

A Rich Feature Vector for Protein-Protein Interaction Extraction from Multiple Corpora

Makoto Miwa¹ Rune Sætre¹ Yusuke Miyao¹ Jun'ichi Tsujii^{1,2,3}

¹Department of Computer Science, the University of Tokyo, Japan
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan.

²School of Computer Science, University of Manchester, UK

³National Center for Text Mining, UK

{mmiwa,rune.saetre,yusuke,tsujii}@is.s.u-tokyo.ac.jp

Abstract

Because of the importance of protein-protein interaction (PPI) extraction from text, many corpora have been proposed with slightly differing definitions of proteins and PPI. Since no single corpus is large enough to saturate a machine learning system, it is necessary to learn from multiple different corpora. In this paper, we propose a solution to this challenge. We designed a rich feature vector, and we applied a support vector machine modified for corpus weighting (SVM-CW) to complete the task of multiple corpora PPI extraction. The rich feature vector, made from multiple useful kernels, is used to express the important information for PPI extraction, and the system with our feature vector was shown to be both faster and more accurate than the original kernel-based system, even when using just a single corpus. SVM-CW learns from one corpus, while using other corpora for support. SVM-CW is simple, but it is more effective than other methods that have been successfully applied to other NLP tasks earlier. With the feature vector and SVM-CW, our system achieved the best performance among all state-of-the-art PPI extraction systems reported so far.

1 Introduction

The performance of an information extraction program is highly dependent on various factors, including text types (abstracts, complete articles, reports, etc.), exact definitions of the information to be extracted, shared sub-topics of the text collections from which information is to be extracted.

Even if two corpora are annotated in terms of the same type of information by two groups, the performance of a program trained by one corpus is unlikely to be reproduced in the other corpus. On the other hand, from a practical point of view, it is worth while to effectively use multiple existing annotated corpora together, because it is very costly to make new annotations.

One problem with several different corpora is protein-protein interaction (PPI) extraction from text. While PPIs play a critical role in understanding the working of cells in diverse biological contexts, the manual construction of PPI databases such as BIND, DIP, HPRD, IntAct, and MINT (Mathivanan et al., 2006) is known to be very time-consuming and labor-intensive. The automatic extraction of PPI from published papers has therefore been a major research topic in Natural Language Processing for Biology (BioNLP).

Among several PPI extraction task settings, the most common is sentence-based, pair-wise PPI extraction. At least four annotated corpora have been provided for this setting: AIMed (Bunescu et al., 2005), HPRD50 (Fundel et al., 2006), IEPA (Ding et al., 2002), and LLL (Nédellec, 2005). Each of these corpora have been used as the standard corpus for training and testing PPI programs. Moreover, several corpora are annotated for more types of events than just for PPI. Such examples include BioInfer (Pyysalo et al., 2007), and GENIA (Kim et al., 2008a), and they can be reorganized into PPI corpora. Even though all of these corpora were made for PPI extraction, they were constructed based on different definitions of proteins and PPI, which reflect different biological research interests (Pyysalo et al., 2008).

Research on PPI extraction so far has revealed that the performance on each of the corpora could

XPG_{p1} protein interacts with multiple subunits of TFIIF_{prot} and with CSB_{p2} protein.

Figure 2: A sentence including an interacting protein pair (p1, p2). (AIMed PMID 8652557, 9th sentence, 3rd pair)

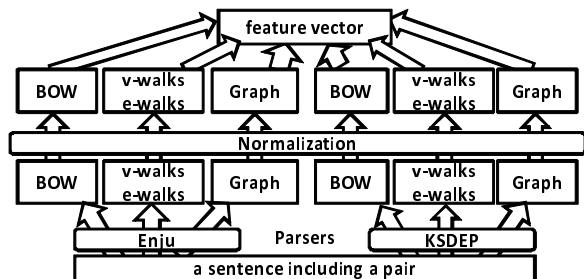


Figure 3: Extraction of a feature vector from the target sentence

of proteins in a sentence is interacting or not. Figure 2 shows an example of a sentence in which the given pair (p1 and p2) actually interacts.

Figure 1 shows the overview of the proposed PPI extraction system. As a classifier using a single corpus, we use the 2-norm soft-margin linear SVM (L2-SVM) classifier, with the dual coordinate decent (DCD) method, by Hsieh et al. (2008). In this section, we explain the two main features: the feature vector, and the corpus weighting method for multiple corpora.

3.1 Feature Vector

We propose a feature vector with three types of features, corresponding to the three different kernels, which were each combined with the two parsers: the Enju 2.3.0, and KSDEP beta 1 (Miyao et al., 2008); this feature vector is used because the kernels with these parsers were shown to be effective for PPI extraction by Miwa et al. (2008), and because it is important to start from a good performance single corpus system. Both parsers were retrained using the GENIA Treebank corpus provided by Kim et al. (2003). By using our linear feature vector, we can perform calculations faster by using fast linear classifiers like L2-SVM, and we also obtain a more accurate extraction, than by using the original kernel method.

Figure 3 summarizes the way in which the feature vector is constructed. The system extracts Bag-of-Words (BOW), shortest path (SP), and graph features from the output of two parsers. The

PROT_M:1, and_M:1, interact_M:1, multiple_M:1, of_M:1, protein_M:1, subunit_M:1, with_M:2, protein_A:1

Figure 4: Bag-of-Words features of the pair in Figure 2 with their positions (B:Before, M:in the Middle of, A:After) and frequencies.

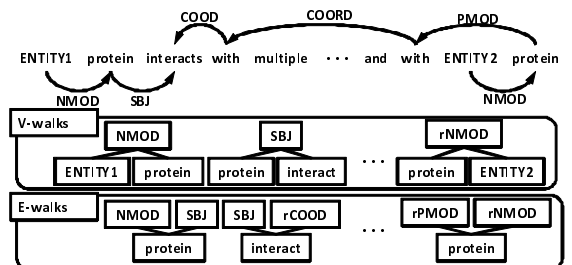


Figure 5: Vertex walks, edge walks in the upper shortest path between the proteins in the parse tree by KSDEP. The walks and their subsets are used as the shortest path features of the pair in Figure 2.

output is grouped according to the feature-type and parser, and each group of features is separately normalized by the L2-norm¹. Finally, all values are put into a single feature vector, and the whole feature vector is then also normalized by the L2-norm. The features are constructed by using predicate argument structures (PAS) from Enju, and by using the dependency trees from KSDEP.

3.1.1 Bag-of-Words (BOW) Features

The BOW feature includes the lemma form of a word, its relative position to the target pair of proteins (Before, Middle, After), and its frequency in the target sentence. BOW features form the BOW kernel in the original kernel method. BOW features for the pair in Figure 2 are shown in Figure 4.

3.1.2 Shortest Path (SP) Features

SP features include vertex walks (v-walks), edge walks (e-walks), and their subsets (Kim et al., 2008b) on the target pair in a parse structure, and represent the connection between the pair. The features are the subsets of the tree kernels on the shortest path (Sætre et al., 2007). Figure 5 illustrates the shortest path between the pair in Figure 2, and its v-walks and e-walks extracted from the shortest path in the parse tree by KSDEP. A v-walk includes two lemmas and their link, while

¹The vector normalized by the L2-norm is also called a unit vector.

an e-walk includes a lemma and its two links. The links indicates the predicate argument relations for PAS, and the dependencies for dependency trees.

3.1.3 Graph Features

Graph features are made from the all-paths graph kernel proposed by Airola et al. (2008). The kernel represents the target pair using graph matrices based on two subgraphs, and the graph features are all the non-zero elements in the graph matrices.

The two subgraphs are a parse structure subgraph (PSS) and a linear order subgraph (LOS). Figure 6 describes the subgraphs of the sentence parsed by KSDEP in Figure 2. PSS represents the parse structure of a sentence. PSS has word vertices or link vertices. A word vertex contains its lemma and its part-of-speech (POS), while a link vertex contains its link. Additionally, both types of vertices contain their positions relative to the shortest path. The “IP”s in the vertices on the shortest path represent the positions, and the vertices are differentiated from the other vertices like “P”, “CC”, and “and:CC” in Figure 6. LOS represents the word sequence in the sentence. LOS has word vertices, each of which contains its lemma, its relative position to the target pair, and its POS.

Each subgraph is represented by a graph matrix G as follows:

$$G = L^T \sum_{n=1}^{\infty} A^n L, \quad (1)$$

where L is a $N \times L$ label matrix, A is an $N \times N$ edge matrix, N represents the number of vertices, and L represents the number of labels. The label of a vertex includes all information described above (e.g. “ENTITY1:NN:IP” in Figure 6). If two vertices have exactly same information, the labels will be same. G can be calculated efficiently by using the Neumann Series (Airola et al., 2008). The label matrix represents the correspondence between labels and vertices. L_{ij} is 1 if the i -th vertex corresponds to the j -th label, and 0 otherwise. The edge matrix represents the connection between the pairs of vertices. A_{ij} is a weight w_{ij} (0.9 or 0.3 in Figure 6 (Airola et al., 2008)) if the i -th vertex is connected to the j -th vertex, and 0 otherwise. By this calculation, G_{ij} represent the sum of the weights of all paths between the i -th label and the j -th label.

	A	B	H	I	L
positive	1,000	2,534	163	335	164
all	5,834	9,653	433	817	330

Table 1: The sizes of used PPI corpora. A:AIMed, B:BioInfer, H:HPRD50, I:IEPA, and L:LLL.

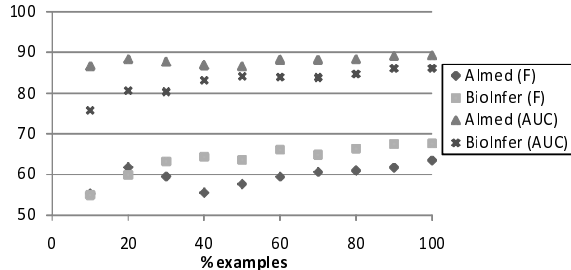


Figure 7: Learning curves on two large corpora. The x-axis is related to the percentage of the examples in a corpus. The curves are obtained by a 10-fold CV with a random split.

3.2 Corpus Weighting for Mixing Corpora

Table 1 shows the sizes of the PPI corpora that we used. Their widely-ranged differences including the sizes were manually analyzed by Pyysalo et al. (2008). While AIMed, HPRD50, IEPA, and LLL were all annotated as PPI corpora, BioInfer in its original form contains much more fine-grained information than does just the PPI. BioInfer was transformed into a PPI corpus by a program, so making it the largest of the five. Among them, AIMed alone was created by annotating whole abstracts, while the other corpora were made by annotating single sentences selected from abstracts.

Figure 7 shows the learning curves on two large corpora: AIMed and BioInfer. The curves are obtained by performing a 10-fold cross validation (CV) on each corpus, with random splits, using our system. The curves show that the performances can benefit from the additional examples. To get a better PPI extraction system for a chosen target, we need to draw useful shared information from external source corpora. We refer to examples in the source corpora as “source examples”, and examples in a target corpus as “target examples”. Among the corpora, we assume that the labels in some examples are incompatible, and that their distributions are also different, but that the feature space is shared.

In order to draw useful information from the source corpora to get a better model for the target

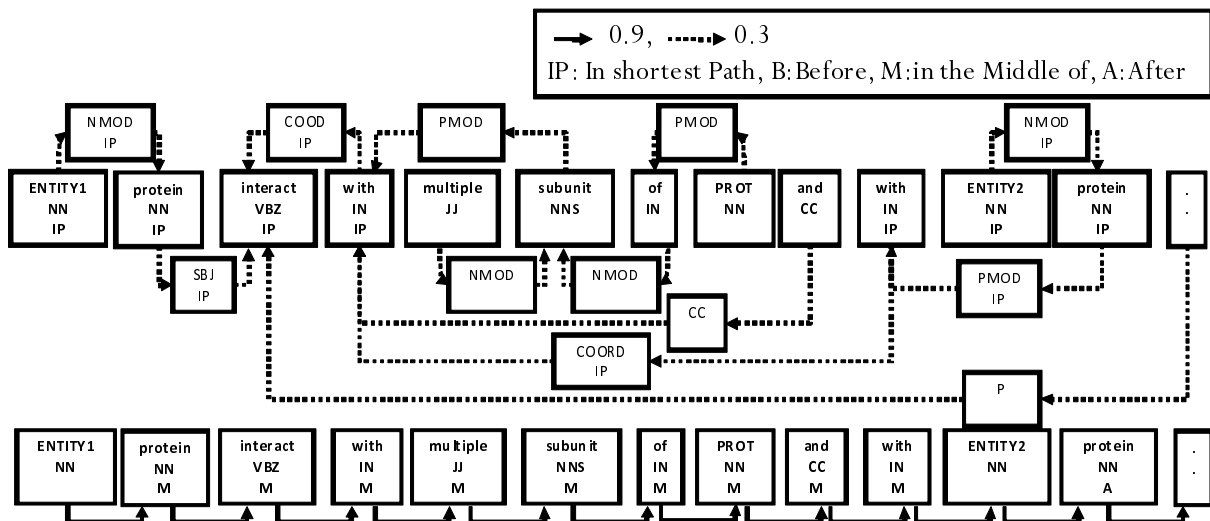


Figure 6: Parse structure subgraph and linear order subgraph to extract graph features of the pair in Figure 2. The parse structure subgraph is from the parse tree by KSDEP.

corpus, we use SVM-CW, which has been used as a DA method. Given a set of instance-label pairs (\mathbf{x}_i, y_i) , $i = 1, \dots, ls + lt$, $\mathbf{x}_i \in \mathbb{R}^n$, and $y_i \in \{-1, +1\}$, we solve the following problem:

$$\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C_s \sum_{i=1}^{ls} \ell_i + C_t \sum_{i=ls+1}^{ls+lt} \ell_i, \quad (2)$$

where \mathbf{w} is a weight vector, ℓ is a loss function, and ls and lt are the numbers of source and target examples respectively. $C_s \geq 0$ and $C_t \geq 0$ are penalty parameters. We use a squared hinge loss $\ell_i = \max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i)^2$. Here, the source corpora are treated as one corpus. The problem, excluding the second term, is equal to L2-SVM. The problem can be solved using the DCD method.

As an ITL method, SVM-CW weights each corpus, and tries to benefit from the source corpora, by adjusting the effect of their compatibility and incompatibility. For the adjustment, these penalty parameters should be set properly. Since we are unaware of the widely ranged differences among the corpora, we empirically estimated them by performing 10-fold CV on the training data.

4 Evaluation

4.1 Evaluation Settings

We used five corpora for evaluation: AImEd, BioInfer, HPRD50, IEPA, and LLL. For the comparison with other methods, we report the F-score (%), and the area under the receiver operating characteristic (ROC) curve (AUC) (%)

using (abstract-wise) a 10-fold CV and a one-answer-per-occurrence criterion. These measures are commonly used for the PPI extraction tasks. The F-score is a harmonic mean of Precision and Recall. The ROC curve is a plot of a true positive rate (TPR) vs a false positive rate (FPR) for different thresholds. We tuned the regularization parameters of all classifiers by performing a 10-fold CV on the training data using a random split. The other parameters were fixed, and we report the highest of the macro-averaged F-scores as our final F-score. For 10-fold CV, we split the corpora as recommended by Airola et al. (2008).

4.2 PPI Extraction on a Single Corpus

In this section, we evaluate our system on a single corpus, in order to evaluate our feature vector and to justify the use of the following modules: normalization methods and classification methods.

First, we compare our preprocessing method with other preprocessing methods to confirm how our preprocessing method improves the performance. Our method produced 64.2% in F-score using L2-SVM on AImEd. Scaling all features individually to have a maximal absolute value of 1, produced only 44.2% in the F-score, while normalizing the feature vector by L2-norm produced 61.5% in the F-score. Both methods were inferior to our method, because the values of features in the same group should be treated together, and because the values of features in the different groups should not have a big discrepancy. Weighting each

	L2	L1	LR	AP	CW
F	64.2	64.0	64.2	62.7	63.0
AUC	89.1	88.8	89.0	88.5	87.8

Table 2: Classification performance on AIMed using five different linear classifiers. The F-score (F) and Area Under the ROC curve (AUC) are shown. L2 is L2-SVM, L1 is L1-SVM, LR is logistic regression, AP is averaged perceptron, and CW is confidence weighted linear classification.

group with different values can produce better results, as will be explored in our future work.

Next, using our feature vector, we applied five different linear classifiers to extract PPI from AIMed: L2-SVM, 1-norm soft-margin SVM (L1-SVM), logistic regression (LR) (Fan et al., 2008), averaged perceptron (AP) (Collins, 2002), and confidence weighted linear classification (CW) (Dredze et al., 2008). Table 2 indicates the performance of these classifiers on AIMed. We employed better settings for the task than did the original methods for AP and CW. We used a Widrow-Hoff learning rule (Bishop, 1995) for AP, and we performed one iteration for CW. L2-SVM is as good as, if not better, than other classifiers (F-score and AUC). In the least, L2-SVM is as fast as these classifiers. AP and CW are worse than the other three methods, because they require a large number of examples, and are un-suitable for the current task. This result indicates that all linear classifiers, with the exception of AP and CW, perform almost equally, when using our feature vector.

Finally, we implemented the kernel method by Miwa et al. (2008). For a 10-fold CV on AIMed, the running time was 9,507 seconds, and the performance was 61.5% F-score and 87.1% AUC. Our system used 4,702 seconds, and the performance was 64.2% F-score and 89.1% AUC. This result displayed that our system, with L2-SVM, and our new feature vector, is better, and faster, than the kernel-based system.

4.3 Evaluation of Corpus Weighting

In this section, we first apply each model from a source corpus to a target corpus, to show how different the corpora are. We then evaluate SVM-CW by comparing it with three other methods (see Section 2) with limited features, and apply it to every corpus.

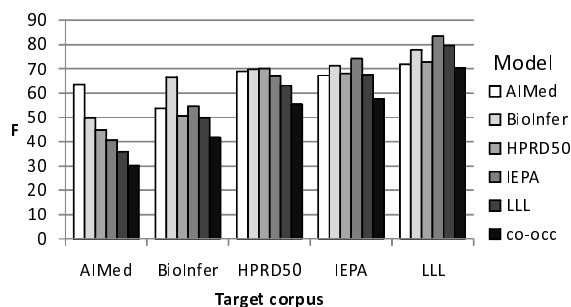


Figure 8: F-score on a target corpus using a model on a source corpus. For the comparison, we show the 10-fold CV result on each target corpus and co-occurrences. The regularization parameter was fixed to 1.

First, we apply the model from a source corpus to a target corpus. Figure 8 shows how the model from a source corpus performs on the target corpus. Interestingly, the model from IEPA performs better on LLL than the model from LLL itself. All the results showed that using different corpora (except IEPA) is worse than just using the same corpora. However, the cross-corpora scores are still better than the co-occurrences base-line, which indicates that the corpora share some information, even though they are not fully compatible.

Next, we compare SVM-CW with three other methods: aSVM, SVD-ASO, and TrAdaBoost. For this comparison, we used our feature vector without including the graph features, because SVD-ASO and TrAdaBoost require large computational resources. We applied SVD-ASO and TrAdaBoost in the following way. As for SVD-ASO, we made 400 auxiliary problems from the labels of each corpus by splitting features randomly, and extracted 50 additional features each for 4 feature groups. In total, we made new 200 additional features from 2,000 auxiliary problems. As recommended by Ando et al. (2005), we removed negative weights, performed SVD to each feature group, and iterated ASO once. Since AdaBoost easily overfitted with our rich feature vector, we applied soft margins (Ratsch et al., 2001) to TrAdaBoost. The update parameter for source examples was calculated using the update parameter on the training data in AdaBoost and the original parameter in TrAdaBoost. This ensures that the parameter would be the same as the original parameter, when the C value in the soft margin approaches infinity.

	aSVM		SVD-ASO		TrAdaBoost		SVM-CW		L2-SVM	
	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC
AIMed	63.6	88.4	62.9	88.3	63.4	88.4	64.0	88.6	63.2	88.4
BioInfer	66.5	85.2	65.7	85.1	66.1	85.2	66.7	85.4	66.2	85.1
HPRD50	71.2	84.3	68.7	80.8	72.6	85.3	72.7	86.4	67.2	80.7
IEPA	73.8	85.4	72.3	83.8	74.3	86.3	75.2	85.9	73.0	84.7
LLL	85.9	89.2	79.3	85.5	86.5	88.8	86.9	90.3	80.3	86.3

Table 3: Comparison of methods on multiple corpora. Our feature vector without graph features is used. The source corpora with the best F-scores are reported for aSVM, TrAdaBoost, and SVM-CW.

	F-score						AUC					
	A	B	H	I	L	all	A	B	H	I	L	all
A	(64.2)	64.0	64.7	65.2	63.7	64.2	(89.1)	89.5	89.2	89.3	89.0	89.4
B	67.9	(67.6)	67.9	67.9	67.7	68.3	86.2	(86.1)	86.2	86.3	86.2	86.4
H	71.3	71.2	(69.7)	74.1	70.8	74.9	84.7	85.0	(82.8)	85.0	83.4	87.9
I	74.4	75.6	73.7	(74.4)	74.4	76.6	86.7	87.1	85.4	(85.6)	86.9	87.8
L	83.2	85.9	82.0	86.7	(80.5)	84.1	86.3	87.1	87.4	90.8	(86.0)	86.2

Table 4: F-score and AUC by SVM-CW. Rows correspond to a target corpus, and columns a source corpus. A:AIMed, B:BioInfer, H:HPRD50, I:IEPA, and L:LLL corpora. “all” signifies that all source corpora are used as one source corpus, ignoring the differences among the corpora. For the comparison, we show the 10-fold CV result on each target corpus.

In Table 3, we demonstrate the results of the comparison. SVM-CW improved the classification performance at least as much as all the other methods. The improvement is mainly attributed to the aggressive use of source examples while learning the model. Some source examples can be used as training data, as indicated in Figure 8. SVM-CW does not set the restriction between C_s and C_t in Equation (2), so it can use source examples aggressively while learning the model. Since aSVM transfers a model, and SVD-ASO transfers an additional feature space, aSVM and SVD-ASO do not use the source examples while learning the model. In addition to the difference in the data usage, the settings of aSVM and SVD-ASO do not match the current task. As for aSVM, the DA assumption (that the labels are the same) does not match the task. In SVD-ASO, the numbers of both source examples and auxiliary problems are much smaller than those reported by Ando et al. (2005). TrAdaBoost uses the source examples while learning the model, but never increases the weight of the examples, and it attempts to reduce their effects.

Finally, we apply SVM-CW to all corpora using all features. Table 4 summarizes the F-score and AUC by SVM-CW with all features. SVM-CW

is especially effective for small corpora, showing that SVM-CW can adapt source corpora to a small annotated target corpus. The improvement on AIMed is small compared to the improvement on BioInfer, even though these corpora are similar in size. One of the reasons for this is that whole abstracts are annotated in AIMed, therefore making the examples biased. The difference between L2-SVM and SVM-CW + IEPA on AIMed is small, but statistically, it is significant (McNemar test (McNemar, 1947), $P = 0.0081$). In the cases of HPRD50 + IEPA, LLL + IEPA, and two folds in BioInfer + IEPA, C_s is larger than C_t in Equation (2). This is worth noting, because the source corpus is more weighted than the target corpus, and the prediction performance on the target corpus is improved. Most methods put more trust in the target corpus than in the source corpus, and our results show that this setting is not always effective for mixing corpora. The results also indicate that IEPA contains more useful information for extracting PPI than other corpora, and that using source examples aggressively is important for these combinations. We compared the results of L2-SVM and SVM-CW + IEPA on AIMed, and found that 38 pairs were described as “interaction” or “binding” in the sentences among 61

	SVM-CW		L2-SVM		Airola et al.	
	F	AUC	F	AUC	F	AUC
A	65.2	89.3	64.2	89.1	56.4	84.8
B	68.3	86.4	67.6	86.1	61.3	81.9
H	74.9	87.9	69.7	82.8	63.4	79.7
I	76.6	87.8	74.4	85.6	75.1	85.1
L	86.7	90.8	80.5	86.0	76.8	83.4

Table 6: Comparison with the results by Airola et al. (2008). A:AIMed, B:BioInfer, H:HPRD50, I:IEPA, and L:LLL corpora. The results with the highest F-score from Table 4 are reported as the results for SVM-CW.

newly found pairs. This analysis is evidence that IEPA contains instances to help find such interactions, and that SVM-CW helps to collect gold pairs that lack enough supporting instances in a single corpus, by adding instances from other corpora. SVM-CW missed coreferential relations that were also missed by L2-SVM. This can be attributed to the fact that the coreferential information is not stored in our current feature vector; so we need an even more expressive feature space. This is left as future work.

SVM-CW is effective on most corpus combinations, and all the models from single corpora can be improved by adding other source corpora. This result is impressive, because the baselines by L2-SVM on just single corpora are already better than or at least comparable to other state-of-the-art PPI extraction systems, and also because the variety of the differences among different corpora is quite wide depending on various factors including annotation policies of the corpora (Pyysalo et al., 2008). The results suggest that SVM-CW is useful as an ITL method.

4.4 Comparison with Other PPI Systems

We compare our system with other previously published PPI extraction systems. Tables 5 and 6 summarize the comparison. Table 5 summarizes the comparison of several PPI extraction systems evaluated on the AIMed corpus. As indicated, the performance of the heavy kernel method is lower than our fast rich feature-vector method. Our system is, to the extent of our knowledge, the best performing PPI extraction system evaluated on the AIMed corpus, both in terms of AUC and F-scores. Airola et al. (2008) first reported results using all five corpora. We cannot directly com-

pare our result with the F-score results, because they tuned the threshold, but our system still outperforms the system by Airola et al. (2008) on every corpus in AUC values. The results also indicate that our system outperforms other systems on all PPI corpora, and that both the rich feature vector and the corpus weighting are effective for the PPI extraction task.

5 Conclusion

In this paper, we proposed a PPI extraction system with a rich feature vector, using a corpus weighting method (SVM-CW) for combining the multiple PPI corpora. The feature vector extracts as much information as possible from the main training corpus, and SVM-CW incorporate other external source corpora in order to improve the performance of the classifier on the main target corpus. To the extent of our knowledge, this is the first application of ITL and DA methods to PPI extraction. As a result, the system, with SVM-CW and the feature vector, outperformed all other PPI extraction systems on all of the corpora. The PPI corpora share some information, and it is shown to be effective to add other source corpora when working with a specific target corpus.

The main contributions of this paper are: 1) conducting experiments in extracting PPI using multiple corpora, 2) suggesting a rich feature vector using several previously proposed features and normalization methods, 3) the combination of SVM with corpus weighting and the new feature vector improved results on this task compared with prior work.

There are many differences among the corpora that we used, and some of the differences are still unresolved. For further improvement, it would be necessary to investigate what is shared and what is different among the corpora. The SVM-CW method, and the PPI extraction system, can be applied generally to other classification tasks, and to other binary relation extraction tasks, without the need for modification. There are several other tasks in which many different corpora, which at first glance seem compatible, exist. By applying SVM-CW to such corpora, we will analyze which differences can be resolved by SVM-CW, and what differences require a manual resolution.

For the PPI extraction system, we found many false negatives that need to be resolved. For further improvement, we need to analyze the cause

	positive	all	P	R	F	AUC
SVM-CW	1,000	5,834	60.0	71.9	65.2	89.3
L2-SVM	1,000	5,834	62.7	66.6	64.2	89.1
(Miwa et al., 2008)	1,005	5,648	60.4	69.3	64.2 (61.5)	87.9 (87.1)
(Miyao et al., 2008)	1,059	5,648	54.9	65.5	59.5	
(Airola et al., 2008)	1,000	5,834	52.9	61.8	56.4	84.8
(Sætre et al., 2007)	1,068	5,631	64.3	44.1	52.0	
(Erkan et al., 2007)	951	4,020	59.6	60.7	60.0	
(Bunescu and Mooney, 2005)			65.0	46.4	54.2	

Table 5: Comparison with previous PPI extraction results on the AIMed corpus. The numbers of positive and all examples, precision (P), recall (R), F-score (F), and AUC are shown. The result with the highest F-score from Table 4 is reported as the result for SVM-CW. The scores in the parentheses of Miwa et al. (2008) indicate the result using the same 10-fold splits as our result, as indicated in Section 4.2.

of these false negatives more deeply, and design a more discriminative feature space. This is left as a future direction of our work.

Acknowledgments

This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan), Genome Network Project (MEXT, Japan), and Scientific Research (C) (General) (MEXT, Japan).

References

- Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross corpus learning. *BMC Bioinformatics*.
- Rie Kubota Ando, Tong Zhang, and Peter Bartlett. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.
- Rie Kubota Ando. 2006. Applying alternating structure optimization to word sense disambiguation. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 77–84, June.
- C. M. Bishop. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press.
- Razvan C. Bunescu and Raymond J. Mooney. 2005. Subsequence kernels for relation extraction. In *NIPS 2005*.
- Razvan C. Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.
- Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP 2002*, pages 1–8.
- Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007. Boosting for transfer learning. In *ICML 2007*, pages 193–200.
- J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. 2002. Mining medline: abstracts, sentences, or phrases? *Pacific Symposium on Biocomputing*, pages 326–337.
- Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *ICML 2008*, pages 264–271.
- Gunes Erkan, Arzucan Ozgur, and Dragomir R. Radev. 2007. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *EMNLP 2007*.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.
- Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2006. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.
- Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathya Keerthi, and S. Sundararajan. 2008. A dual coordinate descent method for large-scale linear SVM. In *ICML 2008*, pages 408–415.
- Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun’ichi Tsujii. 2003. GENIA corpus — a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–i182.
- Jin-Dong Kim, Tomoko Ohta, and Jun’ichi Tsujii. 2008a. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9:10.

- Seonho Kim, Juntae Yoon, and Jihoon Yang. 2008b. Kernel approaches for genic interaction extraction. *Bioinformatics*, 24(1):118–126.
- Suresh Mathivanan, Balamurugan Periaswamy, TKB Gandhi, Kumaran Kandasamy, Shubha Suresh, Riaz Mohmood, YL Ramachandra, and Akhilesh Pandey. 2006. An evaluation of human protein-protein interaction data in the public domain. *BMC Bioinformatics*, 7 Suppl 5:S19.
- Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, June.
- Makoto Miwa, Rune Sætre, Yusuke Miyao, Tomoko Ohta, and Jun’ichi Tsujii. 2008. Combining multiple layers of syntactic information for protein-protein interaction extraction. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 101–108.
- Yusuke Miyao, Rune Sætre, Kenji Sagae, Takuya Matsuzaki, and Jun’ichi Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. In *Proceedings of the 45th Meeting of the Association for Computational Linguistics (ACL’08:HLT)*.
- Claire Nédellec. 2005. Learning language in logic - genic interaction extraction challenge. In *Proceedings of the LLL’05 Workshop*.
- Sinno Jialin Pan and Qiang Yang. 2008. A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, November.
- Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8:50.
- Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. In *BMC Bioinformatics*, volume 9(Suppl 3), page S6.
- Gunnar Ratsch, Takashi Onoda, and Klaus-Robert Müller. 2001. Soft margins for adaboost. *Machine Learning*, 42(3):287–320.
- Rune Sætre, Kenji Sagae, and Jun’ichi Tsujii. 2007. Syntactic features for protein-protein interaction extraction. In *LBM 2007 short papers*.
- Gabriele Schweikert, Christian Widmer, Bernhard Schölkopf, and Gunnar Rätsch. 2008. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In *NIPS*, pages 1433–1440.
- Jun Yang, Rong Yan, and Alexander G. Hauptmann. 2007. Cross-domain video concept detection using adaptive SVMs. In *MULTIMEDIA ’07: Proceedings of the 15th international conference on Multimedia*, pages 188–197.