ESSLLI 2010: Resource-light Morpho-syntactic Analysis of Highly Inflected Languages

Resource-light Approaches to Morphology

Anna Feldman & Jirka Hana

Overview

1. Linguistica
   - Intro
   - Signatures
   - Process
   - Evaluation & Problems
2. Yarowsky & Wicentowski 2000
   - Intro
   - Similarity measures
   - Combination
   - Resources
   - Problems

Linguistica

- (Goldsmith 2001)
- `http://linguistica.uchicago.edu/`
- Learns signatures (paradigms) together with roots they combine with
- Completely unsupervised: input = raw text (5K-500K tokens)
- Assumes suffix-based morphology

Signatures

- Signatures are sets of suffixes that are used with a given set of stems.

NULL.ed.ing	*betray, betrayed, betraying*
NULL.ed.ing.s	*remain, remained, remaining, remains*
NULL.s	*cow, cows*
e.ed.ing.es	*notice, noticed, noticing, notices*

- Similar to but not the same as paradigms:
  - Includes both derivational and inflectional affixes;
  - Purely corpus based, thus often not complete
    See `NULL.ed.ing` vs `NULL.ed.ing.s` above (the corpus contains *remains* but no *betrays*)
- Purely concatenative, so *blow/blew* would be analyzed as *bl* + *ow/ew* (if analyzed at all)

Top English signatures

Rank	Signature	#Stems	Rank	Signature	#Stems
1	NULL.ed.ing	69	16	e.es.ing	7
2	e.ed.ing	35	17	NULL.ly.ness	7
3	NULL.s	253	18	NULL.ness	20
4	NULL.ed.s	30	19	e.ing	18
5	NULL.ed.ing.s	14	20	NULL.ly.s	6
6	's.NULL.s	23	21	NULL.y	17
7	NULL.ly	105	22	NULL.er	16
8	NULL.ing.s	18	23	e.ed.es.ing	4
9	NULL.ed	89	24	NULL.ed.er.ing	4
10	NULL.ing	77	25	NULL.es	16
11	ed.ing	74	26	NULL.ful	13
12	's.NULL	65	27	NULL.e	13
13	e.ed	44	28	ed.s	13
14	e.es	42	29	e.ed.es	5
15	NULL.er.est.ly	5	30	ed.es.ing	5

Process

1. A set of heuristics is used to generate candidate signatures (together with roots they combine with)
2. The MDL metrics is used to accept or reject them

Step 1: Candidate generation – Word segmentation

- Uses heuristics to generate a list of potential affixes:
  - Collect all word-tails up to length six,
  - For each tail $n_1, n_2 \ldots n_k$, compute the following metric (where N_k is the total number of tail of length k):
    $\frac{C(n_1,n_2 \ldots n_k)}{N_k} \log \frac{C(n_1,n_2 \ldots n_k)}{C(n_1)C(n_2) \ldots C(n_k)}$
  - The first 100 top ranking candidates are chosen
- Other heuristics are possible
- Words in the corpus are segmented according to these candidates.
- For each stem collect the list associated suffixes (incl. NULL), i.e., the signature for that stem.
- All signatures associated only with one stem or only with one suffix are dropped.

Step 2: Candidate evaluation

- Not all suggested signatures are useful. They need to be evaluated.
- Use Minimum Description Length to filter them

Minimum description length (MDL)

- Criterion for selecting among models
- Developed by (Rissanen 1989); see also (Kazakov 1997; Marcken 1995)
- According to MDL, the best model is the one which gives the most compact description of the data, including the description of the model itself.
- In our case:
  - A grammar (the model) can be used to compress a corpus.
  - The better the morphological description is, the better the compression is.
- The size of the grammar and corpus is measured in bits.

Evaluation

- Applied to English, French, Italian, Spanish, and Latin.
- Identification of morpheme boundaries in 1000-word corpus
- Evaluated subjectively, because there is no gold standard
- Not always clear where the boundary *should* be:
  aboli-tion vs. abol-ish; Alexand-er, Alex-is, John-son; alumn-i
- English: precision = 85.9 %; recall = 90.4 %

Problems

- Analyzes only suffixes (easily generalizable to prefixes as well).
- Handling stem-internal changes would require significant overhaul.
- All phonological/graphemic changes accompanying inflection, must be factored into suffixes:
  English: *hated* (*hate+ed*) analyzed as *hat-ed*
  Russian: *plak-at'* 'cry$_{inf}$ and *plač-et* 'cry$_{pres.3pl}$' analyzed as *pla-kat' / pla-čet'*
- Considers only information contained in individual words and their frequencies. Ignores any contextual information (reflecting syntactical and semantical information).

Yarowsky & Wicentowski 2000

- Resource-light induction of inflectional paradigms (suffixal and irregular).
- Tested on induction of English/Spanish present-past verb pairs
- Forms of the same lexeme are discovered using a combination of four measures:
  - expected frequency distributions,
  - context similarity,
  - phonemic/orthographic similarity,
  - model of suffix and stem-change probabilities.

Process

1. Estimate a probabilistic alignment between inflected forms
2. Train a supervised morphological analysis learner on a weighted subset of these aligned pairs.
3. Use the result of Step 2 as either a stand-alone analyzer or a probabilistic scoring component to iteratively refine the alignment in Step 1.

Frequency similarity

- Two forms belong to the same lexeme, when their relative frequency fits the expected distribution.
  sing/sang – 1204/1427 – *sing/singed* – 1204/9 – *singe/singed* – 2/9
- The distribution is approximated by the distribution of regular forms.

Frequency similarity

- Two forms belong to the same lexeme, when their relative frequency fits the expected distribution.
  sing/sang – 1204/1427 – *sing/singed* – 1204/9 – *singe/singed* – 2/9
- The distribution is approximated by the distribution of regular forms.

- Works for verbal tense, but sometimes one can expect multimodal distribution.
- For example, for nouns, the distribution is different for count nouns, mass nouns, plurale-tantum nouns, currency names, proper nouns, . . .

Context similarity

- Forms of the same lemma have similar selectional preferences
- Related verbs tend to occur with similar subjects/objects.
- Arguments identified by simple regular expressions.
- Neither recall nor precission is perfect, but with a large corpus this is tolerable.

Context similarity

- Forms of the same lemma have similar selectional preferences
- Related verbs tend to occur with similar subjects/objects.
- Arguments identified by simple regular expressions.
- Neither recall nor precission is perfect, but with a large corpus this is tolerable.

- Works well for verbs, but other POS have much less strict subcategorization requirements.
- Some inflectional categories influence subcategorization, e.g., aspect in Slavic

Form similarity

- Form (phonemic/graphemic) similarity is measured by weighted Levenshtein measure (Levenshtein 1966).

Form similarity

- Form (phonemic/graphemic) similarity is measured by weighted Levenshtein measure (Levenshtein 1966).
- Levenshtein distance (edit distance)
  - Distance between two strings is the minimal number of character substitutions, insertion or deletions
  - Used in many different applications
  - Can be calculated by an efficient dynamic programming algorithm
  - Various modifications exists – additional operations, operations' cost depend on the modified characters, etc.

Form similarity

- Form (phonemic/graphemic) similarity is measured by weighted Levenshtein measure (Levenshtein 1966).
- Levenshtein distance (edit distance)
  - Distance between two strings is the minimal number of character substitutions, insertion or deletions
  - Used in many different applications
  - Can be calculated by an efficient dynamic programming algorithm
  - Various modifications exists – additional operations, operations' cost depend on the modified characters, etc.
- Edit cost operate on character clusters
- Four types of clusters are distinguished: V, V+, C, C+

Morphological Transformation Probabilities

In step k+1, a probabilistic generative model is trained on the basis of the analyzer obtained in step k.

$$
\begin{aligned}
&P(\text{form} \mid \text{root}, \text{suffix}, \text{pos}) = P(a \rightarrow b \mid \text{root}, \text{suffix}, \text{pos}) = \\
&P(cb + s \mid ca, +s, \text{pos}) = P(a \rightarrow b \mid ca, +s, \text{pos}) = \\
&\quad \approx \quad \lambda_1 P(a \rightarrow b \mid \text{last}_3(\text{root}), \text{suffix}, \text{pos}) \\
&\quad + \quad (1-\lambda_1)\lambda_2 P(a \rightarrow b \mid \text{last}_2(\text{root}), \text{suffix}, \text{pos}) \\
&\quad + \quad (1-\lambda_2)\lambda_3 P(a \rightarrow b \mid \text{last}_1(\text{root}), \text{suffix}, \text{pos}) \\
&\quad + \quad (1-\lambda_3)\lambda_4 P(a \rightarrow b \mid \text{suffix}, \text{pos}) \\
&\quad + \quad (1-\lambda_4) P(a \rightarrow \ b)
\end{aligned}
$$

Combination

- Of the four measures, no single model is sufficiently effective on its own.
  English present-past tense verb pairs:

	Iteration	Accuracy
Frequency	1	9.8 %
Levenshtein	1	31.3%
Context	1	28.0 %
F+L+C	1	71.6 %
F+L+C+M	1	96.5%
F+L+C+M	conv	99.2%

- Therefore, traditional classifier combination techniques are applied to merge scores of the four models.

Required resources

1. List of inflectional categories, each with canonical suffixes.
2. A large unannotated text corpus.
3. A list of the candidate noun, verb, and adjective base forms (typically obtainable from a dictionary)
4. A rough mechanism for identifying the candidate parts of speech of the remaining vocabulary, not based on morphological analysis
5. A list of consonants and vowels.
6. Optionally, a list of common function words.
7. Optionally, various distance/similarity tables generated by the same algorithm on previously studied (related) languages - used as seed information.

Problems

- Suffix/tail based
  Generalized by (Wicentowski 2004), but no longer unsurpervised.
- The "rough" mechanism for identifying POS relies on word-order templates. Good for English, not so much for Polish.
- Other problems mentioned above