Information extraction by text classification

Nicholas Kushmerick Edward Johnston Stephen McGuinness
Smart Media Institute, Computer Science Department, University College Dublin
nick@ucd.ie

Abstract

Information extraction and text classification are usually seen as complementary forms of shallow text processing, in that they are aimed at very different tasks. In this paper, we describe two simple but real-world domains in which text classification techniques can be used directly for information extraction. Specifically, we describe systems for extracting information from business cards, and for automatically processing "change of address" email messages, that are based primarily on text classification techniques. Our main technical contribution is a novel integration of hidden Markov models and text classifiers.

1 Introduction

Text classification (TC) is the task of assigning one or more predefined categories to a text document; see [Yan99] for a survey. For example, an email system might assign to messages either the category "junk" or "non-junk", perhaps in order to decide whether to automatically delete the message. As a second example, one might want to assign categories such as "International", "Sport", "Business", *etc.* to newspaper articles. In contrast, information extraction (IE) is the task of finding particular fragments of documents that instantiate some predefined information need or concept; see [CL96] for a survey. For example, an IE system might identify fragments of text expressing the title, company, location and salary details from job advertisements.

TC and IE are both forms of shallow text processing, but interaction or synergy between the techniques has been quite minimal. Many deployed IE systems use some from of TC to ensure that the documents to be processed are likely to contain the expected data. Going the other way, one could envision a TC system that assigns categories based on fragments extracted during an initial IE step.

In this paper, we explore how TC and IE can be integrated more closely. There has been negative results reported on using TC for many challenging IE tasks (*e.g.* [Cal98]). Nevertheless, in this paper we describe two simple but real-world IE applications that successfully use TC techniques to guide extraction decisions. One application involves automatically populating a contact database from business cards; the second involves automatically processing "change of address" email messages. Our preliminary results demonstrate that TC can be used fruitfully for IE in these relatively structured domains.

2 Populating a contact database from business cards

The first task is to convert physical business cards into entries in a database of contact details. The first step in achieving this task is to use optical character recognition techniques to convert a physical card into a text document. We used a corpus of 505 business cards from Cardscan.com,

O	knowledge and know-how
C	TECHSMITH INC.
O	since 1979
N	Seymour Mermelstein
T	President
A	223 Park Street
A	Newton, MA 02158 USA
P	(617) 332-1003
P	fax: (617) 332-1090

Figure 1: An example business card, annoted with the meaning of each field.

a manufacturer of portable scanners. Our business card extraction system based on this training data has been fully implemented; a demonstration is available at `www.smi.ucd.ie/iebc`.

Business card corpus. The 505 physical cards scanned by Cardscan.com resulted in 4249 lines of text, each of which was then manually annotated. Figure 1 shows a representative example. Each line is annotated with one of eleven symbols: **A**ddress, **N**ame, **T**itle, **C**ompany name, company **L**ogo, **P**hone number (either voice or fax), **W**eb address, **E**mail address, **T**elex number, ca**B**le number, and **O**ther miscellaneous text. Note that a card may contain more than one instance of a given field, and need not contain every field.

One complication (not shown in Figure 1) is that the Cardscan.com annotators sometimes marked a line with more than one symbol. For example, the corpus contains a card with the line "BLS" labelled **OCL**, and another line "BLS ENGINEERING INC. TEL:(617) 964-8097" labelled **CP**. These examples indicate the two different reasons why lines were sometimes annotated with multiple symbols. In the first case, the human annotator apparently believes that it is genuinely ambiguous whether the text "BLS" is **O**ther miscellaneous text, the **C**ompany name, or the company **L**ogo. In the second case, the line does in fact contains both a **C**ompany name and a **P**hone number. In our corpus of 505 business cards 9.1% of the lines have such multiple labels. Our current algorithms do not attempt to distinguish between these two reasons for multiple labels, or to predict multiple labels during testing. Instead, we interpret multiple labels as noise. Specifically, during training we treat a line as a training example for each of its labels in turn, and during evaluation we treat a predicted label as correct if it matches any of the "true" labels.

Naïve extraction algorithm. We treat the task of populating a contact database from such a corpus of scanned business cards as a TC problem. The documents to be classified are the individual lines ℓ_i of scanned text, and the categories f are the eleven symbols described above. To process a card, our algorithm computes the probability $\Pr[f|\ell_i]$ that line ℓ_i of the card is labelled f. We use the naïve Bayes text classification algorithm (see [MN98] for a review) to compute $\Pr[f|\ell_i]$. Our algorithm then labels the lines with the sequence $f_1, f_2, \ldots$ that maximizes $\Pr[f_1, f_2, \ldots|\ell_1, \ell_2, \ldots]$. To compute this probability, we assume independence ($\Pr[f_1, f_2, \ldots|\ell_1, \ell_2, \ldots] = \prod_i \Pr[f_i|\ell_i]$) and thus this probability can be easily maximized by maximizing each term in turn.

Our experiments indicate that this algorithm assigns the correct labels 71% of the time, using a cross-validated 50%-50% training/testing split of the corpus.

Taking advantage of structure. This naïve approach is reasonably accurate, but it ignores important structural constraints. Specifically, business card fields tend to occur in stereotypical orders; for example, it is quite likely that the card-holder's **N**ame is followed immediately by a **T**itle, or that **C**ompany names are followed by **A**dresses. On the other hand, it is rare that card-holders' **N**ames are preceded by **A**dresses.

Such constraints are naturally handled by hidden Markov models (HMMs). An HMM is a probabilistic automaton, defined by a set of states S, a prior distribution over states $\pi(s)$ for $s \in S$, a set of tokens T, a token emission distribution $\Pr[t|s]$ for $t \in T$ and $s \in S$, and a state transition distribution $\Pr[s'|s]$ for $s \in S$ and $s' \in S$. HMMs are widely used for IE by using one state for each field to be extracted (as well as additional "background" states that emit unneeded tokens). Standard HMM training procedures are used to tune the model parameters. To perform IE, the Viterbi algorithm is used to efficiently identify the sequence of states that most likely generated the document. See [Lee97, FM00] for examples of such efforts.

In our business card application, the fact that the units to be classified are entire lines (rather than individual tokens) means that we use a somewhat non-standard HMM. Rather than emitting words, states in our HMM emit lines of text, where the probability of emitting a line is estimated using the naïve Bayes classifier as described above, rather than the estimate provided by standard word-based HMM approaches.

Specifically, for each field f we estimate a prior $\pi(f)$ as the fraction of cards starting with field f. Then, for each field f' (as well as a special field EOF indicating the end of the card), we estimate the state transition probability $\Pr[f'|f]$. In terms of the earlier examples, in our corpus we observe that $\Pr[\mathbf{T}|\mathbf{N}] = 0.80$ and $\Pr[\mathbf{A}|\mathbf{C}] = 0.43$, while $\Pr[\mathbf{N}|\mathbf{A}] = 0.02$. Finally, we train a naïve Bayes classifier in order to calculate $\Pr[f|\ell_i]$ for field f and any line of text ℓ_i. The probability that a business card with lines of text $\ell_1, \ldots, \ell_N$ was generated by the field sequence $f_1, \ldots, f_N, f_{N+1}$ is thus $\pi(f_1) \prod_i \Pr[f_i|\ell_i] \Pr[f_{i+1}|f_i]$, where $f_{N+1} \equiv \mathsf{EOF}$.

This HMM approach yields an accuracy of 74% on a cross-validated 50%/50% training/testing split of the data, compared to 71% for the simple classification approach described above.

We can unify the naïve and HMM approaches as follows. The naïve approach ignores structural information. Equivalently, the naïve approach selects precisely the one state most likely to have generated each line, regardless of the structural implications. In contrast, the HMM approach considers all $|S|$ states, even ones that are unlikely to have generated a particular line of text. That is, due to the transition probabilities, the best state sequence might contain states that are relatively unlikely when considered in isolation.

Figure 2 generalizes this relationship, showing the extraction accuracy as a function of the "search depth": at search depth D, the algorithm considers the D most likely states to have generated each line. Counterintuitively, we see that accuracy does not increase monotonically with depth, but rather decreases at intermediate depths and then increases as this mechanism starts to better approximate the full HMM approach. We do not understand this nonmonotonic behavior, and leave a further exploration of these issues to future work.

We believe (though have not yet empirically confirmed) that we can further improve accuracy by considering additional structural constraints. For example, business cards usually contain exactly one **N**ame, so we want to modify our HMM so that only state sequences with exactly one **N**ame have non-zero probability. Unfortunately, such constraints are inherently non-Markovian, and cannot be modeled in an HMM without an infeasibly large state space. To address this problem, we revise the Viterbi extraction framework as follows. Suppose that one wants to compute the most likely state sequence that conforms to some constraint C (*e.g.* C = "extract exactly one **N**ame"). While the most likely sequence might not conform to C, we can heuristically assume that the best

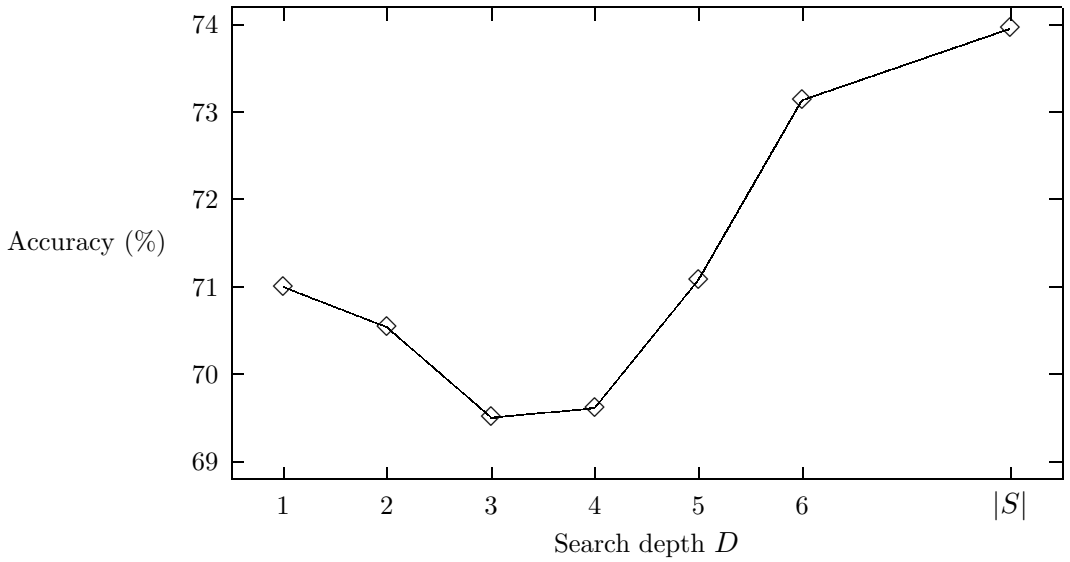

Figure 2: Extraction accuracy for business card domain, as a function of search depth D. $D = 1$ corresponds to the naïve extraction algorithm; $D = |S|$ is the HMM extraction algorithm.

conforming sequence is within the best dozen (say). We have extended the Viterbi algorithm so that it computes the R most probable state sequences in time $O(R \cdot N \cdot |S|)$, compared to the standard Viterbi algorithm that generates the best (*ie*, $R = 1$) state sequence in time $O(N \cdot |S|)$. In our proposed revised framework, we would pick a value for R that we guess is "large enough", invoke the extended Viterbi algorithm to generate the best R state sequences, and select the first the conforms to C.

3 Automatically processing of "change of address" email messages

Figure 3 shows a representative example of a "change of address" (CoA) email message. Our goal is to develop a system that determines whether an email message is in fact such a CoA message, and if so identify the updated address. The intent is that such functionality could be embedded in an email application, enabling the application to prompt the user with a recommended update to his address book. Our CoA system has been fully implemented and integrated into the Pegasus email system; a demonstration is available at `www.smi.ucd.ie/coa`.

We solve this problem in two phases. Using ordinary text classification techniques, we classify the entire email messages as either "CoA" or "non-CoA". If the message is classified as "CoA", we then estimate the probability that each email address in the message is the desired new address, and select the most likely. We now discuss each step in turn, and then present our experimental results.

Message classification. The first step is to classify a message as either "CoA" or "non-CoA". We experimented with both the naïve Bayes and probabilistic indexing [FB91] algorithms. We found that the latter performed somewhat better, and so report results only for this algorithm.

To help the classifier overcome the limitations of bag-of-words model, we enriched the document

From: Robert Kubinsky <robert@lousycorp.com>
Subject: Email update

Hi all - I'm moving jobs and wanted to stay in touch
with everyone so....
My new email address is : robert@cubemedia.com
Hope all is well :)
>>R

Figure 3: An example "change of address" email message.

with unique special tokens for every pair and triplet of adjacent terms. For example, the fact that a message contains the term "new" or "address" is, by itself, not particularly informative, but the two-word phrase "new address" is highly informative.

Address classification. If a message has been classified "CoA", then the next step is to extract the new email address from the text of that message. Note that, unlike many IE tasks, the difficulty is not in identifying candidate fragments—email addresses can be readily identified with a simple regular expression—but deciding which fragments are correct. Our approach is to consider a window of words surrounding each email address, and classify the address as "new" or "old" based on the text in these windows. We used a window of ten words, seven words before the address, and three words after. As before, we then enrich the windows with two- and three-word phrases.

Results. We gathered and manually annotated a corpus of 36 CoA messages and 5720 non-CoA messages. This distribution was chosen to roughly approximate the distribution that might be encountered in reality. The non-CoA messages were simply a set of messages that had been received by the first author over a particular interval, manually examined to remove CoA messages. We obtained additional CoA messages by asking a group of colleagues and friends to compose "artificial but realistic and natural" CoA-style message, *without* telling them about the goal of or techniques used in this research. The CoA messages contain an average of 2.4 email addresses each, of which 39% are the old address.

Our results are indicated in Figure 4, for both the original message/address, and also when enriched with phrase tokens. We report accuracy in terms of the standard precision, recall and F_1 metrics. We see that performance is reasonably good on each step individually, and overall extraction accuracy is 96%. However, as is well known, accuracy can be a misleading indicator. In particular, this 96% accuracy is a weighted average between an accuracy of 66% for the 0.6% of messages that are in fact CoA messages, and 97% accuracy for the 99.4% messages that are non-CoA. Unfortunately, it is difficult to generalize the definitions of precision and recall in this cascaded approach.

These data are based on a cross-validated 50%-50% training/testing split of the corpus of 5756 messages. We note that this performance is thus based on just 18 CoA training messages, a very small sample by the standards of IE and TC. In order to estimate how much improvement might be obtained with additional training data, Figure 5 shows how the F_1 value for each of the two classification steps varies as a function of the fraction of available data used for training. We report here only the results for enriching the documents with the phrase tokens. We see that address classification F_1 improves marginally beyond 50%.

	Words			Phrases		
	P	R	F_1	P	R	F_1
Message classification	.96	.66	.78	.98	.97	.98
Address classification	.96	.62	.76	.98	.68	.80
Overall accuracy				96%		

Figure 4: Results (in terms of precision, recall and F_1) for "change of address" message and address classification, and overall extraction accuracy for the cascaded algorithm.

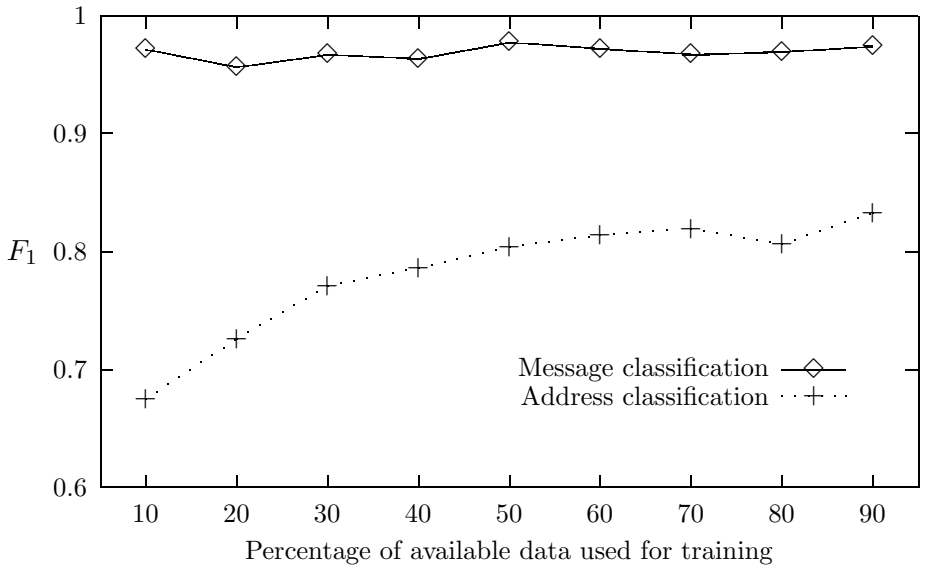

Figure 5: Message and extraction classification F_1 performance, as a function of the amount of available data used for training.

4 Discussion.

We have described two simple but real-world information extraction tasks that can be achieved by text classification techniques. In the case of business cards, we can populate a contact database by classifying individual lines of text. In the "change of address" domain, we extract the new email address by examining words in a window surrounding each candidate. Perhaps the most interesting technical aspect of this work is the novel integration of HMM and statistical text classification techniques, in which HMM states emit token sequences rather than individual tokens.

We do not claim that these techniques can be applied to all IE tasks, but there are some general properties of these tasks that may hold of others. The first key to success is that a relatively small set of candidate fragments can be readily identified. In the case of CoA messages, we can easily find email addresses; in the case of business cards, OCR techniques can easily recover line breaks and other formatting information. A second key to success is that correct candidates can be distinguished from incorrect based on the particular words that occur either in the surrounding context or in the candidates themselves. While not all IE tasks have such properties, we believe there is plenty of "low hanging fruit" as IE systems are deployed in ever more settings.

Acknowledgments. We thank Dexter Sealy of Cardscan.com for providing the business card data, Andrew McCallum for the use of his "Bow" text classification implementation, and Rob Solmer for helpful comments. This research was funded in part by a President's Research Grant from University College Dublin, and grant N00014-00-1-0021 from the US Office of Naval Research.

References

[Cal98] M. E. Califf. *Relational Learning Techniques for Natural Language Information Extraction.* PhD thesis, University of Texas at Austin, 1998.

[CL96] J. Cowie and W. Lehnert. Information extraction. *C. ACM*, 39(1):80–91, 1996.

[FB91] N. Fuhr and C. Buckley. A probabilistic learning approach to document indexing. *ACM Trans. Information Systems*, 9(3):223–248, 1991.

[FM00] D. Freitag and A. McCallum. Information extraction with HMM structures learning by stochastic optimization. In *Proc. 17th Nat. Conf. AI*, pages 584–589, 2000.

[Lee97] T. Leek. Information extraction using hidden Markov models. Master's thesis, University of California, San Diego, 1997.

[MN98] A. McCallum and K. Nigam. A comparison of event model for naive Bayes text classification. In *Proc. AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[Yan99] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1–2):69–90, 1999.