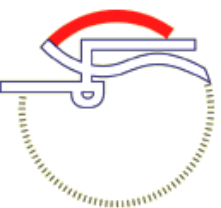


Normalization of “Non-Standard” Words

Final Presentation, Aug 18 1999

Lexical Expansions

Team Members:	Richard Sproat	AT&T Labs
	Alan Black	CMU
	Stan Chen	CMU
	Shankar Kumar	CLSP/JHU
	Mari Ostendorf	Boston University
	Christopher Richards	Williams College



WS 99

“Algorithmic” Expansions

Only one expansion per token

- SLNT, NONE: expand to nothing
- ASWD, PUNC: expand to themselves
- LSEQ: as letters
- NUM: expands integers, floats, roman to string of words
- NORO: 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, BMONEY: as currency

- PRCT: as NUM with “percent”
- EMAIL, URL: treated ASWD (though should not be)
- MSP, FNSP, OTHER: treated ASWD (though should not be), never predicted

Abbreviations (EXPN): Corpus-Dependent *Supervised* Expansions

Without Language Model	6.7%
With Language Model	4.8%

Abbreviations (EXPN): Corpus-Dependent *Unsupervised* Expansions

Problem: given a previously unseen abbreviation, how do you use corpus-internal evidence to find the expansion?

Example: eat-in kit livrm dinrm 17x25 famrm

Elsewhere in Corpus: ... eat-in kitchen ...
 ... living room ...

A Source-Channel Language Model Approach

$$\hat{\mathbf{w}} \approx \operatorname{argmax}_{\mathbf{w}, \mathbf{t}} p(\mathbf{o}, \mathbf{t} | \mathbf{w}) p(\mathbf{w}) \quad (1)$$

$$= \operatorname{argmax}_{\mathbf{w}, \mathbf{t}} \boxed{p(\mathbf{o} | \mathbf{t}, \mathbf{w})} p(\mathbf{t} | \mathbf{w}) \boxed{p(\mathbf{w})} \quad (2)$$

Where:

- \mathbf{o} are the *observed text*
- \mathbf{w} are the *underlying words*
- \mathbf{t} are the *tags*

CART Model for Abbreviations

- **Problem:** predict probability of letter deletion
- 854 pairs of abbreviations and their expansions (from Classifieds): **baseline is 0.54**
- Two CART models:

Features	Model 1	Model 2
Letter -2, -1, +1, +2 (C, S, V, Y, n/a)	+	+
This Letter (C, S, V, Y)	+	+
Boundary -1, +1 (w, m, 0)	+	+
Fate -2, -1 (del, nodel, n/a)	+	+
Fate +1, +2 (del, nodel, n/a)	-	+
	0.85	0.88

- **Model 2** has been compiled into a WFST

Overview of Experiments: 1

- Run on Classified Ad corpus:
 - 307,735 words of training data
 - 76,676 words of test data
- We have from the initial automatic classifications:
 - An estimate of which words are *standard*
 - Non-alphabetic or mixed *non-standard* words
 - An estimate of which words are *alphabetic non-standard*
- Lang. mod. (3-gram LM with modified Kneseer-Ney backoff) is given:
 - Standard words as themselves
 - Non-alphabetic NSW's as their tag (NUM, NTEL ...)
 - Alphabetic NSW's as <UNK>

Overview of Experiments: 2

SO	SO	SO
SHORE	SHORE	SHORE
</s>	</s>	</s>
40	@NUM+PL	@NUM+PL
+	@EXPN	@EXPN
MODERN	MODERN	MODERN
BRK	BRICK	<UNK>
APTS	APARTMENTS	<UNK>
ON	ON	ON
</s>	</s>	</s>
4	@NUM+PL	@NUM+PL
+	@EXPN	@EXPN
ACRES	ACRES	ACRES

Overview of Experiments: 3

- Lexical model is given:
 - Standard words (SW 's) as themselves
 - Alphabetic NSW 's as themselves (i.e. the token)
 - Let:
 - SW be the list of SW 's and SW -bigrams from the training
 - NSW be the list of NSW 's and NSW -bigrams from the training
 - A be the abbreviation model
 - We want to compute $[SW \circ A \circ NSW]^{-1}$
- This WFST generates a weighted lattice of possible expansions of a potential EXPN (alphabetic NSW) for the test data.

Experiment 1

SW is:

- All singleton *SW*'s
- All *SW* bigrams occurring in the data, occurring a minimum of 3 times

Error rate: 33%

Experiment 3

- All singleton SW's
- All SW bigrams occurring in the data, occurring a minimum of 3 times
- A short list of *standard* abbreviations:
aug (August); av (Avenue); blvd (Boulevard) ; ext (extension);
ft (foot, feet); inc (incorporated); l (left); n (north); r (right);
rd (road); st (street); w (west); w/ (with); x (extension); sf
(square foot, square feet); etc (etcetera); n.w (northwest)

Error rate: 24%

Experiment 4

- Same initial setup as Experiment 3
- Rerun the *expander* and the *language model* on the *training data*
- Select the most frequent expansion found for each NSW in the training data as “truth”

Error rate: 19.9%

Experiment 5

- Same initial setup as Experiments 3, 4
- Rerun the *expander* and the *language model* on the *training data*
- Select the most frequent *two* expansions found for each NSW in the training data as “truth”, where we only allow the second alternative if it is at least 0.5 times as frequent as the most frequent.
- Reestimate $p(\mathbf{o}|\mathbf{w}) = p(\mathbf{o}|\mathbf{t}, \mathbf{w})p(\mathbf{t}|\mathbf{w})$ given these data

Error rate: 19.9%

Further Issues

- Need a better model of abbreviation:

OEPN OCEAN PERENNIAL
DALLIN DAVID ALLAIN
MASHPEE MARSH PROPERTIES
SEAVIEW SEASONAL VIEWS
WIGET WITH GUESTS

Some of these can be filtered by disallowing abbreviations that are assigned a high cost by the initial abbreviation model.

- Current abbreviation model makes no use of case information:
BTW (*by the way*); *DR* (*dining room*) ...
- Still lacking a model of what kinds of terms are *likely* to be abbreviated — $p(t|w)$: *BTW* \rightarrow *because the windows*