>0.894</td>
</tr>
<tr>
<td>PL</td>
<td>0.827</td>
<td>0.868</td>
<td>0.861</td>
<td>0.834</td>
<td>0.802</td>
<td>0.933</td>
</tr>
<tr>
<td>FU</td>
<td>0.750</td>
<td>0.720</td>
<td>0.603</td>
<td>0.553</td>
<td>0.608</td>
<td>0.709</td>
</tr>
<tr>
<td>GP</td>
<td>0.946</td>
<td>0.948</td>
<td>0.947</td>
<td>0.946</td>
<td>0.947</td>
<td>0.999</td>
</tr>
<tr>
<td>GN</td>
<td>0.842</td>
<td>0.874</td>
<td>0.849</td>
<td>0.849</td>
<td>0.860</td>
<td>0.964</td>
</tr>
<tr>
<td>Average</td>
<td>0.791 (±0.146)</td>
<td>0.823 (±0.083)</td>
<td>0.814 (±0.128)</td>
<td>0.791 (±0.147)</td>
<td>0.780 (±0.132)</td>
<td>0.899 (±0.113)</td>
</tr>
</tbody>
</table>

Table 7: Best performance for each subcell label extraction and prediction methods, with comparison to the best result from Fyshe et al. 2008 using the same set of PubMed abstracts without any abstract filtering.

<table>
<thead>
<tr>
<th>Organism Type</th>
<th>Match-and-resolve</th>
<th>CMAR</th>
<th>CPAR</th>
<th>Multi-class SVM</th>
<th>Fyshe et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN</td>
<td>0.298</td>
<td>0.723</td>
<td>0.717</td>
<td>0.854</td>
<td>0.849</td>
</tr>
<tr>
<td>PL</td>
<td>0.586</td>
<td>0.821</td>
<td>0.851</td>
<td>0.868</td>
<td>0.842</td>
</tr>
<tr>
<td>FU</td>
<td>0.387</td>
<td>0.527</td>
<td>0.580</td>
<td>0.720</td>
<td>0.710</td>
</tr>
<tr>
<td>GP</td>
<td>0.792</td>
<td>0.947</td>
<td>0.923</td>
<td>0.948</td>
<td>0.950</td>
</tr>
<tr>
<td>GN</td>
<td>0.274</td>
<td>0.741</td>
<td>0.830</td>
<td>0.874</td>
<td>0.886</td>
</tr>
<tr>
<td>Average</td>
<td>0.469 (±0.219)</td>
<td>0.752 (±0.154)</td>
<td>0.780 (±0.134)</td>
<td>0.853 (±0.083)</td>
<td>0.847 (±0.088)</td>
</tr>
</tbody>
</table>

We found that removing the default subcell label when more specific subcell labels are predicted improves the performance of subcell label resolution. For example, in animal, plant or fungi cells, organelle “mitochondrion” is physically surrounded by the major subcellular fluid “cytoplasm”; hence in the GO hierarchy, GO node “mitochondrion” (GO:0005739) is logically part of “cytoplasm” (GO:0005737). Any protein with prediction “mitochondrion” is also labelled with “cytoplasm”. However, according to protein annotation conventions, a protein is labeled as cytoplasmic protein only when it is localized in “cytoplasm” but outside all membrane-bounded organelles, such as “mitochondrion”. Therefore, removing the default label “cytoplasm” when a protein is also predicted as “mitochondrion” is deemed appropriate; the label “cytoplasm” is only retained if it is the only predicted label. This subtle but important observation reduces false positives for the default label “cytoplasm”, hence increasing the F-measure for PL from 0.383 to 0.586 with GOPubMed GO term extraction. In general, removing default labels increases the overall F-measure by 0.04—0.20.

We evaluated the effectiveness of subcell label resolution separately to investigate its impact on the final performance of the match-and-resolve approach. We found that the resolution step is very effective by itself. Using the GO terms present in a protein’s Swiss-Prot (part of UniProtKB) annotation, we performed subcell label resolution to see whether we can recover the protein’s subcellular localization; this method is very similar to the match-and-resolve approach except that the GO terms are given in the query protein’s annotation instead of being extracted from PubMed abstracts. We achieved overall average accuracy 95.84% and average F-measure 0.7904 in resolving subcell labels with annotation GO terms. The result indicates that if perfect GO terms are found, we can effectively resolve them to their true corresponding subcell labels; such effectiveness is due to the use of the GO concept hierarchy in resolving subcell labels.

Nevertheless, we did not observe high F-measure values in our experiments for most match-and-resolve methods, indicating that our matching methods could not, in most cases, capture the perfect GO terms. In fact, the inability to rediscover the annotation GO terms from PubMed entries may
not necessarily be attributed to the defects of the matching approaches — the annotation GO terms in UniProtKB are assigned based on a protein’s human annotation, which may or may not reflect the content of the protein’s associated PubMed entries. Delfs et al. [4] show that only 56% PubMed abstracts contain one or more GO terms and on average only 1.76 terms are found per abstract. However, our experiments do not allow us to conclude with certainty whether the poor results are due to defects in our matching algorithms or to the absence of suitable GO terms in PubMed abstracts.

Since the match-and-resolve approach did not achieve good results in GO term extraction, we also tried predicting GO terms from text features such as bag-of-words and MeSH terms. As expected, the classification results were also poor due to the large number of class labels (GO terms). In conclusion, extracting GO terms from PubMed abstracts remains a difficult task.

5.4 Future Work

Increasing the number of training instances would likely improve the localization prediction accuracy. While the amount of labeled training data is scarce and unlabeled data is abundant, we can use the co-training [1] technique to annotate unlabeled data using high accuracy prediction methods before training our supervised classifiers. We will further evaluate our approaches with other publicly available datasets, and compare our performance with other state-of-the-art systems such as MultiLoc [7].

6. CONCLUSION

We have described experiments with two different categories of methods for discovering protein subcellular localization from their associated PubMed abstracts: supervised classification and match-andResolve. In supervised classification, multi-class SVM outperformed associative classifiers CMAR and CPAR for all organism types, although the differences were not statistically significant; multi-class SVM also outperformed, for certain organism types, methods proposed by a previous study with sophisticated feature expansion. We have shown that associative classifiers can approach the accuracy of SVM when the feature space is small. In the match-and-resolve approach, subcell label resolution is effective if perfect GO terms are given; however, the difficulty of extracting GO terms from PubMed abstracts limits the accuracy of the match-and-resolve approach. All supervised classification methods outperformed match-and resolve methods with various feature sets. However, unlike match-and-resolve methods, supervised classifiers are limited by the availability of training data. Finally, we proposed an effective feature selection method for supervised classification with bag-of-words features; by retaining the topmost 20% features ranked by overall tf.idf values, we preserve classification accuracy while greatly improving classification efficiency.

7. ACKNOWLEDGMENTS

The authors would like to thank Sittichai Jiampojamarn for help with biomedical named entity recognition, Xiaodi Ke for help with associative classification, and the Proteome Analyst Research Group for sharing with us their dataset.

8. REFERENCES