D3.11 E-mail Reply Prediction Tool

Project	SWELL
Project leader	Wessel Kraaij (TNO)
Work package	WP3
Deliverable number	D3.11 (unplanned)
Authors	Rianne Kaptein, Maya Sappelli, Suzan Verberne, John Schavemaker
Reviewers	Miriam Vollenbroek, Saskia van Dantzig, Wessel Kraaij
Date	9-1-2014
Version	0.5
Access Rights	COMMIT/
Status	Updated after review Miriam Vollenbroek and Saskia van Dantzig

SWELL Partners:

Almende, Noldus, Novay, Philips, TNO, Radboud Universiteit Nijmegen, Roessingh Research and Development, Sense-Os, Universiteit Twente,

This publication was supported by the Dutch national program COMMIT.

1 Summary

This document describes a software implementation of an e-mail reply prediction tool which is implemented as a Microsoft Outlook plugin. The tool predicts whether a reply is expected to an e-mail in your e-mail box using machine learning and the training data in your own e-mail box.

Contents

Formatted: Default Paragraph Font, Dutch (Netherlands), Check spelling and grammar

Formatted: Default Paragraph Font, Dutch (Netherlands), Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Dutch (Netherlands), Check spelling and grammar

Formatted: Default Paragraph Font, Dutch (Netherlands), Check spelling and grammar

Formatted: Default Paragraph Font, Dutch (Netherlands), Check spelling and grammar

Formatted: Default Paragraph Font, Dutch (Netherlands), Check spelling and grammar

Formatted: Default Paragraph Font, Dutch (Netherlands), Check spelling and grammar

Formatted: Default Paragraph Font, Dutch (Netherlands), Check spelling and grammar

2 Introduction

In our project SWELL (smart reasoning for well-being at home and at work) we aim to improve well-being at work by supporting knowledge workers. We define knowledge workers as people who use and produce information as their main task, and frequently use computers. Well-working could be defined as "being and feeling in control", with a positive impact on work efficiency and effectiveness, work pleasure, mental and physical health status (see deliverable D3.8).

In order to validate the effectiveness of SWELL concepts, we develop SWELL@Work applications that include the following (see deliverable D3.1b update):

- SWELL board;
- Personal feedback tool: the *Happy Worker App* (HWA);
- E-mail organization assistant;
- SWELL golden demo (including applications above).

The e-mail reply prediction tool described in this deliverable is part of the email organization assistant in the SWELL@work tool. The SWELL Email organization assistant aims to reduce the amount of time spent on email, by automatically categorizing emails in projects, organizing attachments, labeling emails that expect a reply, and adapting email notifications so that unimportant messages do not interrupt the worker. This should lead to better email organization and a more efficient workday.

For this deliverable (D3.11), we implemented the following functionality from the "E-mail organization assistant" :

Predict whether a reply is expected to an e-mail

In the next section we describe the architecture and the user interface of the tool. In Section 4 we describe the features and algorithms used for machine learning. In Section 5 we evaluate the quality of the reply predictor.

3 Reply Prediction Tool

The reply prediction tool is a Microsoft Office Add-in for Outlook. The Add-in was written with the use of the Add-In Express for Office and .NET framework[1]. This framework supports the writing of Microsoft Office extensions by providing version neutrality, easy deployment , and visual designers to customize menus, toolbars, ribbons etc. The plugin was written in C#. For the machine learning component we use Weka (Hall, 2009), a data mining software package written in Java. To connect to the Weka Java library from C#, we use IKVM[2].

After installation of the e-mail plugin a Swell tab is added to the Outlook Explorer Ribbon, which is the view that you get when you open Outlook.

[1] http://www.add-in-express.com/add-in-net/index.php
[2] http://www.ikvm.net/

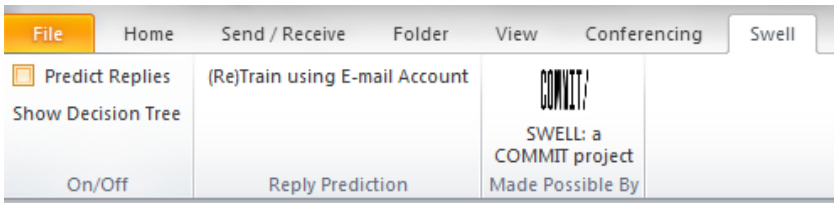

Figure 1 : Outlook Swell tab

The tab contains one checkbox and two buttons:

- Predict Replies. This checkbox turns reply prediction on your mailbox on or off. When you turn it on, for the last 25 mails in your inbox and the last 25 sent mails reply prediction is carried out, and every time your send or receive an e-mail, this e-mail is classified.
- Show Decision Tree. When you click this button a messagebox appears in which the decision tree used to classify your e-mails is visualized.
- (Re)Train using E-mail Account. When you click this button, the plugin goes through all your Outlook mail folders to build a personalized model for reply prediction. Depending on the amount of e-mail in Outlook, this might take some minutes.

When an e-mail is classified as expecting a reply, a Follow Up flag is added to the e-mail setting the due date of the e-mail item to “This Week”. When a reply to the e-mail is sent, the flag is marked as complete.

Limitations:

- Our tool cannot read mails from archived folders.
- During the building of the classifier Outlook cannot be used

4 Machine Learning for Reply Prediction

The approach for our reply prediction is based on the work (Sappelli, Verberne, & Kraaij, 2013). In order to respect the privacy of the (test) users of the application, and to keep the number of features low, we do not use features that require the storage of complete e-mail content. Instead we only use and store features derived from the e-mail contents and meta information. Furthermore, to protect the privacy e-mail addresses of contacts are not stored directly but converted to MD5 hashes.

The following features are used for reply prediction:

1. Incoming or outgoing e-mail (Boolean)
2. Length of e-mail in number of words (Numeric)
3. Contains question marks (Boolean)
4. Number of question words (Numeric)
5. Average reply frequency to sender (number of e-mails/number of replied e-mails) (Numeric)
6. Count of replies to sender (Numeric)
7. Receiver score (Numeric)

1 : e-mail address not in to or cc
2: e-mail address in cc
3: e-mail address in to

8. Number of receivers (Numeric)
9. Is the e-mail already a reply to another e-mail in the conversation (Boolean)

All these features are used to predict the class attribute:

10. Will there be a reply to this e-mail (Boolean)

Most of the features are independent of the language the e-mail is written in. Only the number of question words depends on language. We added Dutch and English question words[3] to our vocabulary here, since these are the languages that most people of our target group use.

The Weka package provides us with a number of classification algorithms that could be used to classify e-mails. In the next section we evaluate the performance of a number of classification algorithms.

5 Evaluation

For the evaluation of the quality of the e-mail predictor, we created a version of the tool with added functionality to create a training and test file from an e-mail account. Error on the training data is not a good indicator of performance on future data, so we have to make a separation between training and test data.

To create the train and test files the mail items are sorted on date, so we train on older mails and predict the newer mails. This is similar to what happens when you use the tool: the model is trained on your e-mail history, and predicts for new e-mails whether a reply is expected.

The size of our training and test data is reasonable. Cross-validation is a procedure to make the most of the available training data and avoids overlapping test sets. In the first step data is split into k subsets of equal size, in the second step each subset is used in turn for testing. For cross-validation it is important that the test data is not used in any way to create the classifier (Witten,2005). However, features 5 and 6, the average reply frequency and the count of replies are dependent on the complete training set. This makes it hard to do cross-validation, since these features would need to be recalculated for each subset of the data.

So, we train on two third of the period of the dataset and test on one third of the period of the dataset. However, since our tool does not have access to archived e-mails, and older e-mails are more likely to be archived, this ratio is not preserved in the collected training and test data. In the table we present the results on four evaluation measures:

1. Accuracy: the percentage of correctly classified e-mails.
2. Precision on the positive class (e-mails where a reply is received): The proportion of the examples which truly have class x among all those which were classified as class x.

[3] Dutch question words:wat,wanneer,waar,welke,wie,waarom,hoe.
English question words: what,when,where,which,who,whom,whose,why,how.

3. Recall on the positive class (e-mails where a reply is received): The proportion of examples which were classified as class x, among all examples which truly have class x, i.e. how much part of the class was captured.
4. F-Measure on the positive class (e-mails where a reply is received): A weighted average of precision and recall, i.e. 2*Precision*Recall / (Precision + Recall)

Since in our dataset the classes are not balanced, i.e. there are less e-mails that receive a reply, accuracy only does not give a realistic picture. We take a closer look at the most important class: e-mails that receive a reply, and report on precision, recall and F-measure for this class.

Each column in the table represents the e-mail data from one mailbox. In the last column we give the average over the three datasets. The size of the training and test set is given, as well as the percentage of positive examples in the set, i.e. the number of e-mails that receive a reply.

Simple classification algorithms often work well. We use 5 different types of common classification algorithms (Witten, 2005) to see which one works well on our dataset:

- NB: Naïve Bayes, a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions.
- J48: J48 Decision Tree, learns a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label.
- SMO: Support Vector Machine, constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.
- RF: Random Forest, an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.
- IB1: Nearest Neighbour Classifier (n=1), an instance is assigned to the class of its single nearest neighbor, i.e. the instance in the training set that is most similar to the instance to be classified.

Table 1 shows the classification results of the different classification algorithms on our dataset.

		Set A	Set B	Set C	Average
# training examples		1086	585	352	674
% training examples replied		12,3	7,5	7,4	9,1
# test examples		1726	2062	610	1466
% test examples replied		11	2,9	5,2	6,4
Classifier					
NB	Accuracy	84,6	89,1	93,1	88,9
	Precision Replied	0,385	0,209	0,414	0,336
	Recall Replied	0,649	0,983	0,750	0,794

Formatted: Font:

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Font: (Default) Calibri, Font color: Black

Formatted Table

Formatted: Font: (Default) Calibri, Font color: Black

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Font: (Default) Calibri, Font color: Black

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Font: (Default) Calibri, Font color: Black

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Font: (Default) Calibri, Font color: Black

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Font: (Default) Calibri, Font color: Black

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Font: (Default) Calibri, Font color: Black

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Font: (Default) Calibri, Font color: Black

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Font: (Default) Calibri, Font color: Black

Formatted: Space After: 0 pt, Line spacing: single

	F-measure Replied	0,483	0,345	0,533	0,454
J48	Accuracy	91,2	97,9	97,0	95,4
	Precision Replied	0,614	0,660	0,769	0,681
	Recall Replied	0,550	0,550	0,625	0,575
	F-measure Replied	0,580	0,600	0,690	0,623
SMO	Accuracy	89,7	97,3	95,9	94,3
	Precision Replied	0,606	0,750	0,733	0,696
	Recall Replied	0,209	0,100	0,344	0,218
	F-measure Replied	0,311	0,176	0,468	0,318
RF	Accuracy	**95,7**	98,1	**98,2**	**97,3**
	Precision Replied	**0,834**	0,636	**0,839**	**0,770**
	Recall Replied	**0,764**	0,817	**0,813**	0,798
	F-measure Replied	**0,798**	0,715	**0,825**	0,779
IB1	Accuracy	94,8	**98,8**	97,7	97,1
	Precision Replied	0,771	**0,757**	0,765	0,764
	Recall Replied	0,759	**0,883**	**0,813**	**0,818**
	F-measure Replied	0,765	**0,815**	0,788	**0,789**

		Set A	Set B	Set C	Average
# training examples		1086	585	352	674
% training examples replied		12,3	7,5	7,4	9,1
# test examples		1726	2062	610	1466
% test examples replied		11	2,9	5,2	6,4
Classifier	-	-	-	-	-
NB	Accuracy	84,6	89,1	93,1	88,9
-	Precision Replied	0,385	0,209	0,414	0,336
-	Recall Replied	0,649	0,983	0,750	0,794
-	F-measure Replied	0,483	0,345	0,533	0,454
J48	Accuracy	91,2	97,9	97,0	95,4
-	Precision Replied	0,614	0,660	0,769	0,681
-	Recall Replied	0,550	0,550	0,625	0,575
-	F-measure Replied	0,580	0,600	0,690	0,623
SMO	Accuracy	89,7	97,3	95,9	94,3
-	Precision Replied	0,606	0,750	0,733	0,696
-	Recall Replied	0,209	0,100	0,344	0,218
-	F-measure Replied	0,311	0,176	0,468	0,318
RF	Accuracy	**95,7**	98,1	**98,2**	**97,3**
-	Precision Replied	**0,834**	0,636	**0,839**	**0,770**
-	Recall Replied	**0,764**	0,817	**0,813**	0,798
-	F-measure Replied	**0,798**	0,715	**0,825**	0,779
IB1	Accuracy	94,8	**98,8**	97,7	97,1
-	Precision Replied	0,771	**0,757**	0,765	0,764
	Recall Replied	0,759	**0,883**	**0,813**	**0,818**
-	F-measure Replied	0,765	**0,815**	0,788	**0,789**

Table 1: Performance of different classification algorithms on e-mail data sets.

Our test persons receive a lot of e-mail on which no reply is needed, for example from mailing lists. Only a small percentage of e-mails receives a reply within the TNO company, in other settings this ratio might be different. The Random Forest and the Nearest Neighbor classifier perform best overall. In our tool we make use of the J48 Decision Tree classifier. This classifier is transparent about how classification results are obtained, and also achieves good results.

In the interface of the tool there is an option to show the decision tree that is used to classify your data. In Figure 1 an example decision tree is shown. The tree indicates how the classifier uses the attributes to make a decision. The leaf nodes indicate which class an instance will be assigned to should that node be reached. The numbers in brackets after the leaf nodes indicate the number of instances assigned to that node, followed by how many of those instances are incorrectly classified as a result.

If it turns out that transparency of the algorithm is not an important feature for the user, it is easy to change the classification algorithm in the tool. In a next version the classification algorithm could even be personalized, whatever classification algorithm works best on your data could be selected.

6 References

Hall, M. F. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter, 11*(1), 10-18.

Sappelli, M., Verberne, S., & Kraaij, W. (2013). Combining textual and non-textual features for e-mail importance estimation. *25th Benelux Conference on Artificial Intelligence (BNAIC 2013).* Delft.

Witten, I.H., and Eibe F. (2005). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Figure 1: An example decision tree

J48 pruned tree

IsAReply = True

| AbsReplyFrequency <= 2: False (27.0/4.0)

| AbsReplyFrequency > 2

| | IncomingMail = True: False (46.0/14.0)

| | IncomingMail = False: True (61.0/24.0)

IsAReply = False

| AbsReplyFrequency <= 0: False (416.0)

| AbsReplyFrequency > 0

| | NrReceivers <= 1

| | | ReceiverSpecificity <= 0: False (10.0)

| | | ReceiverSpecificity > 0

| | | | AvgReplyFrequency <= 0.203704

| | | | | ContainsQuestionMarks = True

| | | | | | IncomingMail = True

| | | | | | | Length <= 1403

| | | | | | | | QuestionWordsCount <= 1

| | | | | | | | | AvgReplyFrequency <= 0.178571: True (10.0/3.0)

| | | | | | | | | AvgReplyFrequency > 0.178571: False (2.0)

| | | | | | | | QuestionWordsCount > 1: True (4.0)

| | | | | | | Length > 1403

| | | | | | | | QuestionWordsCount <= 6

| | | | | | | | | AbsReplyFrequency <= 4

| | | | | | | | | | QuestionWordsCount <= 2: False (5.0)

| | | | | | | | | | QuestionWordsCount > 2: True (2.0)

| | | | | | | | | AbsReplyFrequency > 4: False (8.0)

| | | | | | | | QuestionWordsCount > 6: True (2.0)

| | | | | | IncomingMail = False

| | | | | | | AbsReplyFrequency <= 6: True (12.0/5.0)

| | | | | | | AbsReplyFrequency > 6: False (8.0)

| | | | | ContainsQuestionMarks = False

| | | | | | IncomingMail = True

| | | | | | | MailId <= 363: False (15.0/1.0)

| | | | | | | MailId > 363

| | | | | | | | MailId <= 736: True (6.0/1.0)

| | | | | | | | MailId > 736: False (3.0)

| | | | | | IncomingMail = False

| | | | | | | Length <= 230: True (2.0)

| | | | | | | Length > 230: False (16.0/2.0)

| | | | AvgReplyFrequency > 0.203704: True (31.0/7.0)

| | NrReceivers > 1: False (400.0/23.0)

Number of Leaves : 21

Size of the tree : 41