eGovMon
eGovernment Monitoring
Project no.: Verdikt 183392/S10

Web-mining methods for detecting transparency

D4.1.1

Version: 1.0
Date: 2009-08-31
Author: Morten Goodwin Olsen
Dissemination Level: Public
Status: Final

Abstract:

In this deliverable we present an approach for measuring transparency of Norwegian municipality web sites using learning algorithms. As a proof of concept we have developed an algorithm for detecting the presence of mail records. This implementation detects mail records with an accuracy of 96.5% using only about 3 seconds per page.

Additionally, we have outlined initial ideas of how to implement algorithms for all eGovMon tests which rely on learning algorithms.

This document consists of 23 pages including this cover

Version control:

Version	***Status***	***Date***	***Change***	***Author***
0.1	DRAFT	2009-08-26	Initial Draft.	Morten Goodwin Olsen
0.2	RC	2009-08-30	Updated based on Comments from Annika Nietzio, Mikael Snaprud, Dagfinn Rømen and Lasse Berntzen.	Morten Goodwin Olsen
0.3	RC	2009-08-31	Minor updates from Annika Nietzio	Morten Goodwin Olsen
1.0	Final	2009-11-25	Finalized	Morten Goodwin Olsen

Table of Contents

1 Introduction

One of the key goals of the eGovernment Monitor (eGovMon) project is benchmarking and measuring transparency of local government and public agency web sites in Norway.

The eGovMon transparency indicators defined four modes for testing (D3.2.1):

- **Automatable (Deterministic):** Tests which can be implemented with formal deterministic algorithms such as existence of link to RSS-feed. These tests will always return a true/false answer.
- **Automatable (Learning Algorithm):** Tests which can not be be implemented by simple formal tests. By using learning algorithms, we can test for issues such as existence of mail records, contact information, central phone service, privacy statements, security statements, chat, forums etc. The results from these could either be used directly or as input to an expert who confirms or rejects the results. These tests will return a probability value (number between 0 and 1) specifying the confidence of the classification result.
- **Indirect:** Tests from which the result is not directly available on the web page as HTML, but could be indirectly known. Occasionally, information is only available in document formats eGovMon does not download, behind login etc. Such tests cannot currently be carried out automatically but requires human judgement. However, by detecting which software is used and from which vendor, known features from the software can be extracted.
- **Manual:** Tests which require human judgement either by expert or as surveys sent to the municipalities. Example of such tests would be to determine the existence and properties of administrative processes, how laws and regulations are carried out etc.

We will in this deliverable focus on tests which are Automatable through Learning Algorithms. Note that the architecture for transparency measurements is not part of this deliverable but is addressed in D4.2.1.

eGovMon defined 15 tests which rely on learning algorithms (D3.2.1). Note that D3.2.1 lists additional tests which are not part of this document. The are ether deterministic or manual and does not in any way rely on learning algorithms.

The tests which rely on learning algorithms are:

- **T.1.1:** Does the web site provide contact information of the organisation? If yes, what is the URL of the contact information page?
- **T.1.2:** Is the contact information complete?
- **T.2.1:** Does the web site provide a news page (or section of a page) with information?
- **T.2.3:** Does the web site provide an email newsletter?
- **T.3.1:** Does the web site provide information about the access to the mail record of the organisation? If yes, what it the URL?
- **T.3.2:** Does the web site provide access to the mail record of the organisation? If yes, what it the URL?
- **T.3.3:** Does the online mail records have a search function?
- **T.3.4:** Are the documents from the mail record available online?

- **T.4.1:** Does the web site provide information on the date and time of the next council meeting?
- **T.4.2:** Is the agenda for the next council meeting published on the web site?
- **T.4.3:** Are background documents for political meetings available online?
- **T.5.1:** Does the web site provide information on audio and / or video recordings of political meetings?
- **T.5.3:** Are audio and / or video recordings of council meetings available from the web site?
- **T.5.4:** Are the minutes from political meetings published on the web site?
- **T.6.1:** Does the web site provide a site search function?

This document mainly focuses on outlining a proof-of-concept measuring on transparency using learning algorithms. Additionally, we present a list of expected features as well as a short outline of the remaining tests.

As a proof of concept we have addressed test T.3.1. "Does the web site provide information about the access to mail record of the organisation? If yes, what is the URL?." This was done because we had already training data available. In this deliverable we will show how to detect a mail record automatically using manually classified data.

We believe that similar approaches can be applied for the other automatable transparency tests which focus on learning algorithms.

1.1 Reading instructions

At first we present the reader with basic information regarding classification of web pages, definitions used in this deliverable as well as how we address the problem of detecting mail record pages. In section 5, we describe steps to reach an accuracy of 96.5% within a duration of 3 seconds per web site. Finally, in section 7, we present initial ideas for implementing all 15 eGovMon transparency tests which rely on learning algorithms.

1.2 Related Deliverables

This deliverable is related to the following eGovMon Deliverable:

- **D3.2.1,** Annika Nietzio, eGovMon Transparency Indicators (part 1)
- **D4.2.1,** Morten Goodwin Olsen, eGovMon System Architecture
- **D6.1.1**, Mandana Eibegger, Large Scale Online Reporting Tool specification

2 Objectives

The objective is to define an algorithm which automatically detects the presence of mail records in public governmental web sites as well as the URL to the mail record. As input, the algorithm takes the URL of the web site. Note that some of the other tests will not take web site URL as input, but rather test results from other tests which typically means URLs to some services. A list of input for the learning algorithm tests are listed in section 9.

As from the algorithm we have:

(1) Existence of Mail Record as a boolean true/false value.

(2) URL to the mail record, if it exists.

(3) Probability[1].

(4) Duration of the algorithm.

Note that output (1),(2) and (3) will be stored in the eGovMon database and will be used as the transparency metrics. Output (4) is only part of testing the algorithm. The inputs and outputs are illustrated in Figure 2.1.

We attempt to identify both which algorithms work best, and which features to look for and how the algorithm may be adjusted to increase accuracy and performance.

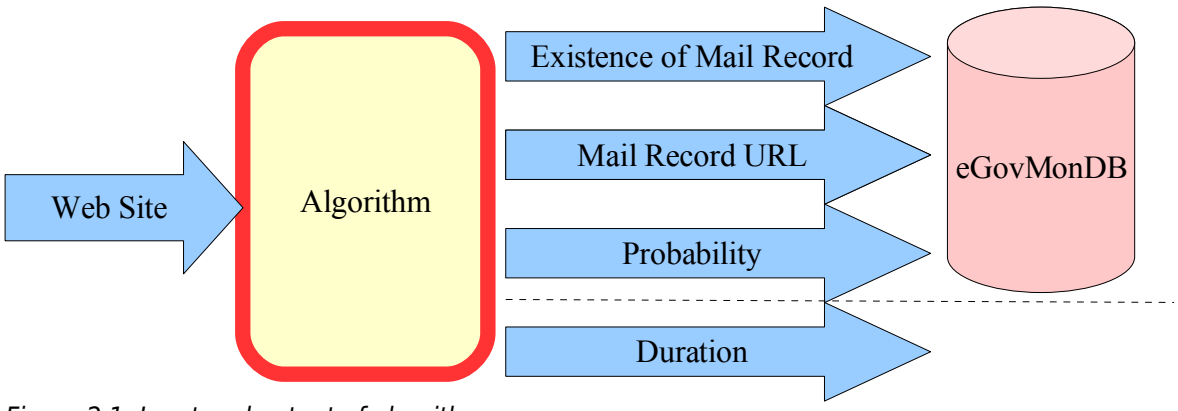

Figure 2.1: Input and output of algorithm

1 For the algorithm addressed in this deliverable probability is available. However, other tests, which may rely on different algorithms, may not use probability. If this is the case, probability will be substituted with some certainty metrics.

In this deliverable we have the following definitions:

What	Notation	Description
Mail Record	p_m	Class of web pages where users may access public governmental mail records.
Not Mail Record	p_n	Class of any other public governmental web page.
Web Page	$\vec{x}_i$	Web page *i* to be classified. Note that web pages are identified as their feature vectors.
Estimated Mail Record	ω_m	Estimation of p_m .
Estimated Not Mail Record	ω_n	Estimation of p_n .
Accuracy		Percentage of which pages correctly classified (Pages classified as part of p_m which truly are in p_m , as well as pages classified as p_n which truly are of p_n). This is further elaborated in Table 1.

Any web page $\vec{x}_i$ is either part of class p_m (mail record) or class p_n (not mail record). The information available from $\vec{x}_i$ is all data normally available from the web page, namely the URL, HTTP header, images, (X)HTML, Style sheets etc.

Note that p_m and p_n are the true classes of mail record and not mail record. It is clear that when classifying any $\vec{x}_i$, we can not know for sure which class $\vec{x}_i$ is located in, and thus we do not have the complete knowledge of p_m and p_n . Thus, based on the data available, we estimate p_m as ω_m and p_n as ω_n . From ω_m and ω_n we can see where we believe $\vec{x}_i$ should be located and classify $\vec{x}_i$ accordingly. However, in a controlled situation, we can check if $\vec{x}_i$ was classified correctly and measure how well the algorithms performs.

The objective can be summarised as: for all pages ($\vec{x}_i$) in a municipality web site, classify the page as ether mail record (p_m) or not mail record (p_n).

3 Algorithms

It should be noted that classification algorithms are divided into three parts ([Pattern]); preprocessing, extracting features and classification.

Figure 3.1: Main stages of classification algorithm

Preprocessing

In our algorithms, preprocessing includes downloading the web page. It also means building the sets ω_m and ω_n based on the training data available. Note that in many cases building of the training sets are done prior to any classification such as during nightly builds.

Extracting Features

Extracting the features from the web page which should be useful for the classification. Such features could for example be the words used in the page. The vector for the page ($\vec{x}_i$) is built.

Classification

Performing the actual classification. In our case this means deciding which class page $\vec{x}_i$ belongs to (ω_m or ω_n) and thus deciding if the page is mail record or not.

4 Training Data

The eGovMon focuses on Norwegian local government / municipality and public agency web sites. Because of this, it seemed natural to focus on data from Norwegian municipality web sites. For training data, we manually extracted the web pages containing mail records, where they existed (p_m). Additionally, as a counter part or control group, we randomly selected one web page and manually confirmed that it did not contain any mail records (p_n). In total this included 326 web pages. 204 web pages with mail records ($|\ p_m\ | = 204$) web pages and 118 web pages without mail records ($|\ p_n\ | = 118$).

5 Feature Extraction

As with any pattern classification approach [Pattern] we need to define the features which identify p_m and p_n well. Based on our knowledge in the area we extracted features we believe identify p_m and p_n - and further confirmed or reject the hypothesis by testing.

Furthermore, when designing an algorithm we need to keep in mind both accuracy and performance. The algorithm need to quickly predict the correct class. In this chapter we present our approach for improving both.

Firstly, it is worth noticing that the most web site contains multiple web pages. In fact, for accessibility evaluation, eGovMon downloads up to 6000 pages from each web site. Further more, 15% of the web sites have significantly more than 6000 web pages. From this we know that it would be impractical, if not impossible, to download and classify every page from a web site. When applicable, we limit the number of pages downloaded from each web site to 6000 as done in [INDREF].

Secondly, the true performance of a classification algorithm dealing with online web pages would depend on downloading as well as processing duration. Note that processing duration include preprocessing (parsing of (X)HTML), feature extraction and classification. For reasons of clarity, we present this here as processing duration.

Note that even though training is an essential part of any classification scheme and the duration of training may be significant, this can be done when new training data is available - and would not have any impact on the processing duration.

The performance for downloading and classifying each web site is therefore:

```
Performance per site
        = (Number of pages to be downloaded) *
          (Download Duration + Processing Duration)
```

To get a realistic measure of download duration, we downloaded 836 random web pages from Norwegian municipalities[2] . The average duration to download one page was 2.9667993535822408 seconds. It should be mentioned that in a crawler architecture downloads would typically run in parallel. However, the individual download duration for each page would be unchanged but many pages could be downloaded in parallel ([D5.3.1-4]).

We have in this chapter used Naïve Bayes as basis for our classification algorithm. One of the characteristics of Naïve Bayes is that the individual features within each web page is treated independently. Thus, we assume that there are no relations between the features other than the fact that they are from the same class (e.g. both from P_m or both from P_n). As an example, using words from each web page as features, the following two sentences: ”We have mail records online, do not e-mail us for requests.” and ”We do not have mail records online, e-mail us for requests.” are treated as identical despite the fact that the sentences have opposite intentions.

It should be noted that traditional Naïve Bayes may not be sufficient for all tests. If some of the tests would for instance order of characters or words to be included (e.g. [NGRAM]). Addressing additional algorithms will be based on which algorithms are most suitable is not part of this deliverable, but will be addressed in documents following.

5.1 Content only

Initially, we started with the simplest approach by applying the algorithm to the words within each web page. I.e. we choose the feature to be the words in the document. The individual words part of p_m and p_n can be seen in Figure 5.1. On the y-axis we can see the 50 most common words part of p_m and p_n . On the x-axis we can see the frequency of the words used in mail records. As an example, the word *postliste* has in p_m a frequency of slightly more than 1.5%. This means

2 Including Mail Record Web Pages

that slightly more than 1.5% of the words in p_m is the word *postliste*. We can further see from Figure 5.1 that there is a clear difference between the words part of p_m and p_n and that, not surprisingly, p_m is more homogeneous than p_n .

By classifying based on words alone , we only received an accuracy of 0.612813370474 and a duration of 2.8357719291340002 seconds per page.

For this approach, every page within the web site would need to be classified (or every page until mail record was found if it existed). If we assume the number of downloads to be similar to the eGovMon accessibility evaluation, 6000 pages per web site, we would have the duration of each web site as:

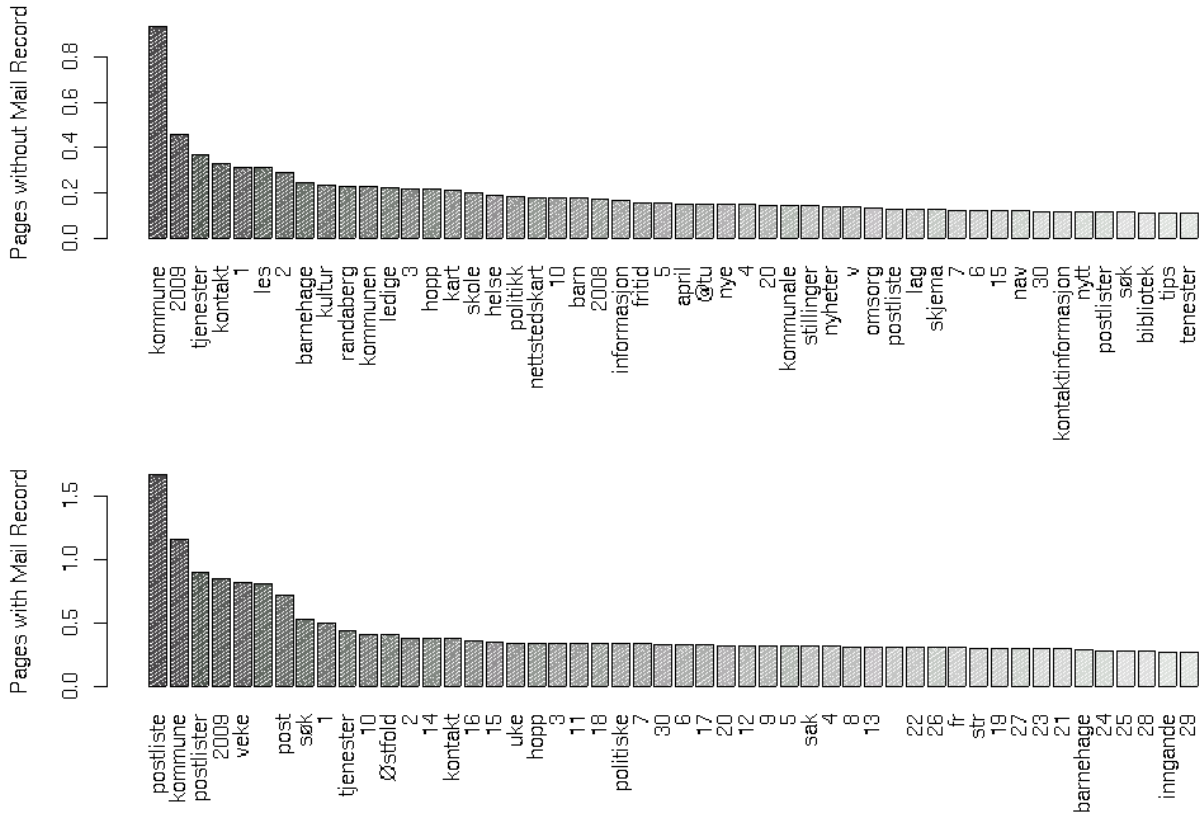

Figure 5.1: Words in P_m *and* P_n

Performance per site
= (Number of pages to be downloaded) *
(Download Duration + Processing Duration)
= 6000 * (2.9667993535822408 + 2.8357719291340002) seconds
= 9 hours and 40 minutes

5.2 Attribute Values

To increase the accuracy, we decided to include attribute values as part of the feature. Such features would typically identify technologies and vendor used.

Example of such feature would be titleTjenester in

It is worth noticing that there are only a few content providers for mail records in Norway. We believe that attribute values such as titleTjenester part of mail records would be characteristic for the mail records and could be used to increase the accuracy. Including attribute values gave us an increased accuracy to 0.656509695291 .

5.3 Only Title

Since the performance of the algorithm is, for any practical purposes, clearly too slow, we decided to reduce the number of features used in the classification. We believe that the title, which is intended to describe the web page, contains a lot of information of whether the page is a mail record or not. The title could in a mail record for example be the following

<title>Mail Record</title>

.

By using only the title we get, as expected, a lower accuracy of 0.560906515581 . However, the duration of classifying each individual page was significantly increased. The average duration for each page was 0.00027525102770004149 seconds[3].

Performance per site
= (Number of pages to be downloaded) *
(Download Duration + Processing Duration)
= 6000 * (2.9667993535822408 + 0.00027525102770004149)seconds
= 4 hours and 57 minutes

5.4 Title and h1

To increase the accuracy, we needed to add more features part of the training/classification. Similar to title, we believe that the high level header information (H1) contains information on whether pages are p_m or p_n .

The h1 could in a mail record for example be the following

<H1>Mail Record</H1>.

By including H1 we increased the accuracy to 0.680790960452 . Note that using only H1 and title we have a better accuracy than using the content and attribute values.

5.5 Leave out two

For testing we have used the leave-one-out method for testing.

For our tasks, some challenges with using leave one out became evident. The problem was that our training data in reality where built in sets. We have collected from each web site one random page and the mail record page. Since these two pages are from the same municipality and same web site they are strongly connected, even though one is from the classes p_n and one from p_m .

Very often, the page to be classified would be wrongly classified as the same as the other page from the same municipality. As an example, the municipality *Nordkapp* has the title their mail record page: ”*Nordkapp Kommune*”. The randomly selected counter page had the title ”*Nordkapp Kommune Hjelp*”. By having $\vec{x}_i$ part of p_n , words such as ”*Nordkapp*” is likely to appear in the counterpart p_m and the classification would get an unfair disadvantage. Any page with the title *Nordkapp Kommune* would be wrongly classified as p_n since the words ”*Nordkapp Kommune Hjelp*” exists

there. This is a practical example where the data is strongly clustered as leave-one-out alone provides an unfair bias [MRUAAEC].

3 It should be noted that the downloading speed by far exceeds the processing speed. The download speed par page cannot in practice be reduced ([D5.3.1-4]). However, the download speed per site can be reduced by e.g. downloading fewer pages. This is addressed in section 5.8.

Because of this, we decided not only to remove the page to be classified from the training data, as is customary in leave-one-out, but to remove all data from the same web site. By doing this, we got a significant accuracy increase to 0.878531073446.

It should be noted that removing pages from the same municipality is easy in a controlled environment However, when the system is running live it may be more challenging. Several possibilities exists which should be investigated further:

- When classifying for one municipality, remove all training data from the same municipality. This means that all web sites will have a specialised classification algorithm. It could be time consuming to rebuild the training data for each web site.
- Have a specialised list of words which should be removed from the training / classification data. These words could include names of the municipalities, names of counties, all proper nouns, etc.

5.6 Removing overlapping words

As can be seen in Figure 5.1 there are several words which appear in both classes, p_m and p_n . These are words that are not unique for each class and work as bias for the algorithms. Example of such words would be Kommune which, according to Figure 5.1, is almost just as likely to appear in p_m and p_n . By removing the overlapping words, similar to how stop words are typically removed, we increased the accuracy to 0.90395480226 .

Note that despite several updates, the performance remained relatively unchanged since section 5.3. The reason is that these updates do not change the computation significantly and thus do not have a significant impact on the processing duration, nor any changes to the number of pages needed to download.

5.7 Including URL file names

The URL to an online service often contains a lot of information by itself. As an example, a file name for a mail record page could be postlist.html. By including the file name in the training/classification, the accuracy was a bit surprisingly decreased to 0.81512605042.

We noticed that many extracted strings are typically part of such file names (extensions), such as html, php, asp, which do not include any information of whether it is mail record or not. By deliberately removing these, the accuracy increased to 0.93288590604.

5.8 URL file name alone

A huge advantage with only using the URL as training classification, is that we do not need to download the page. I.e. we do not need to download 6000 page from each web site. This means we can download the homepage of the site and only use the URL in the links to classify. By doing this we got a slight reduction of accuracy to 0.921686746988 . Even though this was a slight decrease in accuracy, it was a huge increase in performance due to the fact that we only need to download one page from each site.

By URL file name we include directory and file name, but do not include parameters. For the following URL: http://www.example.com/directory/filename.html the page vector $\vec{x}_i$ would consist of the words: directory and filename.html

Performance per site

```
= (Number of pages to be downloaded) *
  (Download Duration + Processing Duration)
= 1 * (2.9667993535822408 + 0.00014583516565167009) seconds
= 3 seconds
```

It should be noted that prior to any classification, eGovMon crawls and detects 6000 pages from each web site (D4.2.1). Another possibility would be that the apply the algorithm on all URLs detected by the Crawler. This might have a positive affect on the performance.

5.9 Including URL parameter

In an attempt to further increase the accuracy, we included, in addition to the file name, attributes in the URL.

Example of attributes values would be recordnr and month in the following.
http://www.example.kommune.no/somepage.html?recordnr=15&month=04

By doing this there was a slight increase in the accuracy to 0.93769470405 .

5.10 Frame Page

Many of the mail records pages consisted of frames. A frame page is special because it includes many URLs (one for each sub page) in contrast to one URL which is typical for normal pages By adjusting the training/classification set to include a URL to the frame page rather than the individual sub page we increase the accuracy to 0.965838509317.

I.e. if we get a random governmental web site as input to our algorithm, we can in 96.5% of the cases find the correct mail record within approximately 3 seconds[4].

5.11 Accuracy of implementations

In Figure 5.2 we present a graph of the accuracy of implementations of the classification algorithm. Note that goal was to both have as high accuracy as possible as well as short duration of evaluation. In the figure we have three different duration speeds; 9 hours and 40 minutes, 4 hours and 57 minutes, and 3 seconds. A complete list of accuracy and duration is available in Appendix A – Summary of Algorithm Development.

4 This test was done on a Fujitsu PRIMERGY RX200 2x Quadcore Intel Xeon DP E5405 on a 100 Mbit connection. Other servers may have different durations.

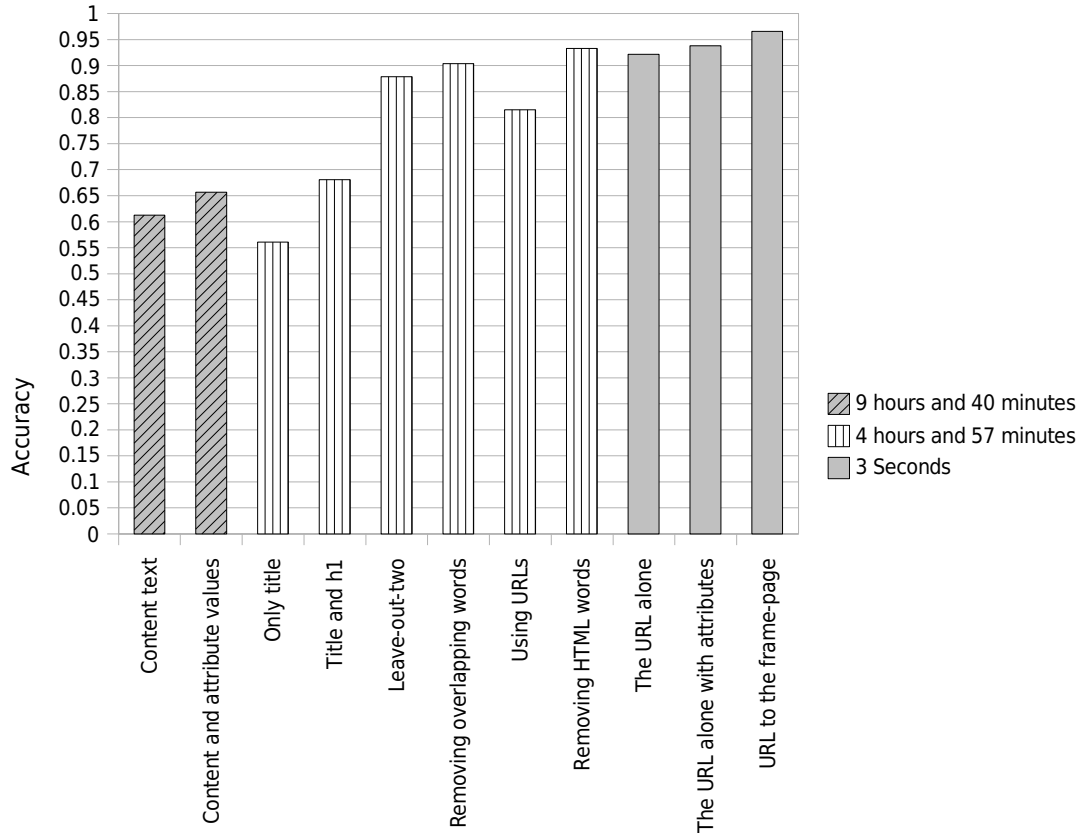

Figure 5.2: Accuracy with the different implementations

5.12 Features Not Used

Several features where tested but had little or no impact on the accuracy or performance. This shows that these features are not identifying mail records well and are therefore just as common in mail record web pages P_m as other web page P_n .

The following features did not contribute anything to the results:

- Any types of dates.
- Number of unique times dates are mentions.
- Any types of months (e.g.: integer - 01, string - January etc.).
- Any types of days (e.g.: integer 1 – string - Monday)
- Link texts.
- Number of links to PDF-documents.
- PDF link text.
- Links with dates, months or days.
- Use of bold text (<b> or tag).
- Predefined keywords (found pages with ”We do not have mail records”).

6 Contributing Words

In Figure 6.1 we present the frequency of contributing words for both p_m and p_n using only the URL is used as features. This is different to Figure 5.1 which contains all words within the page. It is worth noticing that there is a huge difference between these classes. The p_m is, as expected a lot more homogeneous than p_n . For mail records, p_m , we can in Figure 6.1 find words which we naturally identify as mail records (in Norwegian) are: *journal* (English: journal), *innsyn* (English: insight) and *postliste* (English: mail record).

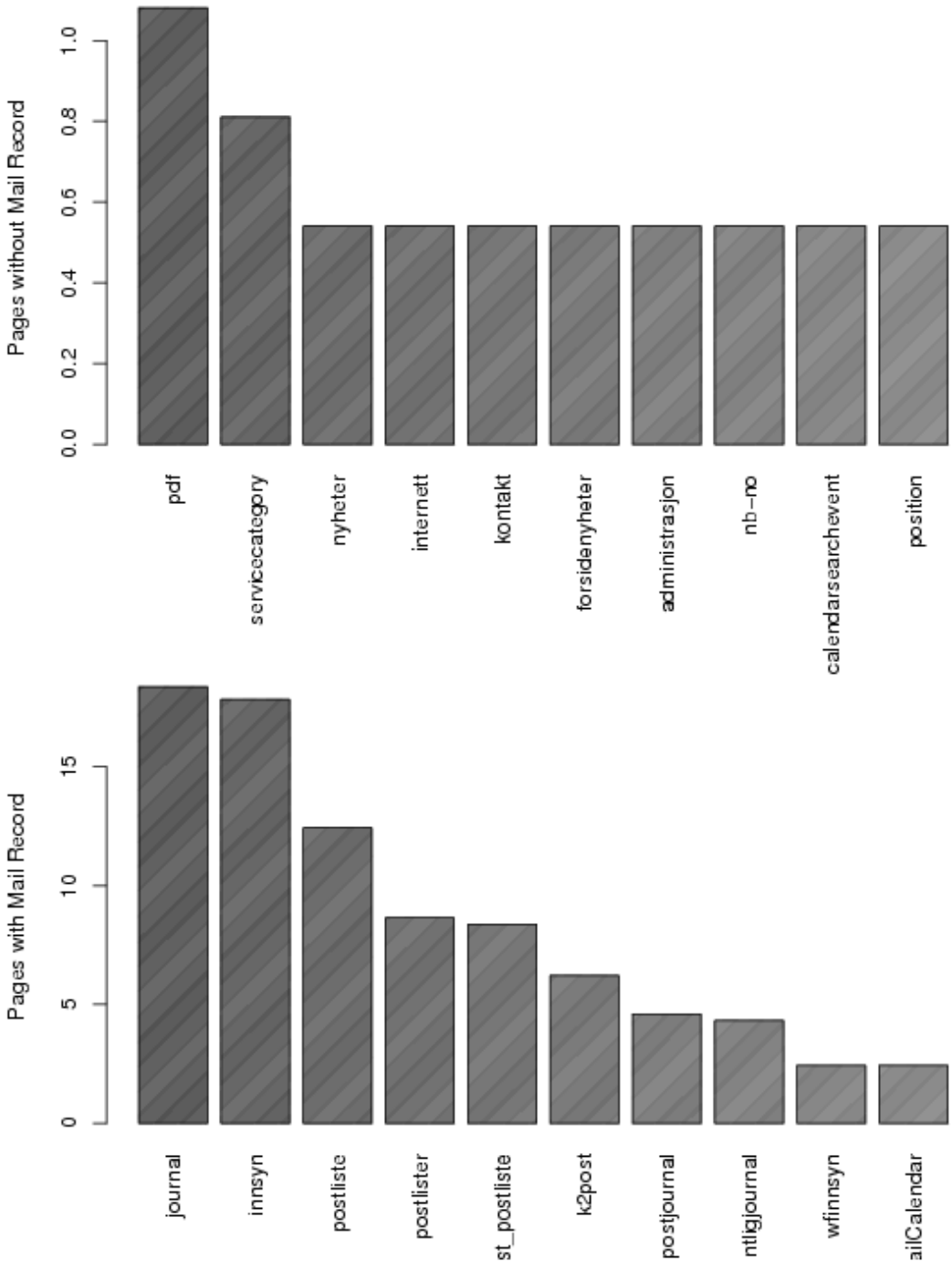

Figure 6.1: Contributing Words

The figure further shows us that we could have applied stemming to the words since both postliste (singular) and *postlister* (plural) are part of the list. However, we should keep in mind our goal of both accuracy and duration as well as the fact that stemming increases the duration. Furthermore, looking at the web pages we could not correctly classify (section Appendix B), we see that stemming would not be useful for increasing accuracy. Furthermore, the frequency for the individual words within p_m is very high. The most common word, journal, has a frequency of more than 18%. This means that more than 18% of the words in the URL of pages classified as p_m are journal. Furthermore, since journal is not visible in p_n, we can confidently say that when the word journal is encountered, we can with confidence say that the page is mail record.

On the other hand, for the remaining pages p_n, the frequency for each word is significantly lower. The word PDF has a little more then 1%. Not surprisingly, from Figure 5.1 we cannot see any words which are typical for pages which are not mail records. I.e these pages are not homogeneous. In Table 1 we show the confusion matrix for this classification. The table shows us 4.4% of the what are truly p_m, are classified as p_n, while only 0.9% of what should be in p_n where classified as p_n. From this we can claim, that the algorithm can confidently assert that a page is in p_m. If it has no words distinct marking them as mail records, it is in the class p_n.

Classified / Correct	p_m	p_n
p_m	0.95588235294117652	0.0085470085470085479
p_n	0.044117647058823532	0.99145299145299148

Table 1: Confusion Matrix

7 Suggested Approaches Learning Algorithm Tests

In this section we provide educated guesses on what is needed as input and features for the remaining tests. It should be noted that input is addressed differently than in D3.2.1, where input is separated into data source, occurrence pattern and input from other tests. In contrast, we do not separate type of input in this deliverable.

Test	**Features which are expected to be useful**	**Input**
T.1.1 Does the web site provide contact information of the organisation? If yes, what is the URL of the contact information page?	Words and numbers (phone number) from web page.	Every web page
T.1.2 Is the contact information complete?	Words and numbers (phone number) from web page.	One web page (From T.1.1)
T.2.1 Does the web site provide a news page (or section of a page) with information?	Words, number of links, link text,RSS link	Every web page

T.2.3 Does the web site provide an email newsletter?	Link text, Words, Subscription Form Fields	Every web page
T.3.1 Does the web site provide information about the access to the mail record of the organisation? If yes, what it the URL?	URL	One web page (home page)
T.3.2 Does the web site provide access to the mail record of the organisation? If yes, what it the URL?	URL	One web page (From T.3.1)
T.3.3 Does the online mail records have a search function?	Form ID, Form Button Text, Words	One web page (From T.3.1)
T.3.4 Are the documents from the mail record available online?	Link text, document types,	One web page (From T.3.1)
T.4.1 Does the web site provide information on the date and time of the next council meeting?	Words, dates	Every web page
T.4.2 Is the agenda for the next council meeting published on the web site?	Words, dates	One web page (From T.4.1)
T.4.3 Are background documents for political meetings available online?	Words, dates, link text, document types	One web page (From T.4.1)
T.5.1 Does the web site provide information on audio and / or video recordings of political meetings?	Types of streaming, words, text	Every web page
T.5.3 Are audio and / or video recordings of council meetings available from the web site?	Formats of links, link texts, embed types, embed descriptions	One web page (From T.5.1)
T.5.4 Are the minutes from political meetings published on the web site?	Dates, Words.	One web page (From T.5.1)
T.6.1 Does the web site provide a site search function?	URL	One web page (home page)

8 Conclusion

We have in this deliverable provided a proof-of-concept of applying learning algorithms for testing. For this approach we successfully identify mail records in 96.5% of the cases within 3 seconds per web site. We have found that classifying based on the URL alone gives both best accuracy and performance as there exists only few content providers for mail records in Norway and most are identifiable by the URL.

References

Pattern: Anil K. Jain, Robert P.W. Duin and Jianchang Mao, Statistical Pattern Recognition: A Review, 2000, Statistical Pattern Recognition: A Review

INDREF: Annika Nietzio, Nils Ulltveit-Moe, Terje Gjøsæter, Morten Goodwin Olsen, UWEM Indicator Refinement, , http://www.eiao.net/resources

D5.3.1-4: Morten Goodwin Olsen, Nils UlltveitMoe,Annika Nietzio, Mikael Snaprud, Anand B. Pillai, Test report of EIAO version 2.2, 2008, http://www.eiao.net/resources

NGRAM: Peter Nather, N-gram based Text Categorization, 2005, http://www.dcs.fmph.uniba.sk/diplomovky/obhajene/getfile.php/Ng-based-tc.pdf?id=1&fid=3&type=application%2Fpdf

MRUAAEC: Lennart Eriksson, Joanna Jaworska, Andrew P. Worth, Mark T.D. Cronin, Robert M. McDowell and Paola Gramatica, Methods for Reliability and Uncertainty Assessment and for ApplicabilityEvaluations of Classification- and Regression-Based QSARs, 2003, http://www.ehponline.org/members/2003/5758/5758.html

9 Appendix A – Summary of Algorithm Development

Change	Accuracy	Performance per site
Content text	0.612813370474	9 hours and 40 minutes
Content and attribute values	0.656509695291	-
Only title	0.560906515581	4 hours and 57 minutes
Title and h1	0.680790960452	-
Leave-out-two	0.878531073446	-
Removing overlapping words	0.90395480226	-
Using URL file names	0.81512605042	-
Removing HTML strings	0.93288590604	-
The URL alone	0.921686746988	3 Seconds
The URL alone with attributes	0.93769470405	-
URL to the frame-page	0.965838509317	-

10 Appendix B – Pages we cannot classify

In this section we present the pages that we cannot classify with our approach. By examining the URLs, It is clear that there is little or no information that these URLs are mail record (belong to P_m). i.e. we cannot see any names such as postliste (English: mail record) as presented seen Figure 6.1. Note that the URLs we cannot classify sums up to 0.85% of the pages.

•http://www.nordreisa.kommune.no/troms/nordreisa/nordreisak.nsf/id/D960D328F33D2898C125754E0036724A?OpenDocument

•http://www.froland.kommune.no/templates/froland/frolandpage.aspxid=1388

•http://www.leikanger.kommune.no/cms/leikanger.nsf/(\$all)/b96c9351e690d5fdc12574f00034b239opendocument

•http://www.hemne.kommune.no/899/206/1528-0.html

•http://www.krodsherad.kommune.no/sider/tekst.aspside=111

•http://lokaldemokrati.asker.kommune.no/templates/ldpostliste.aspxid=33&kom=asker

Note that the last URL contains the file name *ldpostliste.aspx*. Intuitively, it should have been straight forward to classify this as a mail record due to the fact that the Norwegian word for mail record is postliste. However, our algorithm is implemented with complete file names and not parts of file names. To address parts of file names we could have an N-gram character approach ([NGRAM]).