Automatic Fax Routing

Paul Viola, James Rinker, and Martin Law

Microsoft Research
Redmond, WA (USA)

Abstract. We present a system for automatic FAX routing which processes incoming FAX images and forwards them to the correct email alias. The system first performs optical character recognition to find words and in some cases parts of words (we have observed error rates as high as 10 to 20 percent). For all these "noisy" words, a set of features is computed which include internal text features, location features, and relationship features. These features are combined to estimate the relevance of the word in the context of the page *and* the recipient database. The parameters of the word relevance function are *learned* from training data using the AdaBoost learning algorithm. Words are then compared to the database of recipients to find likely matches. The recipients are finally ranked by combining the quality of the matches and the relevance of the words. Experiments are presented which demonstrate the effectiveness of this system on a large set of real data.

1 Introduction

Companies may receive hundreds or thousands of FAXes per day. While many are printed by a conventional FAX machine, a growing number will arrive at computers equipped with FAX modems or through an internet FAX service. One natural mechanism for delivering these FAXes is as email with an attached FAX image file (such as TIFF). Incoming FAX images lack *digital* information about the destination email address (though they may include a small amount of digital information about the sender). Routing the FAX to the correct email address must be performed by hand, by a FAX secretary that examines each FAX image. Though the cost of a digital FAX system is significantly less than a paper FAX system, the time required for routing FAXes in a large organization can lead to significant costs.

This paper describes an automatic system for routing a FAX to a set of incoming addresses. The process proceeds in several steps: the text on the FAX is recognized using an optical character recognition algorithm, the text is examined to "spot" candidate words which are likely to be relevant to the addressee's name, and finally all relevant candidate words are matched to the database of email addresses to determine a set of likely matches.

2 The FAX Routing Task

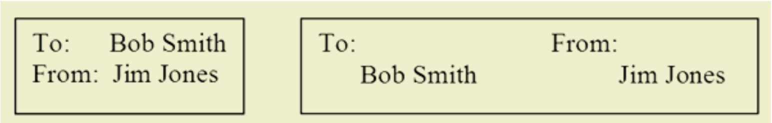
We begin with a description of the task and some observations about typical FAXes.

S. Marinai and A. Dengel (Eds.): DAS 2004, LNCS 3163, pp. 484–495, 2004.

1. We are given a database of email addresses which also includes the first and last name, and optionally a "full name" which is sometimes different from either the first or last name (i.e. the full name can be used to disambiguate people with the same name, and it is used for mailing lists such as "Purchasing Department" or "Legal Affairs"). In our experiments this database included 15,000 recipients.
2. Incoming FAXes come from a very wide range of sources. The quality of these documents is highly varied:
   (a) Most have a structured cover sheet.
   (b) A number are "FAXed back" and actually have the FROM and TO fields reversed (they are intended to be returned to the sender of the original FAX).
   (c) About 40% have handwritten recipient information.
   (d) Low image quality can lead to Optical Character Recognition (OCR) error rates greater than 20%.
   (e) The format of printed cover sheets is often tabular. Since OCR algorithms have difficulty with this type of organization, the extracted document structure (words/lines/paragraphs) is often unreliable.

As mentioned above, in our experiments about 40% of the FAXes contain cover sheets on which the addressee's name is written by hand. In this case conventional OCR cannot be used. On this point we make two observations. In a majority of these cases the cover sheet is a printed document. The printed words on the cover sheet can be used to predict the location of the addressee name, even if the name cannot be recognized. In addition we are currently working to develop and then integrate handwritten text recognition system. ***In this paper we will assume that the recipient information is printed and we will limit discussion to these FAXes.***

At first it may seem as though the FAX routing problem is easy. Our first approach was simple and entirely ad-hoc. The successes and failures of this system motivated our current effort. In this initial approach, the OCR text stream was searched to find anchor keywords such as "To" or "Attention". The immediately following text was then compared to the email address database to find matches with the either the email address or name. Using this technique we were able to correctly route approximately 52% of the FAXes received. While it was possible to propose alternative email hypotheses, this system did not provide an estimate that the given answer was correct.



To: Bob Smith
From: Jim Jones

To: From:
Bob Smith Jim Jones

Fig. 1. On the left is a "good" address, which after OCR is easy to interpret. On the right is a more difficult address block, which after OCR requires additional geometric processing.

We found several sorts of problems with the above system:

1. Missing anchor issues:
   (a) The anchor word was unusual (e.g. “care of”).
   (b) The OCR text stream separated the anchor word from recipient information (e.g. Figure 1)).
   (c) The anchor word is missing entirely.
   (d) The anchor word was corrupted by OCR errors.
2. Poor name matches:
   (a) Name misspelled.
   (b) Name corrupted.
   (c) Name incorrect (e.g. last name is hyphenated in database but on FAX it is one of the two names).

We address the first problem in part by building a robust and flexible statistical word spotter that assigns a relevance score to each word on the page. Words with high score are more likely to contain relevant recipient information. The word spotter is a combination of simple word features, which is *learned* from example FAX data. The word spotter can often spot recipient information when all of the above “missing anchor” issues arise. In a large collection of FAXes the correct recipient information can be spotted more than 95% of the time.

We address the second problem by using an efficient yet robust string matching algorithm. In addition several heuristics are used to deal with matching parts of names.

Finally, information from throughout the FAX is integrated to collect the most robust possible evidence. So if the FAX starts out with a partially corrupt address block (e.g. “Te: Bob Smuth), and the body of the FAX contains “Dear Mr. Smith”, evidence is integrated to conclude that the FAX is for “Bob Smith”.

3 Related Work

In their paper, Lii and Srihari show that the “address block” of a FAX can be extracted using keywords such as TO and ATTENTION [1]. The keywords themselves are found using two heuristics: that they are terminated by a colon or that they are proceeded by a large white space. The position of the keywords, and surrounding words, are used to find the rectangular region which contains the address block.

The FAXAssist system routes FAXes by matching all words in the document to the recipient database using “string edit distance” [2]. The full names in the database are processed to yield common forms such as `LastName, FirstName` and `FirstInitial. LastName`. The match score for each word is further modified using a model for the likely positions of the recipient name.

The name extraction program of Likforman-Sulem, Vaillant, and Yvon uses a collection of features to represent each word on the page [3]. These features are both internal and external. Internal features include tests to see if the word is capitalized, a common word, a common first name, etc. External word features are defined by decomposing the document into blocks. Those blocks which are near words from the recipient class such as TO and ATTENTION are labeled as potentially being a “sender block”. The recipient block is detected similarly. Words in the sender block have the “sender block feature” (likewise with the recipient block). The overall set of features are combined linearly using a set of hand chosen weights.

4 Recipient Information Spotting

The key component of the recipient spotting process is a word scoring function which assigns a score to each word on the FAX page. Machine learning is used to train this function to minimize the error evaluated on a large set of training examples. The scoring function is expressed as a sum of simpler binary word functions which depend on: i) the text of the word; ii) the location of the word; and iii) the spatial relationship to other words. Each feature function has the following form:

$$f_j(w) \in \begin{cases} \alpha_j & \text{if the feature is true} \\ \beta_j & \text{otherwise} \end{cases} \tag{1}$$

Typical examples of binary word functions include:

- Is the word equal to the string “Mr.”?
- Does the word include the substring “.com”?
- Is the word more than 7 inches from the top of the page?
- Is the word within 0.5 inches of the word “Attention”?
- Is the distance to the nearest word greater than 1 inch?

The final word score is computed as $\sum_j f_j(w)$.

Clearly there are many potential binary word functions; we propose a programmatic technique for generating a large set of these combinatorially. In our experiments more than 2000 of these simple binary features are generated before learning is used to select a small set of critical features and to estimate α_j and β_j .

Many of the binary word features are based on an underlying continuous word filter (the term “filter” is chosen to emphasize the continuous nature of the response). Examining the features listed above, three are based on filters with the addition of a threshold:

- Distance of the current word from the top of the page.
- Distance of the current word from the word “Attention”.
- Distance of the current word to the nearest word.

A very large number of filters and features are generated combinatorially from training data and a few basic principles:

1. Word text features:
   (a) One feature is generated for each commonly occurring word in the training database. The feature is true if the FAX word matches this word.
   (b) A single binary feature is generated from all of the words in the email database. The feature is true if the FAX word matches one of these alias words.
   (c) The presence of a common substring from the training data (e.g. ".com" or "@").
2. Location filters: X location, Y location, width and height of the bounding box.
3. Relationship features: true if a common word is within a given distance D which may be 200, 300, or 400 pixels.
   (a) True if word is within D and directly to the left.
   (b) True if word is within D and directly right.
   (c) True if word is within D and either left or above.
4. Relationship filters:
   (a) Distance to the nth nearest word.
   (b) Distance to the nearest word which matches a common string
   (c) The number of words on the current line.

Using the training database to select the 200 most common words and expanding each of the above principles leads to 2128 filters and features.

The AdaBoost machine learning framework is used both to select a subset of the available features, including the filter thresholds, as well as feature scores α_j and β_j [4]. AdaBoost is an extremely simple algorithm which is nevertheless a very effective feature selection mechanism and an efficient learning algorithm. AdaBoost proceeds in rounds; in each round the "best" new feature is selected and added to the classifier. Typically AdaBoost is run for a few hundred rounds to yield a classifier which depends on a few hundred features.

To summarize, the overall framework is to first generate a very large set of features which are related to the classification process, and then second to use AdaBoost to select a small set of these features so that the final classifier is effective and computationally efficient. This basic framework was introduced by Tieu and Viola in their work on image database retrieval and then used by Viola and Jones to produce an extremely fast and effective face detection system [5,6].

4.1 Training the Classifier

The word relevance classifier is trained using a set of labeled data: a set of FAXes upon which OCR has been run, and in which the recipient information has been highlighted. In our experiments we use the ScanSoft SDK [7]. Each word on the FAX is assigned a label $+1$ if it contains information relevant to the recipient identity, and -1 otherwise.

For each FAX in the training set the location of the recipient info is noted by drawing a bounding rectangle, or set of rectangles on the FAX. The rectangle is intended to “highlight” the relevant recipient information. During training, the particulars of the rectangle are discarded. The only information retained is the identity of the words that lie within the rectangle. In practice, there is no attempt to label *all* relevant words. On each FAX only the *most* relevant words are highlighted (usually the contents of the “TO” field). Given that there may be over one hundred words on a FAX page, most of the words are not relevant.

Table 1. Table showing the top scoring features selected by the AdaBoost process.

Score	**Feature**
5.92	Word in current bounding box is in “recipient alias”
4.45	Word in current bbox is a human name
3.59	Word “To” appear on the left (up to 500 pixels)
2.93	Word in current bounding box is “COMPANYNAME” (binary)
2.72	Word “Attn” appears on the left (up to 500 pixels)
2.37	The string ’@’ appears in the current word
2.31	Word in current bounding box is “Business” (binary)
1.68	’Confidence of the word in current bounding box’ ≤ 36
1.61	“Mr” appears on the left (up to 500 pixels)
-5.41	$1601 \leq$ Y co-ordinate of the center of the bounding box
-4.42	The word “Phone” appears on the left (up to 500 pixels)
-3.40	The word “From” appears on the left (up to 500 pixels)
-3.39	$1421 \leq$ Y co-ordinate of the center of the bbox ≤ 1600
-3.32	Distance of 4-th nearest word ≤ 72
-2.52	$1537 \leq$ X co-ordinate of the center of the bounding box
-2.20	$496 \leq$ Width of the bounding box
-1.90	$1157 \leq$ Y co-ordinate of the center of the bbox ≤ 1420
-1.86	$877 \leq$ Confidence of the word in current bounding box

The word scoring algorithm is trained to predict which words appear in the recipient rectangle. The learning problem is therefore binary, each word is given a label of +1 if it is in the rectangle, and -1 otherwise. As described above, AdaBoost selecting from a set of word features is used to predict this label. After training on a set of 2221 FAXes, the correct label is predicted 95% of the time on a separate set of test data. The top features selected for the classifier are shown in Table 1. AdaBoost assigns two scores, or votes, to each selected feature: α if the feature is true and β if false. Since each feature is either true or false is is useful to consider the net score which is $\alpha - \beta$. We sort the features by these net scores.

Positive net scores are associated with relevant words, and negative with irrelevant words. Not surprisingly the most important feature tests if the word is in the recipient alias database. Another expected positive scoring feature tests for the presence of the word “TO” nearby and to the left. The negative features are equally interesting, but less predictable. The most negative states that it is bad to be near the bottom of the page (i.e. 1601 dots assuming 200 DPI is 8 inches). Other negative features penalize words for being near other labels, like “PHONE” and “FROM”. The feature **Distance of 4th nearest word $\leq$ 72** states that if there are 4 words within 0.36 inches, then the word is less likely to be relevant. Essentially, words in the middle of a paragraph are less likely to be relevant. Note that our training set included FAXes with handwritten addresses (about 40%). Often the recipient address was the only handwritten word on the page. This led the system to conclude that words with a low OCR confidence (as handwritten words often are) were more likely to be the address.

The signed scores assigned by AdaBoost are not directly interpretable as a probability. It is not difficult, however to estimate a probability as a logistic function of the score. The parameters of this logistic are estimated using logistic regression on a held out set of labeled examples (held out from the *training* set). The resulting quantity, which ranges from 0 to 1, is approximately the probability that the word is relevant to the recipient identity.

The final classifier yields very good coverage of the FAXes: since a recipient name typically contains two words, the chance of missing both words is very low. There are typically 1 or 2 false positives and generally less than 5 or 10 false positives on a given FAX. The set of relevant words are then matched to the database of alias words as described below. The top scoring words for two typical FAXes are shown in Figures 2 and 3.

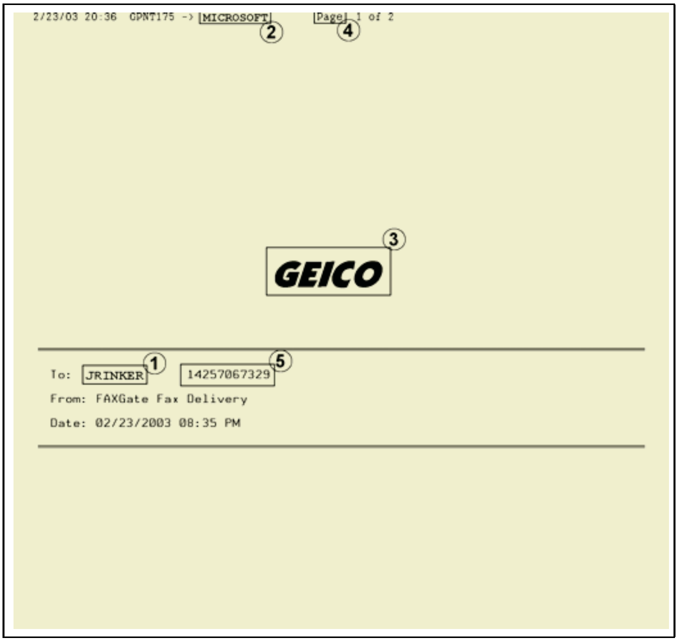
5 Recipient Information Matching

In order to robustly match in the presence of OCR errors, we have chosen to measure the “string edit distance” between words in the alias database and words in the FAXes. The string edit distance between two strings measures the number of characters that must be added to the first string, deleted from the second, or substituted. For example, the string edit distance between “CAATE” and “CAR” is two deletions and one substitution. Based on observation of typical OCR errors, separate costs can be assigned to deletion, addition, and substitution errors.

For use in the FAX routing process, the string edit distance is converted to a match score which ranges from zero to one. We consider this score to be analogous to a probability of the corrupted word given the true word. While we have considered estimating this probability function from data, in this paper we use a surrogate function which yields good results. We define the match score as

$$m(w_1, w_2) = \exp\left(-4\frac{dist(w_1, w_2)}{maxdist(w_1, w_2)}\right) \tag{2}$$

where $dist()$ is the string edit distance and $maxdist()$ is the maximum string edit distance between two strings of this length. Given equal costs for insertion/deletion/substitution the maximum distance is the length of the longer string.



2/23/03 20:36 GPNT175 -> MICROSOFT Page 1 of 2

GEICO

To: JRINKER 14257067329

From: FAXGate Fax Delivery

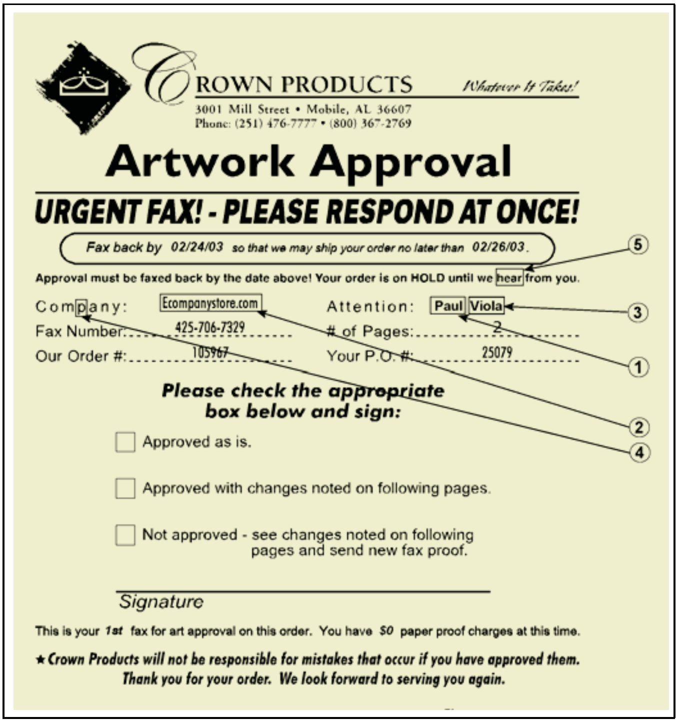
Date: 02/23/2003 08:35 PM

Rank	Word	Score	Probability
1	JRINKER	3.6206	0.632152
2	MICROSOFT	0.4270	0.116035
3	GEICO	-0.8704	0.044132
4	Page	-0.9962	0.040053
5	14257087329	-1.3000	0.031635

Fig. 2. A simple example FAX which shows the top 5 scoring words. Each word is emphasized by enclosing it in a rectangle (not part of the original image), the word rank is shown as a circled number.

String edit distance has a well known dynamic programming solution, which leads to an algorithm with $O(NM)$ complexity, where N is the number of characters in the OCR word and M is the number characters in the alias word [8]. Given K words in the alias database the complexity is naïvely $O(KNM)$ to find the best matching word, a cost which may be prohibitive for large databases. While there is an extremely efficient algorithm for finding the exact match between a word and a large database, a solution for the task of finding the best string under the string edit distance is not quite as clear.

Branch-and-Bound Search. We choose a branch-and-bound search to find the best match, which relies on a cheap underestimate of the string edit distance which we call the “order invariant edit distance”. One guaranteed underestimate of the string edit distance is to ignore the component of the distance that depends on character order. It is computed in the following way. For each word compute a character occurrence vector. The word “CAATE” has two occurrences of A, one of C, one of E and one of T. The word “CAR” has one A, one C, and one



CROWN PRODUCTS *Whatever It Takes!*
3001 Mill Street • Mobile, AL 36607
Phone: (251) 476-7777 • (800) 367-2769

Artwork Approval

URGENT FAX! - PLEASE RESPOND AT ONCE!

Fax back by 02/24/03 so that we may ship your order no later than 02/26/03.

Approval must be faxed back by the date above! Your order is on HOLD until we hear from you.

Company: Ecompanystore.com
Attention: Paul Viola
Fax Number: 425-706-7329
# of Pages: 2
Our Order #: 105967
Your P.O. #: 25079

Please check the appropriate box below and sign:

□ Approved as is.

□ Approved with changes noted on following pages.

□ Not approved - see changes noted on following pages and send new fax proof.

Signature

This is your *1st* fax for art approval on this order. You have *$0* paper proof charges at this time.

★ Crown Products will not be responsible for mistakes that occur if you have approved them. Thank you for your order. We look forward to serving you again.

Rank	Word	Score	Probability
1	Paul	5.16044	0.855890
2	Ecompanystore.com	4.82982	0.819846
3	Viola	1.55514	0.245645
4	p	0.99124	0.171346
5	hear	0.94523	0.166149

Fig. 3. A second more complex FAX which shows the top 5 scoring words.

R. The signed difference between these occurrence vectors is precisely related to the order invariant edit distance: one A and one E must be deleted, and the T must be substituted for an R. This distance is computed $O(N+M)$ time by first measuring the occurrences, and then subtracting the counts for each character.

The branch-and-bound search algorithm is used as follows. Given a FAX word, compute the distance underestimate for each database word, and sort the database elements from least distance to greatest. Starting with the smallest distance database words, compute the true string edit distance for each and reinsert the word in the sorted list based on the true string edit distance (which is always greater than or equal to the underestimate). The first example which is encountered twice (first using the underestimate and then later using the true distance) is the closest example in the database. This is because the true distance for this example is less than the distance underestimate for those examples for which the true distance is not yet known, and less than the true distance for those examples for which the true distance is known.

The complexity of the matching process is no longer deterministic, since it depends on the words in the database. In practice the cost of the match of our database and other typical databases is approximately $O\left(K(N+M)\right)$, since the underestimates are fairly tight.

6 Recipient Information Integration

We have described a scheme for "spotting" the relevant words on a FAX, and a mechanism for matching strings efficiently. Using these modules a set of N relevant words can be extracted from the FAX. Each of these words can be matched efficiently to the database, yielding a set of alias words which match "well". The remaining issue is integration of evidence from these various sources. For this problem we propose two algorithms.

The first algorithm is called "simple weighted score". Each email address in the database is assigned a score using the following formula:

$$s(a) = \sum_w r(w)m(w,a) \tag{3}$$

where $s(a)$ is the score for the alias, the summation is over all words in the document, $r(w)$ is the relevance of the word, and $m(w,a)$ is the best match between the FAX word and one of the text fields in the alias entry (first name, last name, email address, or full name). A simple example: if the string "Bob Smith" appears in the FAX, the evidence for the alias "bob_smith" is good because the word "Bob" votes for the first name of the user bob_smith, while the word "Smith" votes for the last name. Note that the word "Bob" also votes for the alias "bob_riley" and "bob_dean".

The greatest flaw in the simple weighted score is its assumption of word independence. The score for the alias bob_smith is the same if "Bob Smith" is observed or alternatively if "Joe Smith" and "Bob Jones" appear separately.

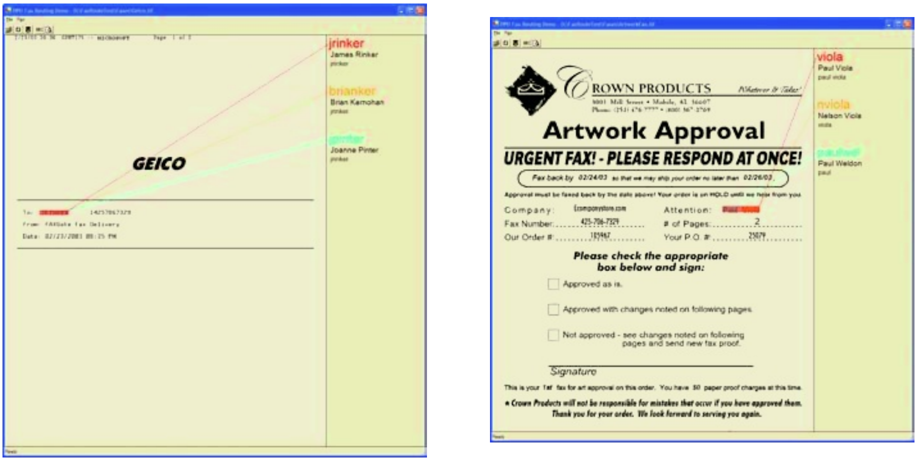
The second proposed matching algorithm is called "contiguous weighted score". It attempts to model the interdependence of words in a FAX document as follows:

$$s(a) = \sum_{w_t, w_{t+1}} C\left(r(w_t),\ r(w_{t+1})\right) m(a, w_t, w_{t+1}) + \sum_w r(w)m(a,w) \tag{4}$$

where w_t, w_{t+1} are contiguous words in the FAX, and $C()$ is a function which combines the confidences of the two words (e.g max, sum, or product), and

$$m(a, w_t, w_{t+1}) = \max \begin{cases} m(\text{'first last'},\ \text{“}w_t\, w_{t+1}\text{”}) \\ m(\text{'last first'},\text{“}w_t\, w_{t+1}\text{”}) \\ m(\text{full\_name},\text{“}w_t\, w_{t+1}\text{”}) \\ m(\text{first},\text{“}w_t\, w_{t+1}\text{”}) \\ m(\text{last},\text{“}w_t\, w_{t+1}\text{”}) \end{cases} \tag{5}$$

Returning to the above example, the FAX string "Bob Smith" now yields a higher score for bob_smith, since there is an additional bonus for matching a



GEICO

To: 14257067329
From: FAXSafe Fax Delivery
Date: 02/23/2003 08:25 PM

CROWN PRODUCTS *Whatever & Takes!*

Artwork Approval

URGENT FAX! - PLEASE RESPOND AT ONCE!

Fax back by 02/24/03 so that we may ship your order no later than 02/26/03.

Approval must be faxed back by the date above! Your order is on HOLD until we hear from you.

Company: Ecompanystore.com Attention:
Fax Number: 425-706-7329 # of Pages: 2
Our Order #: 105967 Your P.O. #: 25079

Please check the appropriate box below and sign:

□ Approved as is.

□ Approved with changes noted on following pages.

□ Not approved - see changes noted on following pages and send new fax proof.

Signature

This is your 1st fax for art approval on this order. You have $0 paper proof charges at this time.

** Crown Products will not be responsible for mistakes that occur if you have approved them. Thank you for your order. We look forward to serving you again.*

Fig. 4. A screen dump of the final system operating on the two FAXes shown above. In this case the top three recipient addresses are shown.

contiguous pair of words. In addition, the last two terms match a pair of FAX words to a single alias entry. This helps in situations where the word boundaries found by OCR are not reliable.

Email aliases that appear directly in a FAX message may require special handling, since the domain may or may not appear (e.g. "pviola" and "pviola@microsoft.com" are equivalent). This is easily handled by adding the alias to the database of alias strings both with and without the appended domain for the receiving organization.

7 Experimental Results

Final routing experiments were performed on a set of 2455 business FAXes received at one company over a period of several months (see Figure 4 for a screen dump). The FAXes varied in type, including personal FAXes intended for employees, purchase orders, and forms filled out by vendors and clients. There were a total of 15,000 addresses in the email database. The distribution followed a predictable Zipf law, in which a large percentage of FAXes were sent to a few of the addresses. In this set of FAXes, there were a total of 723 email addresses (though this was not assumed during testing).

The set of FAXes were randomly separated into two sets of 2221 for "training" and 234 for "testing". The set used for testing was random selected from those which were *not* addressed by hand. On the testing set 95% of the FAXes were routed to the correct recipient. In this experiment that match score of equation 4 is used. The relevance function is computed using the score assigned by the boosted classifier after 300 rounds of boosting.

8 Conclusions

The main contribution of this paper is to present a single unified machine learning process which can used to estimate the relevance of each word on a FAX page. The learning algorithm removes most of detailed engineering and hand tuning required of the system builder, since it optimally combines word text features, word relationship features, and global features such as the location on the page. As a result this system is more likely to be applicable to related problems, such as the extraction of information from other types of scanned documents.

References

1. Lii, J., Srihari, S.N.: Location of name and address on fax cover pages. In: International Conference on Document Analysis and Recognition. (1995) 756–759
2. Tupaj, S., Dediu, H., Alam, H.: Faxassist: an automatic routing of unconstrained fax to email location. In: SPLI Document Recognition and Retrieval VII. (2000)
3. Likforman-Sulem, L., Vaillant, P., Yvon, F.: Proper names extraction from fax images combining textual and image features. In: International Conference on Document Analysis and Recognition. (2003) 545–549
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55** (1997) 119–139
5. Tieu, K., Viola, P.: Boosting image retrieval. In: International Conference on Computer Vision. (2000)
6. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2001)
7. ScanSoft: Scansoft optical character recognition sdk (2002)
8. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. J. ACM **21** (1974) 168–173