An Efficient Active Learning Framework for New Relation Types

Lisheng Fu
May, 2013

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science
Department of Computer Science
New York University

Prof. Ralph Grishman

Prof. Ernest Davis

Abstract

Relation extraction is a fundamental task in information extraction. Different methods have been studied for building a relation extraction system. Supervised training of models for this task has yielded good performance, but at substantial cost for the annotation of large training corpora (About 40K same-sentence entity pairs). Semi-supervised methods can only require a seed set, but the performance is very limited when the seed set is very small, which is not very satisfactory for real relation extraction applications. The trade-off of annotation and performance is also hard to decide in practice. Active learning strategies allow users to gradually improve the model and to achieve comparable performance to supervised methods with limited annotation. Recent study shows active learning on this task needs much fewer labels for each type to build a useful relation extraction application. We feel active learning is a good direction to do relation extraction and presents a more efficient active learning framework. This framework starts from a better balance between positive and negative samples, and boosts by interleaving self-training and co-testing. We also studied the reduction of annotation cost by enforcing argument type constraints. Experiments show a substantial speed-up by comparison to previous state-of-the-art pure co-testing active learning framework. We obtain reasonable performance with only a hundred labels for individual ACE 2004 relation types. We also developed a GUI tool for real human-in-the-loop active learning trials. The goal of building relation extraction systems in a very short time seems to be promising.

Table of Contents

1 Introduction

Relation extraction aims to discover the semantic relationship, if any, between a pair of entities in text.
E.g. *Mr. Smith, a senior programmer at Microsoft...*
[EMP-ORG.Employ-staff ("a senior programmer at Microsoft", "Microsoft")]

This structured information can be used to build higher-level applications such as question answering and other text mining applications.

Relation extraction was intensively studied as part of the multi-site ACE [Automatic Content Extraction] evaluations conducted in 2003, 2004, and 2005. For 2004, six major relation types were defined: Physical (PHYS), Personal/Social (PER-SOC), Employment/Membership/Subsidiary – (EMP-ORG), Agent-Artifact (ART), PER/ORG Affiliation (Other-AFF), GPE Affiliation (GPE-AFF). Each relation mention takes two entity mention arguments in the same sentence. In annotating text, each entity mention pair within one sentence will be labeled if it involves one of the relation types. As part of ACE, substantial hand-annotated corpora marked with entities and relations were produced. For example, the ACE 2004 corpus had in total about 5,000 relation instances (and about 45,000 same-sentence entity pairs not bearing one of these relations). These large training corpora stimulated research on the supervised training of relation extractors, with considerable success: the best systems, when given hand-tagged entities, correctly identify and classify relations with an F score above 70%.

Although supervised methods were effective, annotating a corpus of this size is too expensive in practice to serve as a model for developing new extractors: it requires annotation of 50K instances, of which only a small portion involve the target relation type. In consequence, most research has focused on reducing the annotation cost through semi-supervised learning methods such as bootstrapping systems. However, with limited labeled data, those semi-supervised systems failed to come close to the supervised level of performance. Their performance also varies with the distribution of seeds.

Recent studies have proposed new ways of reducing the annotation cost by using active learning. The advantage of active learning is that it can achieve reasonable performance, and even performance comparable to the supervised version, with few labeled examples, due to its ability to selectively sample unlabeled data for annotation.

Another means of minimizing annotation cost is utilizing large amounts of external unlabeled data. This has been done mostly through semi-supervised learning using multiple views. (Sun and Grishman 2012) proposed a co-testing framework for relation type extension by combining active learning with the analysis of large unlabeled data, and outperformed previous semi-supervised methods and basic active learning methods.

To further reduce the annotation cost and provide an efficient framework for rapidly developing relation extraction models, we combine active learning with semi-supervised methods, provide solutions to the imbalanced seed set and uneven co-testing classifiers, and incorporate argument constraints assistance. Most relation types now achieve reasonable performance with only a hundred labeled instances. Section 2 gives more related work in detail. Section 3 describes the enhancements we have made. Section 4 reports the experimental results and the improvement in performance when only a few instances have been labeled. Section 5 concludes the paper.

2 Related Work

For reducing the cost of annotation in the task of relation extraction, most prior work used semi-supervised learning. (Uszkoreit 2011) introduced a bootstrapping system for relation extraction rules, which achieved good performance under some circumstances. However, most previous semi-supervised methods have large performance gaps from supervised systems, and their performance depends on the choice of seeds (Vyas et al., 2009; Kozareva and Hovy, 2010).

Recent studies have shown the effectiveness of active learning for this task. (Zhang et al., 2012) proposed a unified framework for biomedical relation extraction. They used an SVM as the local classifier and tried

both uncertainty-based and density-based query functions and showed comparable results for the two methods. They also proposed using cosine-distance to ensure the diversity of queries.

(Donmez, Carbonell, & Bennett 2007) presented a dual strategy active learner which was reported to be better than other methods in the trade-off of uncertainty vs density in solving the problem of limited working range (only outperforming other methods in a certain range of number of labels) of different active learning strategies. (Roth and Small 2008) used an analogous method in their pipeline models of active learning of segmentation, entity classification and relation classification at the same time. They also adopted a regularized version of the structured perceptron (Collins 2002) instead of SVM and reported better results in active learning. Their work simulated the whole pipeline in active learning to achieve relation extraction, but had no specific research in the stage of relation extraction in the pipeline.

(Zhang 2010) proposed multi-task active learning with output constraints as a generalization of multi-view learning. The multi-task method relied on constraints on output between different tasks; this might be extended to situations where we need to learn relation sub-types as well as types, but was not applicable when relation extraction is an individual task.

The idea of multi-view learning in the co-testing framework has been used by (Sun and Grishman 2012). They proposed an LGCo-testing framework in which the local view is a maximum-entropy model with local features, and the global view is the global context distribution of the phrases between the two entity mentions of a relation in a large unlabeled corpus. Since the semantic role of a mention pair is highly dependent on the context, using this global view outperformed splitting the local view. The co-testing framework used KL-divergence as the extension of uncertainty. The query function was to select instances of highest relative entropy at each iteration.

None of these methods considered incorporating self-training methods or enforcing entity type constraints to boost active learning. The performance of these methods is more or less limited by the seed set, the improvement of which has not been well studied.

3 Method

3.1 Framework Structure

In active learning, users are asked to judge whether a particular sentence expresses the target relation between two entity mentions (Figure 1). For a fixed number of queries (fixed annotation cost), active learning is expected to achieve as high performance as possible. Our framework starts at a better initial setting (section 3.2), and then interleaves self-training with querying (section 3.3). We adopt the state-of-the-art co-testing based active learning algorithm (Sun and Grishman 2012) with a little tweak for imbalanced classifiers (section 3.4) as our query function. By enforcing entity type constraints to auto-label (section 3.5), the annotation cost could be further reduced. This framework is able to build a bridge between labeled data and unlabeled data more rapidly than previous pure co-testing based active learning.

The overall procedure of our framework is as follows:

Let:
U: unlabeled data
V: labeled data
(Labeled **pos**itive [relation] or **neg**ative [non-relation])
L: Local classifier
G: Global classifier

BEGIN
  // Initial set, section 3.2
  V = seed set
  Add Non-relations to V [see text]
  Train L, G on V

REPEAT
  //Co-testing based on L and G, section 3.3
  P = {x ∈ U | G(x) = **pos** & L(x) = **neg**}
  N = {x ∈ U | G(x) = **neg** & L(x) = **pos**} Select 5 queries from P ∪ N, preferring P;
    FOR each q ∈ queries
      //Entity type rules, section 3.5
      IF q violates entity type constraints
        THEN V += <q, **neg**>
        ELSE V += <q, *user-assigned label*>
      END IF
    END FOR
  Retrain L, G on V
  //Interleaved self-training, section 3.4
  Self-Train using both L, G to obtain positives and negatives and add to V
  Retrain L, G on V
END REPEAT
END

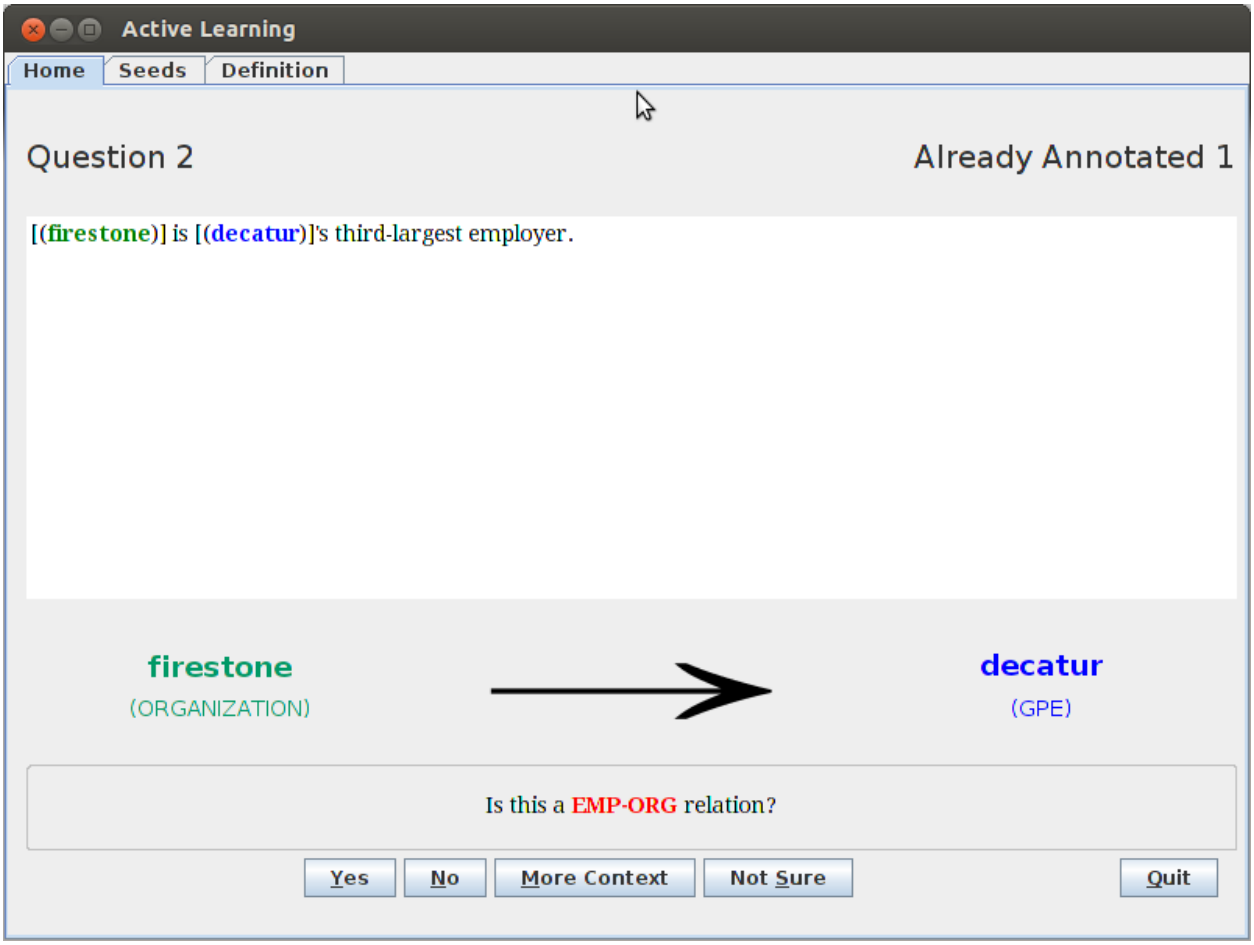

Figure 1. Interactive query to users

Entity 1	Entity 2	Full Sentence

ben bradley	cltv	[(ben bradley)] of chicago affiliate [(cltv)] has the story.
Bush	Cabinet	And, unlike other recent Bush visits to the ranch near Waco, Texas, no plans were made to bring reporters out for picture-taking sessions with prominent visitors or would-be members of a [(Bush)] [(Cabinet)].
legislator	Iran	[(Iran)]'s only Jewish [(legislator)] on Sunday criticized the treatment of non-Muslims in the country, the Islamic Republic News Agency reported.
denise	headline news	[(denise dillon)], "[(headline news)]."
executives	company	Schreiber also says the risk of cultural clashes between the companies is limited, because the companies disclosed publicly where top [(executives)] would stand in the combined [(company)] only a few weeks after the merger was announced.

Table 1. Randomly selected seeds from ACE 2004

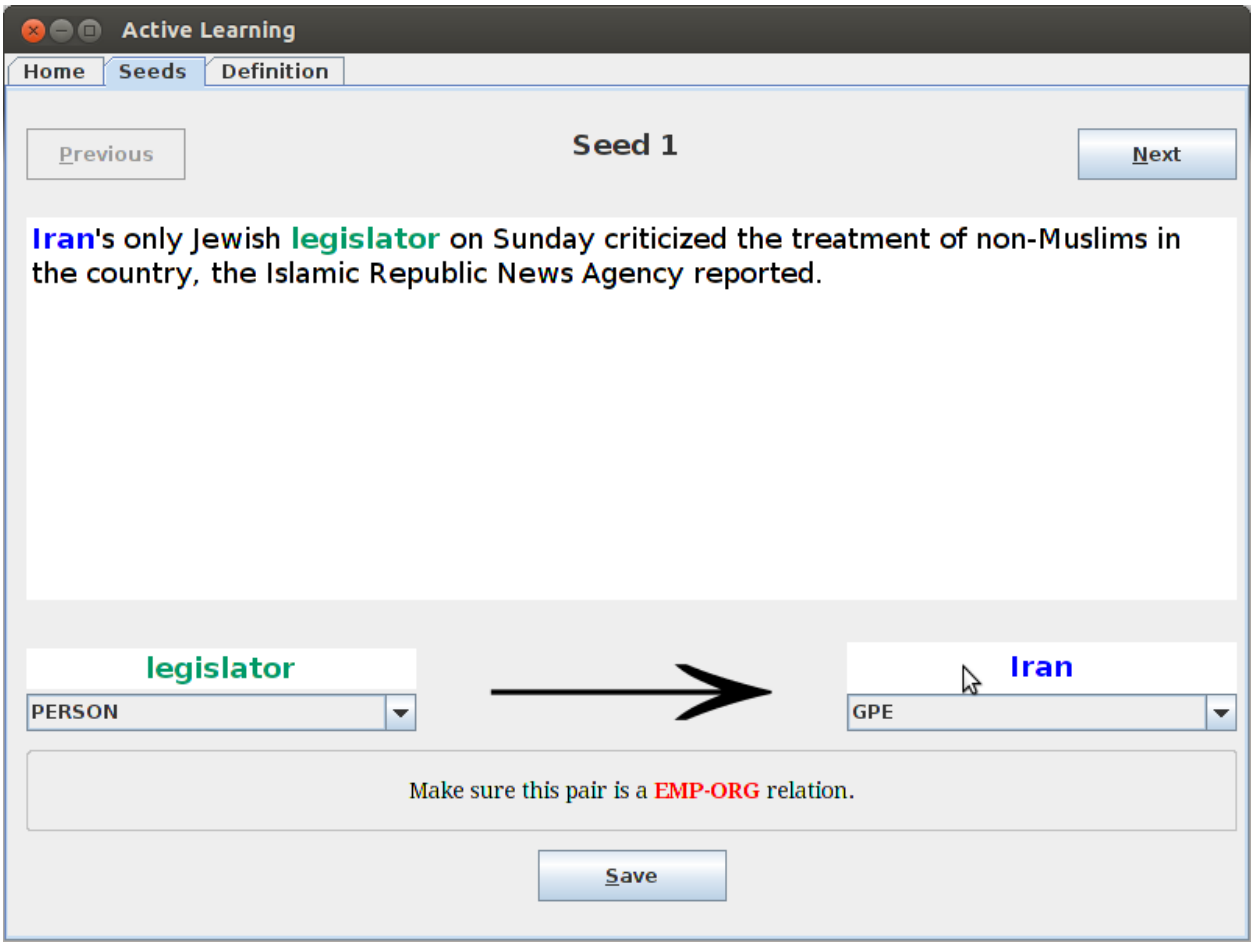

Figure 2. Interactive seed to users

3.2 Balance the Initial Set by Non-relation Selection

To initiate active learning, we require a small amount of seeds (5 in our experiments, e.g. Table 1) for the target relation type. In real applications, this could be interactive and obtained from users (Figure 2). Limited context without full sentences are also acceptable. Diverse seeds will be helpful, but not required. To train the model, we also need negative samples, which could be randomly selected from the corpus based on the assumption that positives are sparse enough. To guarantee no noise in a small negative set, we can also introduce human supervision on this small set. However, this initial set is still far from the real environment which has a lot more negative instances. As a result, this initial model gives poor performance, queries in early iterations look irrelevant to the target relation. We can do limited work to find more positives given the positive seeds (section 3.5 incorporating semi-supervised methods), but we can try to approximate the negative background better by adding a certain amount of high-confident negative samples automatically and then give the model the ability to distinguish most negative samples even in the very beginning.

In fact, the number of non-relation instances (mention pairs that are not the target type) is usually much larger than the number of target type instances. In ACE 2004, it's about 25 times larger than the most frequent relation, EMP-ORG. Random sampling could be used here because of the sparsity of positives. However, in the unlucky cases, the random sampling may introduce too many false positives, which is not acceptable for the initial set, even though active learning can deal with a certain degree of noise. To overcome this problem, we train an initial model by incrementally adding roughly guaranteed non-relations. Since every relation is defined under entity type constraints, we have a subset of the unlabeled data in which the mention pair violates these constraints of the target relation. The instances in this subset are mostly guaranteed as not target relations if human-labeled, and are roughly guaranteed if labeled by a NE tagger. Even if the quality of the NE tagger is limited, this subset will have much higher non-relation ratio than the whole set. By sampling from this subset of non-relations, we safely approximate the non-relation background of the unlabeled data and foster the early learning of the entity type rules. Thus the queries will also be more meaningful to users even at the beginning of the active learning process.

In implementing the sampling, we use the metric of how much of the non-relation subset we have learned instead of specifying a fixed number of instances. We train the model (a basic local feature classifier, the same as that in co-testing, section 3.3) on the labeled instances, apply the classifier to the so-far-unlabeled instances of this subset, and rank the instances by their uncertainty. We repeatedly select the five most uncertain instances, add them to the labeled set, and retrain the model until the model gives mostly correct predictions on classifying the non-relations in this subset. In the experiments, it is tuned to be 99% accurate on non-relations when the model has roughly balanced precision versus recall on target relations. The balanced model will be a better initial model for later active learning. Meanwhile, the way we add non-relations also enforces early learning of entity type constraints.

3.3 Co-testing based query function

When the initial set is ready, we can start selective sampling and ask queries to improve the model. We use a similar co-testing method as LGCo-Testing (Sun and Grishman 2012), the state-of-the-art active learning algorithm for relation type extension, but give preference to the weaker classifier to get benefit in the early interations.

LGCo-Testing uses co-testing based on the local view and the global view to select queries. The local classifier is a Maximum Entropy model that uses a rich set of lexical and syntactic features (from both constituent and dependency parses) as well as semantic type information for the arguments (Table 2). The global classifier relies on global context distribution, and it returns the relation type of the labeled instances to which the unlabeled instance is most similar (Table 3). The instances on which the two classifiers disagree is the contention set, from which queries are selected. Elements of the contention set are ranked by the

KL-divergence. Because of additional knowledge from the global view, this method exceeds other methods in active learning for relation extraction, and thus we choose this method as our query function.

Level	Type	Description	Value
Entity	*ET*	Entity types	*ET1=PERSON; ET2= LOCATION*
	ET12	Combination of ET1 and ET2	*ET12=PERSON--LOCATION*
	ML	Mention levels	*ML1=NAME; ML2= NOMNINAL*
	ML12	Combination of ML1 and ML2	*ML12=NAME--NOMINAL*
	HE	Heads of entities	*HE1=Clinton; HE2= border*
	HE12	Combination of HE1 and HE2	*Clinton--border*
	BagWE	Bag of words of entities	*{President, Clinton}{the, Irish, border}*
Sequence	*WBE1*	Words before entity 1	*{NIL}*
	WB	Words between	*{travel, to}*
	WAE2	Words after entity 2	*{for, an}*
	NUMWB	# words between	*2*
	TPatternET	Token pattern coupled with entity types	*PERSON_traveled_to_LOCATION*
Syntactic Parsing	*PTP*	Path of phrase labels connecting E1 and E2 in the parsing tree	*NP--VP--PP*
	PTPH	PTP augmented with the head word of the top phrase in the path	*NP--S--traveled--VP--PP*
	CPHBE1	Chunk heads before E1	
	{NIL}		
	CPHB	Chunk heads between	*{traveled, to}*
	CPHAE2	Chunk heads after E2	*{for, ceremony}*
	CPP	Path of phrase labels connecting the two enti-ties in the chunking	*NP--VP/S--PP--NP*
	CPPH	CPP augmented with head words	*NP-Clinton-VP/S-traveled-PP-to-NP-border*
	CPatternET	Chunk head pattern with entity types	*PERSON_traveled_to_LOCATION*
Dependency Parsing	*DPathET*	Shortest dependency path connecting the two entities coupled with entity types	*PER_nsubj'_trav-eled_prep_to_LOC*
	ET1DW1	Entity type and the dependent word for E1	*PERSON--traveled*
	ET2DW2	Entity type and the dependent word for E2	*LOCATION--to*
	H1DW1	Head and the dependent word for E1	*Clinton--traveled*
	H2DW2	Head word and the dependent word for E2	*border--to*
	ET12Same NP/PP/VP	Whether E1 and E2 included in the same NP/PP/VP	*false/false/true*

Table 2. Sample features for "[President (Clinton)] traveled to [the Irish (border)] for an …" (From Sun and Grishman 2012)

Traveled to	
Phrase	Similarity
visited	0.779
arrived in	0.763
worked in	0.751
lived in	0.719

Table 3. Sample Phrase Similarity (From Sun and Grishman 2012)

served in	0.686
consulted with	0.672

Even though we are satisfied with additional knowledge from the global view. The global classifier, in practice, is still much weaker than the local classifier. In principle, when the two classifiers are evenly matched, co-testing should work quite well at selecting informative instances. In this case, their settings often favor the local classifier in influencing the selection of examples. The instances in the contention set mostly come from the local classifier. However, in terms of diversity of queries, the global classifier is more capable of discovering unseen instances in the local feature space.

Active learning systems that are based on co-testing may have a similar problem. So we tried to compensate this through giving preference to the weaker classifier (In this case, the global classifier.). We distinguish two cases in the contention set. In case 1, the local classifier gives a positive prediction and the global classifier gives a negative prediction. This is the more frequent case when contention occurs. Because of the limited number of instances of positives, the sparsity of phrases for positives will cause the global classifier to make negative predictions very often.

In case 2, the global classifier predicts positive, and the local classifier predicts negative. This is the interesting case. Imagine that all the target type instances were divided into clusters based on local features. If the local classifier predicts positive, it is very likely that the local model has already been trained on some labeled instances in that cluster. Conversely, if the local model predicts negative for a target type, it is likely that the model does not include any labeled instances in that cluster. Initially it is important to cover more of these local feature clusters. This is similar to density-based strategies in active learning. Such strategies covering the local feature clusters first work better at few labels (Donmez, Carbonell, & Bennett 2007). The global classifier, which is based on a different view of data, has the ability to do this more accurately. To enhance this ability in the initial rounds of learning, we give case 2 priority over case 1 when selecting the five examples to query at each iteration (even if it may result in selecting only case 2 examples). To save the computing time, the selection is only made from top entropy instances (1000 in our experiments). When there is a substantial amount of annotated data, the local feature model will be able to cover the diversity from the global view. At this point there may be no case 2 examples among the top entropy instances, and we will naturally transition to case 1 examples. This actually gives a kind of mixture of uncertainty-based and density-based methods, which is expected to give better overall performance.

3.4 Interleaving Self-training

At each iteration of co-testing, the contention set from the local and global classifier will be the candidate set for queries to be given to users (section 3.3). With new labeled data, we can apply semi-supervised methods to further utilize this new knowledge. In this case, we simply apply self-training on which the two classifiers agree to gain more positive and negative instances without hand-labeling.

However, because of the sparsity of positives we tend to be fairly cautious in bootstrapping positives. By observation of early iteration self-trained results, we roughly pick a confidence threshold (0.8) to the local classifier. Again similarly to what we do in co-testing (section 3.3), we give preference to the global classifier (global phrase similarity). Because of the sparsity of the phrases extracted from the mention pair, the global classifier mostly gives negative results as default when no phrase similarity is detected. So when the global classifier gives positive predictions, it indicates there is global phrase similarity between the unlabeled instance and the labeled target instances, which is usually a high precision result given that the local classifier agrees. So we don't set any threshold on this global classifier. In practice, this is able to find positives much more quickly within a number of iterations. At the initial iterations, this directly improves the performance of the model.

In using those instances which both classifiers agree to be negative, we tend to be greedy. In fact, this is again selecting non-relations from unlabeled data, the same as that in the initial set setting. Nevertheless, in the middle of the active learning, the model is more robust to noisy data, and this agreed negative set is also closer to a pure non-relation set. We directly implement random sampling on this set to emphasize the

diversity since we do not need to worry about the accuracy. For simplicity, we pick the number of self-trained instances (both positive and negative) to be the same as the number of queries (5) at each iteration.

3.5 Entity Type Constraints

Relations are defined within entity type constraints. For instance, the EMP-ORG relation is limited to the types (PER – ORG), (PER – GPE), (ORG – ORG), (ORG – GPE), and (GPE – ORG) in ACE 2004.[1] In building real relation extraction systems for new types, these entity type constraints will be user-defined (Figure 3). In supervised learning, this is usually not a big problem. When the number of instances is large enough, the statistical model will effectively incorporate these entity type constraints as long as entity types are extracted as features. However, in active learning, even with suitable training examples, we will select and present to the user some instances violating these constraints. Applying explicit type filters would save a certain amount of human labeling effort. In practice, this still depends on the quality of the NE tagger. In the experiment section, we show that we can save a certain amount of annotation by using these simple constraints on hand-annotated entities. Since this amount is substantial, especially on some sparse types, we believe it is also helpful when using an imperfect NE tagger. A similar rule can be constructed to reject candidate relations where the two arguments are co-referential.

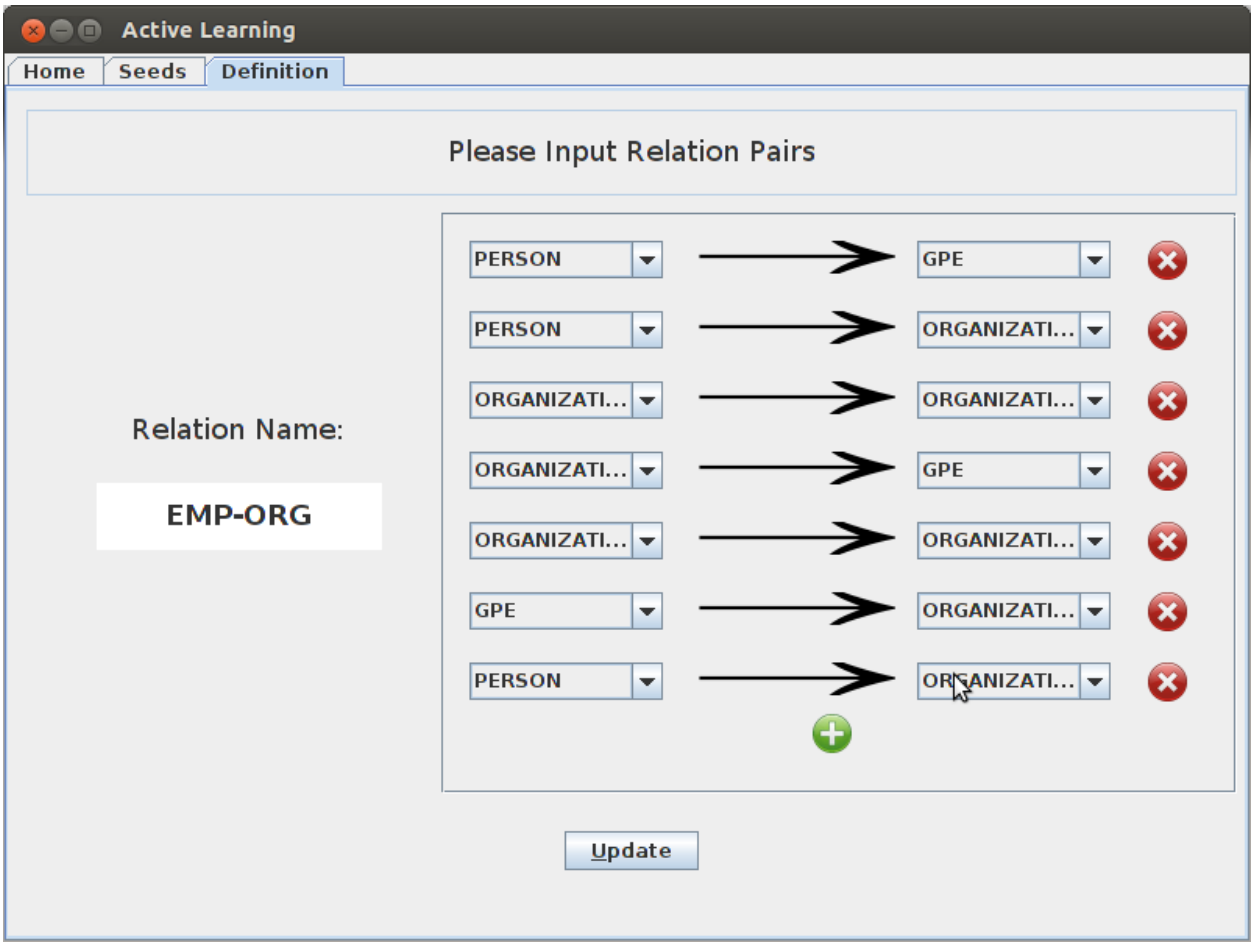

Figure 3. Entity type constraints from users

[1] PER = person, ORG = organization, GPE = geo-political entity: a location with a government.

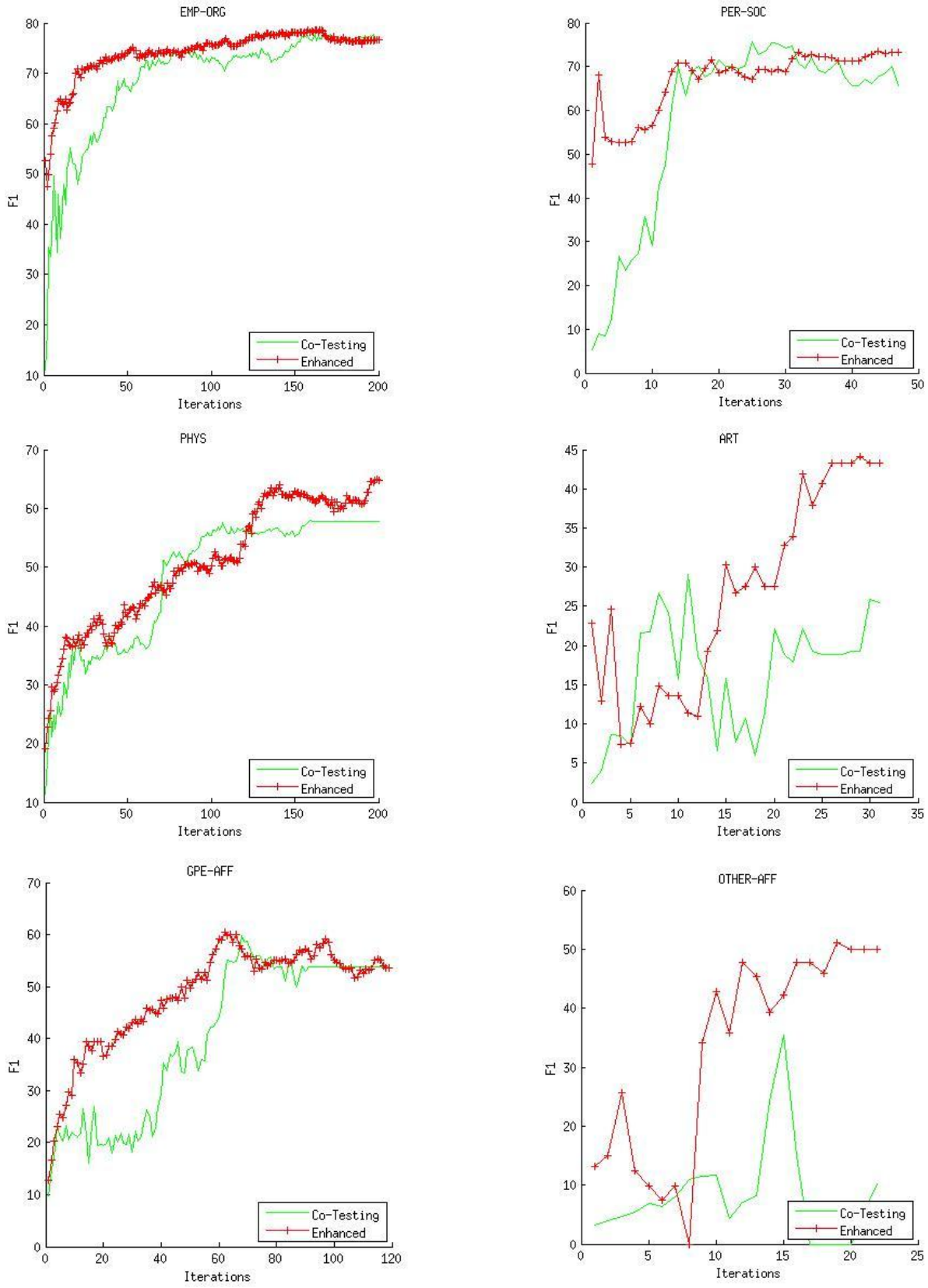

Figure 4: Comparison between baseline and our enhanced framework for all types in the binary case

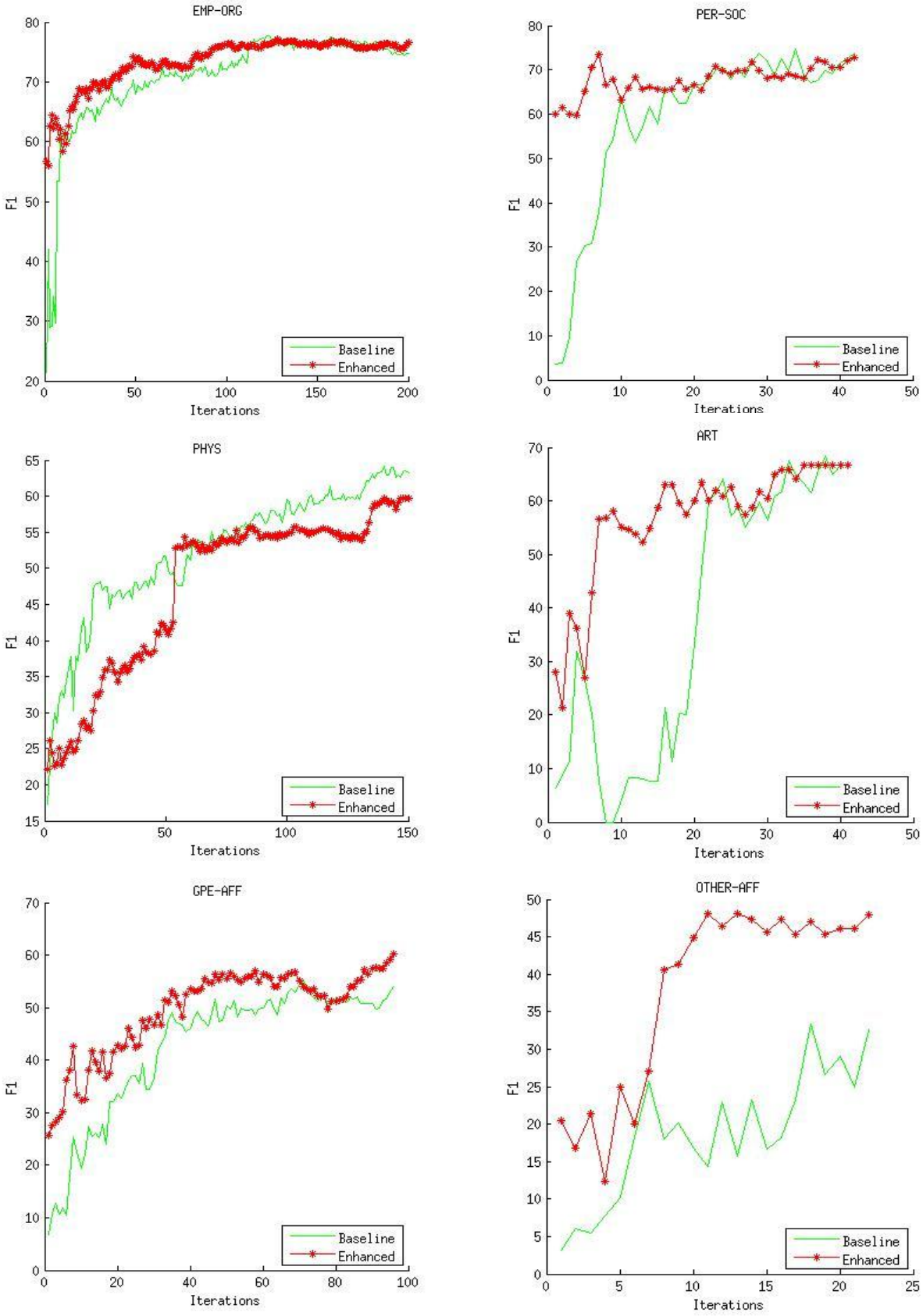

Figure 5: Comparison between baseline and our enhanced framework for all types (with aux)

4 Experiments

4.1 Experimental settings

We use the ACE 2004 corpus to simulate active learning. We pick one relation as the target type, and treat it as unlabeled. We collect all pairs of entity mentions appearing in the same sentence to be the candidates for query. Our task is to find the target relations and obtain reasonable performance using limited hand-labeled data. We use the original tags in the corpus to answer the queries during the active learning process, which simulates hand-labeling. We take randomly selected 4/5 of the corpus as the sampling space for active learning, and the remaining 1/5 as the test set.

4.2 Learning Speed evaluation

We compare our work to the pure co-testing based active learning (Sun and Grishman 2012), and show the F1 measure given the same number of iterations (5 queries per iteration). For random selection of target seeds, we use the same random sequence for both baseline and our framework for fair comparison. In the co-testing framework, the contention set will be empty at some point, which gives the final model of active learning. We show the comparison for each type (Figure 4), and report the overall improvement for early (30) iterations and the final performance (Table 4). The overall result is the average of the F1 measure of all types.

Even though in the non-relation selection, we applied early learning for entity type constraints, during the active learning process, there is still a portion of queries that could be answered automatically by entity type and co-reference rule filters. The hand-labeling cost could thereby be further reduced (Table 5). For some types with fewer instances, the reduction by these filters is substantial. In practice, this has to deal with noise from the NE tagger, but is still helpful as long as there is a decent NE tagger.

	30 iterations		stopping point: iterations	at stopping point	
	baseline	our system		baseline	our system
EMP-ORG	58.13	**71.52**	200	**76.81**	76.66
PHYS	34.63	**41.16**	200	57.85	**64.71**
GPE_AFF	18.18	**43.01**	119	**53.69**	53.68
PER-SOC	**74.29**	68.87	47	65.67	**73.13**
ART	25.93	**43.33**	31	25.45	**43.33**
OTHER-AFF	16.67	**50.00**	22	10.26	**50.00**
Overall	37.97	**52.98**	103	48.29	**60.25**

Table 4. Comparison with baseline (F1 score) in the binary case

Type	# queries in total	#queries that filters apply	Ratio
EMP-ORG	1000	91	9.1%
PHYS	1000	106	10.6%
GPE-AFF	590	84	14.2%
PER-SOC	234	64	27.3%
ART	151	54	35.8%
OTHER-AFF	105	56	53.3%

Table 5. # Instances auto-labeled by type constraints in the binary case

On the whole, our system substantially outperforms the baseline with a small number of labeled examples (100 instances, at the 20th iteration) and with relatively large amount annotation (The final model)

Our system does not work very well on the PHYS relation. It is likely that our non-relation selection strategy fails on this type. This type has rather loose entity type constraints. A lot of pairs are possible. Then we only have a very small subset for non-relation selection, and our strategy that enforces early learning on non-relations works poorly with this small subset of non-relations. As the result, our framework initially got lower scores on this type.

To show the effectiveness of each component of our framework, we display the overall performance comparison at early iterations (Figure 6). At this point, most of the six relations have not reached their stuck point, and so the benefits of the individual components are more evident.

The overall F1 score is the direct average of the F1 scores of six types. Non-relation approximation gives a large improvement since auto-labeling a certain amount of non-relations save quite a lot queries, and the better initial balance of positive and negative examples also makes the model select more informative queries from the beginning. The self-training boosts the system further as it incorporates more instances (especially positives) automatically. Considering the risk of obtaining false instances and the limited diversity of

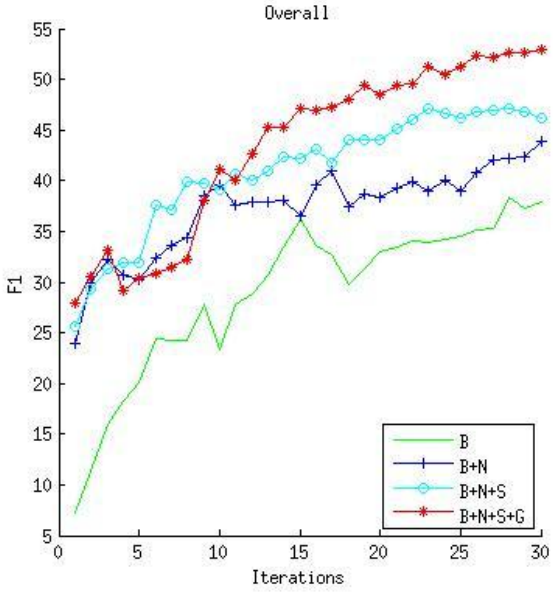

Figure 6: Improvement by different components in the binary case. B: Baseline, N: Non relations, S: Self-Training, G: Preference to the Global View

self-trained instances, only a modest gain can be expected. With further study in the future, improvement from this component to the active learning framework is highly possible. After these, the preference to the weaker classifier (the global view) gives improvement for 10 to 30 iterations. As a trade-off strategy between density and uncertainty, it is common that such methods only outperform the baseline for a certain duration.

With these components and auto-labeling with type constraints, we provide a quite reasonable relation extraction system given only 150 labels.[2] With more labels, we can approximate supervised learning. So we can build a relation extraction system quickly when there is no relation annotation in a new corpus. If we need more relations in this new corpus, we can start the framework again while adding the known (auxiliary) relations. Experiments on this relation type extension also show similar results (Table 6, 7, Figure 5, 7).

[2] Keep in mind that the best systems, trained on thousands of examples, only achieve F scores in the low 70's.

	20 iterations		stopping point: iterations	at stopping point	
	baseline	our system		baseline	our system
EMP-ORG	64.89	**68.32**	200	74.79	**76.63**
PHYS	**47.35**	30.24	150	**63.26**	59.66
GPE_AFF	33.53	**42.86**	96	54.08	**60.19**
PER-SOC	**66.67**	**66.67**	42	**73.76**	72.86
ART	33.30	**60.00**	41	**66.67**	**66.67**
OTHER-AFF	29.09	**46.15**	22	32.65	**48.00**
Overall	45.81	**52.37**	92	60.87	**64.00**

Table 6. Comparison with baseline (F1 score) with auxiliary relations

Type	# queries in total	#queries that filters apply	Ratio
EMP-ORG	1000	50	5.0%
PHYS	770	65	8.4%
GPE-AFF	509	80	15.7%
PER-SOC	210	52	24.8%
ART	224	69	30.8%
OTHER-AFF	135	55	40.7%

Table 7. # Instances auto-labeled by type constraints with auxiliary relations

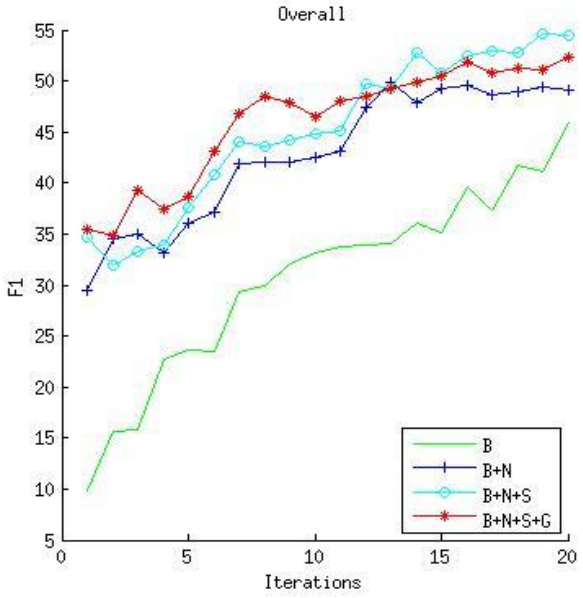

Figure 7: Improvement by different components with auxiliary relations. B: Baseline, N: Non relations, S: Self-Training, G: Preference to the Global View

5 Conclusion

We present a more practically efficient way to do active learning than a pure co-testing based algorithm. The improvement is most pronounced initially, for small numbers of annotations. We can now achieve reasonable performance for extracting relations with very little annotation. Adding a new relation in an hour now seems within reach.

Each component in the framework is still worth further studying. We can consider further enlarge and balance the initial set from the view of non-relation approximation. We can also try more adaptive semi-supervised algorithms to interleave with co-testing. The quality of the global classifier in the co-testing also remains a constraint, so we will be investigating alternative similarity metrics. While the experiments reported here involve simulated active learning, we are now planning real, human-in-the-loop active learning trials.

Acknowledgment

I would like to thank my advisor, Professor Ralph Grishman. In the past year, he gave me guidance of how to do research and write papers. His patient editing, incredible understanding and clear explanations help me a lot.

References

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Pinar Donmez and Jaime G. Carbonell and Paul N. Bennett. 2007. Dual strategy active learning. *In Proceedings. of the European Conference on Machine Learning (ECML).*

Zornista Kozareva and Eduard Hovy. 2010. Not all seeds are equal: Measuring the quality of text mining seeds. *In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL).*

Dan Roth and Kevin Small. 2008. Active learning for pipeline models. *In Proceedings of the 23rd national conference on Artificial intelligence (AAAI)*

Ang Sun and Ralph Grishman. 2012. Active Learning for Relation Type Extension with Local and Global Data Views. *In Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM).*

Hans Uszkoreit. 2011. Learning relation extraction grammars with minimal human intervention: strategy, results, insights and plans. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing (CICLing)*.

Vishnu Vyas, Patrick Pantel, Eric Crestan. 2009. Helping Editors Choose Better Seed Sets for Entity Expansion. *In Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM).*

Hong-Tao ZHANG, Min-Lie HUANG, Xiao-Yan ZHU. 2012. A Unified Active Learning Framework for Biomedical Relation Extraction. In *J. Comput. Sci. Technol.*, 27 (2012), Nr. 6, S. 1302-1313.

Yi Zhang. 2010. Multi-Task Active Learning with Output Constraints. *In Proceedings of the 24th national conference on Artificial intelligence (AAAI)*