HG2052

Language, Technology and the Internet

Lang Identification/Normalization

Francis Bond
Division of Linguistics and Multilingual Studies
http://www3.ntu.edu.sg/home/fcbond/
bond@ieee.org

Lecture 8

Revision of the Web as Corpus

- Direct Query: Search Engine as Query tool and WWW as corpus?
  (Objection: Results are not reliable)
  - Population and exact hit counts are unknown → no statistics possible.
  - Indexing does not allow to draw conclusions on the data.
  - ⊗ Google is missing functionalities that linguists / lexicographers would like to have.

- Web Sample: Use search engine to download data from the net and build a corpus from it.
  - known size and exact hit counts → statistics possible.
  - people can draw conclusions over the included text types.
  - (limited) control over the content.
  - ⊗ sparser data

Direct Query

- Accessible through search engines (Google API, Yahoo API, Scripts)
- Document counts are shown to correlate directly with "real" frequencies (Keller 2003), so search engines can help - but...
  - lots of repetitions of the same text (not representative)
  - very limited query precision (no upper/lower case, no punctuation...)
  - only estimated counts, often hard to reproduce exactly
  - different queries give wildly different numbers

Web Sample

- Extracting and filtering web documents to create linguistically annotated corpora (Kilgarriff 2006)
  - gather documents for different topics (balance!)
  - exclude documents which cannot be preprocessed with available tools (here taggers and lemmatizers)
  - exclude documents which seem irrelevant for a corpus (too short or too long, word lists,...)
  - do this for several languages and make the corpora available

Internet Corpora: Outline

1. Select Seed Words (500)

2. Combine to form multiple queries (6,000)

3. Query a search engine and retrieve the URLs (50,000)

4. Download the files from the URLS (100,000,000 words)

5. Postprocess the data (encoding; cleanup; tagging and parsing)

Sharoff, S (2006) Creating general-purpose corpora using automated search engine queries. In M. Baroni, S. Bernardini (eds.) WaCky! Working papers on the Web as Corpus, Bologna, 2006.

Internet Corpora Summary

- ➢ The web can be used as a corpus
  - ➢ Direct access
    - ∗ Fast and convenient
    - ∗ Huge amounts of data
    - ⊗ unreliable counts
  - ➢ Web sample
    - ∗ Control over the sample
    - ∗ Some setup costs (semi-automated)
    - ⊗ Less data
- ➢ Richer data than a compiled corpus
- ⊗ Less balanced, less markup

Language Identification

What is Language Identification?

- Given a document and a list of possible languages, in what language was the document written? (e.g. English, German, Japanese, Uyghur, ...)
- Orthography? (i.e., does the language have an agreed written form?)
- A solved problem?

An Example

What is the language of the following document:

Seperti diberitakan, Selasa, Megawati optimistis memenangi sengketa pilpres. Sementara itu, Yudhoyono dalam ceramah di kediamannya di Cikeas, Senin malam, menyatakan, tuduhan kecurangan merupakan pencemaran nama baik.

A Second Example

What is the language of the following document:

Revolution is à la mode at the moment in the country, where the joie de vivre of the citizens was once again plunged into chaos after a third coup d'état in as many years. Although the leading general is by no means an enfant terrible per se, *the fledgling economy still stands to be jettisoned down* la poubelle.

Another Example

What is the language of the following document:

Så sitter du åter på handlar'ns trapp och gråter så övergivet.

Yet Another Example

What is the language of the following document:

Nag hmo kuv mus tom khw.

A Harder Example

What is the language of the following document:

11100000101110111001000011110000010111
01110010100011100000101110111001001110

Why do we want Language Identification

- There's more than English out there!
  - circa 2002, $> 30\%$ of the Web was not in English, a number which is continuously growing
  - only $\sim 6\%$ of world's population are native English speakers
  - $< 30\%$ of world's population are competent in English
  - Non-Anglophone communities are rapidly becoming connected

Why Language Identification?

- Language identification provides us with the means to automatically "discover" web data to convert into a corpus over which to learn linguistic (lexical) properties

- Also research on:
  - mining interlinear text (e.g. ODIN)
  - cleaning web text (e.g. CLEANEVAL)

Basic Approaches

- Linguistically-grounded methods
- Similarity-based categorisation and classification
- Feature-based and kernel-based methods

Don't Websites Declare the Language and Encoding?

- These are frequently:
  - not there
  - wrong (e.g. S-JIS, EUC-JP, UTF-8)
- Remember: users are competent "scrollers", but "above the fold" real estate still a premium

Early Attempts: Diacritics

- Intuition: a language has a certain set of "special characters"

- e.g. French vs. English:
  - Once we see one of *à*, *é*, *ô*... we know the document is in French
  - Unless we're talking about a résumé, or a prêt-à-porter fashion show, or...

- Choose a set of "special characters" for each language, and search the document for them

- Advantages:
  - cheap analysis: characters appear, or not
- Disadvantages:
  - overlapping diacritic sets
  - short documents may not contain diacritics
  - only sensible for European languages
  - assumes we know the document encoding

Early Attempts: Discriminating Character n-grams

- Intuition: certain languages have certain strings which only/frequently occur in that language
  - English: "ery "
  - French: "eux "
  - Italian: "cchi"
  - Serbo-Croat: "lj"
- But note, *zucchini, killjoy*...

- Advantages:
  - cheap analysis: sequence appears, or not
- Disadvantages:
  - sequences may occur in multiple languages
  - short documents may not contain given sequence
  - only sensible for alphabet languages

Early Attempts: Stop Word Lists

- Intuition: common words in one language do not occur in another language

- Johnson (1993)

  - List stop words, e.g.
    - English: ***the, a, of, in, by, for***...
    - French: ***le, la, les, de, un, une, à, en***...
    - German: ***ein, das, der, die, in, im***...
  - Document has stop words from one language

- Requires (commonly available?) stop word lists

- Advantages:
  - cheap analysis: words in document $\times$ words in list
- Disadvantages:
  - overlap of stop word sets
  - short documents may not contain stop words
  - only sensible for European languages (?)

Statistical Language ID

- Intuition: **Distribution** of character n-grams is constant across documents in the same language

- Variety of methods:
  - Compare n-gram ranking
  - Compare Bayesian probability of distribution
  - Compare entropy of distribution

- Advantages:
  - language model is independent (?) of document
- Disadvantages:
  - potentially much training data is required
  - classification can be slow
  - domain effects
  - encoding issues make task absurd (or very easy!)

One Example: n-gram Ranking

- For each language in the classification (training) set:
  - Find the frequency of all 1-grams (*A*,*B*,*C*,...), 2-grams (*AA*,*AB*,*AC*,...*BA*,*BB*,*BC*,..., etc.) in the training data
  - Rank each n-gram from most frequent to least frequent (resolve ties)

- To classify a document (test set):
  - Find the frequency of all 1-grams, 2-grams, etc. in the document
  - Rank each n-gram from most frequent to least frequent
  - For each n-gram in the test document:
    - Calculate the "out-of-place" distance between the rank in the test document and the rank in the training language
    - Include (pre-computed) "out-of-range" rank for n-grams not found in training set
  - Sum the distances for each n-gram to a given language to estimate a "language distance"
  - Predict the language that has the least distance to the test document (resolve ties)

N-gram Ranking: Example

- Training data (1-grams only):
  - English: _ , *e* , *t* , *o* , *n* , *i* ...
  - Welsh: _ , *a* , *d* , *y* , *e* , *n* ...
  - Vietnamese: _ , *n* , *h* , *t* , *i* , *c* ...
- Test document: *knowing, having, going*
  - $g(1)$, $n(2) \times 4$
  - $i(3) \times 3$
  - $\_(4)$, $o(5) \times 2$
  - ...

- English:
  - $|\,1-7\,|+|\,2-5\,|+|\,3-6\,|+|\,4-1\,|+|\,5-4\,|$
  - $=16$
- Welsh:
  - $|\,1-7\,|+|\,2-6\,|+|\,3-7\,|+|\,4-1\,|+|\,5-7\,|$
  - $=19$
- Vietnamese:
  - $|\,1-7\,|+|\,2-2\,|+|\,3-5\,|+|\,4-1\,|+|\,5-7\,|$
  - $=13$
- $\rightarrow$ Vietnamese! ...hmm...

Feature–based methods

(Semi-)automatically construct a list of discriminating features (c.f. linguistically grounded methods)

- Monte Carlo sampling of distribution features
- Document similarity using information measures
- Kernel methods

Top performers, but require a level of statistical proficiency beyond this subject!

Encoding Detection

- Intuition: the encoding of a document determines its language
  - If the document is encoded in S-JIS, it is in Japanese
  - GJK $\rightarrow$ Chinese
  - ISO 8859-1 $\rightarrow$???
- One–document, one–encoding much better than one–document, one–language

- Advantages
  - deals with a wide set of languages
  - often need to know encoding anyway
  - relatively small number of encodings ($\sim$100?)
- Disadvantages
  - encoding often does not uniquely identify language
  - especially with Unicode

So, how do they do?

- Most methods report ∼100% accuracy (or precision/recall)
- A solved problem?

What's the Problem?

- Diverse training/test/classification sets between reported results:

- Classification sets contain as few as three languages
  - There are many more languages to be dealt with
  - Obfuscatory impact of many languages is unclear

- Training data can be > 1MB
  - May not be able to find 1MB of training data for many languages
  - Restricts some algorithms to common languages

- Test string can be > 10KB
  - Documents may be **much** smaller than 10KB
  - Impact of performance on small test samples is unclear

Open Issues

- How well do existing techniques support language identification for languages which form the bulk of the more than 7000 languages identified in the Ethnologue?
- Can we treat LangID as an open-class classification problem?

  $\arg\max_{c \in C} lm(c, D)$ vs. $\arg\max_{c \in C \cup C'} lm(c, D)$
- What is the performance of the variety of LangID systems in environments where the amount of gold standard data for training is small (e.g. 50/100/250 words or 50/100/250 characters)?
- Can we move away from a one-to-one view of LangID to a one-to-many view?
  - finer granularity (e.g. sentence, paragraph, section)
  - in quantitative terms (e.g. a document is 95% English, 3% French and 2% Italian)
- Can we move away from IR-style evaluation criteria to produce something more representative of reality?

- gradated judgements for source language
- gradated judgements for resource type
- possibly micro-level markup of the location of different languages in the document

Summary

- What is language identification?
- Why is language identification important?
- What issues arise in language identification?
- What methods are used?
- Why isn't language identification a solved problem?

Language Normalization

How You See the Web

How Web Services See the Web

```
#                                                The Age: national, world, business, entertainment, sport and technology news from Melbourne's leading newspaper. (p1 of 8)

   REFRESH(0300 sec): http://www.theage.com.au/
   #Top stories @theage.com.au

   Welcome to The Age Online. Skip directly to: Search Box, Section Navigation, Content. Text Version.

   Fairfax Digital
   NEWS | MYCAREER | DOMAIN | DRIVE | FINANCE | MOBILE | RSVP

                                                                                member centre | login | register

   IFRAME: http://ffxcam.fairfax.com.au/html.ng/site=age&adspace=100x29

   www.theage.com.au

Korp's lover planned 'murder pack', court told

   Tania Herman [Jesse Hogan 5:39pm] Tania Herman allegedly compiled a list of items in preparation for the attempted murder of Maria Korp. more
     * Herman confessed to me: brother

New laws outlaw bride trafficking

   [4:50pm] Tough new laws could see people who traffic young Australian girls overseas for forced marriages jailed for up to 25 years. more

Qantas staff to testify in Corby case

   [6:52pm] Two Qantas employees arrive in Bali to testify at Schapelle Corby's reopened trial tomorrow. more

MORE TOP STORIES

     * NSW abolishes property vendor duty
     * Domestic violence charges skyrocket
     * Four arrested over illegal tobacco
     * Retrial for three on solicitor's murder
     * Baby whale dies in shark net

World

Special reports

Saddam lawyers see red over court scuffle

   Former Iraqi president Saddam Hussein [7:18pm] Saddam Hussein's legal team boycott proceedings until a man they say attacked him at a hearing is brought to
   justice. more

ADF's top lawyer has Hicks commission in sights

   [6:09pm] The Australian Defence Force's chief lawyer expresses concerns about the US military commission set up to try David Hicks. more
     * Howard refuses to protest over Hicks trial
     * Hicks facing rigged trial, say ex-prosecutors
     * US policy on suspects illegal, says FBI memo
     * Terrorist trials: Vote now

     * London bomb suspect charged in Italy
     * Bush appoints Bolton to UN
     * US teens caught with dismembered body
-- press space for next page --
  Arrow keys: Up and Down to move.  Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

Document Types and Parsing

- Documents come in an ever-increasing range of formats (HTML, PDF, PS, MSWord, Excel, ...)
  - need for robust means to detect document type (resilient to faulty MIME type, metadata, etc)
- Need to be able to extract out basic "semantic" content into common format (text) to index/carry out pre-processing over
- Need to be able to identify the source language(s) of a given document, and its character encoding

Metadata

- Most document types contain metadata of some description:

```
<head>
  <title>CSLI LinGO Lab</title>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
  <meta http-equiv="Content-Style-Type" content="text/css">
  <meta name="keywords" content="linguistic grammars online,
   LinGO, computational linguistics,
   head-driven phrase structure grammar, hpsg, natural language processing,
   parsing, generation, augmentative and alternative communication, aac,
   LinGO Redwoods, multiword expressions, MWE, grammar matrix">
  <meta name="description" content="This page provides information about
   the CSLI Linguistic Grammars Online (LinGO) Lab at Stanford
   University.">
```

- Should we also extract out this data, or is metadata too unreliable to consider using?

What is Our Document "Unit"?

- What is the appropriate granularity of document "unit":
  - an email message?
  - an email message with attachments?
  - an email message with a zip attachment containing multiple documents?
  - an HTML document containing multiple languages?
  - multiple HTML documents encapsulated in frames?
  - a single post in a web user forum "thread"?
  - a single page in a web user forum "thread"?
  - a multi-page web user forum "thread"?

Tokenisation

- **Tokens** are the atomic text elements that we wish to index and use as our units in pre-processing

- **Tokenisation** is the process of converting a text into tokens, e.g.:

```
Tim Berners-Lee's ad hoc pre-processing policy from '92
                          ↓
Tim Berners Lee ad-hoc preprocessing policy from 92
```

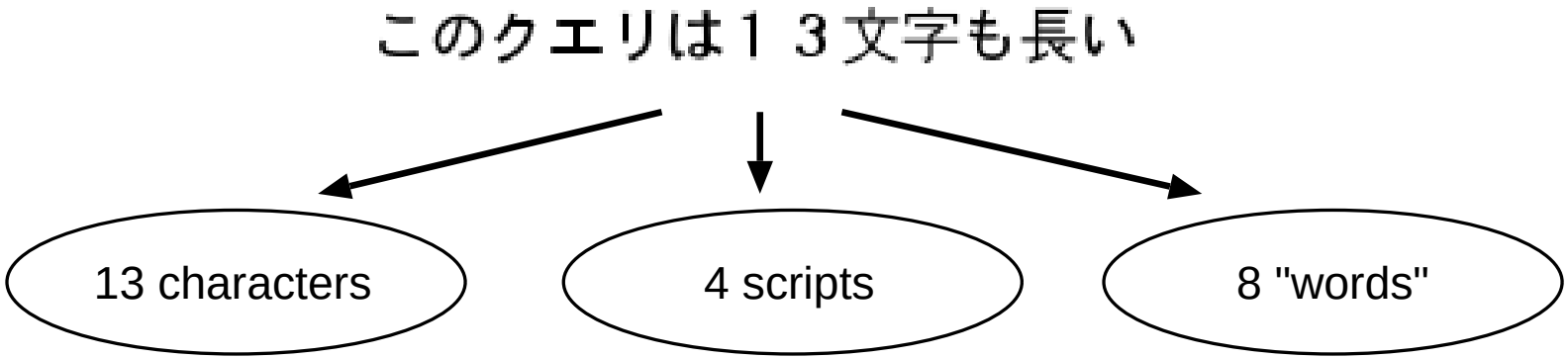
- It is vital that we are **consistent** in tokenising all documents and queries equivalently (why?)

Issues in English Tokenisation

- Hyphenation
  - `Berners-Lee` = one token or two (`Berners Lee`)
  - `tradeoff` vs. `trade-off` vs. `trade off`
- Possessives (`Berners-Lee's` = `Berners-Lee`?)
- Multiword units (`Tailem Bend` = `Tailem-Bend`?)
- The document context will often aid us in making these decisions, but we don't have this luxury with queries AND we need to have a consistent policy for all documents and queries

Tokenisation in Non-segmenting Languages

- What is a "word" in a language such as Thai, Japanese or Chinese?



- How to deal with segmentation ambiguity?

東京都に住んでいます。

ไปหามเท่ดี

Granularity of Tokenisation

- What is the appropriate granularity to index over:
  - sub-characters??
  - characters/character n-grams? (not as silly as it sounds)
  - words/word n-grams (phrases)?
  - some combination of all of these?

- Is it possible to come up with a policy which can be applied consistently across languages (which co-exist within a single "locale")?
  - `raison d'être` = `raison detre`?
  - `resume` = `résumé`?

Token Normalisation

- Tokens are generally further normalised by:
  - normalising numbers, character case, punctuation, etc.
  - eliminating "stopwords"
  - stemming/lemmatisation
  - expanding the token set with synonyms, homonyms, etc.

Number Normalisation

- Dates

  7/10/2006 vs. 10/7/2006 vs. Oct 7, 2006 ...
  2000AD vs 1421 AH vs 2543 (Buddhist) vs Heisei 12 ...

- Amounts

  $700K vs $700,000 vs 0.7 million dollars vs ...
  128.250.37.80 vs. www.cs.mu.oz.au vs. www

- Often indexed as metadata, separate to text tokens

- Occurrences of left-to-right text (e.g. dollar amounts) in right-to-left languages like Hebrew and Arabic

Normalising Case and Punctuation

- The general policy is to reduce all letters to lower case, although this is not always a good idea:
  - SAP vs. sap
  - MoD vs. mod vs. MOD
  - Cardinal Sin vs. cardinal sin
- Punctuation normalisation must be carried out in a language specific fashion in order to accommodate the idiosyncracies of different languages/domains (e.g. `x.id` vs. `xid`)
- Punctuation indicating sentence boundaries generally ignored

Stop Words

- **Stop word** = word which tends to occur with high frequency across all documents and is semantically bleached or promiscuous

  English stop words: `of, the, a, to, not, and, or, ...`

- The general policy for classification is to strip all stop words from documents

  `to be or not to be` $\rightarrow$ `be`

- Stop word lists specific to individual languages (complications with short queries)

- Removing stop words has the spinoff advantage of (moderate) index compression

Discussion

- How might you go about (semi-)automatically identifying stop words in a novel language/domain?

Stemming/lemmatisation

- Basic flavours of word **morphology**:
  - **inflectional morphology**: word-class preserving alternations in word form for a given lexeme (cf. *I* *am*, *you* *are*, *she* *is*, *it can* *be*)
  - **derivational morphology**: description of the process by which a given lexeme is derived from a second lexeme, generally from a different word class (e.g. *a+symmetry+ic* $\rightarrow$ *asymmetric*, *act+ive+ist* $\rightarrow$ *activist*)

- **Stemming** is the process of stripping away affixes to leave the **stem** of the word (often a nonce-word, e.g. *producer* $\rightarrow$ *produc*)

- **Lemmatisation** is the process of recovering the base lexeme of a given word (e.g. *dogs are mammals* $\rightarrow$ *dog be mammal*)

- Obvious "benefits" of stemming and lemmatisation in normalisation:
  - index compression
  - removal of superficial divergences in word form
  - particularly salient when working with languages with rich morphology (e.g. Turkish, Spanish, Inuit)

- Some controversy over whether stemming/lemmatisation hurts or helps in web mining applications; greatest impact over short documents

Porter Stemmer

- Most popular English stemmer currently in use, based on **suffix** stripping only
- Implemented as cascaded set of rewrite rules, e.g.:

  `sses` $\rightarrow$ `ss`
  `ies` $\rightarrow$ `i`
  `ational` $\rightarrow$ `ate`
  `tional` $\rightarrow$ `tion`

- Optionally constrain the algorithm to produce a dictionary-listed stem at each step
- See www.tartarus.org/~martin/PorterStemmer/ for an implementation in your programming language of choice

Decompounding

- In European languages such as German, Dutch and Swedish, compound words are generally single words (e.g. *solar cell* = *zonnecel*; cf. *bathtub*)

- **Decompounding** is the process of splitting a compound word (esp. noun) up into its component tokens (e.g. *zonnecel* $\rightarrow$ *zon cel*)

  generally performed recursively, by way of searching for a concatenation of words which can compound (note: not simply a question of segmentation)

- Decompounding has been shown to have considerable impact in web search applications

Backwards Transliteration

- Languages such as Japanese and Chinese borrow heavily from languages such as English (e.g. names, technical terminology) through the process of **transliteration** (e.g. *computer* $\rightarrow$ *konpyūta*)

- Due to lack of normalisation of the transliteration process, there are commonly multiple transliteration alternatives for a given word (e.g. *konpyūta* vs. *konpyūtā*; *bodī* vs. *badī*)

- Possibilities for normalisation by mapping transliterated words back onto their source language equivalents (**back transliteration**)

Expansion

- Expansion involves abstracting away from a text by way of synonyms and/or homonyms, usually in the form of hand-constructed equivalences:
  - *car* = *automobile*
  - *normalisation* = *normalization*
  - *your* = *you're*
- In practice, this often takes the form of cross-indexing, in indexing any document containing *car* as also containing *automobile*, and vice versa

Summary

- What is tokenisation, and why is it important?
- What complications are then when tokenising over non-segmenting languages?
- What forms of token normalisation are commonly employed over English?
- What is stemming/lemmatisation?
- What other forms of token normalisation are there for non-English languages?
- Do you think the gain from normalisation outweighs the noise introduced?

Acknowledgments

- Many slides from Tim Baldwin's *Web as Data* (Melbourne University 433-352)

- Excellent introduction to Information Retrieval, including web searching: Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008. http://nlp.stanford.edu/IR-book/information-retrieval-book.html *Determining the vocabulary of terms* deals with tokenization/normalization