SICS Technical Report T2009:06
ISSN: 1100-3154

A literature survey of active machine learning in the context of natural language processing

Fredrik Olsson
April 17, 2009
fredrik.olsson@sics.se
Swedish Institute of Computer Science
Box 1263, SE-164 29 Kista, Sweden

Abstract. Active learning is a supervised machine learning technique in which the learner is in control of the data used for learning. That control is utilized by the learner to ask an oracle, typically a human with extensive knowledge of the domain at hand, about the classes of the instances for which the model learned so far makes unreliable predictions. The active learning process takes as input a set of labeled examples, as well as a larger set of unlabeled examples, and produces a classifier and a relatively small set of newly labeled data. The overall goal is to create as good a classifier as possible, without having to mark-up and supply the learner with more data than necessary. The learning process aims at keeping the human annotation effort to a minimum, only asking for advice where the training utility of the result of such a query is high.

Active learning has been successfully applied to a number of natural language processing tasks, such as, information extraction, named entity recognition, text categorization, part-of-speech tagging, parsing, and word sense disambiguation. This report is a literature survey of active learning from the perspective of natural language processing.

Keywords. Active learning, machine learning, natural language processing, literature survey

Contents

Chapter 1

Introduction

This report is a survey of the literature relevant to active machine learning in the context of natural language processing. The intention is for it to act as an overview and introductory source of information on the subject.

The survey is partly called for by the results of an on-line questionnaire concerning the nature of annotation projects targeting information access in general, and the use of active learning as annotation support in particular (Tomanek and Olsson 2009). The questionnaire was announced to a number of emailing lists, including Corpora, BioNLP, UAI List, ML-news, SIGIRlist, and Linguist list, in February of 2009. One of the main findings was that active learning is not widely used; only 20% of the participants responded positively to the question "Have you ever used active learning in order to speed up annotation/labeling work of any linguistic data?". Thus, one of the reasons to compile this survey is simply to help spread the word about the fundamentals of active learning to the practitioners in the field of natural language processing.

Since active learning is a vivid research area and thus constitutes a moving target, I strive to revise and update the web version of the survey periodically.[1] Please direct suggestions for improvements, papers to include, and general comments to fredrik.olsson@sics.se.

In the following, the reader is assumed to have general knowledge of machine learning such as provided by, for instance, Mitchell (1997), and Witten and Frank (2005). I would also like to point the curious reader to the survey of the literature of active learning by Settles (Settles 2009).

[1] The web version is available at <http://www.sics.se/people/fredriko>.

Chapter 2

Approaches to Active Learning

Active machine learning is a supervised learning method in which the learner is in control of the data from which it learns. That control is used by the learner to ask an oracle, a teacher, typically a human with extensive knowledge of the domain at hand, about the classes of the instances for which the model learned so far makes unreliable predictions. The active learning process takes as input a set of labeled examples, as well as a larger set of unlabeled examples, and produces a classifier and a relatively small set of newly labeled data. The overall goal is to produce as good a classifier as possible, without having to mark-up and supply the learner with more data than necessary. The learning process aims at keeping the human annotation effort to a minimum, only asking for advice where the training utility of the result of such a query is high.

On those occasions where it is necessary to distinguish between "ordinary" machine learning and active learning, the former is sometimes referred to as *passive learning* or learning by *random sampling* from the available set of labeled training data.

A prototypical active learning algorithm is outlined in Figure 2.1. Active learning has been successfully applied to a number of language technology tasks, such as

- information extraction (Scheffer, Decomain and Wrobel 2001; Finn and Kushmerick 2003; Jones et al. 2003; Culotta et al. 2006);
- named entity recognition (Shen et al. 2004; Hachey, Alex and Becker 2005; Becker et al. 2005; Vlachos 2006; Kim et al. 2006);
- text categorization (Lewis and Gale 1994; Lewis 1995; Liere and Tadepalli 1997; McCallum and Nigam 1998; Nigam and Ghani 2000; Schohn and Cohn 2000; Tong and Koller 2002; Hoi, Jin and Lyu 2006);

- part-of-speech tagging (Dagan and Engelson 1995; Argamon-Engelson and Dagan 1999; Ringger et al. 2007);
- parsing (Thompson, Califf and Mooney 1999; Hwa 2000; Tang, Luo and Roukos 2002; Steedman et al. 2003; Hwa et al. 2003; Osborne and Baldridge 2004; Becker and Osborne 2005; Reichart and Rappoport 2007);
- word sense disambiguation (Chen et al. 2006; Chan and Ng 2007; Zhu and Hovy 2007; Zhu, Wang and Hovy 2008a);
- spoken language understanding (Tur, Hakkani-Tür and Schapire 2005; Wu et al. 2006);
- phone sequence recognition (Douglas 2003);
- automatic transliteration (Kuo, Li and Yang 2006); and
- sequence segmentation (Sassano 2002).

One of the first attempts to make expert knowledge an integral part of learning is that of query construction (Angluin 1988). Angluin introduces a range of queries that the learner is allowed to ask the teacher, such as queries regarding *membership* ("Is this concept an example of the target concept?"), *equivalence* ("Is X equivalent to Y?"), and *disjointness* ("Are X and Y disjoint?"). Besides a simple *yes* or *no*, the full answer from the teacher can contain counterexamples, except in the case of membership queries. The learner constructs queries by altering the attribute values of instances in such a way that the answer to the query is as informative as possible. Adopting this generative approach to active learning leads to problems in domains where changing the values of attributes are not guaranteed to make sense to the human expert; consider the example of text categorization using a bag-of-word approach. If the learner first replaces some of the words in the representation, and then asks the teacher whether the new artificially created document is a member of a certain class, it is not likely that the new document makes sense to the teacher.

In contrast to the theoretically interesting generative approach to active learning, current practices are based on example-driven means to incorporate the teacher into the learning process; the instances that the learner asks (queries) the teacher to classify all stem from existing, unlabeled data. The *selective sampling* method introduced by Cohn, Atlas and Ladner (1994) builds on the concept of membership queries, albeit from an example-driven perspective; the learner queries the teacher about the data at hand for which it is uncertain, that is, for which it believes misclassifications are possible.

1. Initialize the process by applying base learner B to labeled training data set D_L to obtain classifier C.
2. Apply C to unlabeled data set D_U to obtain ${D_U}'$.
3. From ${D_U}'$, select the most informative n instances to learn from, I.
4. Ask the teacher for classifications of the instances in I.
5. Move I, with supplied classifications, from ${D_U}'$ to D_L.
6. Re-train using B on D_L to obtain a new classifier, C'.
7. Repeat steps 2 through 6, until D_U is empty or until some stopping criterion is met.
8. Output a classifier that is trained on D_L.

Figure 2.1: A prototypical active learning algorithm.

2.1 Query by uncertainty

Building on the ideas introduced by Cohn and colleagues concerning selective sampling (Cohn, Atlas and Ladner 1994), in particular the way the learner selects what instances to ask the teacher about, *query by uncertainty* (*uncertainty sampling, uncertainty reduction*) queries the learning instances for which the current hypothesis is least confident. In query by uncertainty, a single classifier is learned from labeled data and subsequently utilized for examining the unlabeled data. Those instances in the unlabeled data set that the classifier is least certain about are subject to classification by a human annotator. The use of confidence scores pertains to the third step in Figure 2.1. This straightforward method requires the base learner to provide a score indicating how confident it is in each prediction it performs.

Query by uncertainty has been realized using a range of base learners, such as logistic regression (Lewis and Gale 1994), Support Vector Machines (Schohn and Cohn 2000), and Markov Models (Scheffer, Decomain and Wrobel 2001). They all report results indicating that the amount of data that require annotation in order to reach a given performance, compared to passively learning from examples provided in a random order, is heavily reduced using query by uncertainty.

Becker and Osborne (2005) report on a two-stage model for actively learning statistical grammars. They use uncertainty sampling for selecting the sentences for which the parser provides the lowest confidence scores. The problem with this approach, they claim, is that the confidence score says nothing about the state of the statistical model itself; if the estimate of the parser's confidence in a certain parse tree is based on rarely occurring

1. Initialize the process by applying *EnsembleGenerationMethod* using base learner B on labeled training data set D_L to obtain a committee of classifiers C.
2. Have each classifier in C predict a label for every instance in the unlabeled data set D_U, obtaining labeled set $D_U{}'$.
3. From $D_U{}'$, select the most informative n instances to learn from, obtaining $D_U{}''$.
4. Ask the teacher for classifications of the instances I in $D_U{}''$.
5. Move I, with supplied classifications, from $D_U{}''$ to D_L.
6. Re-train using *EnsembleGenerationMethod* and base learner B on D_L to obtain a new committee, C.
7. Repeat steps 2 through 6 until D_U is empty or some stopping criterion is met.
8. Output a classifier learned using *EnsembleGenerationMethod* and base learner B on D_L.

Figure 2.2: A prototypical query by committee algorithm.

information in the underlying data, the confidence in the confidence score is low, and should thus be avoided. The first stage in Becker and Osborne's two-stage method aims at identifying and singling out those instances (sentences) for which the parser cannot provide reliable confidence measures. In the second stage, query by uncertainty is applied to the remaining set of instances. Becker and Osborne (2005) report that their method performs better than the original form of uncertainty sampling, and that it exhibits results competitive with a standard query by committee method.

2.2 Query by committee

Query by committee, like query by uncertainty, is a selective sampling method, the fundamental difference between the two being that query by committee is a multi-classifier approach. In the original conception of query by committee, several hypotheses are randomly sampled from the version space (Seung, Opper and Sompolinsky 1992). The committee thus obtained is used to examine the set of unlabeled data, and the disagreement between the hypotheses with respect to the class of a given instance is utilized to decide whether that instance is to be classified by the human annotator. The idea with using a decision committee relies on the assumption that in order for approaches combining several classifiers to work, the ensemble needs

to be made up from diverse classifiers. If all classifiers are identical, there will be no disagreement between them as to how a given instance should be classified, and the whole idea of voting (or averaging) is invalidated. Query by committee, in the original sense, is possible only with base learners for which it is feasible to access and sample from the version space; learners reported to work in such a setting include Winnow (Liere and Tadepalli 1997), and perceptrons (Freund et al. 1997). A prototypical query by committee algorithm is shown in Figure 2.2.

2.2.1 Query by bagging and boosting

Abe and Mamitsuka (1998) introduce an alternative way of generating multiple hypotheses; they build on *bagging* and *boosting* to generate committees of classifiers from the same underlying data set.

Bagging, short for bootstrap aggregating (Breiman 1996), is a technique exploiting the bias-variance decomposition of classification errors (see, for instance, Domingos 2000 for an overview of the decomposition problem). Bagging aims at minimizing the variance part of the error by randomly sampling – with replacement – from the data set, thus creating several data sets from the original one. The same base learner is then applied to each data set in order to create a committee of classifiers. In the case of classification, an instance is assigned the label that the majority of the classifiers predicted (majority vote). In the case of regression, the value assigned to an instance is the average of the predictions made by the classifiers.

Like bagging, boosting (Freund and Schapire 1997) is a way of combining classifiers obtained from the same base learner. Instead of building classifiers independently, boosting allows for classifiers to influence each other during training. Boosting is based on the assumption that several classifiers learned using a weak[1] base learner, over a varying distribution of the target classes in the training data, can be combined into one strong classifier. The basic idea is to let classifiers concentrate on the cases in which previously built classifiers failed to correctly classify data. Furthermore, in classifying data, boosting assigns weights to the classifiers according to their performance; the better the performance, the higher valued is the classifier's contribution in voting (or averaging). Schapire (2003) provides an overview of boosting.

Abe and Mamitsuka (1998) claim that query by committee, query by bagging, and query by boosting form a natural progression; in query by committee, the variance in performance among the hypotheses is due to the randomness exhibited by the base learner. In query by bagging, the variance is a result of the randomization introduced when sampling from the data set. Finally, the variance in query by boosting is a result of altering the sampling

[1] A learner is *weak* if it produces a classifier that is only slightly better than random guessing, while a learner is said to be *strong* if it produces a classifier that achieves a low error with high confidence for a given concept (Schapire 1990).

according to the weighting of the votes given by the hypotheses involved. A generalized variant of query by bagging is obtained if the *EnsembleGenerationMethod* in Figure 2.2 is substituted with bagging. Essentially, query by bagging applies bagging in order to generate a set of hypotheses that is then used to decide whether it is worth querying the teacher for classification of a given unlabeled instance. Query by boosting proceeds similarly to query by bagging, with boosting applied to the labeled data set in order to generate a committee of classifiers instead of bagging, that is, boosting is used as *EnsembleGenerationMethod* in Figure 2.2.

Abe and Mamitsuka (1998) report results from experiments using the decision tree learner C4.5 as base learner and eight data sets from the UCI Machine Learning Repository, the latest release of which is described in (Asuncion and Newman 2007). They find that query by bagging and query by boosting significantly outperformed a single C4.5 decision tree, as well as boosting using C4.5.

2.2.2 ActiveDecorate

Melville and Mooney (2004) introduce ActiveDecorate, an extension to the Decorate method (Melville and Mooney 2003) for constructing diverse committees by enhancing available data with artificially generated training examples. Decorate – short for *Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples* – is an iterative method generating one classifier at a time. In each iteration, artificial training data is generated in such a way that the labels of the data are maximally different from the predictions made by the current committee of classifiers. A strong base learner is then used to train a classifier on the union of the artificial data set and the available labeled set. If the resulting classifier increases the prediction error on the training set, it is rejected as a member of the committee, and added otherwise. In ActiveDecorate, the Decorate method is utilized for generating the committee of classifiers, which is then used to decide which instances from the unlabeled data set are up for annotation by the human oracle. In terms of the prototypical query by committee algorithm in Figure 2.2, ActiveDecorate is used as *EnsembleGenerationMethod*.

Melville and Mooney (2004) carry out experiments on 15 data sets from the UCI repository (Asuncion and Newman 2007). They show that their algorithm outperforms query by bagging and query by boosting as introduced by Abe and Mamitsuka (1998) both in terms of accuracy reached, and in terms of the amount of data needed to reach top accuracy. Melville and Mooney conclude that the superiority of ActiveDecorate is due to the diversity of the generated ensembles.

2.3 Active learning with redundant views

Roughly speaking, utilizing redundant views is similar to the query by committee approach described above. The essential difference is that instead of randomly sampling the version space, or otherwise tamper with the existing training data with the purpose of extending it to obtain a committee, using redundant views involves splitting the feature set into several sub-sets or *views*, each of which is enough, to some extent, to describe the underlying problem.

Blum and Mitchell (1998) introduce a semi-supervised bootstrapping technique called *Co-training* in which two classifiers are trained on the same data, but utilizing different views of it. The example of views provided by Blum and Mitchell (1998) is from the task of categorizing texts on the web. One way of learning how to do that is by looking at the links to the target document from other documents on the web, another way is to consider the contents of the target document alone. These two ways correspond to two separate views of learning the same target concept.

As in active learning, Co-training starts off with a small set of labeled data, and a large set of unlabeled data. The classifiers are first trained on the labeled part, and subsequently used to tag an unlabeled set. The idea is then that during the learning process, the predictions made by the first classifier on the unlabeled data set, and for which it has the highest confidence, are added to the training set of the second classifier, and vice-versa. The classifiers are then retrained on the newly extended training set, and the bootstrapping process continues with the remainder of the unlabeled data.

A drawback with the Co-training method as it is originally described by Blum and Mitchell (1998) is that it requires the views of data to be conditionally independent and compatible given the class, that is, each view should be enough for producing a strong learner compatible with the target concept. In practice, however, finding such a split of features may be hard; the problem is further discussed in Section 2.3.1.

Co-training *per se* is not within the active learning paradigm since it does not involve a teacher, but the work by Blum and Mitchell (1998) forms the basis for other approaches. One such approach is that of *Corrected Co-training* (Pierce and Cardie 2001). Corrected Co-training is a way of remedying the degradation in performance that can occur when applying Co-training to large data sets. The concerns of Pierce and Cardie (2001) include that of scalability of the original Co-training method. Pierce and Cardie investigate the task of noun phrase chunking, and they find that when hundreds of thousands of examples instead of hundreds, are needed to learn a target concept, the successive degradation of the quality of the bootstrapped data set becomes an issue. When increasing the amount of unlabeled data, and thus also increasing the number of iterations during which Co-training

1. Initialize the process by applying base learner B using each v in views V to labeled training set D_L to obtain a committee of classifiers C.
2. Have each classifier in C predict a label for every instance in the unlabeled data set D_U, obtaining labeled set ${D_U}'$.
3. From ${D_U}'$, select those instances for which the classifiers in C predicted different labels to obtain the *contention set*[2] ${D_U}''$.
4. Select instances I from ${D_U}''$ and ask the teacher for their labels.
5. Move instances I, with supplied classifications, from ${D_U}''$ to D_L.
6. Re-train by applying base learner B using each v in views V to D_L to obtain committe C'.
7. Repeat steps 2 through 6 until D_U is empty or some stopping criterion is met.
8. Output the final classifier learned by combining base learner B, views in V, and data D_L.

Figure 2.3: A prototypical multiple view active learning algorithm.

will be in effect, the risk of errors introduced by the classifiers into each view increases. In Corrected Co-training a human annotator reviews and edits, as found appropriate, the data produced by both view classifiers in each iteration, prior to adding the data to the pool of labeled training data. This way, Pierce and Cardie point out, the quality of the labeled data is maintained with only a moderate effort needed on behalf of the human annotator. Figure 2.3 shows a prototypical algorithm for multi-view active learning. It is easy to see how Corrected Co-training fits into it; if, instead of having the classifiers select the instances on which they disagree (step 3 in Figure 2.3), each classifier selects the instances for which it makes highly confident predictions, and have the teacher correct them in step 4, the algorithm in Figure 2.3 would describe Corrected Co-training.

Hwa et al. (2003) adopt a Corrected Co-training approach to statistical parsing. In pursuing their goal – to further decrease the amount of corrections of parse trees a human annotator has to perform – they introduce *single-sided corrected Co-training*. Single-sided Corrected Co-training is like Corrected Co-training, with the difference that the annotator only reviews the data, parse trees, produced by one of the view classifiers. Hwa et al. (2003) conclude that in terms of parsing performance, parsers trained using some form of sample selection technique are better off than parsers trained

[2]The instance or set of instances for which the view classifiers disagree is called the *contention point*, and *contention set*, respectively.

in a pure Co-training setting, given the cost of human annotation. Furthermore, Hwa and colleagues point out that even though parsing performance achieved using single-sided Corrected Co-training is not as good as that resulting from Corrected Co-training, some corrections are better than none.

In their work, Pierce and Cardie (2001) note that corrected Co-training does not help their noun phrase chunker to reach the expected performance. Their hypothesis as to why the performance gap occurs, is that Co-training does not lend itself to finding the most informative examples available in the unlabeled data set. Since each classifier selects the examples it is most confident in, the examples are likely to represent aspects of the task at hand already familiar to the classifiers, rather than representing potentially new and more informative ones. Thus, where Co-training promotes confidence in the selected examples over finding examples that would help incorporating new information about the task, active learning works the other way around. A method closely related to Co-training, but which is more exploratory by nature, is *Co-testing* (Muslea, Minton and Knoblock 2000, 2006). Co-testing is an iterative process that works under the same premises as active learning in general, that is, it has access to a small set of labeled data, as well as a large set of unlabeled data. Co-testing proceeds by first learning a hypothesis using each view of the data, then asking a human annotator to label the unlabeled instances for which the view classifiers' predictions disagree on labels. Such instances are called the *contention set* or *contention point*. The newly annotated instances are then added to the set of labeled training data.

Muslea, Minton and Knoblock (2006) introduce a number of variants of Co-testing. The variations are due to choices of how to select the instances to query the human annotator about, as well as how the final hypothesis is to be created. The former choice pertains to step 4 in Figure 2.3, and the options are:

Naïve – Randomly choose an example from the contention set. This strategy is suitable when using a base learner that does not provide confidence estimates for the predictions it makes.

Aggressive – Choose to query the example in the contention set for which the least confident classifier makes the most confident prediction. This strategy is suitable for situations where there is (almost) no noise.

Conservative – Choose to query the example in the contention set for which the classifiers makes predictions that are as close as possible. This strategy is suitable for noisy domains.

Muslea, Minton and Knoblock (2006) also present three ways of forming the final hypothesis in Co-testing, that is, the classifier to output at the end of the process. These ways concern step 8 in Figure 2.3:

Weighted vote – Combine the votes of all view classifiers, weighted according to each classifier's confidence estimate of its own prediction.

Majority vote – Combine the votes of all view classifiers so that the label predicted by the majority of the classifiers is used.

Winner-takes-all – The final classifier is the one learned in the view that made the least amount of mistakes throughout the learning process.

Previously described multi-view approaches to learning all relied on the views being *strong*. Analogously to the notion of a strong learner in ensemble-based methods, a strong view is a view which provides enough information about the data for a learner to learn a given target concept. Conversely, there are *weak* views, that is, views that are not by themselves enough to learn a given target concept, but rather a concept more general or more specific than the concept of interest. In the light of weak views, Muslea, Minton and Knoblock (2006) redefine the notion of contention point, or contention set, to be the set of examples, from the unlabeled data, for which the strong view classifiers disagree. Muslea and colleagues introduce two ways of making use of weak views in Co-testing. The first is as tie-breakers when two strong views predict a different label for an unlabeled instance, and the second is by using a weak view in conjunction with two strong views in such a way that the weak view would indicate a mistake made by both strong views. The latter is done by detecting the set of contention points for which the weak view disagrees with both strong views. Then the next example to ask the human annotator to label, is the one for which the weak view makes the most confident prediction. This example is likely to represent a mistake made by both strong views, Muslea, Minton and Knoblock (2006) claim, and leads to faster convergence of the classifiers learned.

The experimental set-up in used by Muslea, Minton and Knoblock (2006) is targeted at testing whether Co-testing converges faster than the corresponding single-view active learning methods when applied to problems in which there exist several views. The tasks are of two types: classification, including text classification, advertisement removal, and discourse tree parsing; and wrapper induction. For all tasks in their empirical validation, Muslea, Minton and Knoblock (2006) show that the Co-testing variants employed outperform the single-view, state-of-the art approaches to active learning that were also part of the investigation.

The advantages of using Co-testing include its ability to use any base learner suitable for the particular problem at hand. This seems to be a rather unique feature among the active learning methods reviewed in this chapter. Nevertheless, there are a couple of concerns regarding the shortcomings of Co-testing aired by Muslea and colleagues that need to be mentioned. Both concerns relate to the use of multiple views. The first is that Co-testing can obviously only be applied to tasks where there exist two views. The

other of their concerns is that the views of data have to be uncorrelated (independent) and compatible, that is, the same assumption brought up by Blum and Mitchell (1998) in their original work on Co-training. If the views are correlated, the classifier learned in each view may turn out so similar that no contention set is generated when both view classifiers are run on the unlabeled data. In this case, there is no way of selecting an example for which to query the human annotator. If the views are incompatible, the view classifiers will learn two different tasks and the process will not converge.

Just as with committee-based methods, utilizing multiple views seems like a viable way to make the most of a situation that is caused by having access to a small amount of labeled data. Though, the question remains of how one should proceed in order to define multiple views in a way so that the they are uncorrelated and compatible with the target concept.

2.3.1 How to split a feature set

Acquiring a feature set split adhering to the assumptions underlying the multi-view learning paradigm is a non-trivial task requiring knowledge about the learning situation, the data, and the domain. Two approaches to the view detection and validation problem form the extreme ends of a scale; randomly splitting a given feature set and hope for the best at one end, and adopting a very cautions view on the matter by computing the correlation and compatibility for every combination of the features in a given set at the other end.

Nigam and Ghani (2000) report on randomly splitting the feature set for tasks where there exists no natural division of the features into separate views. The task is text categorization, using Naïve Bayes as base learner. Nigam and Ghani argue that, if the features are sufficiently redundant, and one can identify a reasonable division of the feature set, the application of Co-training using such a non-natural feature set split should exhibit the same advantages as applying Co-training to a task in which there exists natural views.

Concerning the ability to learn a desired target concept in each view, Collins and Singer (1999) introduce a Co-training algorithm that utilizes a boosting-like step to optimize the compatibility between the views. The algorithm, called CoBoost, favors hypotheses that predict the same label for most of the unlabeled examples.

Muslea, Minton and Knoblock (2002a) suggest a method for validating the compatibility of views, that is, given two views, the method should provide an answer to whether each view is enough to learn the target concept. The way Muslea and colleagues go about is by collecting information about a number of tasks solved using the same views as the ones under investigation. Given this information, a classifier for discriminating between the

tasks in which the views were compatible, and the tasks in which they were not, is trained and applied. The obvious drawback of this approach is that the first time the question of whether a set of views is compatible with a desired concept, the method by Muslea, Minton and Knoblock (2002a) is not applicable. In all fairness, it should be noted that the authors clearly state the proposed view validation method to be but one step towards automatic view detection.

Muslea, Minton and Knoblock (2002b) investigate view dependence and compatibility for several semi-supervised algorithms along with one algorithm combining semi-supervised and active learning (Co-testing), CoEMT. The conclusions made by Muslea and colleagues are interesting, albeit perhaps not surprising. For instance, the performance of all multi-view algorithms under investigation degrades as the views used become less compatible, that is, when the target concept learned by view classifiers are not the same in each view. A second, very important point made in (Muslea, Minton and Knoblock 2002a) is that the robustness of the active learning algorithm with respect to view correlation is suggested to be due to the usage of an active learning component; being able to ask a teacher for advice seems to compensate for the views not being entirely uncorrelated.

Balcan, Blum and Yang (2005) argue that, for the kind of Co-training presented by Blum and Mitchell (1998), the original assumption of conditional independence between views is overly strong. Balcan and colleagues claim that the views do not have to denote conditionally independent ways of representing the task to be useful to Co-training, if the base learner is able to correctly learn the target concept using positive training examples only.

Zhang et al. (2005) present an algorithm called *Correlation and Compatibility based Feature Partitioner, CCFP* for computing, from a given set of features, independent and compatible views. CCFP makes use of feature pair-wise symmetric uncertainty and feature-wise information gain to detect the views. Zhang and colleagues point out that in order to employ CCFP, a fairly large number of labeled examples are needed. Exactly how large a number is required is undisclosed. CCFP is empirically tested and Zhang et al. (2005) report on somewhat satisfactory results.

Finally, one way of circumventing the assumptions of view independence and compatibility is simply not to employ different views at all. Goldman and Zhou (2000) propose a variant of Co-training which assumes no redundant views of the data; instead, a single view is used by differently biased base learners. Chawla and Karakoulas (2005) make empirical studies on this version of Co-training. Since the methods of interest to the present thesis are those containing elements of active learning, which the original Co-training approach does not, the single-view multiple-learner approach to Co-training will not be further elaborated.

In the literature, there is to my knowledge no report on automatic means

to discover, from a given set of features, views that satisfy the original Co-training assumptions concerning independence and compatibility. Although the Co-training method as such is not of primary interest to this thesis, offsprings of the method are. The main approach to active multi-view learning, Co-testing and its variants rely on the same assumptions as does Co-training. Muslea, Minton and Knoblock (2002b) show that violating the compatibility assumption in the context of an active learning component, does not necessarily lead to failure; the active learner might have a stabilizing effect on the divergence of the target concept learned in each view. As regards the conditional independence assumption made by Blum and Mitchell (1998), subsequent work (Balcan, Blum and Yang 2005) shows that the independence assumption is too strong, and that iterative Co-training, and thus also Co-testing, works under a less rigid assumption concerning the expansion of the data in the learning process.

Chapter 3

Quantifying disagreement

So far, the issue of disagreement has been mentioned but deliberately not elaborated on. The algorithms for query by committee and its variants (Figure 2.2) as well as those utilizing multiple views of data (Figure 2.3) all contain steps in which the disagreement between classifiers concerning instances has to be quantified. In a two-class case, such quantification is simply the difference between the positive and negative votes given by the classifiers. Typically, instances for which the distribution of votes is homogeneous is selected for querying. Generalizing disagreement to a multi-class case is not trivial. Körner and Wrobel (2006) empirically test four approaches to measuring disagreement between members of a committee of classifiers in a multi-class setting. The active learning approaches they consider are query by bagging, query by boosting, ActiveDecorate, and Co-testing. The disagreement measures investigated are *margin-based disagreement*, *uncertainty sampling-based disagreement*, *entropy-based disagreement*, and finally a measure of their own dubbed *specific disagreement*. Körner and Wrobel (2006) strongly advocate the use of margin-based disagreement as a standard approach to quantifying disagreement in an ensemble-based setting.

Sections 3.1 through 3.4 deal with the different measures used by Körner and Wrobel (2006), followed by the treatment of *Kullback-Leibler divergence, Jensen-Shannon divergence, vote entropy*, and *F-complement* in Sections 3.5 to 3.8.

3.1 Margin-based disagreement

Margin, as introduced by Abe and Mamitsuka (1998) for binary classification in query by boosting, is defined as the difference between the number of votes given to the two labels. Abe and Mamitsuka base their notion of margins on the finding that a classifier exhibiting a large margin when trained on labeled data, performs better on unseen data than does a classifier that has a smaller margin on the training data (Schapire et al. 1998). Melville

and Mooney (2004) extend Abe and Mamitsuka's definition of margin to include class probabilities given by the individual committee members. Körner and Wrobel (2006), in turn, generalize Melville and Mooney's definition of margin to account for the multi-class setting as well. The margin-based disagreement for a given instance is the difference between the first and second highest probabilities with which an ensemble of classifiers assigns different class labels to the instance.

For example, if an instance X is classified by committee member 1 as belonging to class A with a probability of 0.7, by member 2 as belonging class B with a probability of 0.2, and by member 3 to class C with 0.3, then the margin for X is $A - C = 0.4$. If instance Y is classified by member 1 as class A with a probability of 0.8, by member 2 as class B with a probability of 0.9, and by member 3 as class C with 0.6, then the margin for Y is $B - A = 0.1$. A low value on the margin indicates that the ensemble disagree regarding the classification of the instance, while a high value signals agreement. Thus, in the above example, instance Y is more informative than instance X.

3.2 Uncertainty sampling-based disagreement

Originally, uncertainty sampling is a method used in conjunction with single classifiers, rather than ensembles of classifiers (see Section 2.1). Körner and Wrobel (2006), though, prefer to view it as another way of generalizing the binary margin approach introduced in the previous section. In uncertainty sampling, instances are preferred that receives the lowest *class probability* estimate by the ensemble of classifiers. The class probability is the highest probability with which an instance is assigned a class label.

3.3 Entropy-based disagreement

The entropy-based disagreement used in (Körner and Wrobel 2006) is what they refer to as the ordinary entropy measure (*information entropy* or *Shannon entropy*) first introduced by Shannon (1948). The entropy H of a random variable X is defined in equation 3.1 in the case of a c class problem, that is, where X can take on values $x_1, \ldots, x_c$.

$$H(X) = -\sum_{i=1}^{c} p(x_i) \log_2 p(x_i) \tag{3.1}$$

where $p(x_i)$ denotes the probability of x_i. A lower value on $H(X)$ indicates less confusion or less uncertainty concerning the outcome of the value of X.

3.4 The Körner-Wrobel disagreement measure

The *specific disagreement* measure, here referred to as the *Körner-Wrobel disagreement measure* is a combination of margin-based disagreement M and the maximal class probability P over classes C in order to indicate disagreement on a narrow subset of class values. The Körner-Wrobel disagreement measure, R, is defined in equation 3.2.

$$R = M + 0.5\frac{1}{(|C|P)^3} \tag{3.2}$$

Körner and Wrobel (2006) find that the success of the specific disagreement measure is closely related to which active learning method is used. Throughout the experiments conducted by Körner and Wrobel, those configurations utilizing specific disagreement as selection metric perform less well than the margin-based and entropy-based disagreement measures investigated.

3.5 Kullback-Leibler divergence

The Kullback-Leibler divergence (*KL-divergence, information divergence*) is a non-negative measure of the divergence between two probability distributions p and q in the same event space $X = \{x_1, \ldots, x_c\}$. The KL-divergence, denoted $D(\cdot \parallel \cdot)$, between two probability distributions p and q is defined in equation 3.3.

$$D(p \parallel q) = \sum_{i=1}^{c} p(x_i)\log\frac{p(x_i)}{q(x_i)} \tag{3.3}$$

A high value on the KL-divergence indicates a large difference between the distributions p and q. A zero-valued KL-divergence signals full agreement, that is p and q are equivalent.

Kullback-Leibler divergence to the mean (Pereira, Tishby and Lee 1993) quantifies the disagreement between committee members; it is the average KL-divergence between each distribution and the mean of all distributions. KL-divergence to the mean, D_{mean} for an instance x is defined in equation 3.4.

$$D_{mean}(x) = \frac{1}{k}\sum_{i=1}^{k} D(p_i(x) \parallel p_{mean}(x)) \tag{3.4}$$

where k is the number of classifiers involved, $p_i(x)$ is the probability distribution for x given by the i-th classifier, $p_{mean}(x)$ is the mean probability distribution of all k classifiers for x, and $D(\cdot \parallel \cdot)$ is the KL-divergence as defined in equation 3.3.

KL-divergence, as well as KL-divergence to the mean, has been used for detecting and measuring disagreement in active learning, see for instance (McCallum and Nigam 1998; Becker et al. 2005; Becker and Osborne 2005)

3.6 Jensen-Shannon divergence

The *Jensen-Shannon divergence*, (*JSD*) is a symmetrized and smoothed version of KL-divergence, which essentially means that it can be used to measure the distance between two probability distributions (Lin 1991). The Jensen-Shannon divergence for two distributions p and q is defined in equation 3.5.

$$JSD(p,q) = H(w_1 p + w_2 q) - w_1 H(p) - w_2 H(q) \tag{3.5}$$

where w_1 and w_2 are the weights of the probability distributions such that $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$, and H is the Shannon entropy as defined in equation 3.1.

Lin (1991) defines the Jensen-Shannon divergence for k distributions as in equation 3.6.

$$JSD(p_1, \ldots, p_k) = H(\sum_{i=1}^{k} w_i p_i) - \sum_{i=1}^{k} w_i H(p_i) \tag{3.6}$$

where p_i is the class probability distribution given by the i-th classifier for a given instance, w_i is the vote weight of the i-th classifier among the k classifiers in the set, and $H(p)$ is the entropy as defined in equation 3.1. A Jensen-Shannon divergence value of zero signals complete agreement among the classifiers in the committee, while correspondingly, increasingly larger JSD values indicate larger disagreement.

3.7 Vote entropy

Engelson and Dagan (1996) use *vote entropy* for quantifying the disagreement within a committee of classifiers used for active learning in a part-of-speech tagging task. Disagreement VE for an instance e based on vote entropy is defined as in equation 3.7.

$$VE(e) = -\frac{1}{\log k} \sum_{i=0}^{|l|} \frac{V(l_i, e)}{k} \log \frac{V(l_i, e)}{k} \tag{3.7}$$

where k is the number of members in the committee, and $V(l_i, e)$ is the number of members assigning label l_i to instance e. Vote entropy is computed per tagged unit, for instance per token. In tasks where the smallest tagged

unit is but a part of the construction under consideration, for instance in phrase chunking where each phrase may contain one or more tokens, the vote entropy of the larger unit is computed as the mean of the vote entropy of its parts (Ngai and Yarowsky 2000; Tomanek, Wermter and Hahn 2007a).

Weighted vote entropy (Olsson 2008) is applicable only in committee-based settings where the individual members of the committee has received weights reflecting their performance. For instance, this is the case with Boosting (Section 2.2.1), but not with Decorate (Section 2.2.2).

Weighted vote entropy is calculated similarly to the original vote entropy metric (equation 3.7), but with the weight of the committee members substituted for the votes. Disagreement based on weighted vote entropy WVE for an instance e is defined as in equation 3.8.

$$WVE(e) = -\frac{1}{\log w} \sum_{i=1}^{|c|} \frac{W(c_i, e)}{w} \log \frac{W(c_i, e)}{w} \tag{3.8}$$

where w is the sum of the weights of all committee members, and $W(c_i, e)$ is the sum of the weights of the committee members assigning label c_i to instance e.

3.8 F-complement

Ngai and Yarowsky (2000) compare the vote entropy measure, as introduced by Engelson and Dagan, with their own measure called *F-complement* (*F-score complement*). Disagreement FC concerning the classification of data e among a committee based on the F-complement is defined as in equation 3.9.

$$FC(e) = \frac{1}{2} \sum_{k_i, k_j \in K} (1 - F_{\beta=1}(k_i(e), k_j(e))) \tag{3.9}$$

where K is the committee of classifiers, k_i and k_j are members of K, and $F_{\beta=1}(k_i(e), k_j(e))$ is the F-score, $F_{\beta=1}$ (defined in equation 3.10), of the classifier k_i's labelling of the data e relative to the evaluation of k_j on e.

In calculating the F-complement, the output of one of the classifiers in the committee is used as the answer key, against which all other committee members' results are compared and measured (in terms of F-score).

Ngai and Yarowsky (2000) find that the task they are interested in, base noun phrase chunking, using F-complement to select instances to annotate performs slightly better than using vote entropy. Hachey, Alex and Becker (2005) use F-complement to select sentences for named entity annotation; they point out that the F-complement is equivalent to the inter-annotator agreement between $|K|$ classifiers.

The F-score is the harmonic mean of *precision* (equation 3.11) and *recall* (equation 3.12) such that

$$F = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R} \tag{3.10}$$

where β is a constant used for determining the influence of precision over recall, or vice-versa. β is usually set to 1, which is commonly referred to as F_1 or $F_{\beta=1}$. Precision, P, is defined as the ratio between the number of correctly classified instances and the number of classified instances:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.11}$$

Recall, R, is defined as the ratio between the number of correctly classified instances and the total number of instances:

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3.12}$$

Chapter 4

Data access

There are several issues related to the way that the active learner has access to the data from which it learns. First of all, the seed set of instances used to start the process (e.g., item 1 in Figure 2.1) may have impact on how the learning proceeds (Section 4.1). Further, the way that the learner is provided access to the unlabeled data has implications for the overall setting of the learning process; is the data made available as a stream or as a pool (Section 4.2)? A related question is whether a batch or singletons of unlabeled instances is processed in each learning iteration (Section 4.3).

4.1 Selecting the seed set

The initial set of labeled data used in active learning should be representative with respect to the classes that the learning process is to handle. Omitting a class from the initial seed set might result in trouble further down the road when the learner fits the classes it knows of with the unlabeled data it sees. Instances that would have been informative to the learner can go unnoticed simply because the learner, when selecting informative instances, treat instances from several classes as if they belong to one and the same class.

A related issue is that of instance distribution. Given that the learner is fed a seed set of data in which all classes are represented, the number of examples of each class plays a crucial role in whether the learner is able to properly learn how to distinguish between the classes. Should the distribution of instances in the seed set mirror the (expected) distribution of instances in the unlabeled set?

In the context of text categorization, McCallum and Nigam (1998) report on a method that allows for starting the active learning process without any labeled examples at all. They select instances (documents) from the region of the pool of unlabeled data that has the highest density. A dense region is one in which the distance (based on Kullback-Leibler divergence, defined

in equation 3.3) between documents is small. McCallum and Nigam (1998) combine expectation-maximization (Dempster, Laird and Rubin 1977) and active learning in a pool-based setting (Section 4.2); their results show that the learning in this particular setting might in fact benefit from being initiated without the use of a labeled seed set of documents.

Tomanek, Wermter and Hahn (2007b) describe a three-step approach to compiling a seed set for the task of named entity recognition in the biomedical domain. In the first step, a list of as many named entities as possible is gathered, the source being either a human domain expert, or some other trusted source. The second step involves matching the listed named entities against the sentences in the unlabeled document pool. Third, the sentences are ranked according to the number of diverse matches of named entities to include in the seed set. Tomanek, Wermter and Hahn (2007b) report results from running the same active learning experiment with three different seed sets; a randomly selected set, a set tuned according to the above mentioned method, and no seed set at all. Though the learning curves seem to converge, initially the tuned seed set clearly contributes to a better progression of learning.

Olsson (2008) compares random selection of documents to include in the seed set for a named entity recognition task, to a seed set made up from documents selected based on their distance from the centroids of clusters obtained by K-means clustering. Olsson concludes that neither query by uncertainty, nor query by committee produced better classification results when the seed sets were selected based on clustering. However, the clustering approach taken did affect the variance of the performance of the classifier learned in a positive way.

In experimental settings, a work-around to the seed set selection problem is to run the active learning process several times, and then present the average of the results achieved in each round. Averaging rounds, combined with randomly selecting a fairly large initial seed set – where its size is possibly related to the number of classes – might prove enough to circumvent the seed set problem when conducting controlled experiments. How the issue is best addressed in a live setting is not clear.

4.2 Stream-based and pool-based data access

There are two ways in which a learner is provided access to data, either from a stream, or by selecting from a pool. In stream-based selection used by, among others, Liere and Tadepalli (1997) and McCallum and Nigam (1998), unlabeled instances are presented one by one. For each instance, the learner has to decide whether the instance is so informative that is should be annotated by the teacher. In the pool-based case – used by for example Lewis and Gale (1994), and McCallum and Nigam (1998) – the learner has

access to a set of instances and has the opportunity to compare and select instances regardless of their individual order.

4.3 Processing singletons and batches

The issue of whether the learner should process a single instance or a batch of instances in each iteration has impact on the speed of the active learning process. Since in each iteration, the base learner generates classifiers based on the labeled training data available, adding only one instance at a time slows the overall learning process down. If, on the other hand, a batch of instances is added, the amount of data added to the training set in each iteration increases, and the learning process progresses faster. The prototypical active learning algorithms presented previously, see Figures 2.1, 2.2 and 2.3, respectively, do not advocate one approach over the other. In practice though, it is clearly easier to fit singleton instance processing with the algorithms. Selecting a good batch of instances is non-trivial since each instance in the batch needs to be informative, both with respect to the other instances in the batch, as well as with respect to the set of unlabeled data as a whole.

While investigating active learning for named entity recognition, Shen et al. (2004) use the notions of *informativeness*, *representativeness*, and *diversity*, and propose scoring functions for incorporating these measures when selecting batches of examples from the pool of unlabeled data. Informativeness relates to the uncertainty of an instance, representativeness relates an instance to the majority of instances, while diversity is a means to avoid repetition among instances, and thus maximize the training utility of a batch.

The pool-based approach to text classification adopted by McCallum and Nigam (1998) facilitates the use of what they refer to as *density-weighted pool-based sampling*. The density in a region around a given document – to be understood as representativeness in the vocabulary of Shen et al. (2004) – is quantified as the average distance between that document and all other documents. McCallum and Nigam (1998) combine density with disagreement, calculated as the Kullback-Leibler divergence (equation 3.3), such that the document with the largest product of density and Kullback-Leibler divergence is selected as a representative of many other documents, while retaining a confident committee disagreement. McCallum and Nigam show that density-weighted pool-based sampling used in conjunction with Kullback-Leibler divergence yields significantly better results than the same experiments conducted with pool-based Kullback-Leibler divergence, stream-based Kullback-Leibler divergence, stream-based vote entropy, and random sampling.

Tang, Luo and Roukos (2002) also experiment with representativeness, or density, albeit in a different setting; that of statistical parsing. They

propose to use clustering of the unlabeled data set based on the distance between sentences, the resulting clusters are then used to compute the density of examples. Tang and colleagues define the distance between two sentences based on the parse trees corresponding to the sentences. A parse tree can be uniquely represented by a series of events, each of which is constituted by a parse action and its context. Sentence similarity is calculated as the Hamming edit distance between two sequences of events. The Hamming distance measures the number of substitutions (or errors) required to turn one sequence into the other (Hamming 1950). The results reported by Tang, Luo and Roukos (2002) show that taking density into account helps in keeping the amount of training data needed down, compared to random sampling.

Brinker (2003) addresses the issue of incorporating a diversity measure when selecting batches of instances. Brinker's work is carried out with Support Vector Machines, and his batch selection method is accordingly described in terms of feature vectors in a high-dimensional space. When selecting single instances for querying, an instance with a minimal distance to the classification hyperplane is usually favored, since choosing such an instance will result in halving the version space. When selecting several unlabeled instances, Brinker (2003) argue that picking instances such that their angles are maximal with respect to each other, rather than relative to the decision hyperplane, is a batch selection technique which is both computationally cheap and scalable to large data sets. Brinker (2003) conducts empirical investigations using a number of UCI data sets (Asuncion and Newman 2007), and reports results indicating that previously approaches to active learning with Support Vector Machines are outperformed by his batch selection strategy.

Hoi, Jin and Lyu (2006) present work on large-scale text categorization in which a batch of documents is selected in each learning iteration. Hoi and colleagues report on the development of an active learning algorithm utilizing logistic regression as base learner, capable of selecting several documents at a time, while minimizing the redundancy in the selected batch. The uncertainty of the logistic regression model is measured using Fisher matrix information, something which is claimed to allow for the batch selection problem to be re-cast as an optimization problem in which instances from the unlabeled pool are selected in such as way that the Fisher information is maximized. The notion of Fisher information and Fisher matrix is described by Hoi, Jin and Lyu (2006). Hoi and colleagues carry out experiments on several document collections, using a range of learning methods, and conclude that their active learning approach equipped with the batch selection method is more effective than the margin-based active learning methods tested.

Chapter 5

The creation and re-use of annotated data

Under some circumstances, active learning can evidently be used to identify the most informative units in a corpus. What really takes place is the reordering, and elicitation, of available examples highly biased towards the preferences of the base learner and task in effect. The data produced is often viewed as a side effect of an annotation process that is really intended to produce a classifier with as little human effort as possible. An effect of this classifier-centric view of active learning is that the resulting annotated data may be hard to re-use by base learners other than the one used in the active learning process. Research addressing the re-usability of actively annotated data is reviewed in Section 5.1. Further, a number of research efforts adopting a more data-centric approach to active learning in that it is used as a technique when creating annotated corpora are described in Section 5.2.

5.1 Data re-use

Baldridge and Osborne (2004) use active learning to create a corpus on which to train parsers. In doing that, their principal worry is whether the selected material will be useful if used with a base learner different than the one used to select the data. Indeed, for their particular task, they find that the gains of using active learning may turn out minimal or even negative. The reason lies in how complex it is for a human annotator to assign parse trees to the selected sentences. In response to this finding, Baldridge and Osborne formulate a strategy involving semi-automatic labeling to operate in concert with active learning. The semi-automatic technique makes use of the fact that the parser used can provide ranked partial parses, of which the ones with higher probability than chance is presented to the user in a drop-down list. Baldridge and Osborne (2004) conclude that n-best automation

can be used to increase the possibility of the annotations produced being re-usable.

Tomanek, Wermter and Hahn (2007a) conduct two experiments addressing the re-usability of training material obtained by employing active learning to annotate named entities in a biomedical domain. In their first experiment, the base learners used for selecting data – called *selectors* – and the learning schemes used for testing the data – called *testers* – are varied. By keeping the feature set fixed and using the best selector, generated by a conditional random field base learner, Tomanek and colleagues show that feeding the tester with data generated by faster, albeit worse performing selectors based on maximum entropy and naïve Bayes, still yield results far better than passive learning. Comparable results are reported for the variation of the tester's base learner.

In the second experiment outlined in Tomanek, Wermter and Hahn 2007a, the feature sets used by the selectors are reduced, while that of the tester remain fixed and full. The three reduced feature sets contain, in turn, all but the syntactic features, all but the syntactic and morphological features, and finally, only orthographic features. A committee of conditional random field selectors employs each of the three reduced feature sets. Tomanek, Wermter and Hahn (2007a) show that, even when using selectors in concert with the most reduced feature set, a tester (also based on conditional random fields) still can make use of the data and generate results better than those resulting from passive learning.

Vlachos (2006) approaches the production of marked-up data slightly different than the rest; instead of employing active learning for the purpose of selecting sentences to annotate with names, Vlachos uses it to select the automatically inserted named entity annotations that need to be corrected. Vlachos finds that his approach to use active learning to select errors to correct outperforms active learning for selecting instances to annotate in all cases except for the one where very noisy data had been used to train the initial pre-tagger. Vlachos (2006) points out that his approach is likely to yield data more re-usable than the data created using "ordinary" active learning. This claim is based on the observation that the corpus produced by Vlachos's method contains all data that the initial corpus does, and although only parts of the data is manually corrected, the errors in the uncorrected parts are possibly non-significant to a machine learner. Since the distribution of the data in the resulting corpus is the same as in the original one, the former is likely to be as re-usable as the latter.

5.2 Active learning as annotation support

Olsson (2008) present a three-stage method called BootMark for bootstrapping the marking up of named entities in documents. What differentiates

BootMark from other similar methods, such as Melita (Ciravegna, Petrelli and Wilks 2002), is that BootMark makes use of active learning to select entire documents to annotate. The BootMark method consists of three phases: (1) Manual annotation of a set of documents; (2) Bootstrapping – active machine learning for the purpose of selecting which document to annotate next; (3) The remaining unannotated documents of the original corpus are marked up using pre-tagging with revision. Olsson identifies and emprically investigates five emerging issues pertaining to the realization of BootMark. The issues are related to: the characteristics of the named entity recognition task and the base learners used in conjunction with it; the constitution of the set of documents annotated by the human annotator in phase one in order to start the bootstrapping process; the active selection of the documents to annotate; the monitoring and termination of the active learning; and the applicability of the actively learned named entity recognizer as a pre-tagger for annotating unlabeled data.

Chklovski and Mihalcea (2002) direct their efforts towards collecting a word sense-tagged corpus involving the general public as annotators by using the World Wide Web as communications channel. The active learning component used to select instances to tag is made up from two classifiers created by two different base learners. An instance is picked out for annotation if the labels assigned to it by the classifiers are not equal.

In setting up the experiment, Chklovski and Mihalcea faced two rather uncommon challenges; that of ensuring the quality of annotations provided by a potentially very large and uncontrolled selection of annotators, and that of drawing attention to their task in order to bring in enough annotators. Chklovski and Mihalcea dealt with the first challenge by limiting the number of tags of an item to two, and also by limiting the number of tags assigned to an item to one per contributor. The authors proposed to make the tagging task "game-like", including awarding prizes to the winners, in order for people to be inclined to contribute. Mihalcea and Chklovski (2003) report that after the first nine months of being available on the web, their system had collected 90 000 high-quality tagged items.

Tomanek et al. (2007a; 2007b) describe the Jena annotation environment (Jane), a client-server-based workbench for accommodating and managing the labeling of texts. The task described by Tomanek and colleagues is that of named entity recognition in the immunogenetics domain, although they point out that other types of tasks are possible too. Jane contains tools supporting a single annotator, as well as for managing teams of annotators. The administrative tool facilitating the latter include modules for managing users, creating and editing projects, monitoring the annotation progress and inter-annotator agreement, and deploying the data once the annotation of it is completed. Single annotators have the option to choose focused annotation, that is, being supported by a server-side active annotation module that selects the sentences to be marked-up. Active learning is realized as query

by committee employing three different base learners – conditional random fields (Lafferty, McCallum and Pereira 2001), maximum entropy, and Naïve Bayes – with vote entropy as disagreement quantification. Tomanek et al. (2007a; 2007b) perform a real-world annotation experiment, indicating a reduction in the number of annotations with between 50% and 75%. One conclusion drawn from the experiment is that active learning is particularly advantageous when the instances of interest are sparsely distributed in the texts. The density of named entities, that is, the number of entities per sentence, in the corpus produced in the experiment is almost 15 times greater than the density of names in the test corpus. Another set of conclusions is realized as a list of requirements for facilitating deployment of active learning in practical circumstances:

- the turn-around time for selecting what to annotate needs to be kept short;
- the data produced in the annotation process need to be re-usable by other machine learners; and,
- the criterion for stopping the active learning needs to be sensitive to the progress of the performance in the annotation process.

The requirements list presented by Tomanek and colleagues is one of several manifestations of increasing awareness in the research community of the conditions under which the human annotators operate. Concerns are raised regarding the usefulness of the resulting data sets for tasks other than that which originally created the data.

Chapter 6

Cost-sensitive active learning

The performance of active learning is often measured along the lines of how much data is required to reach some given performance compared to reaching the same performance by learning from randomly sampled data from the same set of unlabeled data. However, only taking the amount of data into consideration is not always appropriate, e.g., when some types of data are harder for the user to annotate than others, or when the acquisition of certain types of unlabeled examples is expensive. In these cases, it is necessary to model the cost of learning as being attributed to other characteristics of the data and the annotation situation than simply the sheer amount of data processed. Cost in this sense is typically derived from monetary, temporal, or effort-based issues. A cost model should reflect the constraints currently in effect; for instance, if annotator time is more important than the presumed cognitive load put on the user, then the overall time should take precedence in the evaluation of the plausibility of the method under consideration. If on the other hand a high cognitive load causes the users to produce annotations with too high a variance, resulting in poor data quality, then the user situation may have to take precedence over monetary issues in order to allow for the recruitment and training of more personnel.

Using a scale mimicking the actions made by the user when annotating the data is one way of facilitating a finer grained measurement of the learning progress. For instance, Hwa (2000) uses the number of brackets required for marking up parse trees in the training data as a measure of cost, rather than using the sheer number of sentences available. Hwa (2000) uses active learning to select sentences to be marked-up with parse trees. The corpus constructed is then used to induce a statistical grammar. The sample selection method used by Hwa is based on a single learner using sentence length and tree entropy as means to select the sentences to annotate. Hwa points out that creating a statistical grammar (parser) is a complex task which differs from the kind of classification commonly addressed by active learning in two significant respects; where a classifier selects labels from a fixed set

of categories for each instance, in parsing, every instance has a different set of possible parse trees. While most classification problems concern a rather limited set of classes, the number of parse trees may be exponential with respect to the length of the sentence to parse. These two characteristics have bearing towards the complexity of the task faced by the human annotator acting as oracle in the learning process. Hwa's aim is to minimize the amount of annotation required by the human in terms of the number of sentences processed, as well as in terms of the number of brackets denoting the structure of each sentence.

Osborne and Baldridge (Osborne and Baldridge 2004; Baldridge and Osborne 2004) distinguish between *unit cost* and *discriminant cost* in their work on ensemble-based active learning for selecting parses. In their setting, unit cost is the absolute number of sentences selected in the learning process. Discriminant cost assigns a variable cost per sentence and concerns the decision an annotator has to make concerning the example parse trees provided by the system.

Culotta et al. (2006) design their system so that segmentation decisions are converted into classification tasks. They use what they refer to as Expected Number of User Actions (ENUA) to measure the effort required by the user to label each example in an information extraction setting. The ENUA is computed as a function of four atomic labeling actions, corresponding to annotating the start and end boundaries, the type of a field to be extracted, as well as to an option for the user to select the correct annotation among k predicted ones. The use of ENUA reflects the authors' goal with the system; to reduce the total number of actions required by the user. Culotta et al. (2006) notice that there is a trade-off between how large k is, that is, how many choices the user is presented with, and the reduction in amount of required user actions caused by introducing the multiple-choice selection.

It seems reasonable to assume that, on average, it must be harder to annotate the units provided by active learning, than it is annotating units randomly drawn from the same corpus simply because the former is, by definition, more informative. Along these lines, Hachey, Alex and Becker (2005) find that the three selection metrics they used in a live annotation experiment yield three distinct annotation time per token/data size curves. Hachey and colleagues measure maximum Kullback-Leibler divergence, average Kullback-Leibler divergence and F-complement for selecting the sentences in which the annotators are to mark up named entities. Hachey, Alex and Becker (2005) demonstrate that the time spent on marking up an example is correlated with its informativeness. Similarly, in the experiments conducted by Ringger et al. (2007), the selection metric resulting in the best performance in terms of amount of data needed to reach a given accuracy, is also the one demanding the most attention from the user; the amount of corrections made by the oracle is clearly larger for the most complex selection method used.

Sometimes failure according to one cost model is success when measured under different model. Ganchev, Pereira and Mandel (2007) design their process so as to require binary decisions only, as opposed to full annotations/corrections, from the user. They show that the effectiveness of their approach is inferior to that of learning from fully manually annotated data. Their semi-automatic method requires the annotator to process more data than he would have had if he had chosen to manually annotate it. This is an effect of reducing the load on the user to binary decisions. On the other hand, Ganchev and colleagues show that less effort is required by the annotator to produce annotations of a quality superior to that of manually tagging. In all, Ganchev, Pereira and Mandel (2007) conclude that, in the conducted experiments, the suggested semi-automatic method reduced annotation time by 58%.

Haertel et al. (2008) show that the active learning strategy which performs the best depends on how the annotation cost is measured. They examine the performance of query by uncertainty and query by committee for the task of part-of-speech tagging under a number of cost models. The models used include what Haertel and colleagues refer to as an hourly cost model. As an example of how a cost model can be used in their particular setting, Haertel et al. (2008) show that when a low tag accuracy is required, random selection of what to annotate is cheapest according to the hourly cost model. On the other hand, query by uncertainty is cost-effective (compared to a random baseline) starting at around 91% tag accuracy, while query by committee is more effective than both the baseline and query by uncertainty at tag accuracies starting at around 80%.

Settles, Craven and Friedland (2008) describe four active learning tasks involving human annotators; the annotating of entities and relations from news-wire text, the classification of images in a content-based image retrieval system, the annotation of speculative vs. definite statements in biomedical text, and the annotation of contact information from email text. Burr and colleagues investigate various aspects of the time required to annotate data, and use that as the cost model in their experiments. They ask questions such as: Are annotation times variable for a given task or domain? Do times vary from one annotator to the next? Can we improve active learning by utilizing cost information? Burr and colleagues conclude that the cost, i.e., time, can vary considerably across the instances of data annotated, that active learning can fail if the variability of the cost of instances are not taken into account, and that, in some domains, it is possible to learn to predict the annotation costs.

Castro et al. (2008) investigate what they refer to as human active learning. They conduct a series of experiments in which humans are to categorize instances as belonging to one of two classes. The conclusions made by Castro and colleagues include that humans are capable of actively selecting informative examples from a pool of unlabeled examples, and when doing so, the

humans learn faster and better than they would have done if the examples were provided to them based on random sampling. However, the human active learning does not show as much improvement as that obtained by active machine learning on the same task. Further, Castro and colleagues conclude that human active learning sensitive to noise, and do not approach the theoretical bounds set out in active machine learning, that is, humans are not as good as machines in selecting informative queries from an unlabeled data set.

Although the work by Castro et al. (2008) is not directly geared towards cost-sensitive learning, my interpretation is that it raises important questions pertaining to the cost of the of labeling data in terms of effort required by the human annotator: Are the examples located between the human upper bound and the machine upper bound are too hard for the human to label? Does this mean that it is not necessary to let the machine select the most informative queries, that is, the ones hardest to label, in order for machine active learning (involving a human) to work optimally? Ultimately, does this mean that the human is the weakest point in active learning in that he cannot assimilate the most informative examples provided to him by the active machine learning algorithm?

The research on cost-sensitive active learning in general, and the results obtained by Castro et al. (2008) in particular, indicate that we should pay more attention to the practical situation of the human annotator when striving to make active learning an operational tool for practitioners of NLP.

Chapter 7

Monitoring, assessing and terminating the learning process

The monitoring, assessing and terminating of the active learning process go hand in hand. The purpose of assessing the learning process is to provide the human annotator with means to form a picture of the learning status. Once the status is known, the annotator has the opportunity to act accordingly, for instance, to manually stop the active learning process.

The purpose of defining the stopping criterion is slightly different than that of assessing the learning process. A stopping criterion is used to automatically terminate the learning process, and ideally the realization of the definition, e.g., the setting of any thresholds necessary, should not hinder nor disturb the human annotator.

It should be remembered that there is a readily available way of assessing the process, and thus also to be able to manually decide when the active learning should be stopped; to use a marked-up, held-out test set on which the learner is evaluated in each iteration. This is the way that active learning is usually evaluated in experimental settings. The drawback of this method is that the user has to manually annotate more data before the annotation process takes off. As such, it clearly counteracts the goal of active learning and should only be considered a last resort.

7.1 Measures for monitoring learning progress

A very common way of monitoring how an active learner behaves is by plotting a learning curve, typically with the classification error or F-score along one axis, commonly the Y-axis, and something else along the other axis. It is that *something else* that is of interest here. The X-axis is usually indicating the amount of data seen – depending on the granularity of choice

for that particular learning task it can be for instance tokens, named entities, or sentences – or the number of iterations made while learning. The purpose of a learning curve is to depict the progress in the learning process; few variations of how to measure progress exist, and consequently there are few differences in how the axes of a graph illustrating a learning curve are labeled.

There are times when the graphical nature of learning curves is not an appropriate means to describe the learning process. Abe and Mamitsuka (1998) calculate the *data efficiency* achieved by using an active learning approach as the ratio between the number of iterations required by a base learner to reach top performance when data is drawn at random, and the number of iterations required for the base learner in an active learning setting to reach the same performance.

Melville and Mooney (2004) defines the *data utilization ratio* – which is similar to the data efficiency introduced by Abe and Mamitsuka (1998) – as the number of instances an active learner requires to reach a target error rate divided by the number that the base learner – Decorate – requires to reach the same error rate. Both *data efficiency* and *data utilization ratio* reflect how good an active learner is at making use of the data.

Baram, El-Yaniv and Luz (2004) propose to use a quantification of the *deficiency* of the querying function with respect to randomly selecting instances from which to learn. The deficiency is defined in equation 7.1.

$$Deficiency_n(A) = \frac{\sum_{t=1}^{n}(Acc_n(L) - Acc_t(A))}{\sum_{t=1}^{n}(Acc_n(L) - Acc_t(L))} \tag{7.1}$$

where t is the training set size, $Acc_n(L)$ is the maximal achievable accuracy when using algorithm L and all available training data, $Acc_t(A)$ is the average accuracy achieved by active learning algorithm A and t amount of training data, and $Acc_t(L)$ is the average accuracy achieved using random sampling and learning algorithm L and t amount of training data. The deficiency measure captures the global performance of active learner A throughout the learning process. Smaller values indicate more efficient learning.

There are, of course, more parameters than data-related ones to consider when using active learning in a practical setting, such as time, money, cognitive load on the user; Chapter 6 brings up a number of issues relating to the cost of annotation.

7.2 Assessing and terminating the learning

A number of different approaches for assessing and deciding when to stop the active learning process have been suggested in the literature. These approaches include to decide on a target accuracy and stop when it has

been reached, to go on for a given number of active learning iterations, or to exhaust the pool of unlabeled data.

Some more elaborate methods monitor the accuracy as the learning process progresses and stop when accuracy deterioration is detected. Schohn and Cohn (2000) observe, while working with Support Vector Machines for document classification, that when instances are drawn at random from the pool of unlabeled data, the classifier performance increases monotonically. However, when Schohn and Cohn add instances according to their active learning selection metric, classifier performance peaks at a level above that achieved when using all available data. Thus, they obtain better performance by training on a subset of data, than when using all data available. Schohn and Cohn (2000) use this observation to form the basis for a stopping criterion; if the best, most informative instance is no closer to the decision hyperplane than any of the support vectors, the margin has been exhausted and learning is terminated. This is an approximation of true peak performance that seem to work well, Schohn and Cohn (2000) claim.

Zhu and Hovy (2007) investigate two strategies for deciding when to stop learning – *max-confidence* and *min-error* – in a word sense disambiguation task. Max-confidence relies on an entropy-based uncertainty measure of unlabeled instances, while min-error is based on the classification accuracy of predicted labels for instances when compared to the labels provided by the human annotator. Thresholds for max-confidence and min-error are set such that when the two conditions are met, the current classifier is assumed to provide high confidence in the classification of the remaining unlabeled data. The experiments carried out by Zhu and Hovy indicate that min-error is a good choice of stopping criterion, and that the max-confidence approach is not as good as min-error.

Zhu, Wang and Hovy (2008a) extend the work presented in (Zhu and Hovy 2007) and introduce an approach called *minimum expected error strategy*. The strategy involves estimating the classification error on future unlabeled instances in the active learning process. Zhu and colleagues test their stopping criterion on two tasks; word sense disambiguation, and text classification. Zhu, Wang and Hovy (2008a) conclude that the minimum error strategy achieves promising results.

In addition to the max-confidence and min-error strategies, Zhu, Wang and Hovy (2008b) introduce and evaluate *overall-uncertainty* and *classification-change*. Overall-uncertainty is similar to max-confidence, but instead of taking only the most informative instances into consideration, overall-uncertainty is calculated using all data remaining in the unlabeled pool. Classification-change builds on the assumption that the most informative instance is the one which causes the classifier to change the predicted label of the instance. Zhu and colleagues realize the classification-change-based stopping criterion such that the learning process is terminated once no predicted label of the instances in the unlabeled pool changes during two

consecutive active learning iterations. Zhu, Wang and Hovy (2008b) propose ways of combining max-confidence, min-error, and overall-uncertainty with classification-change in order to come to terms with the problem of pre-defining the required thresholds. Zhu and colleagues conclude that the proposed criteria work well, and that the combination strategies can achieve even better results.

Vlachos (2008) suggests to use classifier confidence as a means to define a stopping criterion for uncertainty based sampling. Roughly, the idea is to stop learning when the confidence of the classifier, on an external possibly unannotated test set, remains at the same level or drops for a number of consecutive iterations during the learning process. Vlachos shows that the criterion indeed is applicable to the two tasks he investigates; text classification and named entity recognition carried out using Support Vector Machines, maximum entropy models, and Bayesian logistic regression.

Laws and Schütze (2008) investigate three ways of terminating uncertainty-based active learning for named entity recognition; *minimal absolute performance*, *maximum possible performance*, and *convergence*. The minimal absolute performance of the system is set by the user prior to starting the active learning process. The classifier then estimates its own performance using a held-out unlabeled data set. Once the desired performance is reached, the learning is terminated. The maximum possible performance strategy refers to the optimal performance of the classifier given the data. Once the optimal performance is achieved, the process is aborted. Finally, the convergence criterion aims to stop the learning process when the pool of available data does not contribute to the classifier's performance. The convergence is calculated as the gradient of the classifier's estimated performance or uncertainty. Laws and Schütze (2008) conclude that both gradient-based approaches, that is, convergence, can be used as stopping criteria relative to the optimal performance achievable on a given pool of data. Laws and Schütze also show that while their method lend itself to acceptable estimates of accuracy, it is much harder to estimate the recall of the classifier. Thus, the stopping criteria based on minimal absolute performance as well as maximum possible performance are not reliable.

Tomanek and Hahn (2008) examine two ways of monitoring the progression of learning in the context of a query by committee setting for training named entity recognizers. Their first approach relies on the assumption that the agreement within the decision committee concerning the most informative instance selected in each active learning iteration approaches one as the learning process progresses. Tomanek and Hahn refer to this as the *selection agreement*, originally introduced in Tomanek, Wermter and Hahn 2007a. The motivation for using the selection agreement score is that active learning should be aborted when it no longer contributes to increasing the performance of the classifier; at that time, active learning is nothing more than a computationally expensive way of random sampling from the remaining data.

The second approach taken by Tomanek and Hahn is to calculate the agreement within the committee regarding a held-out, unannotated test set. This is referred to as the *validation set agreement*. The idea is to calculate the agreement on a test set with a distribution of names that reflects that of the data set on which the active learning takes place. In doing so, Tomanek and Hahn (2008) aim at obtaining an image of the learning progression that is more true than that obtained by calculating the selection agreement, simply because the distribution of the held-out set, and thus also the validation set agreement score, is not affected by the progression of the learning process in the same manner as the selection agreement score is. Tomanek and Hahn (2008) carry out two types of experiments. In the first type, the human annotator is simulated in the sense that the active learning utilizes pre-annotated data; the annotated training examples supplied to the system are in fact not annotated by a human at the time the system requests assistance in classifying them, but comes from the pre-annotated corpus. In this type of experiment, the amount of data is typically limited. The second type of experiment conducted by Tomanek and Hahn (2008) involves real human annotators who operate on a substantially larger amount of data, approximately 2 million sentences, as opposed to the at most 14 000 sentences used in the experiments with simulated annotators.

Tomanek and Hahn (2008) find that, for the experiments with simulated annotators (using relatively small amounts of data), both the selection agreement curves and the validation set agreement curves can be useful for approximating a learning curve, thus indicating the progression of the learning process. However, for the experiments employing human annotators and large amounts of unlabelled data, the selection agreement does not work at all. Tomanek and Hahn conclude that monitoring the progress of active learning should always be based on a separate validation set instead of the data directly affected by the learning process. Thus, validation set agreement is preferred over selection agreement.

Olsson (2008) proposes an intrinsic stopping criterion (ISC) for committee-based active learning. The criterion is further elaborated by Olsson and Tomanek (2009). The ISC is intrinsic, relying only on the characteristics of the base learner and the data at hand in order to decide when the active learning process may be terminated. The ISC does not require the user to set any external parameters prior to initiating the active learning process. Further, the ISC is designed to work with committees of classifiers, and as such, it is independent of how the disagreement between the committee members is quantified. The ISC does neither rely on a particular base learner, nor on a particular way of creating the decision committee.

The ISC combines the selection agreement and the validation set agreement (Tomanek, Wermter and Hahn 2007a; Tomanek and Hahn 2008) into a single stopping criterion by relating the agreement of the committee on a held-out validation set with that on the (remaining) pool of unlabeled

data. If the selection agreement is larger than the validation set agreement, it is a signal that the decision committee is more in agreement concerning the most informative instances in the (diminishing) unlabeled pool than it is concerning the validation set. This, in turn, implies that the committee would learn more from a random sample from the validation set (or from a data source exhibiting the same distribution of instances), than it would from the unlabeled data pool. Based on this argument, a stopping criterion for committee-based active learning can be formulated as: Active learning may be terminated when the Selection Agreement is larger than, or equal to, the Validation Set Agreement.

Olsson and Tomanek define and empirically test the ISC for committee-based active learning. The results of the experiments in two named entity recognition scenarios show that the stopping criterion is a viable one, representing a fair trade-off between data use and classifier performance. In a setting in which the unlabeled pool of data used for learning is static, terminating the learning process by means of the ISC results in a nearly optimal classifier. The ISC can also be used for deciding when the pool of unlabeled data needs to be refreshed.

Of the approaches to defining a stopping criterion for active learning reviewed, the work described by Tomanek and colleagues, and the work by Olsson is explicitly directed towards committee-based active learning. The other approaches involve single classifier active learning strategies.

Bibliography

Abe, Naoki and Hiroshi Mamitsuka 1998. Query learning strategies using boosting and bagging. *Proceedings of the Fifteenth International Conference on Machine Learning*, 1–9. Madison, Wisconsin, USA: Morgan Kaufmann Publishers Inc.
7, 8, 17, 36

Angluin, Dana 1988. Queries and concept learning. *Machine Learning* 2 (4): 319–342.
4

Argamon-Engelson, Shlomo and Ido Dagan 1999. Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research* 11: 335–360.
4

Asuncion, Arthur and David Newman 2007. UCI Machine Learning Repository. URL: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
8, 26

Balcan, Maria-Florina, Avrim Blum and Ke Yang 2005. Co-training and expansion: Towards bridging theory and practice. *Advances in neural information processing systems 17*, 89–96. Cambridge, Massachusetts, USA: MIT Press.
14, 15

Baldridge, Jason and Miles Osborne 2004. Active learning and the total cost of annotation. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 9–16. ACL, Barcelona, Spain.
27, 32

Baram, Yoram, Ran El-Yaniv and Kobi Luz 2004. Online choice of active learning algorithms. *Journal of Machine Learning Research* 5 (December): 255–291.
36

Becker, Markus, Ben Hachey, Beatrice Alex and Claire Grover 2005. Optimising selective sampling for bootstrapping named entity recognition. Stefan Rüping and Tobias Scheffer (eds), *Proceedings of the ICML 2005 Workshop on Learning with Multiple Views*, 5–11. Bonn, Germany.
3, 20

Becker, Markus and Miles Osborne 2005. A two-stage method for active learning of statistical grammars. *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 991–996. Edinburgh, Scotland, UK: Professional Book Center.
4, 5, 6, 20

Blum, Avrim and Tom Mitchell 1998. Combining labeled and unlabeled data with co-training. *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 92–100. ACM, Madison, Wisconsin, USA.
9, 13, 14, 15

Breiman, Leo 1996. Bagging predictors. *Machine Learning* 24 (2): 123–140 (August).
7

Brinker, Klaus 2003. Incorporating diversity in active learning with support vector machines. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 59–66. Washington DC, USA: AAAI Press.
26

Castro, Rui, Charles Kalish, Robert Nowak, Ruichen Qian, Timothy Rogers and Xiaojin Zhu 2008. Human active learning. *Proceedings of the 22nd annual conference on neural information processing systems*. Vancouver, British Columbia, Canada.
33, 34

Chan, Yee Seng and Hwee Tou Ng 2007. Domain adaptation with active learning for word sense disambiguation. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, 49–56. ACL, Prague, Czech Republic.
4

Chawla, Nitesh V. and Grigoris Karakoulas 2005. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research* 23 (March): 331–366.
14

Chen, Jinying, Andrew Schein, Lyle Ungar and Martha Palmer 2006. An empirical study of the behavior of active learning for word sense disambiguation. *Proceedings of the Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL 2006)*, 120–127. ACL, New York, New York, USA.
4

Chklovski, Timothy and Rada Mihalcea 2002. Building a sense tagged corpus with open mind word expert. *Proceedings of the SIGLEX/SENSEVAL*

Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, 116–122. ACL, Philadelphia, Pennsylvania, USA.
29

Ciravegna, Fabio, Daniela Petrelli and Yorick Wilks 2002. User-system cooperation in document annotation based on information extraction. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*. Siguenza, Spain: Springer Verlag.
29

Cohn, David, Les Atlas and Richard Ladner 1994. Improving generalization with active learning. *Machine Learning* 15 (2): 201–221 (May).
4, 5

Collins, Michael and Yoram Singer 1999. Unsupervised models for named entity classification. *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 100–110. ACL, University of Maryland, College Park, Maryland, USA.
13

Culotta, Aron, Trausti Kristjansson, Andrew McCallum and Paul Viola 2006. Corrective feedback and persistent learning for information extraction. *Journal of Artificial Intelligence* 170 (14): 1101–1122 (October).
3, 32

Dagan, Ido and Sean P. Engelson 1995. Committee-based sampling for training probabilistic classifiers. *Proceedings of the Twelfth International Conference on Machine Learning*, 150–157. Tahoe City, California, USA: Morgan Kaufmann.
4

Dempster, Arthur, Nan Laird and Donald Rubin 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* 39 (1): 1–38.
24

Domingos, Pedro 2000. A unified bias-variance decomposition and its applications. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, 231–238. Stanford University, California, USA.
7

Douglas, Shona 2003. Active learning for classifying phone sequences from unsupervised phonotactic models. *Proceedings of Human Language Technology Conference – North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL 2003)*, 19–21. ACL, Edmonton, Alberta, Canada.

4

Engelson, Sean P. and Ido Dagan 1996. Minimizing manual annotation cost in supervised training from corpora. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 319–326. ACL, Santa Cruz, California, USA.
20

Finn, Aidan and Nicolas Kushmerick 2003. Active learning selection strategies for information extraction. *Proceedings of the International Workshop on Adaptive Text Extraction and Mining (ATEM-03)*, 18–25. Catvat, Dubrovnik, Croatia.
3

Freund, Yoav and Robert E. Schapire 1997. A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and Systems Science* 55 (1): 119–139 (August).
7

Freund, Yoav, Sebastian H. Seung, Eli Shamir and Naftali Tishby 1997. Selective sampling using the query by committee algorithm. *Machine Learning* 28 (2-3): 133–168 (August/September).
7

Ganchev, Kuzman, Fernando Pereira and Mark Mandel 2007. Semi-automated named entity annotation. *Proceedings of the Linguistic Annotation Workshop*, 53–56. ACL, Prague, Czech Republic.
33

Goldman, Sally A. and Yan Zhou 2000. Enhancing supervised learning with unlabeled data. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, 327–334. Stanford, California, USA.
14

Hachey, Ben, Beatrice Alex and Markus Becker 2005. Investigating the effects of selective sampling on the annotation task. *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, 144–151. ACL, Ann Arbor, Michigan, USA.
3, 21, 32

Haertel, Robbie, Eric Ringger, Kevin Seppi, James Carroll and Peter McClanahan 2008. Assessing the costs of sampling methods in active learning for annotation. *Proceedings of the 46th annual meeting of the association for computational linguistics: Human language technologies, short papers (companion volume)*, 65–68. ACL, Columbus, Ohio, USA.
33

Hamming, Richard W. 1950. Error detecting and error correcting codes. *Bell System Technical Journal* 26 (2): 147–160 (April).
26

Hoi, Steven C. H., Rong Jin and Michael R. Lyu 2006. Large-scale text categorization by batch mode active learning. *Proceedings of the 15th International World Wide Web Conference (WWW 2006)*, 633–642. Edinburgh, Scotland.
3, 26

Hwa, Rebecca 2000. Sample selection for statistical grammar induction. *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 45–52. ACL, Hong-Kong.
4, 31

Hwa, Rebecca, Miles Osborne, Anoop Sarkar and Mark Steedman 2003. Corrected co-training for statistical parsers. *Proceedings of the Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*. Washington DC, USA.
4, 10

Jones, Rosie, Rayid Ghani, Tom Mitchell and Ellen Riloff 2003. Active learning for information extraction with multiple view feature sets. *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*. Washington DC, USA.
3

Kim, Seokhwan, Yu Song, Kyungduk Kim, Jeong-Won Cha and Gary Geunbae Lee 2006. MMR-based active machine learning for bio named entity recognition. *Proceedings of the Human Language Technology Conference – North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL 2006)*, 69–72. ACL, New York, New York, USA.
3

Körner, Christine and Stefan Wrobel 2006. Multi-class ensemble-based active learning. *Proceedings of The 17th European Conference on Machine Learning and the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 687–694. Berlin, Germany: Springer-Verlag.
17, 18, 19

Kuo, Jin-Shea, Haizhou Li and Ying-Kuei Yang 2006. Learning transliteration lexicons from the web. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association of Computational Linguistics*, 1129–1136. ACL, Sydney, Australia.
4

Lafferty, John, Andrew McCallum and Fernando Pereira 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the Eighteenth International Confer-*

ence on Machine Learning (ICML-2001), 282–289. Williamstown, Massachusetts, USA.
30

Laws, Florian and Hinrich Schütze 2008. Stopping criteria for active learning of named entity recognition. *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, 465–472. ACL, Manchester, England.
38

Lewis, David D. 1995. A sequential algorithm for training text classifiers: Corrigendum and additional data. *ACM SIGIR Forum* 29 (2): 13–19.
3

Lewis, David D. and William A. Gale 1994. A Sequential Algorithm for Training Text Classifiers. *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 3–12. Dublin, Ireland: ACM/Springer.
3, 5, 24

Liere, Ray and Prasad Tadepalli 1997. Active learning with committees for text categorization. *Proceedings of the fourteenth national conference on artificial intelligence*, 591–597. AAAI, Providence, Rhode Island, USA.
3, 7, 24

Lin, Jianhua 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory* 37 (1): 145–151 (January).
20

McCallum, Andrew and Kamal Nigam 1998. Employing em and pool-based active learning for text classification. *Proceedings of the 15th International Conference on Machine Learning (ICML-98)*, 350–358. Madison, Wisconsin, USA: Morgan Kaufmann.
3, 20, 23, 24, 25

Melville, Prem and Raymond J. Mooney 2003. Constructing diverse classifier ensembles using artificial training examples. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, 505–510. Acapulco, Mexico.
8

Melville, Prem and Raymond J. Mooney 2004. Diverse ensembles for active learning. *Proceedings of the 21st International Conference on Machine Learning (ICML-2004)*, 584–591. Banff, Canada.
8, 17, 36

Mihalcea, Rada and Timothy Chklovski 2003. Open mind word expert: Creating large annotated data collections with web user's help. *Proceedings of the EACL 2003 Workshop on Linguistically Annotated Corpora (LINC 2003)*. EACL, Budapest, Hungary.
29

Mitchell, Tom 1997. *Machine learning*. McGraw-Hill.
1

Muslea, Ion, Steven Minton and Craig A. Knoblock 2000. Selective sampling with redundant views. *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-2000)*, 621–626. Austin, Texas, USA.
11

Muslea, Ion, Steven Minton and Craig A. Knoblock 2002a. Adaptive view validation: A first step towards automatic view detection. *Proceedings of the 19th International Conference on Machine Learning (ICML 2002)*, 443–450. Sydney, Australia.
13, 14

Muslea, Ion, Steven Minton and Craig A. Knoblock 2002b. Active + semi-supervised learning = robust multi-view learning. *Proceedings of the 19th International Conference on Machine Learning (ICML-02)*, 435–442. Sydney, Australia.
14, 15

Muslea, Ion, Steven Minton and Craig A. Knoblock 2006. Active learning with multiple views. *Journal of Artificial Intelligence Research* 27 (October): 203–233.
11, 12

Ngai, Grace and David Yarowsky 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 117–125. ACL, Hong-Kong.
21

Nigam, Kamal and Rayid Ghani 2000. Analyzing the effectiveness and applicability of co-training. *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM 2000)*, 86–93. ACM, McLean, Virginia, USA.
3, 13

Olsson, Fredrik 2008. Bootstrapping Named Entity Annotation by means of Active Machine Learning – A Method for Creating Corpora. Ph.D. diss., Department of Swedish, University of Gothenburg.
21, 24, 28, 39

Olsson, Fredrik and Katrin Tomanek 2009. An intrinsic stopping criterion for committee-based active learning. *Proceedings of the 13th conference on computational natural language learning*. ACL, Boulder, Colorado, USA.
39

Osborne, Miles and Jason Baldridge 2004. Ensemble-based active learning for parse selection. *Proceedings of Human Language Technology Conference – the North American Chapter of the Association for Computa-*

tional Linguistics Annual Meeting (HLT-NAACL 2004), 89–96. ACL, Boston, Massachusetts, USA.
4, 32

Pereira, Fernando C. N., Naftali Tishby and Lillian Lee 1993. Distributional clustering of English words. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 183–190. ACL, Columbus, Ohio, USA.
19

Pierce, David and Claire Cardie 2001. Limitations of co-training for natural language learning from large datasets. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP 2001)*, 1–9. Pittsburgh, Pennsylvania, USA.
9, 11

Reichart, Roi and Ari Rappoport 2007. An ensemble method for selection of high quality parses. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, 408–415. ACL, Prague, Czech Republic.
4

Ringger, Eric, Peter McClanahan, Robbie Haertel, George Busby, Marc Carmen, James Carroll, Kevin Seppi and Deryle Lonsdale 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. *Proceedings of the Linguistic Annotation Workshop*, 101–108. ACL, Prague, Czech Republic.
4, 32

Sassano, Manabu 2002. An empirical study of active learning with support vector machines for Japanese word segmentation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 505–512. ACL, Philadelphia, USA.
4

Schapire, Robert E. 1990. The strength of weak learnability. *Machine Learning* 5 (2): 197–227 (June).
7

Schapire, Robert E. 2003. The boosting approach to machine learning: An overview. D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick and B. Yu (eds), *Nonlinear Estimation and Classification*, Volume 171 of *Lecture Notes in Statistics*, 149–172. Springer.
7

Schapire, Robert E., Yoav Freund, Peter Bartlett and Wee Sun Lee 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics* 26 (5): 1651–1686 (October).
17

Scheffer, Tobias, Christian Decomain and Stefan Wrobel 2001. Active hidden

Markov models for information extraction. *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis (IDA-2001)*, 309–318. Lisbon, Portugal: Springer.
3, 5

Schohn, Greg and David Cohn 2000. Less is more: Active learning with support vector machines. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, 839–846. Stanford University, Stanford, California, USA: Morgan Kaufmann.
3, 5, 37

Settles, Burr 2009. Active learning literature survey. Computer sciences technical report 1648, University of Wisconsin-Madison.
1

Settles, Burr, Mark Craven and Lewis Friedland 2008. Active learning with real annotation costs. *Proceedings of the workshop on cost sensitive learning held in conjunction with the 23rd annual conference on neural information processing systems*. Vancouver, British Columbia, Canada.
33

Seung, H. Sebastian, Manfred Opper and Haim Sompolinsky 1992. Query by committee. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 287–294. Pittsburgh, Pennsylvania, USA: ACM.
6

Shannon, Claude E. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27 (July and October): 379–423 and 623–656.
18

Shen, Dan, Jie Zhang, Jian Su, Guodong Zhou and Chew-Lim Tan 2004. Multi-criteria-based active learning for named entity recognition. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 589–596. ACL, Barcelona, Spain.
3, 25

Steedman, Mark, Rebecca Hwa, Stephen Clark, Miles Osborne, Anoop Sarkar, Julia Hockenmaier, Paul Ruhlen, Steven Baker and Jeremiah Crim 2003. Example selection for bootstrapping statistical parsers. *Proceedings of Human Language Technology Conference – North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL 2003)*, 157–164. ACL, Edmonton, Alberta, Canada.
4

Tang, Min, Xiaoqiang Luo and Salim Roukos 2002. Active learning for statistical natural language parsing. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, 120–127. ACL, Philadelphia, Pennsylvania, USA.
4, 25, 26

Thompson, Cynthia A., Mary Elaine Califf and Raymond J. Mooney 1999. Active learning for natural language parsing and information extraction. *Proceedings of the Sixteenth International Machine Learning Conference (ICML-99)*, 406–414. Bled, Slovenia.
4

Tomanek, Katrin and Udo Hahn 2008. Approximating learning curves for active-learning-driven annotation. *Proceedings of Sixth International Conference on Language Resources and Evaluation (LREC 2008)*. ELRA, Marrakech, Morocco.
38, 39

Tomanek, Katrin and Fredrik Olsson 2009. A web survey on the use of active learning to support annotation of text data. *To appear in: Proceedings of naacl hlt 2009 workshop on active learning for natural language processing*. ACL, Boulder, Colorado, USA.
1

Tomanek, Katrin, Joachim Wermter and Udo Hahn 2007a. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 486–495. ACL, Prague, Czech Republic.
21, 28, 29, 30, 38, 39

Tomanek, Katrin, Joachim Wermter and Udo Hahn 2007b. Efficient annotation with the jena annotation environment (JANE). *Proceedings of the Linguistic Annotation Workshop*, 9–16. ACL, Prague, Czech Republic.
24, 29, 30

Tong, Simon and Daphne Koller 2002. Support vector machine active learning with applications to text classification. *Journal of Machine Learning* 2 (March): 45–66.
3

Tur, Gokhan, Dilek Hakkani-Tür and Robert E. Schapire 2005. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication* 45 (2): 171–186 (February).
4

Vlachos, Andreas 2006. Active annotation. *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, 64–71. ACL, Trento, Italy.
3, 28

Vlachos, Andreas 2008. A stopping criterion for active learning. *Computer, Speech and Language* 22 (3): 295–312 (July).
38

Witten, Ian H. and Eibe Frank 2005. *Data mining: Practical machine learn-*

ing tools with java implementations. 2nd edition. San Fransisco: Morgan Kaufmann.
1

Wu, Wei-Lin, Ru-Zhan Lu, Jian-Yong Duan, Hui Liu, Feng Gao and Yu-Quan Chen 2006. A weakly supervised learning approach for spoken language understanding. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 199–207. ACL, Sydney, Australia.
4

Zhang, Kuo, Jie Tang, JuanZi Li and KeHong Wang 2005. Feature-correlation based multi-view detection. *Computational Science and Its Applications (ICCSA 2005)*, Lecture Notes in Computer Science, 1222–1230. Springer-Verlag.
14

Zhu, Jingbo and Eduard Hovy 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 783–790. ACL, Prague, Czech Republic.
4, 37

Zhu, Jingbo, Huizhen Wang and Eduard Hovy 2008a. Learning a stopping criterion for active learning for word sense disambiguation and text classification. *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008)*, 366–372. Hyderabad, India.
4, 37

Zhu, Jingbo, Huizhen Wang and Eduard Hovy 2008b. Multi-criteria-based strategy to stop active learning for data annotation. *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, 1129–1136. ACL, Manchester, England.
37, 38

Author index