CrowdPillar: Improving Machine Learning By Leveraging Support from the Crowd

Sean Goldberg Sunny Khatri

December 13, 2011

Abstract

We propose a new method for combining machine learning with crowdsourcing. Previous work has attempted to partition the input test data randomly into components to be performed separately by humans and machines to optimize some cost-quality-speed metric. By considering features concerning the entropy of the machine learning output, we can better decide which examples should be kept as correct and which should be sent to the crowd to be fixed. After a review of conditional random fields, entropy, and crowdsourcing, we motivate our sequence selection algorithm and discuss its integration into Amazon Mechnical Turks.

1 Introduction

1.1 Motivation

The current state of the web is changing drastically as more and more data is continuously uploaded and made available to the general public. Media such as books, newspaper articles, and scientific papers that were primarily available in print form can now be found digitized on the world wide web. In addition, the ease of web publication versus traditional publication has increased both the total amount of such media and the speed by which it is made accessible. The format of the majority of the world wide web is through an unstructured collection of text. Organizing this wealth of new information has become a priority and is being led by efforts in Information Extraction, which is the automatic withdrawal of relevant information from a collection of unstructured text, thus allowing it to be cast into a relational form. Such a form of the data allows it to be searched, queried, and analyzed in a more efficient fashion while pure unstructured text allows little more than simple keyword search.

Different tasks are associated with the umbrella concept of Information Extraction. Named Entity Recognition concerns itself with extracting the specific entities such as persons or organizations from the text, even if these entities are referenced in different ways. Coreference Resolution and Relationship Extraction try to find links between these entities. Terminology extraction for finding the relevant terms in a given corpus is also a popular current field of research. There are many machine learning models that have been used for these tasks. Hand-written regular expressions are perhaps the simplest (and least accurate) and try to analyze the grammar of the text and parse entities accordingly. Classifiers such as Naive Bayes and Maximum Entropy Discriminative Models (like Logistic Regression) use a training set to learn which words correspond to which entities and reapply the same rules when inferring upon new data. Given the fact that most words within a text occur in context with surrounding words, the most successful models have been sequence models such as Hidden Markov Models and Conditional Random Fields.

Despite the success of many machine learning models, there is a still a limit to how well they can perform. Most models can only give a probabilistic output that the extracted information corresponds to the true underlying interpretation of that information. The output with max probability is taken as the machines decision. In this paper we consider whether it is possible to improve the confidence in the output decisions by harnessing the most complex machine that currently exists: the human mind itself. Humans can perform information extraction tasks relatively simply compared to their machine counterparts.

While theyre able to achieve a much greater accuracy for these types of tasks, they trade that with being much slower and more costly. Recent startups such as Amazon Mechanical Turks and Crowdflower has made it easier to take advantage of a large collaborative group of people for very cheap, also known as Crowdsourcing.

Despite being quicker and cheaper, crowdsourcing of a task still pales in comparison to the cost and speed of a machine handling the task, especially when there are many examples that most models can handle exceptionally well. The ideal scenario would occur if there was an efficient way to utilize a machine model for tasks that it can do well, and crowdsource the rest to achieve the highest accuracy at the best possible tradeoff of cost, speed, and accuracy.

In this paper, we motivate the idea that the machine gets most incorrect those sequences which it is most uncertain about and introduce features associated with the entropy of the output as a means for quantifying this uncertainty. It will be shown that by asking questions to the crowd based on a small subset of the total testing set we can find and correct most of the parts that have been labeled incorrectly.

This paper is organized as follows. The remainder of the introduction describes the information extraction task of labeling on bibliographic citations and our system design for increasing accuracy. Section 2 introduces the basic components of the system: Conditional Random Fields (CRFs), the specific machine learning model used for our experiments, our entropy model for quantifying the uncertainty of the CRF output, and the crowdsourcing mechanism used to improve results. We discuss the experiments that led us to our entropy model in Section 3. Section 4 outlines how we plan to interface with Amazon Mechanical Turks and how different question layouts and formats affect performance. Sections 5 and 6 conclude with a formal conclusion and future work respectively.

1.2 Bibliographic Labeling Task

The specific Information Extraction problem addressed is that of labeling a set of tokens in a corpus of small sequences, such as addresses or bibliographies. See FIGURE 1 for an example. We effectively want to partition the data into segments according to a label corresponding to the type that data corresponds to.

The ability to automatically extract names, streets, and cities in address or authors, titles, and conferences in bibliographies allows the information to be organized in such a way that makes advanced querying on the data possible. This is a difficult problem because context becomes very important if the text is completely unstructured. Sequence models are primarily used in token labeling and we discuss how to apply CRFs to the problem in the next section.

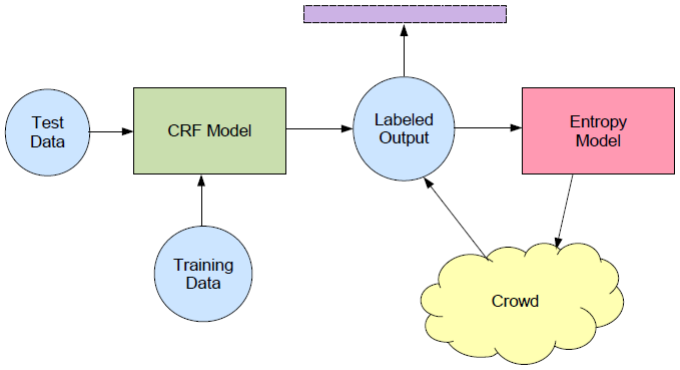

Figure 1: The CrowdPillar system design.

The experiments in this paper are taken over a corpus of 433 bibliographies drawn from the DBLP data set. All bibliographic examples come from this data set as well.

When we discuss accuracy throughout the paper, we are referring to both the precision and recall of the labeling of a sequence. Precision is the ratio of all correct labels to all labels of a particular value. Recall is the ratio of correct labels to the total number of tokens labeled. Since every token is labeled in our experiments, these two values are the same and we reference this single value as accuracy.

The current mean CRF accuracy over the DBLP data set is 90%. The rest of this paper will be focused on methods to increase this number in an efficient and cost-effective way. In the next section, we outline a system designed to do so.

1.3 System Design

See Figure 1 for our proposal of the CrowdPillar system, which merges human(crowdsourcing) and machine (con-

ditional random fields) computation. The CRF is trained on a specific set of data utilizing any number of optimized techniques. How the system is trained is irrelevant to CrowdPillar. The data to be tested is sent through the model to produce some initial output, the entirety of which is passed to our entropy model to select certain sequences or parts of certain sequences to formulate as questions to the crowd. When the answers are received, they are used to modify test data to be clamped to certain truth values designated by the crowd. The data is sent through once more and refined. This can be repeated multiple times until some convergence criterion or accuracy is met.

For a detailed discussion of each one of system components. Please see the Appendix section.

2 Experiments

2.1 Comparing Uncertainty and Accuracy

To provide the best increase in accuracy on the subset of sequences we form into questions, it becomes crucial to pass the most inaccurate ones. The question becomes how to estimate the accuracy of a sequence based entirely on its associated entropy. (FIGURE) is a double histogram containing tokens binned by their entropy value and distinguished by correctness. It becomes clear that tokens labeled correctly are skewed towards low entropy values while those labeled incorrectly are distributed among the higher values. Thus by looking at the entropy of a token we can make a fairly good estimate of whether it was labeled correctly or not. (FIGURE) is a plot of every single token in every sequence of the DBLP data set along along with its entropy. Tokens marked red were correctly labeled by the CRF output and tokens marked blue were incorrectly labeled. Its from this graph that a clear threshold could be set to signify whether a node was considered high entropy or not such that the majority of these corresponded to incorrect nodes. We set this threshold T = 0.2.

To estimate the accuracy on a specific output sequence, we simply counted the fraction of high entropy nodes in that sequence. Indeed, when plotting the fraction of high entropy nodes with the accuracy as in (FIGURE), we see a clear relationship develop. The majority of high accuracy nodes are clustered at the bottom, while all the sequences with a fraction of high entropy above (—-) are scattered along all accuracy values. It are these sequences at the low accuracy values that we would ideally want to send to the crowd. We propose a threshold S = (—-), such that any sequences with a greater fraction of high entropy nodes would be sent to the crowd. These experiments were done on a single data set and the extent to which the thresholds are data dependent has not been determined yet.

2.2 Experiment Setup

(Implementation of CRF)

(Compare asking a question and receiving an answer at a node to clamping it)

(Discuss why we cant give the Tuker the whole sequence and why we want to constrain subset of nodes)

(Describe how we did clamping/Constrained Viterbi)

2.3 Experiment: Best Node to Pick

With the goal of correcting a sequence based resolving the value of one of its tokens, we must decide which token would indeed be the best one to query for an answer. Based on the previous discussion involving accuracy and entropy, the intuitive idea would be to find the token in a sequence with the highest marginal entropy and query that one. To restate, the highest marginal token is the token that, while all others are fixed to their most probable values, is still the one most uncertain. The alternative is just select a node at random. It was discovered in the course of our experiments that for the particular data set being used, there were many occurrences in which clamping a single node led to no increase in accuracy, as it either did not correct a previously incorrect value or by its addition reduced a different from being correct to being incorrect. This behavior is outlined by comparison with the ground truth in (FIGURE). In comparing random node selection versus the highest marginal, we kept only the sequences in which clamping produced some increase in accuracy, whether highest marginal or otherwise. This reduced the data set to a third of its previous size. For those remaining, (FIGURE) shows the discrepancy between asking a question in regards to a node randomly and specifically asking the one with the highest marginal entropy. We see that there are many more nodes with worse performance that the highest marginal entropy, such that a random selection is much more likely to perform worse. It is important to note that the most uncertain node is not always the one whose correction yields the highest effect. There are

some sequences, especially when the number of high entropy nodes is small, that the machine is getting it wrong even though its certain of its answer, at least more certain than some of the other ones it may happen to guess correctly. In our future work, we would like ability to root out these incorrect nodes by means other than just high entropy. In the next section we discuss how the accuracy of individual sequences improved after the clamping process.

2.4 Experiment: Accuracy Increase Through Clamping

(FIGURE) shows the increase in accuracy attained for sequences based on the number of high entropy nodes they have. Its apparent that for sequences with a low number of high entropy nodes and high accuracy, we get a small, but still significant, improvement. The improvement becomes more drastic the greater the number of high entropy nodes (and thus more inaccurate) the sequence is.

(Explain what the graph says)

(Discuss trend that accuracy is improved the most the greater the number of high entropy nodes)

(Discuss why we may be able to get away with clamping more than three)

(Discuss how the three nodes to clamp were chosen: left and right of highest marginal)

(Compare increase with one node vs. increase with three)

(Talk about mean output accuracy before and after clamping)

3 Interface

3.1 XML Generation

(Discuss bridge needed between CRF output to AMT) The output from the CRF is taken and the entropies associated with the predicted output are calculated. Based on the calculated entropies, the required questions is posted in the form of a HIT to Amazon Mechanical Turks.

In order to create a HIT to be posted to AMT, the required sequence to be labelled along with the corresponding label information is required to be formatted in an XML file. Once a question is selected based on entropies, a corresponding XML file is generated and sent to AMT to be posted as a HIT.

(Questions generated from XML file according to prechosen question specification)

3.2 Question Formats

(Discuss trade offs for various question types) There are various ways in which a HIT can be presented to the turkers and that is a primary factor determining the success of a posted HIT in terms of getting correct responses from the turkers. A HIT should be presented in such a way so that it is easier for the turkers to respond and also it should not take much time on the part of the turkers to respond. Another important criteria is that the information presented to the turkers to answer a HIT should be as minimum as possible and still enough so that HIT could be responded properly provided the presented information.

For our task, Some of the ways that are considered are

/subsubsectionWhole Sequence Tagging In this the whole sequence is presented to the turkers to label. With each sequence node a color is associated representing the nodes uncertainty level - red (highly uncertain) yellow (not highly certain) and green (highly certain). Turkers will be allowed to label not only the highly uncertain nodes but also other nodes if they are incorrect according to the turkers. Possible labels can be presented via user interface using radiobuttons or dropdown lists.

/subsubsectionPartial Sequence Tagging Despite of asking the turkers to label the whole sequence, we could provide the turkers with a partial sequence containing the high uncertainty nodes. Although a partial sequence intents to provide the turkers with the minimal information required to label the nodes but might not be true always.

(Discuss why question choice is important)

(Discuss what we want to study in terms of question choice for future work)

4 Future Work

This paper outlines a work in progress and represents partial accomplishment to the finished system outlined earlier on. There are still a number of topics that need to be considered in addition to the final implementation itself.

We were only able to attain a single data set at the time of this writing and one priority to further refine our results on new data. Some of the accuracy improvement is difficult to judge as the CRF was able to attain a high mean accuracy even without the addition of crowdsourcing. Running on far less accurate data may lead to even

better results than described here.

While we showed that asking a question pertaining to the highest marginal nodes were better than random, we would like to consider ways to do even better than marginal. If the sequences are composed of dependent and independent nodes of different connectivity, a scheme for picking out the most connected nodes is desirable. Asking questions in a query-driven manner based on how often a sequence is accessed is also a promising avenue of research.

Finally, the system itself needs to be built, including the bridge between CRF output to XML generation to AMT submission. We presented a number of question formats and it will be crucial to perform an analysis to determine the best ones for each data set.

5 Conclusion

Our results acknowledge the theoretical viability of automatically selecting low accuracy sequences based entirely on features associated with their entropy. We outlined a system by which we use this advantage to efficiently partition data to be solved by humans or computers in unison. We showed that the number of high entropy nodes in a sequence can act as a barometer as to which type of computation should be used to tag it. While this is more of a theoretical finding, we outlined the previous section what it will take to implement our proposed system as well as what factors concerning question design need to be studied further.

6 Appendix

6.1 Conditional Random Fields

Conditional Random Fields (CRFs) are a type of discriminative machine learning sequence model. Whereas Hidden Markov Models (HMMs) are modeled as a joint probability between an output sequence and an underlying hidden state sequence, CRFs directly generate the conditional probability of the hidden sequence given the output sequence without worry about modeling the observation state probabilities. The conditional probability of a total state sequence is expressed as a product of features associated with transitions and observations along the sequence:

(EQUATION 1)

where the represent parameters attached to edge and node potentials. Here we consider a only linear-chain CRFs whose edge potentials are pairwise.

Training a CRF corresponds to adjusting the parameters for each feature to determine how important it is to inferring a hidden state from an observation. Gradient descent is typically used although more complex optimized methods have become common.

A trained CRF can be used on an unlabeled sequence to output the most probable labeling. It does so by way of the same Viterbi algorithm used in HMMs. Viterbi is a dynamic programming algorithm which stores the max probable label transition based on all previous transitions at each step, working out the most probable label path when run over the entire sequence.

In addition to the most probable label sequence, its possible to calculate marginal probabilities for an individual sequence label given all other labels. This method is similar to the standard HMM forward-backward algorithm. (DISCUSS NEW EQUATION WITH M MATRICES)

6.2 Entropy

The origin of entropy dates back to the introduction of thermodynamics and statistical mechanics as a means of describing the total amount of disorder in a physical system. As it applies to information theory and probability theory, entropy is a measure of how uncertain a model is in its output.

For a random variable defined by a probability mass P, the formal definition of its entropy is given by:

(EQUATION)

The main insight to grab from the mathematics is that entropy is a measure of the distribution of possible values of the random variable. If the distribution is skewed towards a small subset of values, the entropy is considered to be low and the resulting confidence of predicting an output to be high. As the distribution becomes more evenly distributed, the entropy reaches a maximum where any prediction isnt any better than a random guess if all possibilities have the same probability.

While it isnt always the case, to a large degree when a machine learning model is forced to make a decision on predicting a random variable where certain values have high probability it gets those correct. When the distribution is more even and its forced to effectively make a guess, it is more likely to get those predictions wrong.

One concept that will become useful when we start considering individual tokens in a sequence is the marginal

entropy. When we make reference to marginal entropy, were referring to the entropy of the marginal distribution of a specific node in question.

If entropy can tell us, which sequences to select as being the most inaccurate, crowdsourcing can tell us how to resolve those sequences.

6.3 Crowdsourcing

CrowdSourcing is a distributed problem solving approach that has recently been very popular. It relies on the fact that a complex problem be better solved by many rather than just a single individual or a small group of individuals as some persons might be able to solve a particular problem efficiently while others can work on problems suitable to them in that way its an open call to community for problem solving. In crowdsourcing a task is broken down and is sent to the crowd to solve. There are various crowdsourcing services, for example, Amazon Mechanical Turks, Crowdflower, etc.. that allows posters to post small tasks and send the response back from the crowd, called the turkers, back to the task posters or generators. In addition for every task been responded the task posters are required to pay certain amount to the turkers for their services. At Amazon Mechanical Turks (AMT), a task is basically called as a HIT -Human Intelligence Task.

The use of Crowdsourcing has been mainly done for those problems that are difficult to be solved by machines, and also requiring high computations hence taking more time, and at the same time easy for humans to solve and where high level of accuracy is desired. Examples could be Given an image, describe the situation presented in the image with a couple of words, Given some contextual information determine whether the given two entities are same or not, etc..

Crowdsourcing provides us a hope of finding the best people suitable to solve a particular problem. One of the major disadvantages, or limitations, of crowdsourcing services is to verify the reliability of the responses provided by the turkers. Some turkers might randomly provide responses and hence a quality metric is required to filter out those random responses. There has been work done to quantify the quality of the turkers from the crowdsourcing services. For example Finin at. el. [1] presents an approach used to provide quality measures to the turkers based on their responses.

The other limitation of crowdsourcing services is we cannot expect responses immediately after a task has been posted. It might take some time to get responses from the turkers and hence might not be of much value is some task is required to be performed in real time and quickly.

The field of machine learning provides an ample opportunity to use crowdsourcing services. The work done and the results produced in machine learning always have some degree of uncertainty associated with them due to which they are bound to provide incorrect results. The uncertainty could be due to many reasons but mainly could be because of insufficient or bad quality training data used to train the learning algorithm. Thus it gives an opportunity to involve human intelligence via crowdsourcing in machine learning to lower down that uncertainty and provide more accurate results.