Introduction

Have you ever imagined using Twitter to help dealing with natural disasters? Today, it is possible. Artificial intelligence combined with social media seems to have no limitations, and I do research in Machine Learning.

What is Machine Learning?

Machine learning is the subfield of Artificial intelligence that gives computers the ability to learn without being explicitly programmed, as it was defined by Arthur Samuel - the American pioneer in the field of computer gaming and artificial intelligence who was actually born in Emporia, Kansas and attended College of Emporia.

Examples of Machine Learning Tasks

Why do we need Machine Learning?

We cannot program everything, and some tasks are difficult to define algorithmically.

Terminology

Supervised Machine Learning is focused on building predictive models given labeled training

data.

Tennis example.

Instances

Labels/Class

Data may come from a variety of sources, for instance, social media networks.

In my research, I use Twitter data, specifically, user-generated tweets about disasters such as floods, hurricanes, terrorist attacks and others, to build specific programs called classifiers that could identify tweets about disasters, and help disaster management teams gather useful information in real time.

Machine Learning algorithms use feature based representations for instances, where each

instance is represented using a collection of features f1; f2; \_ \_ \_ ; fn. An instance is a single object of the world from which a model will be learned, or on which a model will be used (e.g., for prediction). In most machine learning work, instances are described

by feature vectors; some work uses more complex representations (e.g., containing

relations between instances or between parts of instances) [Kohavi and Provost, 1998]. For

example, an instance of the task "identify a given email as spam or non-spam" may be a

text of an email represented as bag-of-words [Mitchell, 1997]. The example of bag-of-words

representation is shown in Table 3.3, Chapter 3. In this work, we use the words "instance"

and "example" interchangeably.

Two major types of learning are distinguished: supervised and unsupervised learning.

In supervised learning the agent is given a training set of N examples, which could be

seen as input-output pairs (x1; y1); (x2; y2); \_ \_ \_ (xN; yN), where each yj was generated by an

unknown function y = f(x). The task is to discover a function h that approximates the true

function f [Russell and Norvig, 2009]. Identifying a given email as spam or non-spam is an

example of a supervised learning task since training a model requires labeled instances, i.e.

emails marked as spam and non-spam.

In unsupervised learning the agent learns patterns in the input even though no explicit

feedback is supplied [Russell and Norvig, 2009]; essentially, it means that only (x1); (x2); \_ \_ \_ (xN)

are provided. Categorizing news articles into topics such as politics, sports, entertainment,

etc. is an example of an unsupervised learning task since it requires \_nding similarity between

di\_erent news articles and clustering them together, with no prior labels provided.

A classi\_er, or a classi\_cation model, is de\_ned as a mapping from unlabeled instances to

(discrete) classes. Classi\_ers have a form (e.g., decision tree) plus an interpretation procedure

(including how to handle unknowns, etc.). Some classi\_ers also provide probability estimates

(scores), which can be thresholded to yield a discrete class decision thereby taking into

account a utility function [Kohavi and Provost, 1998].

In this work, we use the combination of supervised and unsupervised learning methods.

We use supervised learning in the attempt to take advantage of labeled data, and we also

adopt unsupervised learning techniques to make use of unlabeled data.

Background on Disaster Management

1.3 Problem De\_nition and Approach Overview . . . . . . . . . . . . . . . . . . 4

3 Data Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

3.1 Original Dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

3.2 Data Preprocessing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

4 Single-source Domain Adaptation Approach . . . . . . . . . . . . . . . . . . . . . 13

4.1 Problem De\_nition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

4.2 Correlation Alignment Algorithm . . . . . . . . . . . . . . . . . . . . . . . . 14

4.3 Naive Bayes Classi\_er . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

4.4 Experiments and Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

4.4.1 Results for Supervised Learning from Source Data . . . . . . . . . . . 17

4.4.2 Results after Applying Feature Selection . . . . . . . . . . . . . . . . 19

5 Multi-source Domain Adaptation Approach . . . . . . . . . . . . . . . . . . . . . 30

5.1 Problem De\_nition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

5.2 Multi-source Domain Adaptation Algorithm . . . . . . . . . . . . . . . . . . 31

5.3 Experiments and Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

6 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

7 Future Work . . . . . . . . . . . . . . . . .

-----------------------------------------------

A supervised classifier trained on data (training data) from a particular domain (i.e.

disaster) is expected to give accurate predictions on unseen data (testing data) from the

same domain, assuming that the training and test data have similar characteristics. However, labeled data is not easily available for a current target disaster.

However, labeled data from a prior source disaster is presumably available, and can be

used to learn a supervised classifier for the target disaster.

So the idea is to use tweets about previous disasters to train the classifier to identify tweets about a new current on-going disaster for which we don’t have much data right away.

Unfortunately, the source disaster data and the target disaster data may not share the

same characteristics, and the classifier learned from the source may not perform well on

the target. Domain adaptation techniques, which use unlabeled target data in addition to

labeled source data, can be used to address this problem.

We study single-source and multi-source domain adaptation techniques, using Nave Bayes

classifier.

Experimental results on Twitter datasets corresponding to six disasters show that domain

adaptation techniques improve the overall performance as compared to basic supervised

learning classifiers.

Domain adaptation is crucial for many machine learning applications, as it enables the

use of unlabeled data in domains where labeled data is not available.

-----------------

The idea is to use tweets about previous disasters to train the classifier to identify tweets about a new current on-going disaster for which we don’t have much data right away.

However, tweets are very ambiguous because it is human language. For instance, the hashtag #Matthew might refer to the Hurricane Matthew as well as to the Apostle Matthew.

In addition, the classifier trained to identify tweets about “hurricanes” may not perform well identifying tweets about “wildfire”.

The reason for this is because Machine learning is very different from human learning. Humans are able to learn from very few examples and apply the learned knowledge in novel conditions. In contrast, machine learning methods only perform well when given much data.

So I have to apply domain adaptation techniques to adapt the classifier trained on one disaster to perform well on another.

Experimental results show that various domain adaptation techniques improve the overall performance of the classifier.

Thus, domain adaptation becomes essential as it enables the classifier perform well on domains where data is limited or unavailable.

And imagine, next time you are posting about a storm coming up, you might be helping save someone’s life!