

ESLLI 2013: Computational Morphology

Resource-light Approaches to Morphology

Jirka Hana & Anna Feldman

Overview

- 1 Linguistica
 - Intro
 - Signatures
 - Process
 - Evaluation & Problems
- 2 Yarowsky & Wicentowski 2000
 - Intro
 - Similarity measures
 - Combination
 - Resources
 - Problems
- 3 Schone & Jurafsky 2000
 - Algorithm
 - Candidate affixes
 - Computing semantic vectors
 - Subrules
- 4 Cucerzan & Yarowsky 2002

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 $n_1, n_2 \dots n_k$, compute the following metric (where N_k is the total number of tail of length k):
$$\frac{C(n_1, n_2 \dots n_k)}{N_k} \log \frac{C(n_1, n_2 \dots n_k)}{C(n_1)C(n_2) \dots C(n_k)}$$
 - 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; alumni-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).

Linguistica is a strictly concatenative and therefore, it is not suitable for discovering paradigms employing other morphological processes (interfixes, templates, metathesis, deletion, etc.).

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 precision 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 precision 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 .

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

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 unsupervised.
- The “rough” mechanism for identifying POS relies on word-order templates. Good for English, not so much for Polish.
- Other problems mentioned above

Knowledge-free Induction of Morphology using LSA

- unsupervised
- input: a space-separated, unlabeled corpus of English (8M words)
- output: "conflation sets" of morphologically related words

Their algorithm is divided into four parts:

- 1 Hypothesize candidate affixes
- 2 Identify pairs of candidate affixes which may be morphological variants, e.g. (*ed*, *ing*) or (*s*, *NULL*).
- 3 Collect contextual information about all word pairs which share these morphologically variant affixes, e.g. (*walked*, *walking*) or (*walks*, *walk*).
- 4 Determine “morphologically relatedness” for those word pairs with similar semantics (as defined by their ± 50 word context).
- 5 Insert words into a trie and extract potential affixes by observing those places in the trie where branching occurs.

Hypothesize candidate affixes

Two words w_1 and w_2 are said to be p -similar if and only if:

- a. the first p characters of w_1 are the same as the first p characters of w_2
- b. the $p + 1$ characters of w_1 and w_2 are not the same
Ex. *walks* and *walking* are 4-similar, as are *walk* and *walks*.
- c. These pairs are not 5-similar, by rule (a), and not 3-similar, by rule(b).

Selecting candidate pairs

- Identify all pairs of affixes which descend from the same node (e.g. “s”, NULL) and call these pairs rules
- Two words which share the same stem and affix rule form a PPMV (pair of potential morphological variants).
- For example, (“car”, “cars”)

The ruleset of a rule is the set of all PPMVs that have that rule in common. Here, the ruleset of (“s”, NULL) would be the set “cars/car”, “cares/care” The algorithm finds the ruleset for each rule.

Computing semantic vectors

- Decide which of the rulesets that have been generated contain pairs of words which are semantically related.
- S & J don't compute cosine scores directly on each vector in the matrix; rather, they first apply singular value decomposition (SVD) to the matrix (aka Latent Semantic Analysis or LSA; Landauer et al. 1988)
- LSA: The matrix is projected (compressed) into a lower k -dimensional subspace such that the k dimensions of this new subspace are the k most informative dimensions.
- This results in a matrix of "semantic vectors".

Comparing semantic vectors

- Determine if a pair of words in a PPMV are semantically related by defining a normalized cosine score, or NCS
- Compute the NCS between two semantic vectors

Sample normalized cosine scores (NCSs)

PPMV	cos
ally/allies	6.5
car/cars	5.6
dirty/dirt	2.4
rating/rate	0.97
car/cares	-0.14
car/caring	-0.71
car/cared	-0.96
ally/all	-1.3

A score over 2.0 would be rare for a random event

Ruleset-level Statistics

- Determine if a rule is valid (e.g., ("s", NULL) vs. "e", "age")
- So, compute the NCS for PPMVs of a particular rule
- The NCS scores for invalid PPMVs should be distributed normally
- Calculate $\Pr(\text{true})$, the probability that a particular ruleset is valid (=non-random)

Consider the rule (“es”, NULL)

- This rule pairs together “car/cares” which have a low NCS.
- This rule is sometimes valid (“church/churches”, “mash/mashes”, “miss”, “misses”)
- The problem is that we have to decide whether the rule (“es”, NULL) is valid based on members of the ruleset and there will be a lot of incorrect (“es”, NULL) matches (“hat/hates”, “cap/capes”, “sit/sites”)...

...So how can we remedy this?

- Based on the intuition that these rules are phonological variations of other rules, we might expect to find that the (“es”, NULL) rule applies in only specific cases.
- If so, there should be specific environments where we’d find that there were higher than average NCS scores:

Rule/Subrule	Average	Std Dev	# instances
(“es”, NULL)	1.62	2.43	173
(“ches”, “ch”)	2.20	1.66	32
(“shes”, “sh”)	2.39	1.52	15
(“res”, “r”)	-0.69	0.47	6
(“tes”, “t”)	-0.58	0.93	11

Results

- S&J set a threshold (T_5) for determining whether or not to believe that a particular PPMV in a rule set is non-random.

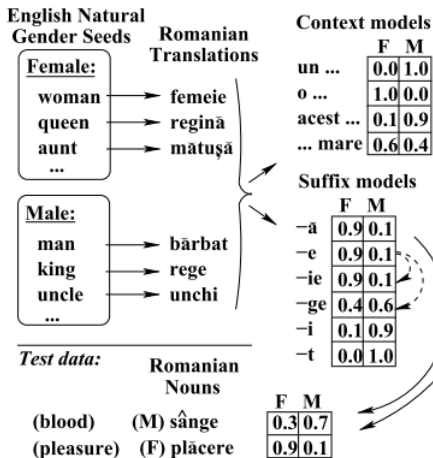
	(Goldsmith) Linguistica	S&J $T_5=0.5$	S&J $T_5=0.7$	S&J $T_5=0.85$
Precision	83.0%	85.0%	90.0%	92.6%
Recall	80.4%	81.8%	79.3%	76.6%
F-Score	81.6%	83.4%	84.3%	83.9%

Schone & Jurafsky 2001: Knowledge Free Induction of Inflectional Morphology

- Extends the Schone and Jurafsky (2000) work
- Includes additional measures because of the shortcomings of semantics alone.
- (“reusability”, “use”) is labeled as a morphological variant but is discarded since the words are not semantically similar enough.
- (“as”, “a”) is deemed acceptable because, since they appear so frequently, neither has much semantic information, so, in that respect, they are semantically very similar.
- Introduction of bad rules: “ho-/∅” \neq “pi-/∅” for “hog/pig” which have very similar semantics [81 unique pairs].

Bootstrapping a Multilingual Part-of-speech Tagger in One Person-day

- Bootstrap a fine-grained, broad-coverage POS tagger in a new language using only one personday of data acquisition effort.
- Resources:
 - ① an online or hard-copy pocket-sized bilingual dictionary
 - ② a basic library reference grammar
 - ③ access to an existing monolingual text corpus in the language
- Induce initial lexical POS distributions from English translations in a bilingual dictionary without POS tags.
- Handle irregular, regular and semi-regular morphology through a robust generative model using weighted Levenshtein alignments.
- Induce grammatical gender via global modeling of context window feature agreement
- Interactively train context and lexical prior models for fine-grained POS tag spaces.



Morfessor

Morfessor (Creutz & Lagus 2002, 2004, 2005)

- splits words into morphemes in a hierarchical fashion
- more suitable for agglutinative languages with a large number of morphemes per word
- an HMM is used to add a simple morphotactic model.

kohonen:etal:2010 modify Morfessor to allow semi-supervised learning.

- Add a set of 0-10,000 correctly segmented words
- Optimize separate weights for unlabeled and labeled data by using a heldout of 500 correctly segmented words

Table : Results of **kohonen:etal:2010** with various size of training data

labeled data size	kohonen:etal:2010						Morfessor	soa
	500	600	800	1.5K	3.5K	10.5K	0	0
English	61.1	65.2	65.6	68.3	69.1	72.9	59.8	66.2
Finnish	49.1	52.7	54.9	56.4	58.2	60.3	44.6	52.5

Paramor

- *Paramor* (**monson:2009**) is a system for unsupervised acquisition of paradigms from a list of words.
- It learns paradigms and a lexicon in several steps.
 - 1 Consider all possible segmentations of words into candidate stems and endings.
 - 2 Creates schemes (partial paradigms with the associated stems) by joining endings that share a large number of associated stems.
 - 3 Similar schemes (as measured by cosine similarity) are merged.
 - 4 Schemes proposing frequent morpheme boundaries not consistent with boundaries proposed by the character entropy measure are