Finite-state formulation of hunspell spell-checkers

Tommi A Pirinen
tommi.pirinen@helsinki.fi

University of Helsinki
Department of Modern Languages
in IMCSIT 2010 CLA
Wisła, Poland

2010-10-20

FSTs in spell-checking

Language model:

- Dictionary: an automaton defininining correct words
- Checking highly efficient by lookup, 100 000—1 000 000 with typical FSTs
- Possible to inject word likelihoods as weights in FST without penalty in efficiency

Error model:

- Filter transducer mapping strings to strings
- e.g. edit distance (KEY, TRY), confusion sets (REP), phonetic folding (PHONE)

FSTs are modifiable on the fly with standard FST tools.

Hunspell dictionaries as FSTs

- The concatenation of prefixes, root and suffixes `Prefix* root Suffix*`
- All flags are included in automata as arcs of special symbols
- Making epsilon arcs back along compounding (e.g. to root or prefixes from COMPOUNDFLAG)
- Further compound etc. flags for morphotactics can be composed or intersected
- Use context rules against flags for hunspell match-delete-insert triplet with simple composition e.g. delete e before X flag if preceded by z: `e:0 -> z _ XFLAG`
- Other simple features: flags to remove word from suggestion dictionary, etc.

Hunspell error models as FSTs

- TRY and KEY: edit distance with different features from addition, deletion, substitution or swap of adjacent letters
- addition and deletion for each a, draw arcs $a : 0$, $0 : a$
- for each alphabet pair a, b, draw arc $a : b$ to auxiliary ending state q_{ab} and afterwards $b : a$ to end state
- edit distance without swaps can be built with 1 state, with swaps Σ^2 states, where Σ is size of TRY or KEY set
- the confusion set of REP is simple one path per replacement
- *all transitions can be weighted to attribute for relative ordering of the models
- *PHONE model: afilter relation from phonemic patterns to pseudophonemic representation and back

Combining language models and error models

- error model is filter mapping wrong forms to correct ones
- the erroneous input is transformed to correct variants using composition over error model and language model
- if both are weighted, weight combining is done by fst algebra

c	t	a	0	input
c	t:a	a:t	10	error model
c	a	t	1	language model
c	a	t	11	result

correcting simple typo by composition and tropical (penalty) weighting

Faithfulness of the hunspell FSTs at the moment

A work-in-progress:

Table: Difference between hunspell and FST with 2 errors

Language	**Hunspell**					f FST				
	1	$2-4$	$5-\infty$	Fp	M	1	$2-4$	$5-\infty$	Fp	M
Occitan	164	34	3	4	85	234	35	:5	0	16
Kurdish	200	26	4	6	6	214	17	4	0	7
Interlingua	817	95	2	13	73	814	92	17	3	72
Hungarian	338	45	6	16	21	354	38	8	6	22

Thank you.
(Demo of command line tools and the oowriter here)

slides and materials available through author's website

`http://www.helsinki.fi/%7Etapirine/`