Procesamiento del Lenguaje Natural, núm. 35 (2005), pp. 375-381 recibido 29-04-2005; aceptado 01-06-2005

Designing an active learning based system for corpus annotation

Bertjan Busser
Tilburg University
The Netherlands
g.j.busser@uvt.nl

Roser Morante
Tilburg University
The Netherlands
r.morante@uvt.nl

Resumen: En este artículo revisamos algunos resultados de experimentos realizados con técnicas de Active Learning para establecer las bases del diseño de un sistema de anotación de corpus. A partir de los datos experimentales diseñamos un sistema modular que posibilita un aprendizaje rápido en una primera fase y que permite pasar a una segunda fase de aprendizaje más lento, pero más preciso. El sistema está diseñado para realizar una tarea de anotación de roles semánticos.
Palabras clave: aprendizaje activo, anotación de corpus, etiquetado de roles semánticos

Abstract: In this paper we review some Active Learning experimental results in order to set up the basis for designing an active learning based system for corpus annotation. Based on the experimental data we design a modular system that allows for initially learning fast, but that it is capable of switching to a slower and more precise learning strategy. The system is designed to perform a semantic role labelling task.
Keywords: active learning, corpus annotation, semantic role labelling

1 Introduction

In this paper we review some experimental evidence, as a starting point to propose a design for an Active Learning (AL) system for automatic Semantic Role Labelling (SRL) on a Spanish corpus.

AL is an appropriate method for annotating large amounts of raw data, capable of functioning within the constraints that a task like labelling a corpus for SRL imposes, such as a limited availability of labelled data and the cost of expert human annotators.

The data we present show that there might be a trade off between learning faster and learning better. Accordingly, we design a modular system that allows for initially learning fast but that it is capable of switching to a slower and more precise learning strategy.

In section 2 we introduce AL. In section 3 we describe the constraints given the task to annotate a large corpus with minimal initial annotation. In section 4 we review some experimental results. Finally in section 5 we propose a design for a system to annotate a large corpus using AL.

2 Active Learning

AL (Thompson, Califf, and Mooney, 1999) is a technique that uses example selection methods for algorithm training. It can be used with different algorithms. A main characteristic of AL for corpus annotation is that it allows to start from a small annotated corpus, as opposed to traditional machine learning of natural language processing (NLP), which typically requires large amounts of annotated data to learn a task adequately.

Another important characteristic is that AL allows for human intervention at any time during the learning process, whereas the traditional approach is a batch type process, which does not allow for influencing the learning process and thus the quality of the annotation.

2.1 General description of Active Learning

AL essentially is providing Machine Learning with a feedback loop. The feedback loop is used to control the learning of the main learning algorithm. The two most straightforward applications are semi-automatic selection of examples to label, and reduction (or compression) of the initial dataset.

In both methods, dataset reduction and selection of examples for labelling, a decision about the utility of the current example is made based on the information the learning algorithm has already learned, as well as the characteristics of the current example.

The AL process is an iterative process, which typically starts out by training the learning algorithm on a labelled bootstrap corpus. Ideally, the labelled bootstrap cor-

ISSN: 1135-5948

pus is as small as possible, often only a few examples. After the initial training, the AL process is applied to each of the remaining examples in turn. Ideally, if an example is accepted for further processing, this example is labelled (in the case of selecting examples for labelling), added to the bootstrap corpus, and the learning algorithm is retrained with the enhanced bootstrap corpus, before processing further examples. This optimal case is often very computationally intensive. It is therefore more common to retrain after a fixed number of processed or accepted examples.

The general AL algorithm is independent of the learning algorithm and example selection criteria used. It can, in principle, be combined with any learning algorithm and selection criterium, and thus allow for maximum optimization for specific tasks.

2.2 Learning strategies

AL can be combined with any learning strategy. It is independent of the learning algorithm(s) used, the parameter settings, the data characteristics (features), and even the number of classifiers. A classifier is a learning algorithm with a specific parameter setting and trained on specific data.

Using more than one classifier, also known as Query By Committee (Melville and Mooney, 2004), is a good way to avoid weaknesses of individual classifiers. A Committee usually consists of an odd number of classifiers to avoid tie breaking problems, and those classifiers should be as diverse as possible.

Another approach using multiple classifiers, and possible to combine with AL, are agents. Agents are classifiers for a specific function, such as disambiguating one specific word (Word Sense Disambiguation) (Hendrickx and van den Bosch, 2001).

In choosing a learning algorithm for a practical AL task there are also practical considerations. An algorithm that is capable of incremental learning is preferable for AL, but not essential. An algorithm that needs disproportionally long training times may not be practical.

2.3 Example selection

There are many possible selection criteria, but the actual choice is often limited by the learning algorithm used. For instance, an often mentioned criterium is the *confidence score*. The confidence score of an example is the probability that the labelling of the current example is correct. However, if the used algorithm does not provide enough information to determine the confidence score this criterium is not usable.

Another often used criterium is agreement, the proportion or number of individual classifiers that agree on a certain labelling of the current example. This criterium applies to Query By Committee.

Universally applicable criteria are rare; random selection is one.

3 Task based Constraints

Large amounts of raw linguistic data are available. However, these data are not immediately useful for most purposes. Annotated data, on the other hand, are scarce and expensive. Having annotated corpora is essential for all NLP applications. The annotation task that we will focus on is a SRL task, for a large corpus (described in 3.1). Semantic tagging of corpus is relevant for information extraction, automatic summarization, question-answering systems, and for all applications that need semantic interpretation.

SRL consists in recognizing the arguments of the verbs in a sentence and assigning semantic roles to them. For every verb all the constituents in the sentence that have a semantic role as argument (Agent, Patient, Instrument, etc.), or as adjunct (Locative, Temporal, Manner, Cause, etc.) are labelled. Most of the existing SRL systems carry out the task in two stages: (1) Recognition of arguments: it consists in analyzing the sentence syntactically in order to define the limits of the constituents that will be arguments of the verb. This is how the information to train classifiers is obtained. This stage requires preprocessing the text. (2) Labelling: classifier algorithms are used to assign roles automatically. These algorithms need training that will be carried out with annotated corpora.

The reference for a SRL task are the results of the CoNLL-2004 Shared Task (Carreras and Màrquez, 2004), in which different machine systems compete to perform a preestablished SRL task using an English corpus. As (Carreras and Màrquez, 2004) put it, for English, different machine learning models have been applied: purely probabilis-

tic models, maximum entropy models, generative models, decision trees, support vector machines, memory based learning, voted perceptrons. The results obtained by these models are not precise enough so as to apply them to real tasks, like the labelling of a 70 million word corpus. The highest result in the CoNLL-2004 Shared Task (Carreras and Màrquez, 2004) was a F_1 rate of 71.72 in the development set, and of 69.49 in the test set, obtained by (Hacioglu et al., 2004).

3.1 Corpus description

The corpus to which the techniques will be applied is the EFE corpus of Spanish, of 70.082.709 words, that contains the news of the EFE agency of news of the year 1994. The corpus is made available by the research group TALP of the Universitat Politècnica de Catalunya (http://www.talp.upc.es/).

4 Review of experimental results on example selection

In this section we will review experimental results on example selection from various sources. Example selection has been, and is, of serious interest for the machine learning community, since correctly distinguishing relevant material from irrelevant material for learning is essential for effective learning. Consequently, there is a large body of literature related to this topic. We will focus on a few studies which compare selection methods on closely related tasks.

4.1 Data editing

Daelemans and Van Den Bosch (Daelemans and den Bosch, 2005) experiment with CPS, case prediction strength, and random selection in editing data on three different NLP tasks. Data editing in this case is example selection. CPS is an information measure, closely related to entropy and information gain. It measures the importance of an individual example, the case, for predicting the right answer in a group of similar examples. The authors apply data editing on three tasks: GPLURAL, generating the German plural from a German singular noun; DIMIN, generating Dutch diminutives from a Dutch noun; and PP, attachment of a prepositional phrase to either a noun or a verb phrase in English.

The procedure applied is ranking the training data in three different ways: from low to high CPS, from high to low CPS, and randomly. The learning algorithm, in this case IB1 with MVDM, is then applied to the first 10 %, 20 %, 30 %, ... , 100 % of each ranked dataset, and tested against a constant, held-out test set. The results of these series of experiments are displayed in Figure 1.

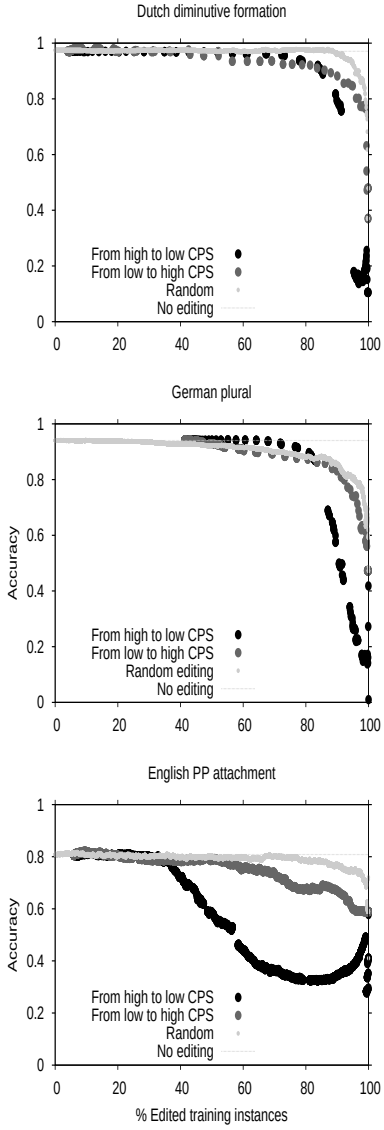

Figure 1: From Daelemans and Van Den Bosch (Daelemans and den Bosch, 2005). Generalization accuracies by IB1 on GPLURAL, DIMIN, and PP with incremental percentages of edited example tokens, according to the CPS editing criterion, from high to low CPS and vice versa, and using random incremental editing.

Summarizing these graphs, we note that in all cases random selection gives the steepest learning curves. In these graphs a 100 % edited training instances means 0 % trai-

ning data, or, in other words, the beginning of learning. The progress of the learning process should be read from right to left. The light gray line represents random selection, and starts out at the highest level in the extreme right of all graphs. Also, initially, it increases the fastest in all graphs.

4.2 Active Learning topic agents

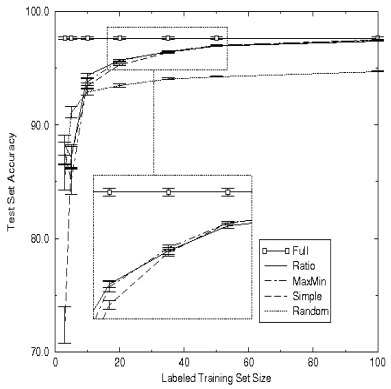

Figure 2: From Tong and Koller (Tong and Koller, 2001). Average test set accuracy over the ten most frequently occurring topics when using a pool of 1000.

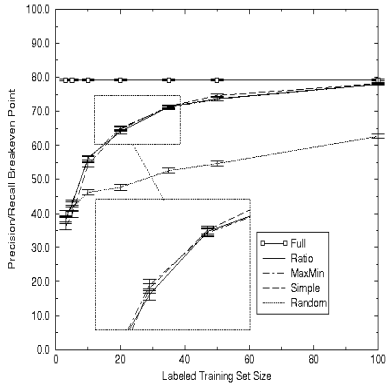

Figure 3: From Tong and Koller (Tong and Koller, 2001). Average test set precision/recall breakeven point over the ten most frequently occurring topics when using a pool of 1000.

Tong and Koller (Tong and Koller, 2001) apply AL to training topic agents for automatically classifying the topic of newswire stories from the Reuters corpus. The agents are SVM-based classifiers trained on TFIDF-weighted word frequency vectors, and are binary classifiers (yes/no decision for a particular topic). The selection criteria used are Simple Margin, MaxMin Margin, and Ratio Margin, and random selection.

SVMs model the instance space in hyperplanes, where each hyperplane corresponds to a certain decision, in this case a particular topic. The margin of this hyperplane is the distance from the center to the edge of the hyperplane. The center of this hyperplane might colloquially be interpreted as corresponding to the most prototypical instantiation of this topic.

Simple Margin is a selection criterium which minimizes the distance to a center, and therefore selects the examples which most closely correspond to the hypothetical prototype. MAXMIN MARGIN and RATIO MARGIN both minimize the growth of either of two opposing margins of the negative and positive hyperplane. They typically select examples lying between the two hyperplanes in order to more precisely define the borders of these, albeit by different criteria.

Summarizing these graphs, in Figure 2 random selection shows the steepest learning curve initially, when using accuracy as the performance measure. In Figure 3, using a precision/recall break-even point as performance measure, random selection is the worst selection measure.

4.3 Active Learning for reduction of data

In our experiments we have focused on reducing the data set size while keeping the performance constant. In Figure 4 we present a graph showing results on Dutch grapheme phoneme conversion, based on the TreeTalk-D system (Busser, 1998). We have applied various selection criteria on this data, including IB2, and two variants of our own implementation of AL.

IB2 is an ancestor of AL, which attempts data set reduction by first classifying each labelled example and only learning it if it was classified wrong. Our baseline variant does the same, but selects each example randomly from the whole set, and removes examples from the data set only when they are used for learning.

Our random variant is a combination of AL and a form of co–training (Blum and Mitchell, 1998). It uses a manually labelled test set to evaluate the quality of an automatically generated labelling. Like in the previous technique, it selects a random example from the data set to learn, but it tests performance before and after learning it on a random test

set of a 1000 examples. This random test set is generated separately for each example. If performance increases after learning, the example is removed from the data set, if not, it is unlearned. It is also much more computationally expensive than the other two methods, so we were not able to complete the full curve.

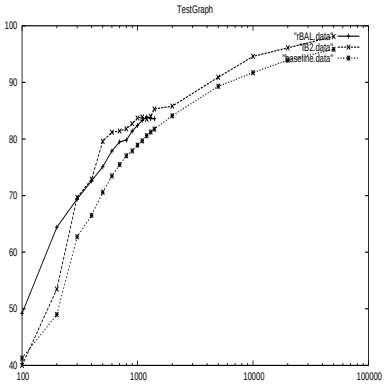

Figure 4: Active Learning for data set reduction in grapheme phoneme conversion.

Summarizing these results, our random variant has a much more irregular curve than the other two. Based on this graph we would have to conclude that IB2 shows the best result, but the random variant might outperform it given enough resources.

In other unpublished experiments we applied QBC with various selection criteria such as majority voting, thresholded majority voting, and thresholded entropy, but results in these experiments were singularly unpromising.

4.4 Discussion

While there are many different selection criteria (we reviewed only a few of these), we propose to draw some conclusions which apply strictly to annotating a large corpus. The constraints in this task make this possible. Generally speaking, there are three constraints from a methodological perspective. The first, and most important constraint is the limited amount of hand-labelled, "gold standard" data available. A second constraint is that the resulting system should attain reasonable performance levels as quick as possible, both for practical reasons and for further experimentation. A third constraint is that in the resulting system it should ideally be possible to plug in various learning algorithms to facilitate experimentation; it is very hard to predict which algorithm is optimal for a certain task.

Furthermore, the task at hand, SRL, has two characteristics which function as constraints. SRL is usually approached as a two phase process, (1) isolating relevant arguments, and (2) categorizing those arguments, which implies that the material to be hand labelled must be carefully selected, taking into account the learning that will be applied; if agents are selected as the learning strategy then the hand-labelled corpus has to contain certain minimum numbers of hand-labelled examples for each agent to be able to attain statistical usability. A second consideration is that SRL usually is done on a corpus labelled with Part-Of-Speech (POS) tags, and constituent tags (chunks). Since this labelling will have to be automated, the system will have to be able to (learn to) deal with errors in the POS tags and chunks.

The limited availability of hand-labelled data rules out bootstrapping the system with a relatively large amount of reliable data in order to attain reasonable performance immediately. Most likely it will also rule out large scale co-training, since this needs reasonable amounts of labelled data to initially train on, and reasonable amounts of labelled data to evaluate the system with. However, it will probably be possible to have one or a small number of small in-the-loop evaluation corpora, and a reasonable held-out test set to evaluate the system with.

We reviewed two sources that showed that random selection facilitates rapid learning of a system, and random selection is also truly independent of the algorithm used. However, based on this literature, caution is necessary, because different measurements lead to different results. Tong and Koller (Tong and Koller, 2001) show that when using an F-score related performance measure, random selection does not necessary learn either the fastest or the best. This might be caused by the characteristics of their algorithm and selection criteria, or by their task, or both. Informed selection criteria, such as MaxMin Margin or CPS, represent introducing a certain bias to the system, distorting the example space. The effects of this bias may or may not result in a positive effect on performance, depending on a variety of factors, including the algorithm used and the specific task. Therefore it seems advisable to use several performance measures to be able to

keep track of these effects.

The utility of examples, or in-the-loop performance, should ideally be measured on an independent corpus and not as internal consistency or inter-concept coherence based on the already learned concept representation because, at least initially, the already learned concept representation might not be very accurate. Furthermore, an independent in-the-loop evaluation corpus allows for using performance measure that require a minimum number of observations to be statistically significant, such as F-score related measures.

5 A system for corpus annotation

Given the conclusions we reach in the previous section, in this section we propose a modular system taking into account the task characteristics, in particular the small amount of labelled data available. The system will initially be geared for fast learning of the task to allow for further experimentation. Switching learning strategy will be possible using the modularity of the system.

5.1 Corpus and general system

To start with, the corpus has to be divided into three subcorpora:

- "Gold Standard" corpus: the part of the corpus manually annotated. It will have approximately 200.000 words.
  - Evaluation Corpus: this is the part of the corpus which will be used to evaluate the system with (EC). In the machine learning literature this would be the held-out test set. We will use a 100.000 words corpus for this purpose.
  - In-The-Loop Evaluation Corpus (ITL-EC): it is the part of the corpus in which the results of every iteration are evaluated, which should also be manually annotated. It does not necessarily have to be very large, but it is desirable that it is a valid random sample of the total example space to make accurate measurements. We might use various randomly selected parts of this corpus for different evaluations in the process. As an upper bound we will use maximally 50.000 words for this purpose.
  - Seed Corpus: this 50.000 word part of the "Gold Standard" corpus will be used to initially bootstrap the system.
- To-be-annotated corpus (AC): the part of the corpus that will be semiautomatically annotated. It will be divided into subparts of, for example 1 million words, to facilitate processing.

Active learning is an iterative process. In the first iteration the classifier is trained with the SC. Next the classifier classifies a subpart of the AC. As a result, information will be obtained about what examples the system can automatically annotate and what examples the system cannot annotate.

It is necessary to establish a methodology to determine what is a classifiable and an unclassifiable example. We will use the In-The-Loop Evaluation Corpus to isolate productive examples, that is those examples, classified automatically, with which the system classifies more, or at least not less, examples in the In-The-Loop Evaluation Corpus correctly, and take those examples to be classifiable correctly.

The examples that the system cannot classify productively will be annotated by a human and incorporated to the SC. As a check we will also manually review a small random sample of the examples that the system can classify productively before entering them in the SC for the next iteration.

The EC is used in order to evaluate if in every iteration the classification results improve. This might be used as a stop criterium; if the performance does not increase anymore, the AL process can be stopped. If the performance is acceptable, the system might be switched to fully automatic classification. If it is not acceptable at that point, it is possible to switch strategies, for example by using a different algorithm or selection criterium, and continue the AL process.

5.2 System details

Initially we will start measuring accuracy and an F-score related measure. Since not all algorithms support manipulating decision threshold (see (Tong and Koller, 2001)), we will use either traditional F-score with $\beta = 1$ or true positive / false positive rate (Fawcett, 2005) which also allows for finding a break-

even point between them like (Tong and Koller, 2001).

Initially we will use random selection due to the indications that it facilitates fast learning as well as being algorithm independent. However, we will do this exclusively on the productive, automatically classified examples, and always select all the manually labelled examples (see section 5.1). If the learning rate of the system drops, according to the evaluation per iteration, we can switch to another criterium such as a productivity threshold.

All of the knowledge and most of the code necessary to implement such a system using KNN, is already available for Timbl (see http://ilk.uvt.nl). We will therefore start with the IB1 algorithm (pure KNN) in the Timbl software package. We will explore the possibility of integrating SVMs (Support Vector Machines) in this system early in this project.

In all aspects we will keep the system sufficiently modular so that we can plug in other performance measures, selection criteria, and algorithms as needed.

Acknowledgements

We would like to thank Antal van den Bosch and Walter Daelemans for kindly providing the graphics and data in Figure 1, and Simon Tong and Daphne Koller for the graphics in Figures 2 and 3. We also would like to thank M. Antònia Martí and CLiC for their support with the Spanish data. Finally, we thank three anonymous reviewers for their comments.

References

Blum, A. and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*. Morgan Kaufmann Publishers.

Busser, G.J. 1998. Treetalk-d: A machine learning approach to dutch word pronunciation. In P. Sojka, V. Matousek, K. Pala, and I. Kopecek, editors, *Proceedings TSD Conferenc*, pages 3–8, Czech Republic. Masaryk University.

Carreras, X. and Ll. Màrquez. 2004. Introduction to the conll-2004 shared task: Semantic role labeling. In *Proceedings of the CoNLL Shared Task 2004*, Boston MA, USA.

Daelemans, W. and A. Van den Bosch. 2005. *Memory-based language processing*. Cambridge University Press, Cambridge, UK.

Fawcett, T. 2005. Roc graphs with instance varying costs. submitted to. *Pattern Recognition Letters Special Issue on ROC Analysis in Pattern Recognition*.

Hacioglu, K., S. Pradhan, J.H. Martin, and D. Jurafsy. 2004. Semantic role labeling by tagging syntactic chunks. In *Proceedings of the CoNLL Shared Task 2004*, Boston MA, USA.

Hendrickx, I. and A. van den Bosch. 2001. Dutch word sense disambiguation: Data and preliminary results. In *Proceedings of Senseval-2*, Toulouse.

Melville, P. and R. J. Mooney. 2004. Diverse ensembles for active learning. In *Proceedings of the 21st International Conference on Machine Learning*, pages 584–591, Banff, Canada.

Thompson, C. A., M.E. Califf, and R.J. Mooney. 1999. Active learning for natural language parsing and information extraction. In *Proceedings of the 16th International Conference on Machine Learning*, pages 406–414.

Tong, S. and D. Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66.