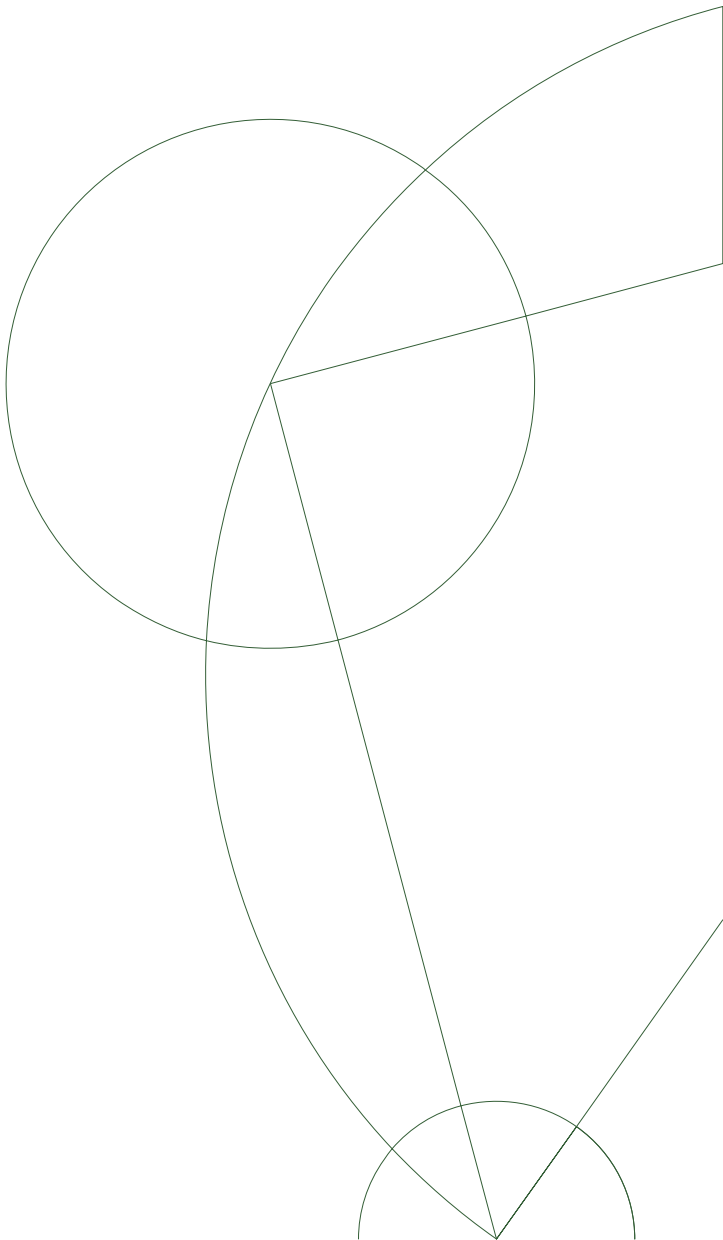
FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

# Master Thesis

Sune Andreas Dybro Debel

# Deep Multi-Task Learning

For Relation Extraction



Dirk Hovy

March 2, 2017

**Abstract**

*Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.*

# Contents

# Part 1

# Introduction

## 1.1 Motivation

## 1.2 Problem Statement

# Part 2

# Background

In this part, we describe the information extraction problem and the challenges it poses. Moreover, we formally describe the supervised machine learning. Specifically, we discuss the challenges of noise and overfitting, and show the usefulness of Vapnik-Chervonenkis analysis.

## 2.1 Information Extraction

In natural language processing, information extraction is the problem of extracting structured information from unstructured text. Many practical information extraction problems fall in one of two categories: **named entity recognition**, or **relation extraction** (Jurafsky and Martin, 2009). Here, we introduce each of them, and explain why they are difficult.

### 2.1.1 Named Entity Recognition

A named entity is roughly anything that has a proper name. The task in named entity recognition is to label mentions of entities such as people, organisations or places occurring in natural language. As an example, consider the sentence:

Jim bought 300 shares of Acme Corp. in 2006.

A named entity recognition system designed to extract the entities *person* and *organisation* should ideally assign the labels:

$$[\text{Jim}]_{person} \text{ bought 300 shares of } [\text{Acme Corp.}]_{organisation} \text{ in 2006.}$$

This is a difficult problem because of two types of ambiguity. Firstly, two distinct entities may share the same name and category, such as *Francis Bacon* the painter and *Francis Bacon* the philosopher. Secondly, two distinct entities can have the same name, but belong to different categories such as *JFK* the former American president and *JFK* the airport near New York.

Named entity recognition can be framed as a sequence labelling problem. A common approach is to apply so called tokenisation to the text, i.e finding boundaries between words and punctuation, and associate each token with a label indicating which entity it belongs to. BIO (figure 2.1) is a widely used labelling scheme in which token labels indicate whether the token is at the **B**eginning, **I**nside, or **O**utside an entity mention.

| Jim | bought | 300 | shares | of | Acme | Corp | . | in | 2006 | . |
|---|---|---|---|---|---|---|---|---|---|---|
| B-PER | O | O | O | O | B-ORG | I-ORG | I-ORG | O | O | O |

*Figure 2.1: A sentence labeled with the BIO labels for named entity recognition.*

### 2.1.2 Relation Extraction

Relation extraction refers to the problem of identifying relationships such as *Family* or *Employment* between entities. As an example, consider the sentence:

Yesterday, New York based Foo Inc. announced their acquisition of Bar Corp.

Imagine we have designed a relation extraction system that recognises the relation *MergerBetween(organisation, organisation)* between two mentions of organisations. Ideally, we would like that system to extract the relation *MergerBetween(Foo Inc., Bar Corp.)* from the above sentence.

Because relation extraction is concerned with relationships between named entities, many systems that perform relation extraction applies named entity recognition first as a pre-processing step. This approach is sometimes called **pipelining**. An alternative to pipelining is **end-to-end** relation extraction, where relations and entities are extracted simultaneously. Pipeline approaches can suffer from the problem of **error propagation**, where the system erroneously assigns a label in the named entity recognition step, which later causes it to make an error in the relation extraction step.

### 2.1.3 Accuracy Measures

Information extraction systems are often evaluated empirically by applying them to collections of text, so called corpora, in which $N$ mentions of named entities or relations are known. Accuracy measures used in such tests are usually defined in terms of:

**True positives ($tp$)** The number of true named entities or relations correctly labeled by the system.

**True negatives ($tn$)** The number of true non-entities or non-relations correctly labeled by the system.

**False positives ($fp$)** The number of true non-entities or relations incorrectly labeled by the system.

**False negatives ($fn$)** The number of true named entities or relations incorrectly labeled by the system.

The distribution of labels used in both named entity recognition and relation extraction is often highly imbalanced. Consider for example the BIO labelling scheme in figure 2.1. Most words will be outside a mention of a named entity, and will have the label `O`. Using simple accuracy $\frac{tp+tn}{N}$ as a performance metric is therefore not very informative, since a useless system which labels all tokens with `O` would achieve high performance.

**Precision** and **recall** are more appropriate performance metrics for this reason. Precision $\frac{tp}{tp+fp}$ is the fraction of true named entities or relations of all named entities or relations that were extracted by the system. This is equal to 0 when none of the information extracted by the system was correct and 1 when all of it was correct.

Recall $\frac{tp}{tp+fn}$ is the fraction of true named entities or relations that were extracted by the system. This is 0 when none of the extracted information was correct, and

1 when all of the extracted information was correct, and no true named entities or relations were incorrectly labeled.

To get a single number that summarises the performance, precision $p$ and recall $r$ are often combined into a single metric, the $F1$ measure, defined as the harmonic mean of precision and recall $\frac{2pr}{p+r}$.

## 2.2 Supervised Machine Learning

Most modern solutions to the information extraction problems in 2.1 are based on supervised machine learning techniques. In this setting, a system learns to recognise the named entities or relations between them from examples provided by a human annotator. In this section we formally describe this approach and introduce important theoretical tools for understanding supervised machine learning.

### 2.2.1 The Supervised Learning Problem

A set $\mathcal{D}_{train}$ of $N$ training examples $(\mathbf{x}_i, \mathbf{y}_i)$ of inputs $\mathbf{x}_i$ and corresponding labels $\mathbf{y}_i$ is created by a human annotator. Each $\mathbf{x}_i$ belongs to an input space $\mathcal{X}$, for example the set of all english sentences. Each $\mathbf{y}_i$ belongs to a space $\mathcal{Y}$ of labels, for example the set of all sequences of BIO tags. As designers of the learning system, we specify a set $\mathcal{H}$ of functions $h : \mathcal{X} \mapsto \mathcal{Y}$. We want to find a $h \in \mathcal{H}$ that can automatically assign labels to a new set of un-labeled inputs $\mathcal{D}_{test} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{X}\}$ at some point in the future.

In supervised machine learning, we want to use an algorithm to learn a function $h$ from $\mathcal{D}_{train}$ that performs well on $\mathcal{D}_{test}$, as measured by some performance measure $e$. In classification problems such as named entity recognition or relation extraction where $\mathcal{Y}$ is discrete, we typically use binary error $e(\mathbf{y}_1, \mathbf{y_2}) = \mathbb{I}[\mathbf{y}_1 \neq \mathbf{y}_2]$. Importantly, we are not explicitly interested in the performance of $h$ on $\mathcal{D}_{train}$ (Abu-Mostafa et al., 2012).

We can formalise the preference for functions $h$ that perform well on examples outside of the training set with a quantity known as **generalisation error**.

**Definition 2.2.1** (generalisation error)**.** Let $P(\mathbf{x}, \mathbf{y})$ be a joint probability distribution over inputs $\mathbf{x} \in \mathcal{X}$ and labels $\mathbf{y} \in \mathcal{Y}$. Let $e(\mathbf{y}_1, \mathbf{y_2}) = \mathbb{I}[\mathbf{y}_1 \neq \mathbf{y}_2]$ be the binary error measure that measures agreement between labels $\mathbf{y}_1$ and $\mathbf{y}_2$. Then the generalisation error $E$ of a function $h : \mathcal{X} \mapsto \mathcal{Y}$ is defined as:

$$E(h) = \mathbb{E}_{\mathbf{x},\mathbf{y} \sim P(\mathbf{y},\mathbf{x})}[e(h(\mathbf{x}), \mathbf{y})]$$

Now, formally, the objective of supervised machine learning is to find a function $h$ in a space of functions $\mathcal{H}$ that minimises $E(h)$. We see the process generating the data as random, but with a behaviour describable by a distribution $P(\mathbf{x}, \mathbf{y})$. Unfortunately, this distribution is unknown. However, we can use sampled data $\mathcal{D} = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \sim P(\mathbf{x}, \mathbf{y})\}$ to estimate $E(h)$ with a quantity known as **empirical error**:

**Definition 2.2.2** (empirical error)**.** Let $\mathcal{D}$ be a set of $N$ examples $\{(\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i, \mathbf{y}_i \sim P(\mathbf{x}, \mathbf{y})\}$. Then the empirical error $\hat{E}$ is defined as:

$$\hat{E}(h, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} e(h(\mathbf{x}_i), \mathbf{y}_i)$$

Because $\mathcal{D}$ is a random quantity, it's dangerous to use $\hat{E}$ to estimate $E$. We risk that the samples are not representative of $P(\mathbf{x}, \mathbf{y})$, leading us to believe that $h$ is great,

when in fact it's terrible. If we assume that the samples in $\mathcal{D}$ are drawn independently from $P(\mathbf{x}, \mathbf{y})$, that is the choice of any one sample did not change the probabilities of the rest samples, and that $h$ is independent of $\mathcal{D}$, that is $h$ was not specifically chosen based on the sample, we can use **Hoeffding's inequality** to bound the risk that this happens:

**Theorem 2.2.1** (Hoeffding's inequality)**.** let $E(h)$ be defined as in definition 2.2.1, and let $E(h, \mathcal{D})$ be defined as in definition 2.2.2. Then:

$$\mathbb{P}\left(|E(h) - \hat{E}(h, \mathcal{D})| \geq \epsilon\right) \leq 2e^{-2N\epsilon^2}$$

The inequality tells us that the probability that $E$ is more than $\epsilon$ away from $\hat{E}$ decreases exponentially in $\epsilon$ and $N$. In other words, the more samples in $D$, the less likely it is that $E$ will be misleading.

Estimating $E$ with a sample that is independent of $h$ is a technique called validation. Because $\mathcal{D}_{train}$ is used to select $h$, it cannot be used to estimate $E$ by Hoeffding's inequality, and we need more sophisticated techniques to understand the relationship between $\mathcal{D}_{train}$ and $E$.

In validation, the sample provided by a human annotator is instead split into two datasets, $\mathcal{D}_{train}$, which we intend to use to choose a good $h$, and $\mathcal{D}_{validate}$, which is not used for selecting the model. Since $\mathcal{D}_{validate}$ is independent of $h$, Hoeffding's inequality applies and it can be used to estimate $E$.

The central question in supervised machine learning is *how can we best define $\mathcal{H}$ and use $\mathcal{D}_{train}$ to make $E$ small?* Answering this question is the objective of a field of research known as **statistical learning theory**.

### 2.2.2 Statistical Learning Theory

We would like to know how best to define $\mathcal{H}$ and use $\mathcal{D}_{train}$ in order to make $E$ small. $\mathcal{D}_{train}$ is the only information we have about $P(\mathbf{x}, \mathbf{y})$, and therefore also the only information we have about $E$. A straight forward idea would be to find a $h$ that minimises the **training error** $\hat{E}(h, \mathcal{D}_{train})$ under the assumption that this $h$ will also minimise $E$.

As we argued in section 2.2, using $\hat{E}$ to estimate $E$ can be misleading. Moreover, because $D_{train}$ is used to specifically choose $h$ that makes $\hat{E}$ small, the guarantees provided by Hoeffding's inequality no longer holds, and therefore it may be possible to select $h$ such that $\hat{E}(h, \mathcal{D}_{train})$ is small and $E$ is large, even when we have a large number of training examples.

The phenomena where training error is small but generalisation error is large is known as **overfitting**. As the name implies, it happens when $h$ is selected based on harmful idiosyncrasies of $\mathcal{D}_{train}$. The harmful idiosyncrasies of $\mathcal{D}_{train}$ can be seen as variation from one sample to another which is not a consequence of the natural variation expected from sampling from $P(\mathbf{x}, \mathbf{y})$, but rather the product of **noise**.

In general, noise comes in two forms. The first form is known as **stochastic noise**. This type of noise is introduced by non-determinism in the relationship between $\mathbf{x}$ and $\mathbf{y}$. If for example the text used as input to named entity recognition comes from scanned documents, stochastic noise may be introduced by scratches or dust. Clearly selecting $h$ that also uses these features of the data will not be beneficial for generalisation.

The second type of noise is called **deterministic noise**. This type of noise may be introduced when the relationship between $\mathbf{x}$ and $\mathbf{y}$ is deterministic, but even the best $h$ in $\mathcal{H}$ can't represent this deterministic relationship exactly. For a finite sample, the $h$ which fits the sample very well will therefore likely not generalise well.

The risk of overfitting is linked to the diversity of $\mathcal{H}$: The more diverse $\mathcal{H}$ is, the greater the risk that there exists a $h \in \mathcal{H}$ that will overfit $\mathcal{D}_{train}$.

A **dichotomy** is a central concept in measuring the diversity of $\mathcal{H}$. A dichotomy is a specific sequence of $N$ labels. For example, if $\mathcal{Y} = \{0, 1\}$, and $N = 3$, then (0 1 0) is a dichotomy, and so is (1 0 0). We have listed all dichotomies for $N = 3$ in figure **??**.

(0 0 0)
(1 0 0)
(0 1 0)
(0 0 1)
(1 1 0)
(0 1 1)
(1 0 1)
(1 1 1)

*Figure 2.2: All dichotomies for $\mathcal{Y} = \{0, 1\}$ and $N = 3$.*

Dichotomies allow us to group similar functions. There may be infinitely many functions in $\mathcal{H}$, but on a specific $\mathcal{D}_{train}$, many of them will produce the same dichotomy. This allows us to quantify the diversity of $\mathcal{H}$ in terms of the number of dichotomies it's able to realise on a set of $N$ points. This is achieved by a measure known as the **growth function**.

**Definition 2.2.3** (growth function)**.** Let $\mathcal{H}(N) = \{(h(\mathbf{x}_1), \ldots, h(\mathbf{x}_N)) \mid h \in \mathcal{H}, \mathbf{x}_i \in \mathcal{X}\}$ be the set of all dichotomies generated by $\mathcal{H}$ on $N$ points, and let $|\cdot|$ be the set cardinality function. Then the growth function $m$ is:

$$m(N, \mathcal{H}) = \max |\mathcal{H}(N)|$$

In words, the growth function measures the maximum number of dichotomies that are realisable by $\mathcal{H}$ on $N$ points. To compute $m(N, \mathcal{H})$, we consider any choice of $N$ points from the whole input space $\mathcal{X}$, select the set that realises the most dichotomies and count them.

The growth function allows us to account for redundancy in $\mathcal{H}$. If two functions $h \in \mathcal{H}$ and $g \in \mathcal{H}$ realise the same dichotomy on $\mathcal{D}$, then any statement based only on $\mathcal{D}$ will be either true or false for for both $h$ and $g$. This makes it possible to group the events $\hat{E}(h, \mathcal{D})$ *is far away from* $E(h)$ and $\hat{E}(g, \mathcal{D})$ *is far away from* $E(g)$, and thereby avoiding to overestimate the probability of either event occurring.

If $\mathcal{H}$ is infinite, the number of redundant functions in $\mathcal{H}$ will also be infinite, since the number of dichotomies on $N$ points is finite. If $m(N, \mathcal{H})$ is polynomial, the number of redundant functions in $\mathcal{H}$ will be so large as to make the probability that $\hat{E}$ is far away from $E$ very small.

This line of reasoning is the basis of the Vapnik-Chervonenkis bound, which bounds $E(h)$ in terms of $\hat{E}(h, \mathcal{D}_{train})$:

**Theorem 2.2.2** (Vapnik-Chervonenkis bound)**.** Let $m(N, \mathcal{H})$ be defined as in definition 2.2.3, $E(h)$ as 2.2.1, and $\hat{E}(h, \mathcal{D})$ as in 2.2.2. Then, with probability $1 - \delta$:

$$E(h) \leq \hat{E}(h, \mathcal{D}_{train}) + \sqrt{\frac{8}{N} \ln \frac{4m(2N, \mathcal{H})}{\delta}}$$

The term $\sqrt{\frac{8}{N} \ln \frac{4m(2N,\mathcal{H})}{\delta}}$ is the discrepancy between the training and generalisation error. It tells us that by increasing the number of training examples $N$, we can afford to use a more diverse $\mathcal{H}$, which will make it more likely that it contains a $h$

that fits $\mathcal{D}_{train}$ well, making $\hat{E}(h, \mathcal{D}_{train})$ small.

Moreover, it tells us that $E(h)$ will be close to $\hat{E}(h, \mathcal{D}_{train})$ if $m(N, \mathcal{H})$ is small. Intuitively, this tells us that a set $\mathcal{H}$ that contains simple functions, where by simple we mean functions that realise a small number of dichotomies, will make it easier to choose $h$ such that generalisation error will be close to training error.

On the other hand, having a set $\mathcal{H}$ that can realise a large number of dichotomies, will make it easier to find a function that will make $\hat{E}(h, \mathcal{D}_{train})$ small. Therefore, the optimal $\mathcal{H}$ must balance the tradeoff between $m$ and $\hat{E}$.

## 2.3 Summary

In this section we have seen that the purpose of named entity recognition is to identify mentions of entities such as people, organisations and places in natural language. The purpose of relation extraction systems is to identify relationships between them.

We have seen that simple accuracy is uninformative as an evaluation measure in information extraction, and described the alternative precision and recall.

We have described the formal setting of on supervised machine learning. We have discussed concepts such as overfitting and noise, diversity of the set of functions $\mathcal{H}$ from which to choose $h$, and its impact on training and generalisation error.

# Part 3

# Neural Networks

# Part 4

# Multi-Task Learning

## 4.1 Representation Learning

### 4.1.1 Auto Encoders

### 4.1.2 Word Embeddings

## 4.2 Deep Multi-Task Learning

# Part 5

# Experiment

## 5.1 Auxiliary Tasks

## 5.2 Network Architecture

# Part 6

# Results

# Part 7

# Discussion

# Part 8

# Conclusion

# Bibliography

Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data*. AMLbook.com, 2012.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Pearson Education, international edition, 2009.

# Appendix A

# Appendix

## A.1 First