Multi-Task Active Learning for Linguistic Annotations

Roi Reichart[1*] **Katrin Tomanek**[2*] **Udo Hahn**[2] **Ari Rappoport**[1]

[1]Institute of Computer Science
Hebrew University of Jerusalem, Israel
{roiri|arir}@cs.huji.ac.il

[2]Jena University Language & Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena, Germany
{katrin.tomanek|udo.hahn}@uni-jena.de

Abstract

We extend the classical single-task active learning (AL) approach. In the multi-task active learning (MTAL) paradigm, we select examples for several annotation tasks rather than for a single one as usually done in the context of AL. We introduce two MTAL meta-protocols, alternating selection and rank combination, and propose a method to implement them in practice. We experiment with a two-task annotation scenario that includes named entity and syntactic parse tree annotations on three different corpora. MTAL outperforms random selection and a stronger baseline, one-sided example selection, in which one task is pursued using AL and the selected examples are provided also to the other task.

1 Introduction

Supervised machine learning methods have successfully been applied to many NLP tasks in the last few decades. These techniques have demonstrated their superiority over both hand-crafted rules and unsupervised learning approaches. However, they require large amounts of labeled training data for every level of linguistic processing (e.g., POS tags, parse trees, or named entities). When, when domains and text genres change (e.g., moving from commonsense newspapers to scientific biology journal articles), extensive retraining on newly supplied training material is often required, since different domains may use different syntactic structures as well as different semantic classes (entities and relations). Consequently, with an increasing coverage of a wide variety of domains in human language technology (HLT) systems, we can expect a growing need for manual annotations to support many kinds of application-specific training data.

Creating annotated data is extremely labor-intensive. The Active Learning (AL) paradigm (Cohn et al., 1996) offers a promising solution to deal with this bottleneck, by allowing the learning algorithm to control the selection of examples to be manually annotated such that the human labeling effort be minimized. AL has been successfully applied already for a wide range of NLP tasks, including POS tagging (Engelson and Dagan, 1996), chunking (Ngai and Yarowsky, 2000), statistical parsing (Hwa, 2004), and named entity recognition (Tomanek et al., 2007).

However, AL is designed in such a way that it selects examples for manual annotation with respect to a *single* learning algorithm or classifier. Under this AL annotation policy, one has to perform a separate annotation cycle for each classifier to be trained. In the following, we will refer to the annotations supplied for a classifier as the annotations for a single *annotation task*.

Modern HLT systems often utilize annotations resulting from different tasks. For example, a machine translation system might use features extracted from parse trees and named entity annotations. For such an application, we obviously need the different annotations to reside in the same text corpus. It is not clear how to apply the single-task AL approach here, since a training example that is beneficial for one task might not be so for others. We could annotate

[*] Both authors contributed equally to this work.

Proceedings of ACL-08: HLT, pages 861–869,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

the same corpus independently by the two tasks and merge the resulting annotations, but that (as we show in this paper) would possibly yield sub-optimal usage of human annotation efforts.

There are two reasons why multi-task AL, and by this, a combined corpus annotated for various tasks, could be of immediate benefit. First, annotators working on *similar* annotation tasks (e.g., considering named entities and relations between them), might exploit annotation data from one subtask for the benefit of the other. If for each subtask a separate corpus is sampled by means of AL, annotators will definitely lack synergy effects and, therefore, annotation will be more laborious and is likely to suffer in terms of quality and accuracy. Second, for *dissimilar* annotation tasks – take, e.g., a comprehensive HLT pipeline incorporating morphological, syntactic and semantic data – a classifier might require features as input which constitute the output of another preceding classifier. As a consequence, training such a classifier which takes into account several annotation tasks will best be performed on a rich corpus annotated with respect to all input-relevant tasks. Both kinds of annotation tasks, similar and dissimilar ones, constitute examples of what we refer to as *multi-task* annotation problems.

Indeed, there have been efforts in creating resources annotated with respect to various annotation tasks though each of them was carried out independently of the other. In the general language UPenn annotation efforts for the WSJ sections of the Penn Treebank (Marcus et al., 1993), sentences are annotated with POS tags, parse trees, as well as discourse annotation from the Penn Discourse Treebank (Miltsakaki et al., 2008), while verbs and verb arguments are annotated with Propbank rolesets (Palmer et al., 2005). In the biomedical GENIA corpus (Ohta et al., 2002), scientific text is annotated with POS tags, parse trees, and named entities.

In this paper, we introduce *multi-task active learning* (MTAL), an active learning paradigm for multiple annotation tasks. We propose a new AL framework where the examples to be annotated are selected so that they are as informative as possible for a *set* of classifiers instead of a single classifier only. This enables the creation of a single combined corpus annotated with respect to various annotation tasks, while preserving the advantages of AL with respect to the minimization of annotation efforts.

In a proof-of-concept scenario, we focus on two highly dissimilar tasks, syntactic parsing and named entity recognition, study the effects of multi-task AL under rather extreme conditions. We propose two MTAL meta-protocols and a method to implement them for these tasks. We run experiments on three corpora for domains and genres that are very different (WSJ: newspapers, Brown: mixed genres, and GENIA: biomedical abstracts). Our protocols outperform two baselines (random and a stronger one-sided selection baseline).

In Section 2 we introduce our MTAL framework and present two MTAL protocols. In Section 3 we discuss the evaluation of these protocols. Section 4 describes the experimental setup, and results are presented in Section 5. We discuss related work in Section 6. Finally, we point to open research issues for this new approach in Section 7.

2 A Framework for Multi-Task AL

In this section we introduce a sample selection framework that aims at reducing the human annotation effort in a multiple annotation scenario.

2.1 Task Definition

To measure the efficiency of selection methods, we define the *training quality* TQ of annotated material S as the performance p yielded with a reference learner X trained on that material: $TQ(X, S) = p$. A selection method can be considered better than another one if a higher TQ is yielded with the same amount of examples being annotated.

Our framework is an extension of the Active Learning (AL) framework (Cohn et al., 1996)). The original AL framework is based on querying in an iterative manner those examples to be manually annotated that are most useful for the learner at hand. The TQ of an annotated corpus selected by means of AL is much higher than random selection. This AL approach can be considered as *single-task AL* because it focuses on a single learner for which the examples are to be selected. In a multiple annotation scenario, however, there are several annotation tasks to be accomplished at once and for each task typically a separate statistical model will then be trained. Thus, the goal of *multi-task AL* is to query those examples for human annotation that are most informative for *all* learners involved.

2.2 One-Sided Selection vs. Multi-Task AL

The naive approach to select examples in a multiple annotation scenario would be to perform a single-task AL selection, i.e., the examples to be annotated are selected with respect to one of the learners only.[1] In a multiple annotation scenario we call such an approach *one-sided* selection. It is an *intrinsic* selection for the reference learner, and an *extrinsic* selection for all the other learners also trained on the annotated material. Obviously, a corpus compiled with the help of one-sided selection will have a good TQ for that learner for which the intrinsic selection has taken place. For all the other learners, however, we have no guarantee that their TQ will not be inferior than the TQ of a random selection process.

In scenarios where the different annotation tasks are highly dissimilar we can expect extrinsic selection to be rather poor. This intuition is demonstrated by experiments we conducted for named entity (NE) and parse annotation tasks[2] (Figure 1). In this scenario, extrinsic selection for the NE annotation task means that examples where selected with respect to the parsing task. Extrinsic selection performed about the same as random selection for the NE task, while for the parsing task extrinsic selection performed markedly worse. This shows that examples that were very informative for the NE learner were not that informative for the parse learner.

2.3 Protocols for Multi-Task AL

Obviously, we can expect one-sided selection to perform better for the reference learner (the one for which an intrinsic selection took place) than multi-task AL selection, because the latter would be a compromise for all learners involved in the multiple annotation scenario. However, the goal of multi-task AL is to minimize the annotation effort over all annotation tasks and not just the effort for a single annotation task.

For a multi-task AL protocol to be valuable in a specific multiple annotation scenario, the TQ for all considered learners should be

1. better than the TQ of random selection,
2. and better than the TQ of any extrinsic selection.

In the following, we introduce two protocols for multi-task AL. Multi-task AL protocols can be considered *meta-protocols* because they basically specify how task-specific, single-task AL approaches can be combined into one selection decision. By this, the protocols are independent of the underlying task-specific AL approaches.

2.3.1 Alternating Selection

The *alternating selection* protocol alternates one-sided AL selection. In s_j consecutive AL iterations, the selection is performed as one-sided selection with respect to learning algorithm X_j. After that, another learning algorithm is considered for selection for s_k consecutive iterations and so on. Depending on the specific scenario, this enables to weight the different annotation tasks by allowing them to guide the selection in more or less AL iterations. This protocol is a straight-forward compromise between the different single-task selection approaches.

In this paper we experiment with the special case of $s_i = 1$, where in every AL iteration the selection leadership is changed. More sophisticated calibration of the parameters s_i is beyond the scope of this paper and will be dealt with in future work.

2.3.2 Rank Combination

The *rank combination* protocol is more directly based on the idea to combine single-task AL selection decisions. In each AL iteration, the usefulness score $s_{X_j}(e)$ of each unlabeled example e from the pool of examples is calculated with respect to each learner X_j and then translated into a rank $r_{X_j}(e)$ where higher usefulness means lower rank number (examples with identical scores get the same rank number). Then, for each example, we sum the rank numbers of each annotation task to get the overall rank $r(e) = \sum_{j=1}^{n} r_{X_j}(e)$. All examples are sorted by this combined rank and b examples with lowest rank numbers are selected for manual annotation.[3]

[1] Of course, all selected examples would be annotated w.r.t. all annotation tasks.

[2] See Section 4 for our experimental setup.

[3] As the number of ranks might differ between the single annotation tasks, we normalize them to the coarsest scale. Then we can sum up the ranks as explained above.

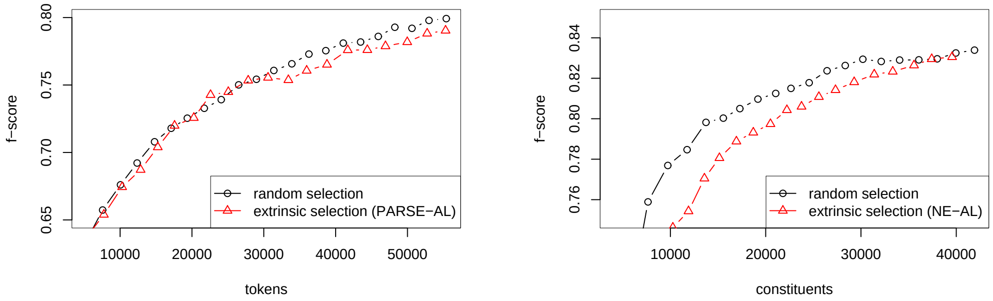

Figure 1: Learning curves for random and extrinsic selection on both tasks: named entity annotation (left) and syntactic parse annotation (right), using the `WSJ` corpus scenario

This protocol favors examples which are good for all learning algorithms. Examples that are highly informative for one task but rather uninformative for another task will not be selected.

3 Evaluation of Multi-Task AL

The notion of training quality (TQ) can be used to quantify the effectiveness of a protocol, and by this, annotation costs in a single-task AL scenario. To actually quantify the overall training quality in a multiple annotation scenario one would have to sum over all the single task's TQs. Of course, depending on the specific annotation task, one would not want to quantify the number of examples being annotated but different task-specific units of annotation. While for entity annotations one does typically count the number of tokens being annotated, in the parsing scenario the number of constituents being annotated is a generally accepted measure. As, however, the actual time needed for the annotation of one example usually differs for different annotation tasks, normalizing exchange rates have to be specified which can then be used as weighting factors. In this paper, we do not define such weighting factors[4], and leave this challenging question to be discussed in the context of psycholinguistic research.

We could quantify the overall efficiency score E of a MTAL protocol P by

$$E(P) = \sum_{j=1}^{n} \alpha_j \cdot TQ(X_j, u_j)$$

where u_j denotes the individual annotation task's number of units being annotated (e.g., constituents for parsing) and the task-specific weights are defined by α_j. Given weights are properly defined, such a score can be applied to directly compare different protocols and quantify their differences.

In practice, such task-specific weights might also be considered in the MTAL protocols. In the alternating selection protocol, the numbers of consecutive iterations s_i each single task protocol can be tuned according to the α parameters. As for the rank combination protocol, the weights can be considered when calculating the overall rank: $r(e) = \sum_{j=1}^{n} \beta_j \cdot r_{X_j}(e)$ where the parameters $\beta_1 \dots \beta_n$ reflect the values of $\alpha_1 \dots \alpha_n$ (though they need not necessarily be the same).

In our experiments, we assumed the same weight for all annotation schemata, thus simply setting $s_i = 1, \beta_i = 1$. This was done for the sake of a clear framework presentation. Finding proper weights for the single tasks and tuning the protocols accordingly is a subject for further research.

4 Experiments

4.1 Scenario and Task-Specific Selection Protocols

The tasks in our scenario comprise one semantic task (annotation with named entities (NE)) and one syntactic task (annotation with PCFG parse trees). The tasks are highly dissimilar, thus increasing the potential value of MTAL. Both tasks are subject to intensive research by the NLP community.

The MTAL protocols proposed are meta-protocols that combine the selection decisions of the underlying, task-specific AL protocols. In our scenario, the task-specific AL protocols are

[4] Such weighting factors not only depend on the annotation level or task but also on the domain, and especially on the cognitive load of the annotation task.

committee-based (Freund et al., 1997) selection protocols. In committee-based AL, a committee consists of k classifiers of the same type trained on different subsets of the training data.[5] Each committee member then makes its predictions on the unlabeled examples, and those examples on which the committee members disagree most are considered most informative for learning and are thus selected for manual annotation. In our scenario the example grain-size is the sentence level.

For the NE task, we apply the AL approach of Tomanek et al. (2007). The committee consists of $k_1 = 3$ classifiers and the vote entropy (VE) (Engelson and Dagan, 1996) is employed as disagreement metric. It is calculated on the token-level as

$$VE_{tok}(t) = -\frac{1}{\log k} \sum_{i=0}^{c} \frac{V(l_i,t)}{k} \log \frac{V(l_i,t)}{k} \quad (1)$$

where $\frac{V(l_i,t)}{k}$ is the ratio of k classifiers where the label l_i is assigned to a token t. The sentence level vote entropy VE_{sent} is then the average over all tokens t_j of sentence s.

For the parsing task, the disagreement score is based on a committee of $k_2 = 10$ instances of Dan Bikel's reimplementation of Collins' parser (Bickel, 2005; Collins, 1999). For each sentence in the unlabeled pool, the agreement between the committee members was calculated using the function reported by Reichart and Rappoport (2007):

$$AF(s) = \frac{1}{N} \sum_{i,l \in [1 \ldots N], i \neq l} fscore(m_i, m_l) \quad (2)$$

Where m_i and m_l are the committee members and $N = \frac{k_2 \cdot (k_2 - 1)}{2}$ is the number of pairs of different committee members. This function calculates the agreement between the members of each pair by calculating their relative f-score and then averages the pairs' scores. The disagreement of the committee on a sentence is simply $1 - AF(s)$.

4.2 Experimental settings

For the NE task we employed the classifier described by Tomanek et al. (2007): The NE tagger is based on Conditional Random Fields (Lafferty et al., 2001) and has a rich feature set including orthographical, lexical, morphological, POS, and contextual features. For parsing, Dan Bikel's reimplementation of Collins' parser is employed, using gold POS tags.

In each AL iteration we select 100 sentences for manual annotation.[6] We start with a randomly chosen seed set of 200 sentences. Within a corpus we used the same seed set in all selection scenarios. We compare the following five selection scenarios: Random selection (*RS*), which serves as our baseline; one-sided AL selection for both tasks (called *NE-AL* and *PARSE-AL*); and multi-task AL selection with the alternating selection protocol (*alter-MTAL*) and the rank combination protocol (*ranks-MTAL*).

We performed our experiments on three different corpora, namely one from the newspaper genre (`WSJ`), a mixed-genre corpus (`Brown`), and a biomedical corpus (`Bio`). Our simulation corpora contain both entity annotations and (constituent) parse annotations. For each corpus we have a pool set (from which we select the examples for annotation) and an evaluation set (used for generating the learning curves). The `WSJ` corpus is based on the WSJ part of the PENN TREEBANK (Marcus et al., 1993); we used the first 10,000 sentences of section 2-21 as the pool set, and section 00 as evaluation set (1,921 sentences). The `Brown` corpus is also based on the respective part of the PENN TREEBANK. We created a sample consisting of 8 of any 10 consecutive sentences in the corpus. This was done as `Brown` contains text from various English text genres, and we did that to create a representative sample of the corpus domains. We finally selected the first 10,000 sentences from this sample as pool set. Every 9th from every 10 consecutive sentences package went into the evaluation set which consists of 2,424 sentences. For both `WSJ` and `Brown` only parse annotations though no entity annotations were available. Thus, we enriched both corpora with entity annotations (three entities: person, location, and organization) by means of a tagger trained on the English data set of the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003).[7] The `Bio` corpus

[5] We randomly sampled $L = \frac{3}{4}$ of the training data to create each committee member.

[6] Manual annotation is simulated by just unveiling the annotations already contained in our corpora.

[7] We employed a tagger similar to the one presented by Settles (2004). Our tagger has a performance of ≈ 84% f-score on the CoNLL-2003 data; inspection of the predicted entities on

is based on the parsed section of the GENIA corpus (Ohta et al., 2002). We performed the same divisions as for Brown, resulting in 2,213 sentences in our pool set and 276 sentences for the evaluation set. This part of the GENIA corpus comes with entity annotations. We have collapsed the entity classes annotated in GENIA (cell line, cell type, DNA, RNA, protein) into a single, biological entity class.

5 Results

In this section we present and discuss our results when applying the five selection strategies (RS, NE-AL, PARSE-AL, alter-MTAL, and ranks-MTAL) to our scenario on the three corpora. We refrain from calculating the overall efficiency score (Section 3) here due to the lack of generally accepted weights for the considered annotation tasks. However, we require from a good selection protocol to exceed the performance of random selection and extrinsic selection. In addition, recall from Section 3 that we set the alternate selection and rank combination parameters to $s_i = 1, \beta_i = 1$, respectively to reflect a tradeoff between the annotation efforts of both tasks.

Figures 2 and 3 depict the learning curves for the NE tagger and the parser on WSJ and Brown, respectively. Each figure shows the five selection strategies. As expected, on both corpora and both tasks intrinsic selection performs best, i.e., for the NE tagger NE-AL and for the parser PARSE-AL. Further, random selection and extrinsic selection perform worst. Most importantly, both MTAL protocols clearly outperform extrinsic and random selection in all our experiments. This is in contrast to NE-AL which performs worse than random selection for all corpora when used as extrinsic selection, and for PARSE-AL that outperforms the random baseline only for Brown when used as extrinsic selection. That is, the MTAL protocols suggest a tradeoff between the annotation efforts of the different tasks, here.

On WSJ, both for the NE and the parse annotation tasks, the performance of the MTAL protocols is very similar, though ranks-MTAL performs slightly better. For the parser task, up to 30,000 constituents MTAL performs almost as good as does PARSE-AL. This is different for the NE task where NE-AL clearly outperforms MTAL. On Brown, in general we see the same results, with some minor differences. On the NE task, extrinsic selection (PARSE-AL) performs better than random selection, but it is still much worse than intrinsic AL or MTAL. Here, ranks-MTAL significantly outperforms alter-MTAL and almost performs as good as intrinsic selection. For the parser task, we see that extrinsic and random selection are equally bad. Both MTAL protocols perform equally well, again being quite similar to the intrinsic selection. On the BIO corpus[8] we observed the same tendencies as in the other two corpora, i.e., MTAL clearly outperforms extrinsic and random selection and supplies a better tradeoff between annotation efforts of the task at hand than one-sided selection.

Overall, we can say that in all scenarios MTAL performs much better than random selection and extrinsic selection, and in most cases the performance of MTAL (especially but not exclusively, ranks-MTAL) is even close to intrinsic selection. This is promising evidence that MTAL selection can be a better choice than one-sided selection in multiple annotation scenarios. Thus, considering all annotation tasks in the selection process (even if the selection protocol is as simple as the alternating selection protocol) is better than selecting only with respect to one task. Further, it should be noted that overall the more sophisticated rank combination protocol does not perform much better than the simpler alternating selection protocol in all scenarios.

Finally, Figure 4 shows the disagreement curves for the two tasks on the WSJ corpus. As has already been discussed by Tomanek and Hahn (2008), disagreement curves can be used as a stopping criterion and to monitor the progress of AL-driven annotation. This is especially valuable when no annotated validation set is available (which is needed for plotting learning curves). We can see that the disagreement curves significantly flatten approximately at the same time as the learning curves do. In the context of MTAL, disagreement curves might not only be interesting as a stopping criterion but rather as a switching criterion, i.e., to identify when MTAL could be turned into one-sided selection. This would be the case if in an MTAL scenario, the disagree-

WSJ and Brown revealed a good tagging performance.

[8] The plots for the Bio are omitted due to space restrictions.

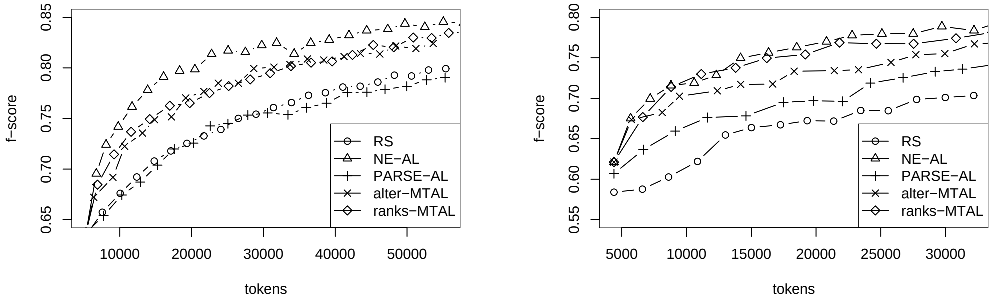

Figure 2: Learning curves for NE task on `WSJ` (left) and `Brown` (right)

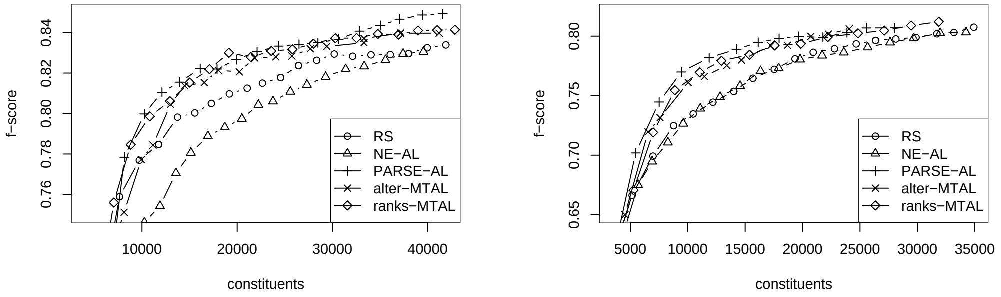

Figure 3: Learning curves for parse task on `WSJ` (left) and `Brown` (right)

ment curve of one task has a slope of (close to) zero. Future work will focus on issues related to this.

6 Related Work

There is a large body of work on single-task AL approaches for many NLP tasks where the focus is mainly on better, task-specific selection protocols and methods to quantify the usefulness score in different scenarios. As to the tasks involved in our scenario, several papers address AL for NER (Shen et al., 2004; Hachey et al., 2005; Tomanek et al., 2007) and syntactic parsing (Tang et al., 2001; Hwa, 2004; Baldridge and Osborne, 2004; Becker and Osborne, 2005). Further, there is some work on questions arising when AL is to be used in real-life annotation scenarios, including impaired inter-annotator agreement, stopping criteria for AL-driven annotation, and issues of reusability (Baldridge and Osborne, 2004; Hachey et al., 2005; Zhu and Hovy, 2007; Tomanek et al., 2007).

Multi-task AL is methodologically related to approaches of decision combination, especially in the context of classifier combination (Ho et al., 1994) and ensemble methods (Breiman, 1996). Those approaches focus on the combination of classifiers in order to improve the classification error rate for one specific classification task. In contrast, the focus of multi-task AL is on strategies to select training material for multi classifier systems where all classifiers cover different classification tasks.

7 Discussion

Our treatment of MTAL within the context of the orthogonal two-task scenario leads to further interesting research questions. First, future investigations will have to focus on the question whether the positive results observed in our orthogonal (i.e., highly dissimilar) two-task scenario will also hold for a more realistic (and maybe more complex) multiple annotation scenario where tasks are more similar and more than two annotation tasks might be involved. Furthermore, several forms of *interdependencies* may arise between the single annotation tasks. As a first example, consider the (functional) interdependencies (i.e., task similarity) in higher-level semantic NLP tasks of relation or event recognition. In such a scenario, several tasks including entity annotations and relation/event annotations, as well as syntactic parse data, have to be incorporated at the same time. Another type of (data flow) inter-

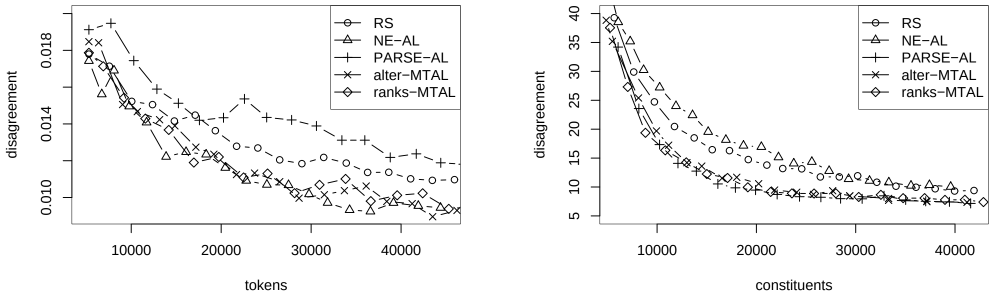

Figure 4: Disagreement curves for NE task (left) and parse task (right) on `WSJ`

dependency occurs in a second scenario where material for several classifiers that are data-dependent on each other – one takes the output of another classifier as input features – has to be efficiently annotated. Whether the proposed protocols are beneficial in the context of such highly interdependent tasks is an open issue. Even more challenging is the idea to provide methodologies helping to predict in an arbitrary application scenario whether the choice of MTAL is truly advantageous.

Another open question is how to measure and quantify the overall *annotation costs* in multiple annotation scenarios. Exchange rates are inherently tied to the specific task and domain. In practice, one might just want to measure the time needed for the annotations. However, in a simulation scenario, a common metric is necessary to compare the performance of different selection strategies with respect to the overall annotation costs. This requires studies on how to quantify, with a comparable cost function, the efforts needed for the annotation of a textual unit of choice (e.g., tokens, sentences) with respect to different annotation tasks.

Finally, the question of *reusability* of the annotated material is an important issue. Reusability in the context of AL means to which degree corpora assembled with the help of any AL technique can be (re)used as a general resource, i.e., whether they are well suited for the training of classifiers other than the ones used during the selection process.This is especially interesting as the details of the classifiers that should be trained in a later stage are typically not known at the resource building time. Thus, we want to select samples valuable to a *family* of classifiers using the various annotation layers. This, of course, is only possible if data annotated with the help of AL is reusable by modified though similar classifiers (e.g., with respect to the features being used) – compared to the classifiers employed for the selection procedure.

The issue of reusability has already been raised but not yet conclusively answered in the context of single-task AL (see Section 6). Evidence was found that reusability up to a certain, though not well-specified, level is possible. Of course, reusability has to be analyzed separately in the context of various MTAL scenarios. We feel that these scenarios might both be more challenging and more relevant to the reusability issue than the single-task AL scenario, since resources annotated with multiple layers can be used to the design of a larger number of a (possibly more complex) learning algorithms.

8 Conclusions

We proposed an extension to the single-task AL approach such that it can be used to select examples for annotation with respect to several annotation tasks. To the best of our knowledge this is the first paper on this issue, with a focus on NLP tasks. We outlined a problem definition and described a framework for multi-task AL. We presented and tested two protocols for multi-task AL. Our results are promising as they give evidence that in a multiple annotation scenario, multi-task AL outperforms naive one-sided and random selection.

Acknowledgments

The work of the second author was funded by the German Ministry of Education and Research within the STEMNET project (01DS001A-C), while the work of the third author was funded by the EC within the BOOTSTREP project (FP6-028099).

References

Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annotation. In *Proceedings of EMNLP'04*, pages 9–16.

Markus Becker and Miles Osborne. 2005. A two-stage method for active learning of statistical grammars. In *Proceedings of IJCAI'05*, pages 991–996.

Daniel M. Bickel. 2005. Code developed at the University of Pennsylvania, `http://www.cis.upenn.edu/~dbikel/software.html`.

Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145.

Michael Collins. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

Sean Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. In *Proceedings of ACL'96*, pages 319–326.

Yoav Freund, Sebastian Seung, Eli Shamir, and Naftali Tishby. 1997. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168.

Ben Hachey, Beatrice Alex, and Markus Becker. 2005. Investigating the effects of selective sampling on the annotation task. In *Proceedings of CoNLL'05*, pages 144–151.

Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. 1994. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75.

Rebecca Hwa. 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML'01*, pages 282–289.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Eleni Miltsakaki, Livio Robaldo, Alan Lee, and Aravind K. Joshi. 2008. Sense annotation in the penn discourse treebank. In *Proceedings of CICLing'08*, pages 275–286.

Grace Ngai and David Yarowsky. 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *Proceedings of ACL'00*, pages 117–125.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002. The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of HLT'02*, pages 82–86.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Roi Reichart and Ari Rappoport. 2007. An ensemble method for selection of high quality parses. In *Proceedings of ACL'07*, pages 408–415, June.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of JNLPBA'04*, pages 107–110.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew Lim Tan. 2004. Multi-criteria-based active learning for named entity recognition. In *Proceedings of ACL'04*, pages 589–596.

Min Tang, Xiaoqiang Luo, and Salim Roukos. 2001. Active learning for statistical natural language parsing. In *Proceedings of ACL'02*, pages 120–127.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CONLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL'03*, pages 142–147.

Katrin Tomanek and Udo Hahn. 2008. Approximating learning curves for active-learning-driven annotation. In *Proceedings of LREC'08*.

Katrin Tomanek, Joachim Wermter, and Udo Hahn. 2007. An approach to text corpus construction which cuts annotation costs and maintains corpus reusability of annotated data. In *Proceedings of EMNLP-CoNLL'07*, pages 486–495.

Jingbo Zhu and Eduard Hovy. 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of EMNLP-CoNLL'07*, pages 783–790.