Linguistics 384: Language and Computers

Topic 3: Text classification

Scott Martin*

Dept. of Linguistics, OSU

Autumn 2007

* The course was created together with Chris Brew, Markus Dickinson and Detmar Meurers.

Outline

Introduction

- Language Identification
- Spam

Outline

Introduction
- Language Identification
- Spam

Language Technology
- Rule-based approaches
- Statistical approaches
- Devious spam

Outline

Introduction

- Language Identification
- Spam

Language Technology

- Rule-based approaches
- Statistical approaches
- Devious spam

Practical aspects

Introduction: Document classification

- Identifying junk e-mail (spam) vs. wanted e-mail (ham) is essentially a task of **document classification**.
- Document classification means taking documents and a set of relevant categories and figure out which documents belong into which category.
  - For example, email sent to the New York Times could be classified into letters to the editor, new subscription requests, complaints about undelivered papers, job inquiries, proposals to buy ad pages, and other
- Can we do such classification tasks automatically?
  - An example: Language identification

Language identification

- We can attempt to classify documents according to the language a document is (mostly) written in.
- Can sometimes tell by
  - which characters are used,
    - e.g. *Liebe Grüße* uses ü and ß → German
  - which character encoding is being used
    - e.g., ISO 8859-8 is used to encode Hebrew characters → text is written in Hebrew
- But how can you tell if you are reading English vs. Japanese transliterated into the Roman alphabet? Or Swedish vs. Norwegian? And all phonetically transcribed text is encoded in the same IPA encoding!
- Consider what you base your guess on when I ask whether the following is Portuguese or Polish: *Czy brak planów zagospodarowania hamuje rozwój Warszawy?*

Language identification

N-grams

- One simple technique for identifying languages is to use **n-grams** = stretch of *n* tokens (i.e., letters or words):
  - Go through texts for which we know which language they are written in and store the n-grams of letters found, for a certain *n*.
    - e.g., extracting the trigrams (3-grams) for the last sentence we'd get: *Go* , *o t*, *th*, *thr*, *hro*, *rou*, . . .
  - This provides us with an indication of what sequences of letters are possible in a given language (and how frequent they occur).
    - e.g., *thr* is not a likely Japanese string.
- How do we make this more concrete?

Language identification

Frequency distributions

- Store a **frequency distribution** of trigrams, i.e., how many times each n-gram appears for a given language.

n-gram	English	Japanese
aba	12	54
ace	95	10
act	45	1
arc	8	0
...	...	

- Now, apply the frequency distribution to a new text and use it to help calculate the probability of the text being a particular language.
  - Compare each n-gram to see if it is more likely to be English or Japanese.
  - See which language won the most comparisons.

Language identification

Different techniques

- Although n-grams do not capture abstract linguistic knowledge, they are a simple and surprisingly effective technique, used throughout computational linguistics.
- Another simple technique for language identification would be to look for keywords in the documents, e.g., *capture* → English, *je* → French, etc.
  - Requires knowledge which words are the best indicators for a particular language.
  - Words occurring frequently and independent of the topic of the text are best, e.g., so-called function words like articles (e.g., in English *the*, *a*, . . .), complementizers (e.g., in English *that*, *whether*, *if*, . . .).

From language to spam identification

- The general idea of looking for recurring patterns of language carries over to identifying spam.
- **spam** = e-mail we don't want, usually only loosely directed to us, including unsolicited commercial e-mail
- Structure of discussion:
  - The issue and its social context
  - Language technology: rule and statistical methods
  - Devious spam
  - What you can do about spam

The issue

- Spam consumes
  - a significant fraction of total Internet bandwidth, which causes both a slowdown of other traffic, and possibly raises overall bandwidth cost.
  - a large amount of storage space on mail servers, sometimes actually making it temporarily impossible for legitimate messages to be received.
  - a significant portion of the time and effort of people who use email to communicate.
- Spam can be the vehicle of identity theft campaigns, other types of fraud, and virus propagation.
  - See http://www.secretservice.gov/alert419.shtml for an explanation of Nigerian Advance Fee Fraud.

(based on *Spam: The Phenomenon* by Colin Fahey,

http://www.spiralsolutions.net/spam_topics/)

How spam works

- A spammer obtains email addresses, e.g., by sending out robots to collect e-mail addresses from web-sites and newsgroups, or by buying (legally or illegally created) address databases
- To that collection of addresses, the spammer often automatically generates other possibilities. e.g., "I've found `smith.1@osu.edu` and `smith.12@osu.edu`. What if I try other `smith.#@osu.edu` combinations?"
- A message is sent out. The spammers are aware of various filters and so try to make their messages devious.

(cf. http://www.philb.com/spamex.htm)

The social context

- Spammers are trying to make money by selling a product
- Sending email is virtually free, even if millions of messages are sent
- Enough people fall for spam to make it worthwhile

- But the negative consequences of spam on our resources are well-established, so how can the problem be addressed
    - Laws don't seem to work well: spammers use other countries, are hard to trace.
    - Checking to see if a human is on the other end before accepting an e-mail takes extra time and effort.
    - Charging for e-mails would mean the end to e-mail as we know it.

Language Technology

- Set up **spam filters** = programs which classify incoming mail into ham vs. spam, saving the latter in a junk-mail folder (or just delete it).
- Spam filters can be set up to filter mail
  - for an individual account → can take user specific properties into account
  - for an entire site
- Two general types of language technology can be used for this:
  - Rule-based filters
  - Statistical filters

Basic filtering

In setting up an e-mail account, you generally can set up the use of several folders and direct message accordingly.

- Send all mail with `espn.com` in the sender address to a separate *sports* folder.
  ⇒ Store messages you don't need immediate access to.
- Delete all mail from `viagra@spam.com`
  ⇒ If you get mail from an address which never sends anything good (i.e., always spam), you never want to see it. You've effectively **blacklisted** it.
- Send all mail from my brother directly to my inbox.
  ⇒ Some messages you'll always want to see right away. You **whitelist** these.

Rule-based filters

This is basically **rule-based filtering** = filtering e-mail based on set rules.
But rule-based spam filters can be more sophisticated:

- can **weight** patterns detected by the rules:
  - e.g., 3 points for *viagra* in the header, 2 for originating from a hotmail account, -2 points for a ".edu" address, . . .

  ⇒ When you pass some threshhold of points, it's marked as spam.
- can use information about systems it knows about:
  - e.g., This html message came from Outlook, but Outlook can't send pure html messages

Spam example

Spam detection software (here: spamassassin) has identified this incoming email as possible spam. It provides:

- Content preview:

  > *Email Marketing Email more than 2,500,000+ TARGETED prospects EVERYDAY! That's over 75,000,000+ prospects per month (and growing!). Our Optin email safelists are 100% Optin and 100% legal to use. Your ad will reach only those prospects who have requested to be included in Optin safelists for people interested in new business opportunities, products and services. [. . .]*

- Content analysis details: (11.2 points, 5.0 required)

Rules

pts	rule name	description
0.1	HTML-TAG-EXISTS-TBODY	BODY: HTML has “tbody” tag
0.1	HTML-FONTCOLOR-RED	BODY: HTML font color is red
0.1	HTML-FONTCOLOR-BLUE	BODY: HTML font color is blue
0.1	MIME-HTML-ONLY	BODY: Message only has text/html MIME parts
0.0	HTML-MESSAGE	BODY: HTML included in message
0.1	HTML-FONT-BIG	BODY: HTML has a big font
0.1	HTML-LINK-CLICK-HERE	BODY: HTML link text says “click here”
0.2	NORMAL-HTTP-TO-IP	URI: Uses a dotted-decimal IP address in URL
0.0	FORGED-HOTMAIL-RCVD	Forged hotmail.com ’Received:’ header found

Rules (cont.)

pts	rule name	description
3.0	NO-RDNS-DOTCOM-HELO	Host HELO'd as a big ISP, but had no rDNS
1.6	FORGED-MUA-OUTLOOK	Forged mail pretending to be from MS Outlook
1.1	FORGED-OUTLOOK-TAGS	Outlook can't send HTML in this format
0.0	CLICK-BELOW	Asks you to click below
1.9	MIME-HEADER-CTYPE-ONLY	'Content-Type' found without required MIME headers
1.7	HTML-MIME-NO-HTML-TAG	HTML-only message, but there is no HTML tag
1.1	FORGED-OUTLOOK-HTML	Outlook can't send HTML message only

Problems with Rule-based filters

Rule-based filters are quite intuitive and can be highly effective, but they also have drawbacks:

- Someone has to identify a pattern and specify a rule matching it (with high precision/recall).
- The more rules there are, the better it detects, but the slower it runs.
- Rule-based filters by nature are a step behind the spammers:
  - rules can only be developed once a pattern has been observed in spam, and
  - once a spammer knows a rule, they will can try to bypass it.

Statistical filters

- Statistical filters have been proposed in place of or in addition to rule based ones.
- Instead of providing hand-written rules, one provides large sets of examples, one set with messages known to be spam, another with messages known to be ham.
- How it works:
  - Count up occurrences of words in previous e-mails:
    - How many times does X appear in something flagged as spam?
    - How many times does X appear in something which isn't spam? (i.e., is ham)
  - From these counts, we calculate the **spam probability** of a word.

Calculating probability example

- Setup
  - *cash* appears in 203 e-mails, 200 of which are spam, 3 of which are real.
  - In total, there are 1500 messages, 1000 spam mails and 500 real e-mails.
- So, in 20% of spam messages (200/1000), *cash* appears, while it appears in only 0.6% of real messages (3/500).
- We calculate the probability of *cash* appearing in spam as: $0.20/(0.006 + 0.20)$ = 0.971, i.e., about 97%

Detecting spam

- We calculate this probability for every word.
- When a new e-mail comes in, we extract all the words and find their probabilities.
- We pick the 15 (or so) words which are the best and the worst indicators of spam (farthest from the middle) i.e., Pick the 15 words which give the strongest indication as to the true contents of the message.
- Combine these probabilities into a single probability
- If the probability is high enough (maybe 90% or more), call it spam.

Detecting spam example

So, let's say that you get an e-mail from me saying:

> *Hey, class, I just heard about a great opportunity in Nigeria to study and even make money.*
>
> *. . .*
>
> *I've also put the quiz online and asked one of the linguistics students to take it for a test drive so we can be pretty sure it works.*
>
> *Scott*

Example continued

- We extract words with high probabilities of being spam: *opportunity*, *Nigeria*, *money*, . . .
- and words with low probabilities of being spam: *linguist*, *quiz*, and possibly *Scott* [it's often hard to realistically fake an acquaintance's name]

We combine these probabilities, and it turns out that *opportunity* and *money* are indicators of spam, but *quiz* and *linguistics* are very good indicators of non-spam.

Recalculating

Note that at some point, this non-spam e-mail will itself be used in recalculating probabilities for words.

- That is, the spam filter is continually **learning** what is spam and thus adapting to new spam techniques
- As with general document classification, this idea of **machine learning** is very important & widely-used.

Machine learning = computer learns how to behave based on previously-seen data.

Some perks of statistical filtering

Paul Graham (http://www.paulgraham.com/wfks.html) list of the benefits of statistical filters:

1. They're effective: they tend to catch 99% of spam.
2. They generate few **false positives** = real e-mails mistakenly treated as spam
3. They learn.
4. They let the user define what spam is → one person's spam is another person's golden opportunity
   e.g., I may hate the espn.com messages I get, but others want to know when fantasy football starts up
5. They're hard to trick → two ways to fake the statistical filters: use fewer bad words, or use more innocent words.
   ⇒ But the innocent words are defined by the user.

Devious spam

- Spam filters try to distinguish spam from ham, using rules and patterns of word occurrences that it has learned about.
- Spammers want to disguise their messages so that they trigger none (or only few) of the rules and do not contain occurrences of words typical for spam.
- Emails are often encoded in HTML (hypertext markup language), so we need to talk about this encoding before we can take a closer look at various spammer tricks.

HTML

The Hypertext Markup Language (HTML) provides meta-information which tells a web browser or mail reader how a document is structured and how it should be displayed.

- HTML markup has beginning and end tags
  - <b>Example</b>: tells the browser to render the text Example in bold, i.e. as **Example**
- An HTML tag can have attributes
  - For example, *color* is an attribute of the *font* tag.
  - <font color=”blue”>Language</font> makes *Language* appear blue

Tricks with spaces and characters

Make words which are good indicators for spam look less like words:

- Space out words to make them unrecognizable to word detectors
  e.g., M O R T G A G E
- Other characters can be used instead to space things out
  e.g., F*R*E*E V'I'A'G'R'A O!NL#I$N%E

⇒ Spam detection software needs to keep up with spammers' tricks for encoding words.

Trick characters

If you can alter characters, words won't appear as the same words which are frequently found in spam.

- Replace letters that look like numbers with numbers
  e.g., V1DE0 T4PE M0RTG4GE
- Use accented characters in English
  e.g., Fántàstìc – earn mõnéy thrôugh unçõlleçted judgments

⇒ Spam detection software needs to undo these mappings

Split words with empty HTML tags

- Make it so that a single suspect word isn't seen as a single word by the detector—but it is seen by the human as a single word.
  e.g., milli<! xe64 >onaire

⇒ **Lesson:** Filters are going to need to understand HTML very well.

Invisible Ink

Spammers do things which can mess up your spam filter by secretly including words which make the e-mail sound legitimate, but which the e-mail user never sees.

- Add some real random words before HTML.
  suspensory obscure aristocratical meningorachidian unafeared brahmachari
  <html>
- Write white text on a white background
  <font color=”white”>suspensory obscure aristocratical meningorachidian unafeared brahmachari</font>

⇒ Spam filters should include in their calculation exactly what the user sees.

Do you see what I see?

One especially devious tactic involves taking English text and dividing it vertically

- Take the English text and instead of printing it out horizontally, print it vertically in a table
- The result will look like English to the user, but will only be word fragments to the parser.

⇒ Again, filter needs to see what the human sees.

Hiding the contents in other media

- Intead of encoding a message in a text, spammers
  - send images
  - send http links to images

    Note: By having each spam message load a different image name, the image loading can function as a message to the spammer signaling this message has been read.
  - send programs (javascript), which when executed get the text from another computer, essentially loading a web page
- Relies on the mail reader to be able to display images and execute programs.

⇒ Very hard to detect as spam, but since the use of these features for benign purpuses is not common, one can just switch off the loading of images and deny execution of programs in general.

What to do?

So, now that spammers are adding “good” words and hiding “bad” ones, what can we do?

- Just throw our hands up and start looking into these great mortgage deals. ;-)
- Mix statistical filters (considers the good) and rule-based filters (still finds the bad).
- Work to make sure that the filters see what the human sees.

What you can do about spam

Negatives

- Don't ever buy anything advertised through spam—if everyone observed this, spamming would not pay off and stop existing.
- Be careful about:
  - Asking to be taken off a list.
    Clicking on "remove me," or replying to spam mail will let them know your e-mail is valid.
  - Posting to a newsgroup which publicly archives their messages
  - Marking (or, more likely, not unmarking) that box when signing up for an account which says something like "I'd like to receive offers . . . "
  - Posting your e-mail on your website or in newsgroups.

What you can do about spam

Positives

- Things you can do:
  - Create accounts specifically used for newsgroups and such
  - Make your e-mail address on your website readable *only* to humans.
    e.g., gee.1ATosuPERIOD—and don't forget that "edu" at the end
  - Use a properly configured spam filter (e.g., the free spamassassin is very well configurable)
  - Avoid loading images in emails by default, if possible.