# Reducing the annotation cost for Natural Language Processing

Andreas Vlachos
*Computer Laboratory, University of Cambridge*
*av308@cl.cam.ac.uk*

## Introduction

*An important issue in applying statistical natural language processing techniques is the need for annotated training material. While the explosion of the WWW has created a big amount of textual information in electronic format, very little part of it is annotated. The lack of annotation becomes the main bottleneck when attempting new tasks or porting existing techniques to new domains or languages that don't have resources available.*

*In order to overcome the lack of annotated material, various methods were developed that either take advantage of extant resources to create training material automatically, or use some seed patterns and iteratively bootstrap a classifier. However, such methods have limitations because the material generated is noisy therefore harming the performance and they still require some manually created resources.*

*A different approach to this issue is the use of active learning (AL). In this framework, the supervised classifier selects the instances that are likely to be the most informative to train on. Active learning has been used successfully in many tasks with a variety of classifiers and the savings in annotated instances used to achieve a certain performance level were substantial.*

*Recently though, there has been some scepticism concerning active learning. It was pointed out that the data selected by active learning using a certain learning method might not be useful to train a different learning method. This is a significant problem, since ideally we would like to be able to use the data produced by active learning to train a variety of models, otherwise called reusability of the data. Another point of criticism is that all of the literature in active learning is based on simulation experiments, i.e. experiments in which all the data is annotated beforehand and the authors report savings in annotation cost comparing with annotating the whole data. Such results are not realistic, because they require all of the data to be annotated in the first place, which is exactly what we aim to avoid. Also, it is important to take into account how real life annotators work in order to estimate the actual savings in annotation costs.*

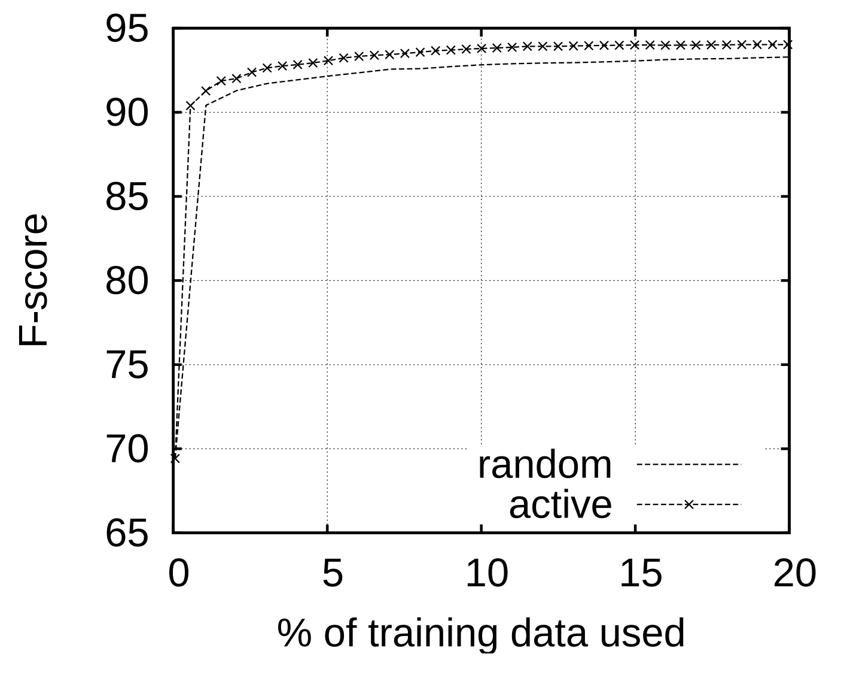
*In my phd I am trying to improve the way training material is generated by combining ideas from unsupervised learning and active learning. The end goal is to provide efficient ways to reduce the annotation cost for NLP tasks so that more people can reap its benefits.*

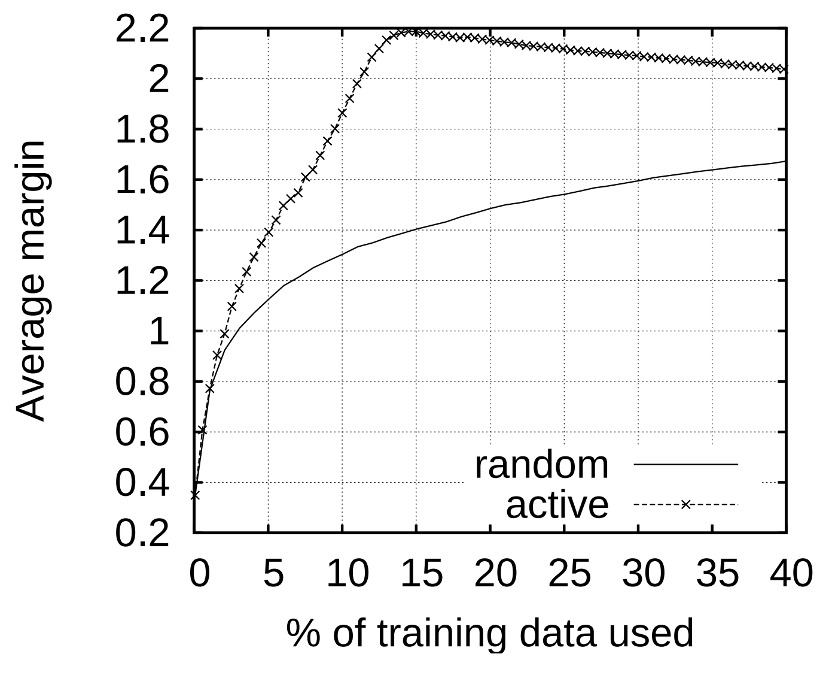
## *Work so far*

### *Active learning stopping criterion*

*The first issue that we dealt with was the definition of an active learning stopping criterion that does not require a pre-annotated dataset. An obvious stopping criterion for active learning would be to measure the performance of the trained classifier on an annotated dataset and terminate the procedure when the performance ceases to improve or it improves at a non-satisfactory rate, or when we cannot afford any more annotation. However, this might not be ideal. Apart from the costs involved in creating a test dataset, there is also the risk that it might not be representative of what can be learnt from the pool of unlabelled data. We suggested to track the confidence of the classifier, which during the most polpular active learning approach used in NLP, uncertainty based sampling exhibits a rise-peak-drop pattern, with the drop occuring when the pool of unlabelled data has been exhausted.*

We confirmed the applicability of the stopping criterion suggested by applying it to text classification using support vector machines (graph below) and bayesian logistic regression.





### Bootstrapping

In order to deal with the lack of annotated data, in the context of the ongoing FlySLIP project, we developed gene name recognizers using training data that was created automatically. In order to achieve this we used freely available text and a dictionary of gene names and their synonyms, both gathered by the FlyBase database curators. The resulting annotated material contained noise, but our expectation was that the statistical learning models would be able to overcome this. Two statistical gene name recognizers were built using the training material generated. The first one is a Hidden Markov Model which depends on heavily on lexical information. The second sytem employs Conditional Random Fields combined with features extracted from the output of a domain-indendent syntactic parser. The performance of both systems was comparable with systems trained on manually annotated material (more than 70% F-score), while the cost invoved in generating their training material automatically is very little.

### Active annotation

In order to reduce the noise in the automatically generated training material, we developed active annotation. In this framework, training data is generated automatically initially and then the statistical learner discovers errors in the data which are corrected by a human.. This procses resembles active learning in terms of employing human effort but it avoids data selection by a specific classifier which might not be useful in training a different system. In experiments on biomedical named entity recognition, we were able to improve the performance of the system used substantially by correcting very few instances in the training data compared to correcting errors randomly.

## Next steps

*The next step in our work will be to incorporate clustering in order to achieve better results. While clustering has been used before in the context of active learning and unsupervised learning, recent advances lead to bayesian non-parametric clustering algorithms, such as the Indian Buffet Process and the Pittman-Yor process. These algorithms have properties that render them particularly suitable for NLP, such as the ability to group instances in multiple clusters and being able to replicate the Zipfian distribution of natural language. In addition, unlike other clustering algorithms commonly used, they are able to discover the number of clusters which suits the data. We plan to extend them in order to be able to take into account human input, so that a generic initial clustering can be adapted to a particular task. From there, training data can be generated in order to build systems that can deal with a variety of NLP tasks.*