Token to Words

Expanding identified token to words

- □ numbers+type = word list
- □ homographs+type = words
- □ symbols broken down and pronounced
- □ unknown words: as word or letter sequence

```
(define (token_to_words token name)
  (cond
   ((string-matches name "[0-9]+'s") ;; e.g. 1950's
     (item.set_feat token "token_pos" "year")
     (append
      (builtin_english_token_to_words token (string-before name "'s"))
      (list '((name "'s")(pos nnp)))))
   ((string-matches name "[0-9]+-[0-9]+")
    ;; e.g.  12-14
    ;; split into two numbers
    ;; identify type of one number (ordinal/cardinal)
    ;; expand woth ``to'' between them
   )
   ....
   (t
     ;; just a simply word
     (builtin_english_token_towords token name))))
```

Example token rule

for “$120 million”

```
(define (token_to_words token name)
 (cond
  ((and (string-matches name "\\$[0-9,]+\\(\\.[0-9]+\\)?")
        (string-matches (item.feat token "n.name")
                        ".*illion.?"))
   (append
    (english_token_to_words token (string-after name "$"))
    (list
     (item.feat token "n.name"))))
  ((and (string-matches (item.feat token "p.name")
                        "\\$[0-9,]+\\(\\.[0-9]+\\)?")
        (string-matches name ".*illion.?"))
   (list "dollars"))
  (t
   (english_token_to_words token name))))
```

Text modes

If we know the type of text being synthesizing
(e.g. email, Latex, HTML) we can tailor the processing.

- mode specific tokenizing
- using tokens to direct synthesis (emphasis, selecting voices etc.)
- mode specific lexical items.
- mode specific syntactic forms.

Explicit markup and/or Custom models

Festival text modes

Customizable modes for synthesis.
Each mode can have

- ☐ A (Unix) filter program to extract/delete information
- ☐ An `init_function` on entering the mode.
- ☐ An `exit_function` on exiting the mode.

An example text mode for email

A filter to extract , from line, subject and body from email message

```
#!/bin/sh
#  Email filter for Festival tts mode
#  usage: email_filter mail_message >tidied_mail_message
grep "^From: " $1
echo
grep "^Subject: " $1
echo
sed '1,/^$/ d' $1
```

setup mode specific token functions

```
(define (email_init_func)
 "Called on starting email text mode."
 (set! email_previous_t2w_func token_to_words)
 (set! english_token_to_words email_token_to_words)
 (set! token_to_words email_token_to_words))

(define (email_exit_func)
 "Called on exit email text mode."
 (set! english_token_to_words email_previous_t2w_func)
 (set! token_to_words email_previous_t2w_func))
```

```
(define (email_token_to_words token name)
  "Email specific token to word rules."
  (cond
   ((string-matches name "<.*@.*>")
     (append
      (email_previous_t2w_func token
       (string-after (string-before name "@") "<"))
      (cons
       "at"
       (email_previous_t2w_func token
        (string-before (string-after name "@") ">")))))
```

```
((and (string-matches name ">")
      (string-matches (item.feat
                       token "whitespace")
                      "[ \t\n]*\n *"))
 (voice_don_diphone)
 nil ;; return nothing to say
)
(t  ;; for all other cases
  (if (string-matches (item.feat
                       token "whitespace")
                      ".*\n[ \n]*")
      (voice_rab_diphone))
  (email_previous_t2w_func token name))))
```

```
(set! tts_text_modes
   (cons
    (list
      'email   ;; mode name
      (list         ;; email mode params
       (list 'init_func email_init_func)
       (list 'exit_func email_exit_func)
       '(filter "email_filter")))
    tts_text_modes))
```

```
From: Alan W Black <awb@cstr.ed.ac.uk>
Subject: Example mail message
Date: Wed, 27 Nov 1996 15:32:54 GMT


 Alan W. Black writes on 27 November 1996:
 >
 >
 > I’m looking for a demo mail message for Festival, but can’t seem to
 > find any suitable.  It should at least have some quoted text, and
 > have some interesting tokens like a URL or such like.
 >
 > Alan

Well I’m not sure exactly what you mean but awb@cogsci.ed.ac.uk
has an interesting home page at http://www.cstr.ed.ac.uk/~awb/ which
might be what you’re looking for.

Alan

 > PS. Will you attend the course?

I hope so

bye for now
```

Reading addresses

Smith, Bobbie Q, 3337 St Laurence St, Fort Worth, TX 71611-5484, (817)839-3689
Anderson, W, 445 Sycamore Way NE, Lincoln, NE 98125-5108, (212)404-9988

Mark-up languages

- Building special text modes might be too difficult
- Need general method for general markup:
  – breaks, voice changing
  – pronunciations, date/time identifies
- All synthesizers include this but are incompatible
- Proposal of *general* method:
  – SGML/XML based
  – *basic* tags only
  – cf. JSML, VoiceXML

```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE speech mark up//EN"
        "Sable.v0_2.dtd"
[]> <SABLE> <SPEAKER NAME="male1">

The boy saw the girl in the park <BREAK/> with the telescope.
The boy saw the girl <BREAK/> in the park with the telescope.

Some English first and then some Spanish.
<LANGUAGE ID="SPANISH">Hola amigos.</LANGUAGE>
<LANGUAGE ID="NEPALI">Namaste</LANGUAGE>

Good morning <BREAK /> My name is Stuart, which is spelled
<RATE SPEED="-40%">
<SAYAS MODE="literal">stuart</SAYAS> </RATE>
though some people pronounce it
<PRON SUB="stoo art">stuart</PRON>.  My telephone number
is <SAYAS MODE="literal">2787</SAYAS>.

I used to work in <PRON SUB="Buckloo">Buccleuch</PRON> Place,
but no one can pronounce that.

By the way, my telephone number is actually
<AUDIO SRC="http://att.com/sounds/touchtone.2.au"/>
<AUDIO SRC="http://att.com/sounds/touchtone.7.au"/>
<AUDIO SRC="http://att.com/sounds/touchtone.8.au"/>
<AUDIO SRC="http://att.com/sounds/touchtone.7.au"/>.
```

SABLE: for marking emphasis

What will the weather be like today in Boston?

`It will be <emph>rainy</emph> today in Boston.`

When will it rain in Boston?

`It will be rainy <emph>today</emph> in Boston.`

Where will it rain today?

`It will be rainy today in <emph>Boston</emph>.`

But we need a richer markup

- SABLE is quite limited:
  - Now embodied in SSML, VoiceXML and JSML
- Concept to speech is richer:
  - translation and generation systems
  - Syntactic, Semantic
  - Anaphoric, Rhetorical, Speech act etc.
- Mark up should be:
  - abstract not low-level
  - e.g *type=question* not
  - *pitch rise at end*

Data: four domains

nantc : press-wire news data

classifieds : real estate ads from on-line newspapers

pc110 : palmtop mailing list (e-mail like)

rfr : rec.food.recipes USENET messages

Corpus	nantc	ads	pc110	rfr
total # tokens	4.3m	415k	264k	209k
# NSWs	377k	180k	72k	46k
% NSW	8.8%	43.4	27.3	22.0

	EXPN	abbreviation, contractions	adv, N.Y, mph, gov't
alpha	LSEQ	letter sequence	CIA, D.C, CDs
	ASWD	read as word	CAT, proper names
	MSPL	misspelling	geogaphy
	NUM	number (cardinal)	12, 45, 1/2, 0.6
	NORD	number (ordinal)	May 7, 3rd, Bill Gates III
	NTEL	telephone (or part of)	212 555-4523
	NDIG	number as digits	Room 101,
N	NIDE	identifier	747, 386, I5, PC110, 3A
U	NADDR	number as street address	5000 Pennsylvania, 4523 Forbes
M	NZIP	zip code or PO Box	91020
B	NTIME	a (compound) time	3.20, 11:45
E	NDATE	a (compound) date	2/2/99, 14/03/87 (or US) 03/14/87
R	NYER	year(s)	1998 80s 1900s 2003
S	MONEY	money (US or otherwise)	$3.45 HK$300, Y20,000, $200K
	BMONY	money tr/m/billions	$3.45 billion
	PRCT	percentage	75%, 3.4%
O	SLNT	not spoken, word boundary	word boundary or emphasis character:
T			M.bath, KENT*REALTY, _really_, ***Added
H	PUNC	not spoken, phrase boundary	non-standard punctuation: "..." in
E			DECIDE...Year, "***" in $99,9K***Whites
R	FNSP	funny spelling	slloooooww, sh*t
	URL	url, pathname or email	http://apj.co.uk, /usr/local, phj@teleport.com
	NONE	token should be ignored	ascii art, formating junk

Data: NSW distributions

	Domains			
	nantc	classifieds	pc110	rfr
ASWD	83.49	28.64	64.60	72.36
LSEQ	9.10	3.00	22.60	2.11
EXPN	7.41	68.36	12.80	25.53

	Domains			
	nantc	classifieds	pc110	rfr
NUM	66.11	58.26	43.77	97.90
NYER	19.06	0.70	0.51	0.27
NORD	9.37	3.37	4.45	0.11
NIDE	2.24	5.83	37.41	0.47
NTEL	1.25	25.92	1.32	0.02

Hand labeling

- Each NSW presented in context
  - Three words either side
- One letter choice of TAG
  - or explicit expansion
  - splits "WinNT" $\rightarrow$ "Win" "NT"
- Test of inter-labeler agreement
  - 3 labelers nantc, 2268 samples, $\kappa = 0.81$
  - 9 labelers ads, 622 samples, $\kappa = 0.84$
- Labeling held as XML markup

```
Today I bought a Sony<W NSW="LSEQ"> NP-F530,</W><W NSW="SPLT"><WS
NSW="NUM"> 1350</WS><WS NSW="EXPN">maH.</WS></W> Like your<W
NSW="NIDE"> 550</W> it is slightly larger than the native<W
NSW="LSEQ"> IBM</W> battery pack.  It's been<W NSW="NUM"> 3</W> hours
now on it's first charge - I am charging in the <W NSW="LSEQ"> PC110.
</W>
```

Can we find NSWs?

- Tokens not in lexicon
- Plus
  - single character tokens
  - "punctuation"
  - common abbreviations (in lexicon)
- Misses homographic abbreviations/standard words
  - "sun", "Jan"
  - also domain specific ones, "kit" and "named"

Domain Dependent?	Detection Algorithm	Precision//Recall			
		nantc	ads	pc110	rfr
No	non-lexical	55/79	96/79	80/65	76/82
No	+ sct + abbrevs	44/93	95/91	70/90	73/96
Yes	++ abbrevs	39/93	92/92	60/91	46/97

Theoretical models

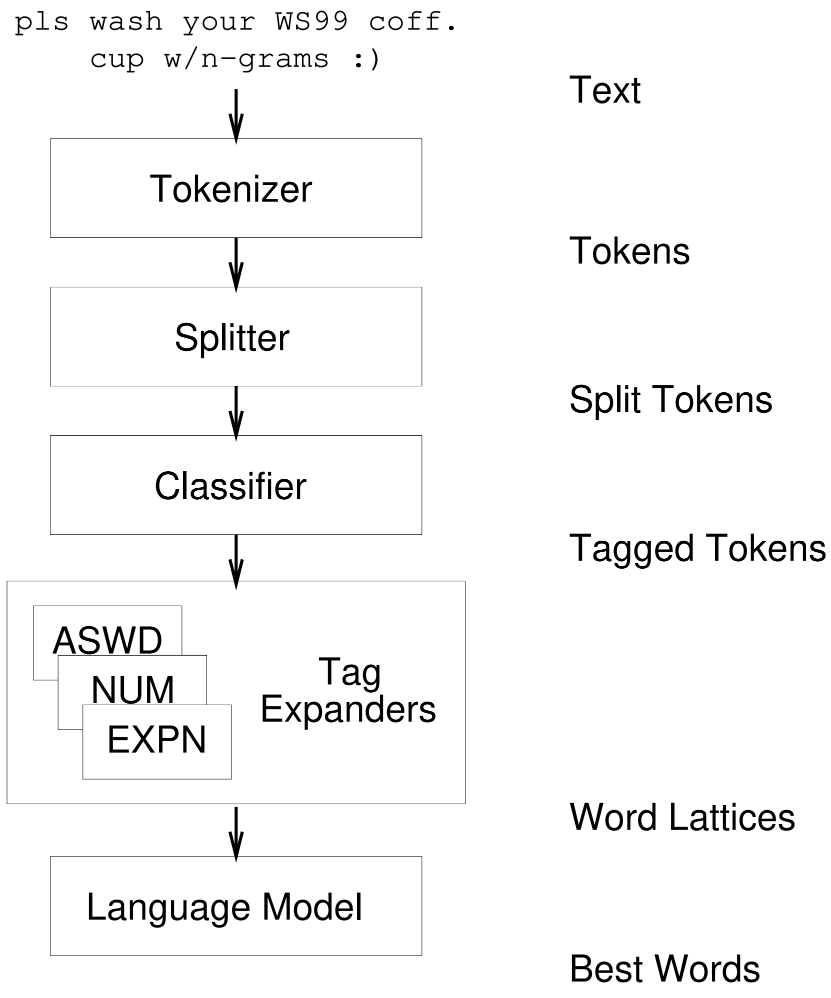
- Source-channel model:

$$\begin{aligned}\hat{\mathbf{w}} &= \operatorname{argmax} p(\mathbf{w}|\mathbf{o}) && (1)\\ &= \operatorname{argmax} p(\mathbf{o}|\mathbf{w})p(\mathbf{w}) && (2)\end{aligned}$$

- Direct approach:

$$\hat{\mathbf{w}} = \operatorname{argmax} p(\mathbf{w}|\mathbf{o}) \qquad (3)$$

Architecture



Splitting

- whitespace separated tokens isn't fine enough
- Further splitting is required:

  1500km $\rightarrow$ 1500 km
  and/or $\rightarrow$ and / or
  WinNT $\rightarrow$ Win NT

- Ideally deterministic, domain independent
- Simple regular expressions

Splitting

	NANTC	classifieds	pc110	RFR
Recall	98.89	94.96	87.66	98.88
Precision	74.41	87.32	81.68	89.51
Split Correct	92.54	85.99	74.11	89.54
Total Correct	98.45	95.19	92.97	98.40

Misses:

– ESANDWICH, 3400sq.ft, xjack, 11/2

“False” positives:

– 1-3pm, w/d, R-Ariz, PC-110

Tag classification

Assign EXPN, NUM, NORD etc to NSWs:

- □ domain independent features:
  – all caps, no vowels, numeric etc.
- □ domain dependent features:
  – alphabetic sub-classifier for EXPN, ASWD and LSEQ

Tested CART and Maximum Entropy models

Alphabetic tag sub-classification

NSW tag $\mathbf{t}$ for alphabetic observations $\mathbf{o}$
NATO: ASWD, **PCMCIA**: LSEQ, **frplc**: EXPN

- $$p(\mathbf{t}|\mathbf{o}) = \frac{p_t(\mathbf{o}|\mathbf{t})p(\mathbf{t})}{p(\mathbf{o})}$$

  where $\mathbf{t} \in [ASWD, LSEQ, EXPN]$.
- $p_t(\mathbf{o}|\mathbf{t})$ estimated by a letter trigram model
  $$p_t(\mathbf{o}|\mathbf{t}) = \prod_{i=1}^{N} p(l_i|l_{i-1}, l_{i-2}),$$
- $p(\mathbf{t})$ prior from data or uniform
- normalized by
  $$p(\mathbf{o}) = \sum_t p_t(\mathbf{o}|\mathbf{t})p(\mathbf{t})$$

Alphabetic tag sub-classification

LLM features are fed into overall classifier through 6 features

Token	$p(\text{ASWD}\mid\mathbf{o})$	$p(\text{LSEQ}\mid\mathbf{o})$	$p(\text{EXPN}\mid\mathbf{o})$	p_{max}	t_{max}	*diff 1-2*
mb	0.0001	0.0038	**0.9962**	0.9962	EXPN	0.9924
Grt	0.0024	0.0000	**0.9976**	0.9976	EXPN	0.9952
NBA	0.0017	**0.9983**	0.0000	0.9983	LSEQ	0.9966
Cust	**0.5456**	0.0000	0.4544	0.5456	ASWD	0.0912

Using LLM features alone

Domain	NANTC	ads	pc110	RFR
Baseline	83.9[ASWD]	80.53[EXPN]	63.77[ASWD]	69.98[ASWD]
Uniform	88.92	98.5	90.83	97.36
Unigram	95.72	98.74	92.27	97.92

Full tag classification

Accuracy	NANTC	ads	pc110	RFR
No LLM Feats	97.7	92.7	90.9	97.3
All LLM feats	98.1	93.5	91.8	96.8

Algorithmic expansions

- SLNT, NONE: expand to nothing
- ASWD, PUNC: expand to themselves
- LSEQ: as letters
- NUM: expands integers, floats, roman to string of words
- NORD: expands to ordinals
- NYER: as number pairs (except 00 and 000)
- NADDR, NZIP, NTEL, NDATE, NTIME: specific expanders
- NIDE: letters as letters, numbers as pairs
- MONEY, BMONY: as currency
- PRCT: as NUM with "percent"
- EMAIL, URL: treated ASWD (though should not be)
- MSPL, FNSP, OTHER: treated ASWD (though should not be), never predicted

EXPN expansions

How to find the expansion of an abbreviation:
– "wbfpl" → "wood burning fireplace"
– "BR" → "bedroom"
– "Fl" → "Florida" or "Floor"

Not simple lists:
– 32 different abbrevs for "bedroom"
– Productive: SQH, SB, Newingtn

In *supervised* case use labelled expansions
error rate:

without language model 6.7%
without language model 4.8%

What about *unsupervised* case?

- ☐ Assume expanded form somewhere in corpus
- ☐ Build letter deletion model from known EXPNs
  – CART predicts prob of letter deletion (88% accuracy)
  – convert CART to WFST
  – compute

$$[SW \circ A \circ NSW]^{-1} \tag{4}$$

  – build a WFST for weighted lattice of possible expansions of a potential NSW.

Unsupervised prediction of expansions

1. All singleton SWs + bigrams > 3 times: 33% error rate
2. as 1 plus standard abbrevs: 24%
3. as 2 but
   – expand on training set
   – use language model
   – select most frequent expansion alone : 19.9%
4. as 3 but
   – select best 2 and reestimate probs: 19.9%

Further issues in EXPN expansions

1. Need better model of expansion:

OEPN	OPEN PERENNIAL
DALLIN	DAVID ALLAIN
MASHPEE	MARSH PROPERTIES
SEAVIEW	SEASONAL VIEWS
WIGET	WITHGUESTS

2. Current ignoirng case (unsupervised)
3. What is *likely* to be abbreviated
   – $p(\mathbf{t}|\mathbf{w})$: $BTW \rightarrow$ because the windows

Language Modeling

- Grand schemes:
  - trigger models
  - maximum entropy
- Simple smoothed backed off trigrams
- Applied to pseudo-words:
  ... lives at 123 Norman St. ...
  ... lives at NADDR Norman St. ...

Baseline results

LDC tools : LDC text conditioning tools

Festival : 1.4.0 released text analyzer

	LDC tools		Festival	
	TER	WER	TER	WER
nantc	–	2.88	1.00	1.38
classifieds	–	30.81	30.09	33.48
pc110	–	22.36	14.37	32.62
rfr	–	9.06	6.28	16.19

Domain dependent model

- ☐ domain independent splitter
- ☐ CART tag classifier with letter language model features
- ☐ EXPNs by WFST
- ☐ Language model

	festival		m4	
	TER	WER	TER	WER
nantc	1.00	1.38	0.39	0.82
classifieds	30.09	33.48	7.00	9.71
pc110	14.37	32.62	3.66	9.25
rfr	6.28	16.19	0.94	2.07

Removing components

m4.nolm: no language model (most prob EXPN)

m4.noef: no letter language models feats

m4.noefln: no LM and no LLM feats

	m4		m4.nolm		m4.noef		m4.noeflm	
	TER	WER	TER	WER	TER	WER	TER	WER
nantc	0.39	0.82	0.39	0.81	0.38	0.78	0.38	0.78
classifieds	7.00	9.71	6.82	9.70	7.55	10.39	7.41	10.42
pc110	3.66	9.25	3.63	9.25	3.93	10.90	3.90	10.90
rfr	0.94	2.07	0.93	2.06	0.88	2.07	0.88	2.07

Giving truth

m4.nosplt: uses hand labeled splits

m4.nost: uses hand labeled splits and actual tags

	m4		m4.nosplt		m4.nost	
	TER	WER	TER	WER	TER	WER
nantc	0.39	0.82	0.20	0.44	0.03	0.06
classifieds	7.00	9.71	5.40	6.35	3.15	4.24
pc110	3.66	9.25	2.58	4.61	0.49	0.75
rfr	0.94	2.07	0.59	1.11	0.16	0.24

Cross-domain models

m4.domin: nantc models

m4.dominE: nantc models with domain EXPNs

	festival		m4		m4.domin		m4.dominE	
	TER	WER	TER	WER	TER	WER	TER	WER
nantc	1.00	1.38	0.39	0.82	0.39	0.82	0.39	0.82
classifieds	30.09	33.48	7.00	9.71	25.20	29.11	19.69	21.18
pc110	14.37	32.62	3.66	9.25	12.35	18.69	12.09	18.07
rfr	6.28	16.19	0.94	2.07	2.71	4.66	2.32	4.14

Unsupervised domain models

Building models from unlabeled data

- ☐ Label tokens with nantc CART tag classifier
- ☐ Relabel alphabetics with best LLM prediction
- ☐ Build EXPN expander from plain text and labeled EXPNs
- ☐ Build words with best EXPN expansion
- ☐ Build LM from full expanded words
- ☐ Run with multiple EXPNs and LM to choose

	TER	WER
m4	7.00	9.71
us1.lm	12.50	13.40
us1.nolm	12.64	13.50
us2.EXPNlist	10.58	13.51
m4.dominE	19.69	21.18

NSW model for new domains

- ☐ Models for specific domains
- ☐ Standard text analyzers fail
- ☐ Can build models from unlabeled data

57 ST E/1st & 2nd Ave Huge
drmn 1 BR 750+ sf, lots of sun &
clsts. Sundeck & lndry facils. Askg
$187K, maint $868, utils
incld. Call Bkr Peter 914-428-9054.

Results

- Marked up databases
- Tools to help label databases
- Tools and methods for building models
- 4 domain models
- Text expander better than LDC or Festival
- Tools and methods for building unsupervised models

But what if there are no spaces?

- Chinese, Japanese etc. don't use whitespace
- But still need to to tokenize

Some techniques

Requires lexicon of words

- ☐ Take longest match in lexicon (that gives partition)
- ☐ or find

$$\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} p(\mathbf{w}|) \qquad (5)$$

- ☐ Lattice of all possible partitions and find most probable

Number pronunciation

In languages with gender, declensions etc.

1 niño → un niño (one boy)
1 niña → una niña (one girl)

1 hermano → un hermano (one brother)
1 hermana → una hermana (one sister)

Can't just look at a/o ending letter

1 pais → un pais (one country)
1 ra'iz → una ra'iz (one root)

Slavic languages have *many* variations for numbers making it harder.

End of Text Analysis

From strings of characters to lists of words

- ☐ Tokenize string of chars
- ☐ Chunk into utterance sized chunks
- ☐ Identify token types (homographs, numbers etc)
- ☐ Expand tokens with token to word rules