Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics
& Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

Method of extracting sentences about protein interaction from the literature on protein structure analysis using selective transfer learning

Shun Koyabu, Riku Kyougoku, and Takenao Ohkawa
Graduate School of System Informatics
Kobe University
1-1 Rokkodai, Nada, Kobe 657-8501, Japan
Email: [s.koyabu, ohkawa]@cs25.scitec.kobe-u.ac.jp

***Abstract*—With the progress of research on structural analysis of proteins, a large number of studies have been conducted on extracting the protein interaction information from literature. For automatic extraction of interaction information, the machine learning approach is useful. Generally, linguistic features obtained directly from the literature are used for learning, but a non-linguistic feature such as the atomic distance calculated from the protein structure data is often very effective for learning and classification. We call this type of feature a "key feature" in this study.**

In the machine learning approach, preparing enough training instances to train the classifier is important, but this often requires great cost. In such a situation, transfer learning is one of the better approaches. However, it is difficult to apply a simple transfer learning algorithm to a task in which the key feature cannot be prepared in the source domain.

In this study, we propose a new transfer learning method called STEK (Selective Transfer learning based on Effectiveness of a Key feature). In this method, we focus on the effectiveness of the key feature, and divide a set of instances into two categories. One is a set of instances applying transfer learning and the other is a set of instances avoiding the use of transfer learning. The proposed method with the InstPrune algorithm showed stably high precision, recall and F-measure on average.

***Index Terms*—Protein interaction imformation extraction, Machine learning, Transfer learning**

I. Introduction

The research on the structual analysis of proteins has progressed at rapid speed, and the relation between structures and functions of proteins and their interaction mechanisms have been clarified. Such knowlegde has been stored in the form of documents in many scientific articles. To make effective use of such knowledge, a large number of studies have been conducted on extracting the protein interaction information from the literature[1] .

For automatic extraction of the interaction information, the machine learning approach is useful[2]. In this approach, the classifier is trained from training instances based on features provided from the literature to decide whether each sentence has interaction information or not[3]. Linguistic features are used in general, but a non-linguistic feature such as the atomic distance calculated from the protein structure data is often very effective for learning and classification[4]. We call this type of feature a "key feature" in this study.

In the machine learning approach, although preparing enough training instances to train the classifier is important, making a large amount of training instances often requires too much cost. In such a situation, transfer learning, in which knowledge in one or more source domains is transferred and used to improve learning tasks in a target domain, is one of the better approaches[5]–[8]. However, it is difficult to apply the simple transfer learning algorithm to a task in which the key feature cannot be prepared in the source domain.

In this study, for effective extraction of the sentences including protein interaction from the literature, we propose a new transfer learning method called STEK (Selective Transfer learning based on Effectiveness of a Key feature). In this method, we focus on the change of the classification results with or without the key feature (that is, the effectiveness of the key feature), and divide a set of instances into two categories; one is a set of instances applying transfer learning and the other is a set of instances avoiding the use of transfer learning.

II. Methods

A protein binds to its interaction partners at the interaction site, and then expresses various functions. Information about protein interaction, e.g. which site on the protein contributes to the interaction, what the interaction partner is, and what kind of interaction is observed, is important for protein function analysis. These types of information are described in much literature of protein structure analysis from the PDB database (protein structure database). We call a sentence including interaction information *interaction sentence* and in this paper, we propose a method of extracting interaction sentences from the literatures.

The following sentence is an example of an interaction sentence[9]:

- "In addition, a histidine residue interacts with the phosphonate oxygen atoms in each of the structures."

978-1-4673-4358-9/12/$31.00 ©2012 IEEE

A. Features

To extract interaction sentences, we use the machine learning approach, in which positive instances (namely interaction sentences) and negative ones are used for the classifier to learn to distinguish between positive and negative. A set of the literature, in which each sentence is given a positive or negative label and named entity tags are attached to the words in the sentence, is used as a corpus for the learning. Table I shows the list of named entity tags. *Substance* means a material such as a protein or residue, and *situation* means a situation such as the place where the reaction happens or the kind of reaction.

TABLE I
LIST OF NAMED ENTITY TAGS

category	named entity tag	contents	examples
substance	<protein>	name of a protein	γ-DSPA
	<residue>	name of a residue	Trp215
	<chain>	chain information	the same A chain
	<ion>	name of an ion	the calcium ion
	<domain>	name of a domain	the CDR loops H2
	<peptide>	name of a peptide	chloromethyl ketone
	<group>	name of a group	the cyclohexene ring
	<atom>	name of an atom	oxygen
	<tert_structure>	name of a tertiary structure	the γ-autolysis loop
	<sec_structure>	name of a secondary structure	β-strand
	<chemical>	name of substate	progesterone
	<molecule>	name of a molecule	a water molecule
situation	<interaction>	interaction information	hydrogen bonds
	<function>	function information	the immune system
	<status>	status information	the standard orientation
	<misc>	misciblility information	membranes
	<reaction>	reaction information	proteolysis

The following features for representing each instance (sentence) are often used to train classifiers.

(1) Frequently observed words in interaction sentences
Words that are directly related to the interaction (e.g.;“bind”,“interact”,“active site”,“salt bridge”) are frequently observed in the interaction sentences. Therefore, whether or not these words are included in the sentence is used as a feature.

(2) Frequently observed phrases in interaction sentences
The phrase “between A and B” is often used to indicate interaction between the residue and the interaction target in interaction sentences. In addition, the following is also often used: “<residue> (*) [VERB] (*) <tag> | <tag> (*) [VERB] (*)<residue>”, where <tag> is any named entity tag, [VERB] is any verb, and (*) is a wild card. Therefore, whether or not such important phrases are included in the sentence is used as a feature.

(3) Important words in a paragraph
If the interaction sentence is considered an important sentence in a paragraph, the important words extracted from every paragraph are often observed in the interaction sentence. The important words in a paragraph means the words that are frequently used in this paragraph but are rarely used in other paragraphs. Therefore, whether or not such important words are included in the sentence is used as a feature. $w_{i,j}$, the importance of word i in paragraph j is defined as the following expression based on popular TF-IDF-like measure.

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{pf_i})$$

where $tf_{i,j}$(term frequency) is the number of occurrence of word i in the paragraph, pf_i is the number of paragraphs in which word i appears in a target article, and *N* is the number of all paragraphs in the target article.

(4) Frequently observed patterns in interaction sentences
While the phrases introduced in feature (2) are predefined fixed patterns, we consider more flexible patterns generated automatically that are frequently (empirically over five occurrences) observed in the target interaction sentences. The pattern is enumarated by using a meta-pattern “(<tag>, *or* [VERB]) (*) (<tag>, *or* [VERB]) (*) (<tag>, *or* [VERB]) ” as a template. Whether or not these patterns are included in the sentence is used as a feature.

(5) Atomic distance
Other than the four kinds of features based on NLP stated above, we also consider an important (non-linguistic) feature in terms of the atomic distance in the protein structure as useful information peculiar to protein structure analysis articles. When a residue of a protein interacts with its interaction partner (residue, compounds, etc.) the distance between the residue and the target is closed. The names of residues or other compounds that perhaps interact with each other are often described in the interaction sentences. Therefore if the atomic distance between residues and partners can be calculated, whether the distance is smaller than the threshold is used as a feature. The distance can be calculated in advance from three-dimensional coordinate data of atoms in PDB files.

The atomic distance feature is much more effective in learning and classification in comparison with other features based on NLP, and we call this type of feature a “key feature” in this study. The number of features depends on the automatically generated patterns, but is usually about 300.

B. Transfer learning

Training instances of sufficient quantity are required for a good classification result in learning. However, it requires much time and cost because the training instances need to be closely checked by an expert. To solve this problem, transfer learning is one of the better approaches. Transfer learning means using some knowledge from another domain (called a source domain) to get a better classification result with the target domain. The following methods are often used.

- Augment method
  Daumé III proposed a very simple method for transfer learning based on the augmentation of the feature space[10]. The augmentation depends on whether the instance is from the source or from the target domain. In this method, for a feature vector x in the original feature space, mappings Φ^s and Φ^t are defined for the source and target domains respectively, and the classifier is trained on these mapped feature spaces:
  $$\begin{aligned} \Phi^s(\mathbf{x}) &= <\mathbf{x}, \mathbf{x}, \mathbf{0}> \\ \Phi^t(\mathbf{x}) &= <\mathbf{x}, \mathbf{0}, \mathbf{x}> \end{aligned}$$
  where **0** is a zero vector of length $|\mathbf{x}|$.
- Instance pruning
  Jiang proposed a method using a part of instances in the source domain that are suitable for learning in the target domain[11]. The algorithm of this method is shown as follows.
  1) Train a classifier using training instances of the target domain.
  2) Classify all training instances of the source domain using this classifier.
  3) Calculate the predictive reliability for the instances whose class label was misestimated.
  4) Delete instances that have the highest *N* predictive reliability from the source domain. N is a cutoff parameter whose value is usually determined by a preliminary experiment.
  5) Train a classifier using training instances of the target domain and all instances of the source domain except for the deleted instances.
  6) Classify instances of the target domain using this classifier.

  This method, deleting misclassified instances that have high predictive reliability, can use only instances to help learning of the target domain. This method depends on parameter *N*, and the experiment by Jiang shows the best case is deleting all misclassified instances of the source domain.

When applying the transfer learning to extraction of interaction sentences, there are two problems.

1) As to the target domain, the interaction information at the residue or the atom levels is described and corresponding named entity tags (such as <residue>, <atom>, and <group>) are used. However, as to the other domain (source), the interaction information at the protein level may be described. Therefore, we cannot assume the identical feature vector from these domains.
2) The key feature is very useful for learning and classifying. However, we cannot be provided the key feature for the instances in the source domain if the interaction at the protein level is treated or there is no structure information in the source domain.

To solve the first problem, we prepare a feature mapping table by considering category of the tags (substance or situation) and occurrence frequency of the tags, in which the features in the source and the target domains representing similar concepts are related to each other[12]. To cope with the second problem, we propose a new framework for the effective extraction of the protein interaction information from the literature, called STEK (Selective Transfer learning based on Effectiveness of a Key feature).

C. Selective transfer learning based on effectiveness of a key feature

The key feature is very useful for the extraction of the interaction sentence, but it is impossible to use for transfer learning because other domain corpora do not include the protein structure data. In the STEK approach, the instances that the key feature has a significant effect on are classified by a classifier trained with the key feature, and the other instances are classified by a classifier using transfer learning without the key feature. The following is the general flow of STEK.

1) Divide a set of target instances (namely, previously unseen instances in the target domain) into two categories; one is a set of instances for which the key feature works effectively (denoted by S_A) and the other is a set of instances for which the key feature has little effect (denoted by S_B) by using a classifier with the key feature.
2) Classify the instances in S_A by using a classifier that is trained from training instances in only the target domain using the key feature (classifier C_A).
3) Classify the instances in S_B by using a classifier that is trained from training instances of both domains without the key feature (classifier C_B).
4) Merge the classification results in steps 3) and 4).

One of these classifiers is selected based on effectiveness of the key feature for each target instance. The proposed method focuses on the difference of the classification results, assuming the key feature value is inverted in order to evaluate its effectiveness. First, by using the key feature, we classify the target instances by the classifier trained using training instances in only the target domain without transfer learning. Then, we invert the value of the key feature of the target instances (0→1, 1→0) and classify them again by the same classifier. Finally, we divide the set of the target instances into eight categories (1PP, 1PN, and so on) based on the following three attributes: the value of the original key feature, the classification result before inversion of the key feature and the result after inversion of the key feature. Table II summarizes the definition of the eight categories of divided instances. Figure 1 shows the flow of dividing target instances.

Next, for the classification, the proposed method focuses on the difference of the classifiers with or without the key feature so that the target instance can be classified using a more useful classifier. In other words, we classify each set of instances by using a more suited classifier selected from classifiers C_A or

TABLE II
DEFINITION OF EIGHT CATEGORIES OF DIVIDED INSTANCES

Group name	Key feature's value before changing	Classify result before changing	Classify result after changing
1PP	1	positive	positive
1PN	1	positive	negative
1NP	1	negative	positive
1NN	1	negative	negative
0PP	0	positive	positive
0PN	0	positive	negative
0NP	0	negative	positive
0NN	0	negative	negative

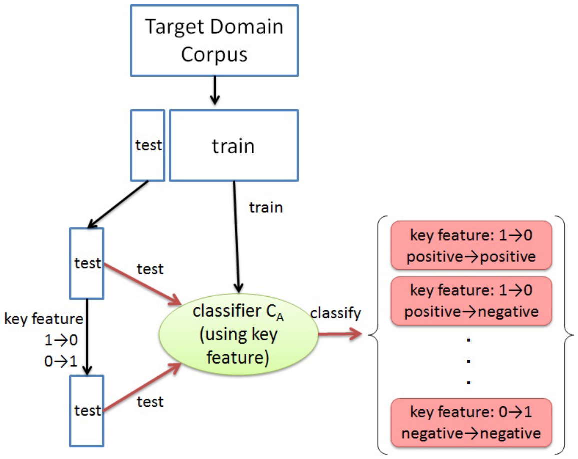
Target Domain Corpus
test
train
train
test
test
key feature
1→0
0→1
test
classifier C_A (using key feature)
classify
key feature: 1→0 positive→positive
key feature: 1→0 positive→negative
key feature: 0→1 negative→negative

Fig. 1. Flow of dividing target instances

C_B, then merge both results into the final output. Figure 2 shows the framework of dividing and merging.

We explain the property of the key feature. Because the value of the key feature is “1”, the atom distance can be calculated and is smaller than the threshold, and this is strong ground for classifying the instances into positive. In addition, it is desirable that the instance that has large effectiveness of the key feature is classified by classifier C_A. Therefore, an instance, which has the value “1” for the key feature and the classification result is positive, should be classified using the classifier C_A because it has large effectiveness and the key feature is useful. Also, an instance, which has the value “1” for the key feature and the classification result is negative, is classified using classifier C_B because it has little effectiveness and the key feature is not useful.

In contrast, because a the key feature value of “0” gives various interpretations, e.g. the atom distance cannot be calculated, the distance between atoms is not close, or the sentence has no interaction information, it is not very strong ground for classifying the instance as negative or positive. For this reason, it is not desirable that the instance that has large effectiveness of the key feature is classified by classifier C_A because of overestimation of the ambiguous key feature. Therefore, an instance, which has the value “0” for the key feature and

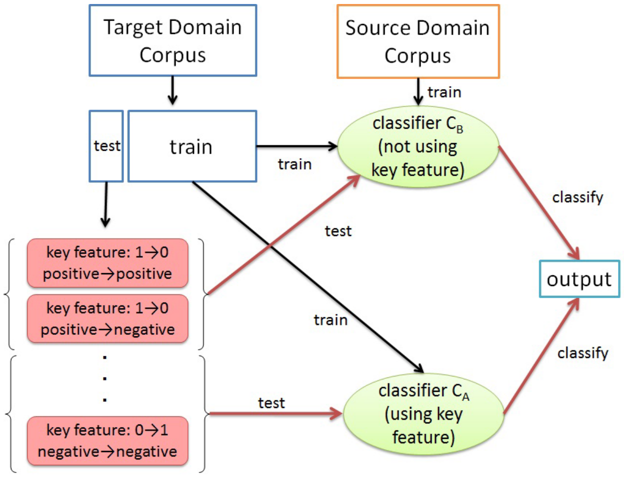
Target Domain Corpus
Source Domain Corpus
train
test
train
train
classifier C_B (not using key feature)
classify
test
output
key feature: 1→0 positive→positive
key feature: 1→0 positive→negative
train
classify
key feature: 0→1 negative→negative
test
classifier C_A (using key feature)

Fig. 2. Flow of STEK framework of dividing and merging

the classification result has been changed, is classified using classifier $C_B$because this instance is given large effectiveness of the key feature and the classification result may be wrong. Also an instance, which has the value “0” for the key feature and the classification result has not been changed, is classified using classifier C_A because it has small effectiveness of the key feature and the classification result may be correct.

In consideration of the above, we select a suitable classifier as follows for each divided category.

1PP Use classifier C_A because the key feature has large effectiveness and its value is “1”.
1PN Use classifier C_A because the key feature has large effectiveness and its value is “1”.
1NP Use classifier C_B because the key feature has little effectiveness and its value is “1”.
1NN Use classifier C_B because the key feature has little effectiveness and its value is “1”.
0PP Use classifier C_A because the key feature has little effectiveness and its value is “0”.
0PN Use classifier C_B because the key feature has large effectiveness and its value is “0”.
0NP Use classifier C_B because the key feature has large effectiveness and its value is “0”.
0NN Use classifier C_A because the key feature has little effectiveness and its value is “0”.

In summary, instances that belong to 1PP, 1PN, 0PP and 0NN are classified by classifier C_A with the key feature, and instances that belong to 1NP, 1NN, 0PN and 0NP are classified by classifier C_B using the transfer learning algorithm without the key feature. Finally, these results are merged.

Figure 3 shows the STEK algorithm, where D_t is a target domain and D_s is a source domain.

III. RESULTS AND DISCUSSION

To evaluate the proposed method, the target corpus in the target domain was constructed from manually curated fourteen articles taken from the Protein Data Bank entry of the PDB_ID

```
Procedure : classification using STEK
1 $C_A$.trained(training instances of $\mathrm{D}_t$) with key feature.
2 $C_A$.classify(target instances of $\mathrm{D}_t$).
3 $C_A$.classify(target instances of $\mathrm{D}_t$ where key feature value is inverse).
4 Divide target instances of $\mathrm{D}_t$ into 8 categories
        ([key  feature, 2's result, 3's result]):(e.g.[1PP],[0NP]...).
5 $C_B$.trained(instances of $\mathrm{D}_s$ and $\mathrm{D}_t$) without key feature.
6 $C_B$.classify(instances in [1PP],[1NN],[0PN],[0NP]).
7 $C_A$.classify(instances in [1PN],[1NP],[0PP],[0NN]).
8 Merge (6's result, 7's result).
```

Fig. 3. STEK algorithm

shown in Table III. The named entity tags and the class labels are attached to all of the sentences in the target corpus manually. Table III also shows the number of sentences and the number of interaction sentences (namely the sentences with positive labels) out of all sentences. Additionally, we use the BioEvent corpus containing 800 Medline abstracts, which was introduced at *BioNLP'09 Shared Task*[1] to apply transfer learning as the corpus in the source domain[13].

TABLE III
THE LITERATURE USED IN EXPERIMENTS

PDB ID	num of sentences	num of interaction sentences
1a0h	359	26
1a0q	295	23
1a3l	272	23
1a3r	299	21
1a4j	190	13
1a5a	113	10
1a5h	296	39
1a5i	324	73
1a5v	277	20
1a5y	291	33
1a5z	428	8
1a26	243	13
2a2g	365	13
2a39	312	4

Bio-event means a specific kind of interaction between biological entities, especially proteins or genes and the tag <protein> is attached to each of the entities. Table IV shows the difference of corpora.

TABLE IV
DIFFERENCE OF CORPORA

	Target Corpus	BioEvent Corpus
Number of the literature	14	800
Number of sentences	4064	7566
Number of positive sentences	319	2450
Interaction level	mainly residue	mainly protein
Named entity tag	12 types	<protein> only

To confirm the effectiveness of STEK for the domain including the key feature, we compare the accuracy of the following five methods.

- **Method 1: Target domain only (TargetOnly)** The conventional method without the source domain. The key feature is available.
- **Method 2: Transfer learning (Augment)** Transfer learning method without STEK. The augment method is used for the transfer learning. The key feature in the target domain is available, but in the source domain the key feature value is assumed to be "0".
- **Method 3: Transfer learning (InstPrune)** Transfer learning method without STEK. Instance pruning is used for the transfer learning. The key feature in the target domain is available, but is unavailable in the source domain.
- **Method 4: STEK (STEKAugment)** The proposed method with the augment method.
- **Method 5: STEK (STEKInstPrune)** The proposed method with the instance pruning.

In all five methods, the decision tree algorithm implemented in *Weka*[2] is utilized for constructing classifiers. Our experimental envioronment is a single PC with a 3.4GHz Core i7 2600 processor (quad cores), 16GB memory and Microsoft Windows 7 Professional operationg system. All algorithms are implemented in Java. The average calculation time for training classifiers in STEK is several minutes. The threshold value of minimal distance in feature (5) is 5.0[4], and the value of cutoff parameter N in instance pruning is 3,000.

One of fourteen articles shown in Table III is selected for the test, and the rest are used for training in the target domain. In addition, all of the sentences in The BioEvent corpus are regarded as training instances in the source domain. Precision, recall and F-measure for the test set, namely the selected article in the target domain, are calculated, and the average value of the fourteen trials is evaluated as the final result.

TABLE V
EXPERIMENT RESULTS

Method	Precision	Recall	F-measure
TargetOnly	0.8154	0.7618	0.7877
Augment	0.8369	0.7398	0.7854
InstPrune	0.8511	0.7524	0.7987
STEKAugment	0.7033	0.8621	0.7746
STEKInstPrune	0.8119	0.8094	**0.8106**

Table V shows the results of the evaluation of the five methods. STEKInstPrune shows high recall and F-measure values; in particular, F-measure is the best score among the five methods. STEKAugument shows a better result than others in recall, which means that many more interaction sentences are extracted.

We focus on the instances for which the classifier C_B has been selected in STEK, in other words, the instances for

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/index.shtml

[2] http://www.cs.waikato.ac.nz/ml/weka/

which the key feature does not work effectively. Comparing the classification results for these instances by classifiers C_A and C_B, we clarify the significance of the transfer learning for such instances.

TABLE VI
CLASSIFICATION RESULTS FOR INSTANCES FOR WHICH THE CLASSIFIER C_B HAS BEEN SELECTED

Method	True positive	False positive	False negative	True negative
TargetOnly	1	2	75	3574
Augment	33	62	43	3514
InstPrune	17	7	60	3568

TABLE VII
CLASSIFICATION ACCURACY FOR INSTANCES FOR WHICH THE CLASSIFIER C_B HAS BEEN SELECTED

Method	Precision	Recall	F-measure
TargetOnly	0.3333	0.0132	0.0253
Augment	0.3338	**0.4342**	**0.3837**
InstPrune	**0.7083**	0.2208	0.3366

Table VI shows the classification result of the conventional method without transfer learning (TargetOnly) and the methods with transfer learning (Augment and InstPrune). The TargetOnly method shows that TP (true positive) is small and FN (false negative) is large, which implies that the positive instances are scarcely extracted. Both in Augment and InstPrune, TP increases and FN decreases, which shows that the positive instances that cannot be extracted by TargetOnly are extracted.

Table VII summarizes extraction accuracy. Both in Augment and InstPrune, F-measure value is drastically improved in comparison with the one in the TargetOnly method. This result suggests that classifier C_B has a great advantage over classifier C_A for classifying the instances to which STEK decides to apply classifier C_B.

IV. CONCLUSION

In this paper, we proposed a novel method for extracting interaction sentences from the literature using transfer learning based on effectiveness of a key feature. STEK has the following notable features.

- STEK divides instances into two sets; one has effectiveness of the key feature, the other does not.
- STEK selects two classifiers for each set; one uses only the target domain with the key feature, the other uses both target and source domains for transfer learning.

As a result of the interaction sentence extraction experiment, using STEK with the InstPrune algorithm shows stably high precision, recall and F-measure on the average, in particular F-measure. The result is much better than other conventional methods.

For future works, we will explore the effect of bias (e.g. texts of difference length, the size of positive and negative sentences, and so on) on the proposed method, and apply other classification algorithms, such as Supprt Vector Machines. Additionally, we plan to apply the proposed method to much larger scale data (e.g. the set of articles referred from all entries in PDB). In our experiments, tags were manually and carefully attached in order to evaluated the effectiveness of the proposed method itself. However, automatically tagging schema may be required for the large scale experiment, which is one of other remaining works.

ACKNOWLEDGMENT

This work was partly supported by Grant-in-Aid for Scientific Research (B) (24300056) from MEXT.

REFERENCES

[1] R. Bunescu *et al.*, “Learning to extract proteins and their interactions from medline abstracts,” in *Proc. of International Conference on Advances in Computer Science and Technology (ACST 2004)*, Washington, DC, Aug. 2003, pp. 46–53.

[2] M. A. Munna and T. Ohkawa, “A method to extract sentences with protein functional information from literature by iterative learning of the corpus,” *IPSJ Transactions on Database*, vol. 47, no. SIG17(TBIO1), pp. 22–30, 2006.

[3] K. Miyanishi, M. Takeuchi, T. Ozaki, and T. Ohkawa, “Iterative learning with feature update for extracting sentences containing protein function information,” in *7th Atlantic Symposium on Computational Biology Genome Informatics*, SaltLake, USA, Jul. 2007, pp. 96–120.

[4] Y. Kaneta, M. A. Munna, and T.Ohkawa, “A method for extracting sentences related to protein interaction from literature using a structure databse,” *The Institute of Electrical Engineers of Japan*, vol. 125, no. 5, pp. 690–697, 2005.

[5] J. Baxter, “A bayesian/information theoretic model of learning to learn via multiple task sampling,” *Machine Learning*, vol. 28, pp. 7–39, 1997.

[6] S. Thrun, “Is learning the n-th thing any easier than learning the first?” in *Advances in Neural Information Processing Systems (NIPS) 8*, Cambridge, MA, Dec. 1996, pp. 640–646.

[7] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, pp. 41–75, 1997.

[8] A. Arnold, R. Nallapati, and W. W. Cohen, “A comparative study of methods for transductive transfer learning,” in *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, Washington, DC, USA, Oct. 2007, pp. 77–82.

[9] J. L. Buchbinder *et al.*, “A comparison of the crystallographic structures of two catalytic antibodies with esterase activity,” *Molecular Biology*, vol. 282, pp. 1033–1041, 1998.

[10] H. D. III, “Frustratingly easy domain adaptation,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, Jun. 2007, pp. 256–263.

[11] J. Jiang and C. Zhai, “Instance weighting for domain adaptation in nlp,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, Jun. 2007, pp. 264–271.

[12] W. Dai *et al.*, “Translated learning: Transfer learning across different feature spaces,” in *NIPS*, Vancouver, B.C., Canada, Dec. 2008, pp. 353–360.

[13] J. D. Kim *et al.*, “Overview of bionlp 0̀9 shared task on event extraction,” in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, Boulder, Colorado, Jun. 2008, pp. 1–9.