BIOINFORMATICS

Vol. 1 no. 1 2003
Pages 1–10

Extracting Synonymous Gene and Protein Terms from Biological Literature

Hong Yu[1] and Eugene Agichtein[1]

[1]Department of Computer Science, Columbia University, New York, NY, USA

ABSTRACT

Motivation: Genes and proteins are often associated with multiple names. More names are added as new functional or structural information is discovered. Because authors can use any one of the known names for a gene or protein, information retrieval and extraction would benefit from identifying the gene and protein terms that are synonyms of the same substance.

Results: We have explored four complementary approaches for extracting gene and protein synonyms from text, namely the unsupervised, partially supervised, and supervised machine-learning techniques, and the manual knowledge-based approach. We report results of a large scale evaluation of these alternatives over an archive of biological journal articles. Our evaluation shows that our extraction techniques could be a valuable supplement to resources such as SWISSPROT, as our systems were able to capture gene and protein synonyms not listed in the SWISSPROT database.

Data Availability: The extracted gene and protein synonyms are available at http://synonyms.cs.columbia.edu/

Contact: {hongyu,eugene}@cs.columbia.edu

INTRODUCTION

Genes and proteins often have multiple names; as biological research progresses, additional names may be given for the same substance, or different names may be found to represent the same substance. For example, the protein *lymphocyte associated receptor of death* has several synonyms including *LARD*, *Apo3*, *DR3*, *TRAMP*, *wsl*, and *TnfRSF12*. Authors often use different names to refer to the same gene or protein across articles or sub-domains. Identifying these name variations would benefit information retrieval and information extraction systems. Recognizing the alternate names for the same substance would help biologists to find and use relevant literature.

Many biological databases such as GenBank[†] and SWISSPROT[‡] include synonyms; however, these databases may not be always up to date. We found that even biology experts disagree with some of the synonyms that were listed in the SWISSPROT database. Furthermore, to our knowledge, gene and protein synonyms and thesauri are mainly constructed by laborious manual curating and review. Therefore, it is desirable to automate this process due to the increasing number of discovered genes and proteins.

The problem of extracting gene and protein synonyms from text requires first to identify gene or protein names in the text, and then to determine whether these names are synonymous. In this work we focus on the second part of the problem, namely, on extracting pairs of gene or protein names that are considered to be synonyms of each other. We rely on existing state-of-the-art *taggers* for gene and protein name entity identification. Having identified the gene or protein entities in text, we can apply different methods for extracting the synonymous ones.

In this study we adapt and explore four novel complementary approaches for extracting synonymous gene and protein names from biological literature. We present an in-depth study of state-of-the-art techniques over a large collection of recent journal articles. We develop a scalable methodology for evaluating the quality of extracted synonyms. Our experimental results show that our techniques result in extracting novel gene and protein synonyms that were not present in the SWISSPROT database[§]. Our extraction techniques could be used to improve search and analysis of biological literature, and to aid human curators of biology resources.

BACKGROUND AND RELATED WORK

Extracting gene and protein synonyms from biological literature is an important problem with significant practical benefits. Synonymous gene and protein names represent the same biological substances. This might be recognized if the substances in question exhibit identical biological functions or the same gene or amino acid sequences[¶].

Recent computational linguistics research on synonym detection has mainly focused on detecting semantically related words rather than exact synonyms, by measuring

[†] http://www.psc.edu/general/software/packages/genbank/genbank.html

[‡] http://www.ebi.ac.uk/swissprot/

[§] All extracted gene and protein synonyms are available on the web at http://snowball.cs.columbia.edu/.

[¶] We found that some biologists espouse a more broad definition of synonyms (e.g., homology).

the similarity of surrounding contexts. For example, these approaches may identify "beer" and "wine" as related words because both have similar surrounding words such as "drink", "people", "bottle" and "make" (e.g., (Dagan et al., 1995; Li and Abe, 1998; Lin, 1998)). A different approach exploited WORDNET(Fellbaum, 1999), a large lexical database for English words, to evaluate semantic similarity of any two concepts based on their distance to other concepts that subsume them in the taxonomy (Resnik, 1995).

In the biomedical domain, most approaches for synonym identification appear to be restricted to the actual content of the strings in question, and ignore the surrounding context. One such approach used a semi-automatic method to identify multi-word synonyms in UMLS (the Unified Medical Language System, a large biomedical taxonomy (Humphreys and Lindberg, 1993)), by linking terms as candidate synonyms if they shared any words (Hole and Srinivasan, 2000). For example, the term "cerebrospinal fluid" leads to "cerebrospinal fluid protein assay." The candidate synonym terms then were evaluated by human curators. A different approach employed a trigram matching algorithm to identify similar multi-word phrases. In this study, the phrases are treated as documents made up of character trigrams. The "documents" are then represented in the vector space model, and similarity is computed as the cosine of the angle between the corresponding vectors (Wilbur and Kim, 2001). Several other systems (e.g., (Liu and Friedman, 2003; Pakhomov, 2002; Park and Byrd, 2001; Schwartz and Hearst, 2003; Yoshida et al., 2000; Yu et al., 2002)) applied rule-based, statistical or machine-learning approaches for mapping abbreviations to their full forms. To our knowledge, few studies have attempted to automatically identify synonymous relations among gene or protein abbreviations. In contrast, our techniques can identify synonyms of abbreviations as well as full names.

We approach the synonym extraction problem by applying *information extraction* and *text classification* techniques, where the desired *structured* information are the synonym pairs that are "hidden" in the biological literature. As we describe next, machine learning techniques have been successfully used to adapt text analysis systems to new domains. The following studies provided the foundation for our current work.

Information Extraction and Machine Learning

One of the major challenges in information extraction is the large amount of manual labor involved in constructing and tuning the extraction system. Many biological information extraction systems (e.g., (Friedman et al., 2001; Rindflesch et al., 2000; Thomas et al., 2000; Yu et al., 2002)) build upon domain knowledge and domain-specific rules. To reduce manual effort, one approach is to build a powerful and intuitive graphical user interface for training the system, so that domain experts can quickly adopt the system for each new task (Yangarber and Grishman, 1998). This approach still needs substantial manual labor to port the system to new domains.

Machine-learning methods are becoming increasingly popular in text analysis. These methods significantly reduce required manual labor by automatically acquiring rules from labeled and unlabeled data. For example, supervised machine learning techniques such as Support Vector Machines (SVMs) were found to be highly effective for text classification (Joachims, 1998). In previous work, other supervised learning techniques have been applied to information extraction from unstructured and semi-structured text (e.g., (Califf and Mooney, 1998; Kushmerick et al., 1997; Muslea et al., 1998; Soderland, 1999)). While supervised machine learning approaches usually need less manual labor than knowledge-based approaches, they still require significant manual effort because of their dependency on a manually *labeled* training corpus. Several approaches attempt to reduce manual effort in annotating the training corpus. For example, instead of tagging entire documents, one approach required only marking the documents as either relevant or irrelevant for the extraction task (Riloff, 1996). This approach requires less manual labor, but the effort involved is still substantial.

For this reason, the general *partially-supervised* approaches have become an attractive alternative. A partially-supervised system typically starts with a relatively small number of manually labeled examples and proceeds to acquire new training examples *automatically*. Some of the early applications of partially-supervised learning using *bootstrapping* include identifying word hyponyms and hypernyms in natural language text (Hearst, 1992), and word sense disambiguation (Yarowsky, 1995).

More recently, the *co-training* framework was proposed for combining unlabeled and labeled examples to boost performance of a learning algorithm (e.g., for web page classification (Blum and Mitchell, 1998)). A related approach was subsequently used for classifying named entities (e.g., company names) in text (Collins and Singer, 1999). A different approach for named entity classification used multi-stage bootstrapping (Riloff and Jones, 1999). Another variation of the bootstrapping technique was applied for disambiguating gene, protein, and RNA terms in biological literature (Hatzivassiloglou et al., 2001).

Bootstrapping has also been used for extracting structured *relations* from text. The partially supervised *DIPRE* method was proposed for automatically acquiring patterns and relations from the pages on the web (Brin, 1998). More recently, the *Snowball* information extraction system (Agichtein and Gravano, 2000) extended the basic *DIPRE* method by incorporating automatic pattern and tuple evaluation for extracting relations from large text collections. A similar bootstrapping-based technique was independently developed for traditional information extraction (Yangarber et al., 2000).

In this study, we explore four complementary approaches to extracting gene and protein synonyms from text. In the next section we describe our implementations of the *unsupervised*, *partially-supervised*, *supervised*, and the *hand-constructed* systems that we developed for extracting gene and protein synonyms from biological literature.

EXTRACTING SYNONYMOUS GENE AND PROTEIN TERMS

We now present our approaches for extracting synonymous genes and proteins from biological literature. To extract these synonyms we first need to identify, or *tag*, the genes and proteins as they appear in the text. This task is accomplished by pre-processing the corpus. Having identified the gene and protein entities, we can apply complementary approaches for determining which of the entities are synonyms of each other. First, we outline an *unsupervised* approach. Next, we describe our adaptation of *Snowball*, a *partially-supervised* information extraction system. We then present a *supervised* machine learning method based on a state-of-the-art automatic text classification tool. Next, we describe the extraction system that was *manually constructed* by a biology expert. Finally, we present a new *combined* system, where the output of the manually constructed system is augmented with the output of the machine learning-based systems.

Pre-processing: Gene and Protein Tagging

Identifying gene or protein names in biological literature has been a rich area of research. As part of our system development we compared three alternative gene taggers: the tagger described in (Fukuda et al., 1998), the tagger described in (Proux et al., 1998), and *Abgene* (Tanabe and Wilbur, 2002). We emphasize that the purpose of our study was not to systematically evaluate gene taggers, but to identify the one that is best suited for our synonym identification task. We examined all three taggers over our training corpus and found that *Abgene* performed best for our extraction task. Therefore, we use *Abgene* as the tagger of choice for all of the subsequent experiments.

We also observed that gene or protein synonyms usually occur within the same *sentence*. Therefore, we *segment* the corpus into sentences using the publicly available *SentenceSplitter* system[‖]. After sentence segmentation, only the pairs of genes that appear within the same sentence will be considered as potential synonyms by any of the following extraction techniques. Additionally, we observed that the authors tend to specify gene and protein synonyms in the first few pages of the article. Therefore, our systems examined only the first 4Kb of text of each article for potential synonyms.

Contextual Similarity: An Unsupervised Approach

We adopted the synonym detection method –to which we will refer as **Similarity**– for identifying synonyms based on *contextual similarity*(Dagan et al., 1995). We chose **Similarity** over the others (e.g., (Li and Abe, 1998),(Lin, 1998)) because **Similarity** has been shown to be both *robust* and *general* (Dagan et al., 1995).

The contextual similarity approach finds sets of words that appear in similar contexts. The main observation is that synonyms of a word t can be detected by finding words that appear in the same contexts as t. If the contexts of t_1 and t_2 are *similar*, then t_1 and t_2 are considered synonyms. More formally, we define the *context* of a term t as all words that occur within a d word window from t (e.g., $d = 5$)[**]. In order to separate chance co-occurrence from the words that tend to appear together, the method uses *mutual information* to weight each word w in the context of t. The mutual information $I(t, w)$ is defined as $log_2(\frac{P(t,w)}{P(t)\cdot P(w)})$, and calculated as:

$$I(t, w) = log_2(\frac{N}{d} \cdot \frac{freq(t, w)}{freq(t) \cdot freq(w)})$$

where N is the size of the corpus in words, and d is the size of the window. Note that $I(t, w) \neq I(w, t)$ because $freq(t, w)$ (i.e., the number of times w appears to the *right* of t) is not symmetric. Using mutual information, we can now define the similarity *Sim* between two terms t_1 and t_2, based on their respective contexts as:

$$\frac{\sum_{w \in lexicon} min(I(w, t_1), I(w, t_2)) + min(I(t_1, w), I(t_2, w))}{\sum_{w \in lexicon} max(I(w, t_1), I(w, t_2)) + max(I(t_1, w), I(t_2, w))}$$

where w ranges over the complete *lexicon* of all of the words that appear in the respective contexts of t_1 and t_2. The value of the similarity *Sim(*t_1, t_2*)* indicates whether t_1 and t_2 are synonyms.

It is not feasible to compute $Sim(t_1, t_2)$ for all choices of t_1 and t_2, since this would require *O(*$|$*lexicon*$|^3$*)* running time. We implemented the heuristic search algorithm to compute a close approximation of the set of most similar terms for a given term t_1 (Dagan et al., 1995). Figure 1 reports some of the synonym sets extracted by **Similarity** from a biological journal archive.

The original **Similarity** method (Dagan et al., 1995) was designed to find contextual synonyms for all words in the corpus. In contrast, we are only interested in computing

[‖] Available from http://l2r.cs.uiuc.edu/~cogcomp/index_research.html.

[**] We distinguish the *left* context of t (i.e., words that appear within d words *before* t), and the *right* contexts of t because of asymmetry of English grammar. For clarity of the presentation, we omit this distinction.

Term t_1	*List of the top ranked synonyms t_2 together with Sim(t_1, t_2)*
multigene	superfamily (0.94) subfamily (0.92) subclass (0.89)
definite	unambiguous (0.94) unequivocal (0.92) rigorous(0.88)
question	intriguing (0.98) possibility (0.94) issue (0.90)

Fig. 1. Some similar term sets extracted by the **Similarity** system from a biological journal archive.

synonyms of *gene* and *protein* terms. Therefore, the **Similarity** system uses a modification of the search algorithm to only search for contextual similarity between terms g_1 and g_2 if both g_1 and g_2 were tagged by the *Abgene* tagger as genes. The confidence *Conf(s)* of a candidate synonym pair $s(g_1, g_2)$ is simply the value of similarity $Sim(g_1, g_2)$. We consider only the top k most similar terms for each term g_1 (we set $k = 5$).

While the unsupervised approach is attractive because it does not require manual training, many of the extracted gene and protein pairs are likely to be false positives. Therefore, we would like to incorporate some domain knowledge without requiring significant manual effort. With this goal in mind, we adapted a partially supervised information extraction system for our synonym identification problem.

Snowball: A Partially-Supervised Approach

The *Snowball* system (Agichtein and Gravano, 2000) uses a *bootstrapping* approach for extracting structured *relations* from unstructured (natural language) text. *Snowball* was designed to operate over large text collections and to require minimal human input. As shown in Figure 2, *Snowball* starts with a small set of user-provided seed tuples for the relation of interest, and automatically generates and evaluates patterns for extracting new tuples. In our study, the relation to be extracted is *Synonym (Gene1, Gene2)*.

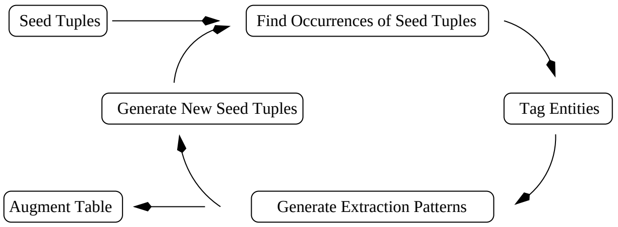


Fig. 2. The architecture of *Snowball*, a partially-supervised information extraction system.

As initial input, *Snowball* only requires a set of user-provided *seed* (i.e., example) tuples in the target relation (i.e., a set of known gene or protein synonym pairs). For this problem, we extended *Snowball* to also make use of *negative* examples (i.e., co-occurring genes and protein expressions known *not* to be synonyms of each other). *Snowball* then proceeds to find occurrences of the positive seed tuples in the collection. These occurrences are converted into *extraction patterns*, which are subsequently used to extract new tuples from the documents, and the process iterates by augmenting the seed tuples with the newly extracted tuples.

A crucial step in the extraction process is the generation of patterns to find new tuples in the documents. Given a set of seed tuples (e.g., $< g_1, g_2 >$), and having found the text segments where g_1 and g_2 occur close to each other, *Snowball* analyzes the text that "connects" g_1 and g_2 to generate patterns. *Snowball*'s patterns incorporate *entity* tags (i.e., the *GENE* tags assigned by the tagger during the preprocessing). For example, a pattern would be generated from a context "*<GENE> also known as <GENE>*". *Snowball* represents the left, middle, and right "contexts" associated with an extraction pattern as *vectors* of weighted terms (where terms can be arbitrary strings of non-space characters). During extraction, to match text portions with patterns, *Snowball* also associates an equivalent set of term vectors with each document portion that contains two entities with the correct tags (i.e., a pair of *GENE*s).

After generating patterns, *Snowball* scans the collection to discover new tuples by matching text segments with the most similar pattern (if any). Each candidate tuple will then have a number of patterns that helped generate it, each with an associated degree of match. *Snowball* uses this information, together with information about the selectivity of the patterns, to decide what candidate tuples to actually add to the table that it is constructing. Intuitively, we can expect that newly extracted synonyms for "known" genes should match the known synonyms for these genes. Otherwise, if the newly extracted synonym is "unknown" (i.e., a potential false positive), the pattern is considered to be less "selective" and its confidence is decreased. For example, if *Snowball* extracted a new synonym pair $s =< g_a, g_b >$, we check if there exists a set of high confidence previously extracted synonyms for g_a, e.g., $< g_a, g_1 >$, $< g_a, g_2 >$. If g_b is equal to either g_1 or g_2, s is considered a positive match for the pattern, and an "unknown" match otherwise. Note that this confidence computation "trusts" tuples generated on earlier iterations more than newly extracted tuples. Additionally, if the pattern P matches a known negative example tuple, the confidence of P is further decreased. More formally, *Snowball* defines *Conf(P)*, the confidence of a pattern P as:

$$\{\log_2(P_{positive})\frac{P_{positive}}{(P_{positive} + P_{unknown} \cdot w_{unk} + P_{negative} \cdot w_{neg}}\}$$

where $P_{positive}$ is the number of positive matches for P, $P_{unknown}$ is the number of "unknown" matches, and $P_{negative}$ is the number of negative matches, adjusted respectively by the w_{unk} and w_{neg} weight parameters (set during system tuning). The confidence scores are normalized so that they are between 0 and 1.

Snowball calculates the confidence of the extracted *tuples* as a function of the confidence values and the number of the patterns that generated the tuples. Intuitively, *Conf(s)*, the confidence of an extracted tuple s, will be high if s is generated by several highly selective patterns. More formally, the *confidence* of s is defined as:

$$Conf(s) = 1 - \prod_{i=0}^{|P|} (1 - (Conf(P_i) \cdot Match(C_i, P_i)))$$

where $P = \{P_i\}$ is the set of extraction patterns that generated s, and C_i is the context associated with an occurrence of s that matched P_i with degree of match $Match(C_i, P_i)$. After determining the confidence of the candidate tuples, *Snowball* discards all tuples with low confidence. These tuples could add noise into the pattern generation process, which would in turn introduce more invalid tuples, degrading the performance of the system. The set of tuples to use as the seed in the next *Snowball* iteration is then $Seed = \{s | Conf(s) > \tau_t\}$, where $\tau_t = 0.6$ is a threshold tuned during system development.

The original **Snowball** system was designed to operate with few user-provided example tuples. When many labeled examples are available, *supervised* methods have performed well. Next, we present our adaptation of a supervised text classification system for synonym extraction.

Text Classification: A Supervised Approach

We can use supervised machine learning to build a *text classifier* to identify synonymous genes and proteins. We start with the same user-provided positive and negative example gene and protein pairs as were used as the initial examples for **Snowball**. We then automatically create the training set of example *contexts* where these gene and protein pairs occur. These contexts are assigned either a positive weight of 1.0 or a negative weight of w_{neg} (tuned as part of system development).

We can now *train* the classifier to distinguish between the "positive" text contexts (i.e., those that contain an example synonym pair), and the "negative" text contexts. Thus, a classifier would be able to distinguish previously unseen text contexts that contain synonym pairs (e.g., "A, also known as B"), from the contexts that do not express the synonymy relation (e.g., "A regulates B").

We chose a state-of-the-art text classification tool *SVMLight* (Joachims, 1998)[††] which has been shown to be effective for text classification. The resulting system, to which we refer as **SVM**, uses as features the same terms and term weights used by *Snowball* for training and prediction. We used the *rbf*, or the radial basis kernel function option of the *SVMLight* package, which performed best in our preliminary experiments over the development corpus.

After the classifier is trained, **SVM** examines every text context C surrounding pairs of identified gene and protein terms in the collection. If the classifier determines C to be an instance of the "positive" (i.e., synonym) class, the corresponding pair of genes or proteins s is assigned the initial confidence score $Conf_0(s)$, equal to the score that the classifier assigned to C. The confidence scores are normalized so that the final confidence of the candidate synonym pair s, $Conf(s)$, is between 0 and 1. Note that **SVM** does not combine evidence from multiple occurrences of the same gene or protein pair: When s occurs in multiple contexts, $Conf(s)$ is assigned based on the single "most promising" text context of s.

GPE: The Hand-Crafted Extraction System

As a final –and the most labor-intensive– extraction approach, we used a previously constructed hand-crafted system called **GPE** (Yu et al., 2002) which was built specifically for extracting synonymous gene and protein expressions. The construction of **GPE** begins with a set of known synonymous gene or protein names. The domain expert examines the contexts where these example gene or protein pairs occur, and manually generates patterns to describe these occurrences. For example, the expert decided that the strings "known as" and "also called" would work well as extraction patterns. Using these manually constructed patterns, **GPE** scans the collection for new synonyms. For example, **GPE** identified the synonymous set *Apo3, LARD, DR3, wsl* from the sentence "...Apo3 (also known as LARD, DR3, and wsl)...". Since **GPE** does not use gene or protein taggers, many pairs of strings that are not genes or proteins can be extracted. To avoid such false positives, **GPE** uses heuristics and knowledge-based filters. After filtering, each extracted synonym pair s is assigned a confidence $Conf(s) = 1$.

The Combined System

While **GPE** requires labor-intensive tuning by a biology expert, it can extract a small high quality set of synonyms (Yu et al., 2002). In contrast, both **Snowball** and **SVM** induce extraction patterns automatically, allowing them to capture synonyms that may be missed by **GPE**. On the other hand, **Snowball** and **SVM** are also likely to extract more false positives, resulting in the lower "quality" of the extracted synonyms. We can exploit the advantages of both the knowledge-based and machine learning-based techniques in a combined system. We now present our **Combined** system that integrates the output

[††] Available at: http://www-ai.cs.uni-dortmund.de/SOFTWARE/SVM_LIGHT/svm_light_v3.02.eng.html

of **Snowball**, **SVM**, and **GPE**[‡‡].

We can combine outputs of the individual extraction systems in different ways (e.g., (Dietterich, 2000)). In our implementation of **Combined**, we assume that each system is an independent predictor, and that the confidence score assigned by each system to the extracted pair corresponds to the probability that the extracted synonym pair is correct. We can then estimate the probability that the extracted synonym pair $s = < p_1, p_2 >$ is correct as (1 $-$ the probability that all systems extracted s incorrectly):

$$Conf(s) = 1 - \prod_{E \in Systems} (1 - Conf_E(s))$$

where $Conf_E(s)$ is the confidence score assigned to s by the individual extraction system E. This combination function quantifies the intuition that agreement of multiple extraction systems on a candidate synonym pair s indicates that s is a true synonym.

EXPERIMENTAL SETUP AND EVALUATION METHODOLOGY

We evaluated **Similarity**, **Snowball**, **SVM**, **GPE**, and **Combined** over a collection of 52,000 recent journal articles from *Science*, *Nature*, *Cell*, *EMBO*, *Cell Biology*, *PNAS*, and the *Journal of Biochemistry*. The journal archives are maintained by the *GeneWays* Project (Friedman et al., 2001) at Columbia University. The collection was separated into two disjoint sets of articles: the **development** collection, containing 20,000 articles, and the **test** collection, containing 32,000 articles.

System Tuning

The **Similarity**, **Snowball**, and the **SVM** systems were tuned over the unlabeled **development** collection articles. The tuning consisted of changing the parameter values (e.g., the size of the context window d) in a systematic manner to find a combination that appeared to perform best on the **development** collection. The final parameter values used for the subsequent experiments over the **test** collection are listed in Figure 3.

User-Provided Examples

Note that our machine-learning based systems do not require manually labeled articles. Instead, approximately 650 *known* gene and protein synonym pairs, previously compiled from a variety of sources, were used as positive examples for the **Snowball** and **SVM** systems. Some of these did not occur in the collections, and thus did not contribute to the system training. Additionally, a set of *negative* examples were compiled by a biology expert by examining the contexts of some commonly co-occurring, but not synonymous, genes and proteins in the **development** collection.

Parameter	*Value*	*Description*
window d	5	Size of the text context (in words) to consider
$\|seed\|$	650	Number of user-provided example pairs (for **Snowball** and **SVM**)
$\|seed_{neg}\|$	28	Number of negative user-provided example pairs (for **Snowball** and **SVM**)
MaxIterations	2	Number of iterations (for **Snowball**)
w_{neg}	2	Relative weight of *negative* pattern matches (for **Snowball** and **SVM**)
w_{unk}	0.1	Relative weight of *unknown* pattern matches (for **Snowball**)

Fig. 3. Final values of the **Similarity**, **Snowball**, and **SVM** system parameters.

One of the goals of our evaluation is to determine whether the extraction approaches that we compare generalize to new document collections. Therefore, the only information that we retained from the tuning of the **Similarity**, **Snowball**, and **SVM** systems were the values of the system parameters (Figure 3). During the "test" stage of our experiments, both **Snowball** and **SVM** systems were re-trained from scratch over the unlabeled articles in the **test** collection, by starting with the same initial example gene and protein pairs described above.

Evaluation Metrics

Our evaluation focuses on the quality of the extracted set of synonym pairs S_e: (1) how comprehensive is S_e, and (2) how "clean" the pairs in S_e are. To compare the alternative extraction systems, we adapt the *recall* and *precision* metrics from information extraction.

Recall: The fraction of all of the synonymous gene and protein pairs that *appear* in the collection, S_{all}, and were *captured* in the extracted set S_e, is defined as:

$$Recall = \frac{|S_e \cap S_{all}|}{|S_{all}|}$$

Precision: The fraction of the *real* synonym pairs in S_e is defined as:

$$Precision = \frac{|S_e \cap S_{all}|}{|S_e|}$$

Note that all of the compared extraction systems assign a confidence score between 0 and 1 to each extracted synonym pair. It would be useful to know the precision of the systems at various confidence levels. Therefore, we calculate *precision at* c, where c is the threshold for the minimum confidence score assigned by the extraction system. The *precision at* c is then defined as the precision of

[‡‡] As we will discuss, **Similarity** did not perform well, and therefore was not included in the **Combined** system.

the subset of the extracted synonyms with the confidence score $\geq c$. We define *recall at* c equivalently.

Evaluation Methodology

For small text collections, we could inspect all documents manually and compile the sets of all of the synonymous genes in the collection by hand. Unfortunately, this evaluation approach does not scale, and becomes infeasible for the kind of large document collections for which automatic extraction systems would be particularly useful. The problem with exhaustive evaluation is two-fold: (1) the extraction systems tend to generate many thousands of synonyms from the collection (which makes it impossible to examine all of them to compute *precision*), and (2) since modern collections typically contains thousands of documents, it is not feasible to examine all of them to compute *recall*.

Estimating Precision: To estimate precision at c, for each system's output S_e we randomly select 20 candidate synonym pairs from S_e with confidence scores (0.0-0.1, 0.1-0.2, ..., 0.9-1.0)[§§]. As a result, each system's output is represented by a sample of approximately 200 synonym pairs. Each sample (together with the supporting text context for each extracted pair) is given to two biology experts to judge the correctness of each extracted pair in the sample. Having computed the precision of the extracted pairs for each range of scores, we estimate *precision at* c as the average of the evaluated precision scores for each confidence range, weighted by the number of extracted tuples within each confidence score range.

Estimating Recall: To compute the exact recall of a set of extracted synonym pairs S_e, we would need to manually process the entire document collection to compile all synonyms in the collection. Clearly, this is not feasible. Therefore, we use a set of known *correct* synonym pairs that appear in the collection, which we call the *GoldStandard*. To create this *GoldStandard*, we use SWISSPROT. From this well structured database, we generate a table of synonymous gene and protein pairs by parsing the "DE" and "GN" sections of protein profiles. Unfortunately, we cannot use this table as is, since some of the pairs may not occur at all in our collection. We found that synonym expressions tend to appear within the same sentence. Therefore, the *GoldStandard* consists of synonymous genes and proteins (as specified by SWISSPROT) that co-occur in at least one sentence in the collection, and were recognized by the *Abgene* tagger. We found a total of 989 such pairs.

Unfortunately, we found that we did not agree with many of these synonym pairs. We consider synonymous gene or protein names to be those that *represent the same genes or proteins*. However, SWISSPROT appears to consider a broader range of synonyms. For example, SWISSPROT synonyms included different genes or proteins that had a similar function, that belong to the same family, that were different subunits, and those that were functionally related (Figure 4). Note that we judged the synonym pairs based solely on the information in our corpus and did not perform any biological experiments.

System	**Tagging**	**Similarity**	**Snowball**	**SVM**	**GPE**
Time	7 hours	40 minutes	2 hours	1.5 hours	35 minutes

Fig. 5. Running times of **Tagging**, **Similarity**, **Snowball**, **SVM**, **GPE** (**test** collection)

To create the *GoldStandard*, we asked six biology experts (all with PhDs in biology) to evaluate gene and protein pairs listed as synonyms in SWISSPROT, and judge whether they considered the pairs as synonyms. Each expert evaluated between 100 to 989 pairs. Each candidate synonym pair was judged by at least two experts, and was included in the *GoldStandard* if at least one of the experts agreed with the SWISSPROT classification[¶¶] . Experts disagreed with SWISSPROT on 318 pairs, and were unsure of additional 83. As a result, we included a total of 588 confirmed synonym pairs in the *GoldStandard*. The agreement was 0.61 among experts, 0.83 between experts and SWISSPROT, and 0.77 overall. The resulting *GoldStandard* is used to estimate recall as the fraction of the *GoldStandard* synonym pairs captured.

RESULTS

In this section we compare the performance of **Similarity**, **Snowball**, **SVM**, **GPE**, and **Combined** on the recall and precision metrics over the **test** collection described above. The experiments were performed on a dual-CPU 1.2Ghz Athlon machine with 2Gb of RAM. We report the running times of each system in Figure 5. Note that the **Tagging** and preprocessing (i.e., identifying the gene and protein terms in the collection) were performed once for the complete collection, and were not required again for the subsequent experiments.

Figure 6(a) reports recall of all systems. **Similarity** performs poorly, with recall less than 0.09 for all confidence scores. In contrast, **Snowball** and **SVM** have the highest recall for confidence scores below 0.4 (reaching 0.72 for **Snowball** and 0.38 for **SVM**), while **GPE** has

[§§] If there are fewer than 20 extracted synonyms with the required confidence score, we select all of the ones that match.

[¶¶] The evaluated synonym pairs are available at http://synonyms.cs.columbia.edu/SPROT/.

Relationship Type	*SWISSPROT Synonyms*	*Context*
Family Related	*GRPE, MGEL*	"... requires the nucleotide release factors, grpe and mge1..."
Fragment	*PS2, ALG3*	"...as ps-2 c-terminal_109-amino acid fragment (alg3) is essential in the death process..."
Subunits	*P40, P38*	"...baculoviruses encoding individual rf-c subunits p140, p40, p38, p37, and p36) yielded..."
Homologous	*GRIP-1, TIF2*	"...shown that grip-1 , the murine homologous of tif2..."
Functionally Related	*CDC47, MCM2*	"and cdc47 , cdc21 , and mis5 form another complex, which relatively weakly associates with mcm2."

Fig. 4. Four types of apparent gene and protein relationships (with an example and literature context of each) that were designated by SWISSPROT as synonyms: *Family Related*, *Subunits*, *Homologous*, and *Functionally Related*.

the best recall (0.14) of any individual system for the higher confidence scores. Note that **GPE** always assigned the $Conf(s) = 1$ to all extracted candidate pairs, and is therefore represented by a single data point in each plot. **Combined** has the highest recall of all systems for all confidence scores. For example, at confidence score $c = 0.4$, **Combined** recall is more than double that of any individual system.

We report the precision of all systems for varying confidence scores in Figure 6(b). **Similarity** has extremely low precision (less than 0.01) and therefore is not shown. Our experiments indicate that **Similarity** performed well for more common terms (Figure 1), but performed poorly on identifying gene and protein synonyms as it tends to extract pairs of genes that are *related*, but not synonymous. Both **Snowball** and **SVM** extract synonyms with over 0.9 precision at their highest confidence scores. **GPE** also has the precision of 0.9. The confidence scores that both **Snowball** and **SVM** assign to their extracted pairs are correlated with the actual precision. For example, while the precision at $c = 0.8$ of **Snowball** is 0.9, precision at $c = 0.1$ is 0.1. **Snowball** has higher precision than **SVM** for all confidence score values. Also note that while both **Snowball** and **SVM** have sharp drops in precision between the confidence scores of 0.4 and 0.7, the **Combined** confidence score is more smooth, and appears to be a better predictor of the precision.

Figure 6(c) reports the values of precision vs. recall for all systems. Both **Snowball** and **SVM** clearly trade off precision for high recall. Even though **Snowball** is able to achieve the recall of almost 0.72, the corresponding precision is 0.07. In contrast, **GPE** has at most 0.14 recall. As we conjectured, combining these complementary approaches in **Combined** resulted in a significant gain: While **Combined** has the highest precision of all systems, it is also able to achieve the highest recall of 0.8.

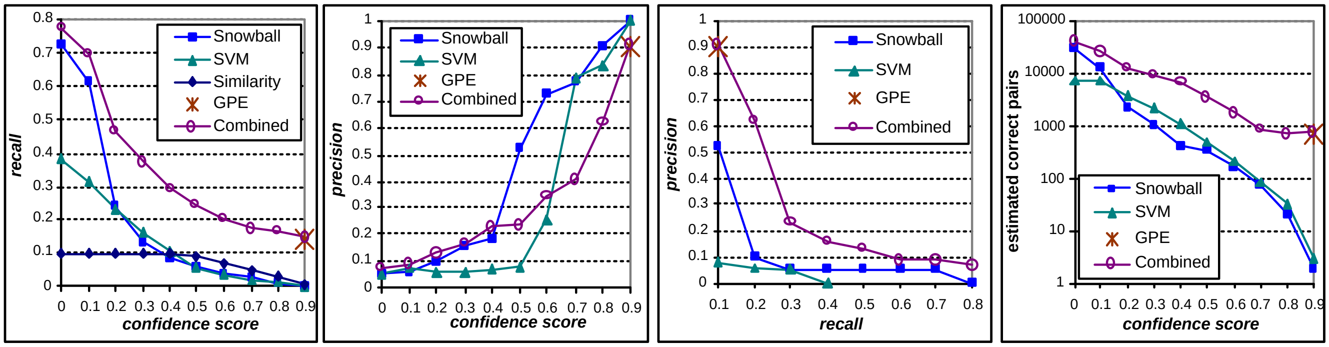
To complement the reported recall figures, we also estimated the number of all real synonym pairs extracted by each system for each confidence score c (Figure 6(d)). These values were calculated by multiplying the number of pairs extracted by the system with the score $\geq c$ by the corresponding precision at c. Despite exhibiting lower precision values, **Snowball** and **SVM** extract a significantly larger set of real synonyms than **GPE**. Similarly, **Combined** extracts the largest estimated number of real synonyms. For example, we estimate **Combined** to have extracted almost 10,000 correct synonyms at the confidence score of 0.4, which is more than ten times the estimated number of synonyms extracted by **Snowball**, **SVM**, or **GPE** individually. In summary, **Combined** is the best performing system on all metrics, and significantly improves over the manually constructed **GPE**.

DISCUSSION

We evaluated the four different extraction approaches over a large collection of biological journal articles. Our extraction results are particularly valuable as we found that many of the synonyms that we extracted do not appear in SWISSPROT. Of the 148 extracted synonym pairs that were manually judged as correct by the experts during our evaluation, 62 (or 42%) were not listed as synonyms in SWISSPROT. This leads us to predict that out of the approximately 10,000 correct synonym pairs extracted by **Combined** with confidence score ≥ 0.4 (Figure 6(d)), we would find more than 4,000 novel synonym pairs.

Our results show that machine learning-based approaches were responsible for the significant improvement of **Combined** over the manually constructed knowledge-based system. **Snowball** and **SVM** are –by design– more flexible, and therefore can detect cases on which **GPE** failed. For example, **Snowball** extracted the pair <*EIF4G*, *P220*> from the text fragment: "*...eIF4G, also known as eIF4 or p220, binds both eIF4A...*", which was not captured by **GPE**. While both **SVM** and **Snowball** contributed to the improved performance of **Combined**, **Snowball** has an additional advantage of generating intuitive human-readable patterns (Figure 7) that can be potentially examined and filtered by a domain expert.

There are many ways to improve our system. For example, we found that the small number of *negative* examples significantly improved performance of *Snowball*. Further experiments on varying the size and composition of the initial example tuple sets may result in additional improvements. We do not differentiate between gene and protein



(a) (b) (c) (d)

Fig. 6. Recall and precision of **Similarity**, **Snowball**, **SVM**, **GPE**, and **Combined**: Recall vs. confidence score (a), Precision vs. confidence score (b), precision vs. recall (c), and the estimated number of correct synonym pairs extracted by each system vs. confidence score (d).

Conf	*Left*	*Middle*	*Right*
0.75	-	<(0.55> <ALSO 0.53> <CALLED 0.53>	-
0.54	-	<ALSO 0.47> <KNOWN 0.47> <AS 0.47>	-
0.47	-	<(0.54> <ALSO 0.54> <TERMED 0.54>	-

Fig. 7. Some **Snowball** patterns automatically discovered from the **test**, collection with the associated pattern *confidence* scores.

terms with the same name, and it may be beneficial to apply the approach of (Hatzivassiloglou et al., 2001) for disambiguation. We may also explore more ways to enhance gene and protein name entity identification, which is likely to further improve extraction quality.

Our approaches extract synonyms from a collection of biological literature, and therefore the quality of the extracted relation depends in part on the collection consistency. We found some conflicting statements in our collections. For example, the following two statements are taken from two different articles in our **test** collection: while the first text fragment suggests that the proteins *PC1* and *PC3* are different substances, another article indicates that *PC1* and *PC3* are synonyms for the same substance:

> “ *...the positive cofactors (pcs) pc1, pc2, pc3, and p15.*”
>
> “*... hydra pc1 (also called pc3) ...*”

Lacking additional information, it is difficult to make a decision whether *PC1* and *PC3* are synonyms. We plan to explore this problem further in our future work.

CONCLUSIONS AND FUTURE WORK

In this paper we have addressed an important problem of extracting gene and protein synonyms from biological literature. We have adapted and evaluated complementary synonym extraction approaches that span the spectrum from an unsupervised approach to a hand-crafted knowledge-based system. We performed a large-scale evaluation of the competing approaches which show that our extraction techniques can be used as a valuable supplement to resources such as SWISSPROT.

As part of our analysis, we discovered some inconsistent statements in the articles. In future work we may incorporate some of the proposed methods for resolving inconsistencies (e.g., (Magnini et al., 2002), (Krauthammer et al., 2002)). Additionally, we may incorporate work of (Hatzivassiloglou et al., 2001) for disambiguation between genes and proteins that share the same name.

We speculate that the machine learning-based approaches for synonym identification, particularly **Snowball**, could be applied successfully for extracting other biological relations such as relationships between genes and proteins, small molecules, drugs, and diseases. We plan to explore this further in the future.

ACKNOWLEDGEMENTS

We thank Dr. Luis Gravano for his advice, support, and many fruitful discussions, and Dr. Andrey Rzhetsky for making the *GeneWays* article collection available. We thank Lorrie Tanabe and Dr. W. John Wilbur for allowing us to use the *Abgene* tagger. We thank Drs. Xiaowei Chen, Hongping Du, Claire He, Laura Luo, Youxing Qu and Andrey Rzhetsky for evaluating the extraction output. We also thank Noemie Elhadad, David Evans, Vasileios Hatzivassiloglou, Roy Byrd, and the anonymous referees for their helpful comments. The second author acknowledges support of the National Science Foundation under Grants No. IIS-97-33880 and IIS-98-17434.

REFERENCES

Agichtein, E. and L. Gravano (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the ACM International Conference on Digital Libraries*.

Blum, A. and T. Mitchell (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of ICML*.

Brin, S. (1998). Extracting patterns and relations from the World-Wide Web. In *Proceedings of the SIGMOD Workshop on the Web and Databases (WebDB)*.

Califf, M. E. and R. J. Mooney (1998). Relational learning of pattern-match rules for information extraction. In *Proceedings of the AAAI Symp. on Applying Machine Learning to Discourse Processing*.

Collins, M. and Y. Singer (1999). Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Dagan, I., S. Marcus, and S. Markovitch (1995). Contextual word similarity and estimation from sparse data. *Computer, Speech and Language*.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science*.

Fellbaum, C. (1999). *WordNet:An Electronic Lexical Database*. MIT Press.

Friedman, C., P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky (2001). Genies: A natural-language processing system for the extraction of molecular pathways from complete journal articles. *Bioinformatics 17 Suppl 1*.

Fukuda, K., A. Tamura, T. Tsunoda, and T. Takagi (1998). Toward information extraction: identifying protein names from biological papers. In *Proceedings of the Pacific Symp. on Biocomputing*.

Hatzivassiloglou, V., P. Duboue, and A. Rzhetsky (2001). Disambiguating proteins, genes, and RNA in text: a machine learning approach. *Bioinformatics 17 Suppl 1*.

Hearst, M. (1992). Automatic acquistion of hyponyms from large text corpora. In *Proceedings of COLING*.

Hole, W. and S. Srinivasan (2000). Discovering missed synonyms in a large concept-oriented metathesaurus. In *Proceedings of the AMIA Symposium*.

Humphreys, B. and D. Lindberg (1993). The UMLS project: making the conceptual connection between users and the information they need. *Bulletin of the Med. Lib. Association 81*.

Joachims, T. (1998). Making large-scale support vector machine learning practical. *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

Krauthammer, K., P. Kra, I. Iossifov, S. Gomez, G. Hripcsak, V. Hatzivassiloglou, C. Friedman, and A. Rzhetsky (2002). Of truth and pathways: chasing bits of information through myriads of articles. *Bioinformatics 18 Suppl 1*.

Kushmerick, N., D. S. Weld, and R. B. Doorenbos (1997). Wrapper induction for information extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Li, H. and N. Abe (1998). Word clustering and disambiguation based on co-occurrence data. In *Proceedings of COLING*.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of ACL*.

Liu, H. and C. Friedman (2003). Mining terminological knowledge in large biomedical corpora. In *Proceedings of the Pacific Symp. on Biocomputing*.

Magnini, B., M. Negri, R. Prevete, and H. Tanev (2002). Is it the right answer? exploiting web redundancy for answer validation. In *Proceedings of ACL*.

Muslea, I., S. Minton, and C. Knoblock (1998). STALKER: Learning extraction rules for semistructured web-based information sources. In *Proceedings of AAAI-98 Workshop on AI and Information Integration*.

Pakhomov, S. (2002). Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical text. In *Proceedings of ACL*.

Park, Y. and R. Byrd (2001). Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of EMNLP*.

Proux, D., F. Rechenmann, L. Julliard, V. Pillet, and B. Jacq (1998). Detecting gene symbols and names in biological texts: A first step toward pertinent information extraction. In *Proceedings of Workshop on Genome Informatics*.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial I ntelligence (IJCAI)*.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 1044–1049.

Riloff, E. and R. Jones (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*.

Rindflesch, T., L. Tanabe, J. Weinstein, and L. Hunter (2000). EDGAR: extraction of drugs, genes and relations from the biomedical literature. In *Proceedings of the Pacific Symp. on Biocomputing*.

Schwartz, A. and M. Hearst (2003). A simple algorithm for identifying abbreviation definitions in biomedical text. In *Proceedings of the Pacific Symp. on Biocomputing*.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning 34*(1-3).

Tanabe, L. and W. Wilbur (2002). Tagging gene and protein names in biomedical text. *Bioinformatics 18*.

Thomas, J., D. Milward, C. Ouzounis, S. Pulman, and M. Carroll (2000). Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of the Pacific Symp. on Biocomputing*.

Wilbur, W. and W. Kim (2001). Flexible phrase-based query handling algorithms. In *Proceedings of the ASIST*.

Yangarber, R. and R. Grishman (1998). NYU: Description of the Proteus/PET system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.

Yangarber, R., R. Grishman, P. Tapanainen, and S. Huttunen (2000). Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of Conference on Applied Natural Language Processing ANLP-NAACL*.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the ACL*.

Yoshida, M., K. Fukuda, and T. Takagi. (2000). PNAD-CSS: a workbench for constructing a protein name abbreviation dictionary. *Bioinformatics 16*.

Yu, H., C. Friedman, and G. Hripcsak (2002). Mapping abbreviations to full forms in biomedical articles. *Journal of American Medical Information Association 9*, 262–72.

Yu, H., V. Hatzivassiloglou, C. Friedman, A. Rzhetsky, and W. J. Wilbur (2002). Automatic extraction of gene and protein synonyms from medline and journal articles. In *Proceedings of the AMIA Symposium*.