Discovering Morphological Paradigms from Plain Text Using a Dirichlet Process Mixture Model

Dreyer et al. (2011)

Amey Chaugule
achaugu2@illinois.edu

Introduction

- Statistical NLP is often very difficult for morphologically rich languages.
- One must learn lexical features individually for each word form as it is not possible to generalise across inflections.
- This paper proposes a mostly unsupervised generative probabilistic model to capture morphological relationships.

Introduction

- The inference algorithm reconstructs *token, type* & *grammar* about a language's morphology.
- **Tokens:** Each word in the corpus has 3 tags. Ex. *Broken* (1) POS – Verb (2) Inflection – past participle and (3) Lexeme – *break*.
- **Types:** This is a morphological paradigm, which in our case is a grid of all the inflected forms of a some lexeme.
- **Grammar:** Parameter θ describes the general patterns of the language. Mote Carlo EM is used to estimate this.

Overview of the Model

Modeling Morphological Alternations

- Given a lemma *x* we could predict its inflected form *y*.
- This joint distribution is a family which can be described by this log-linear model :

$$p(x, y) = \sum_a p(x, y, a) \propto \sum_a \exp(\vec{\theta} \cdot \vec{f}(x, y, a))$$

- *f* is local feature vector and parameter ϑ could penalise or reward specific features.

Overview of the Model

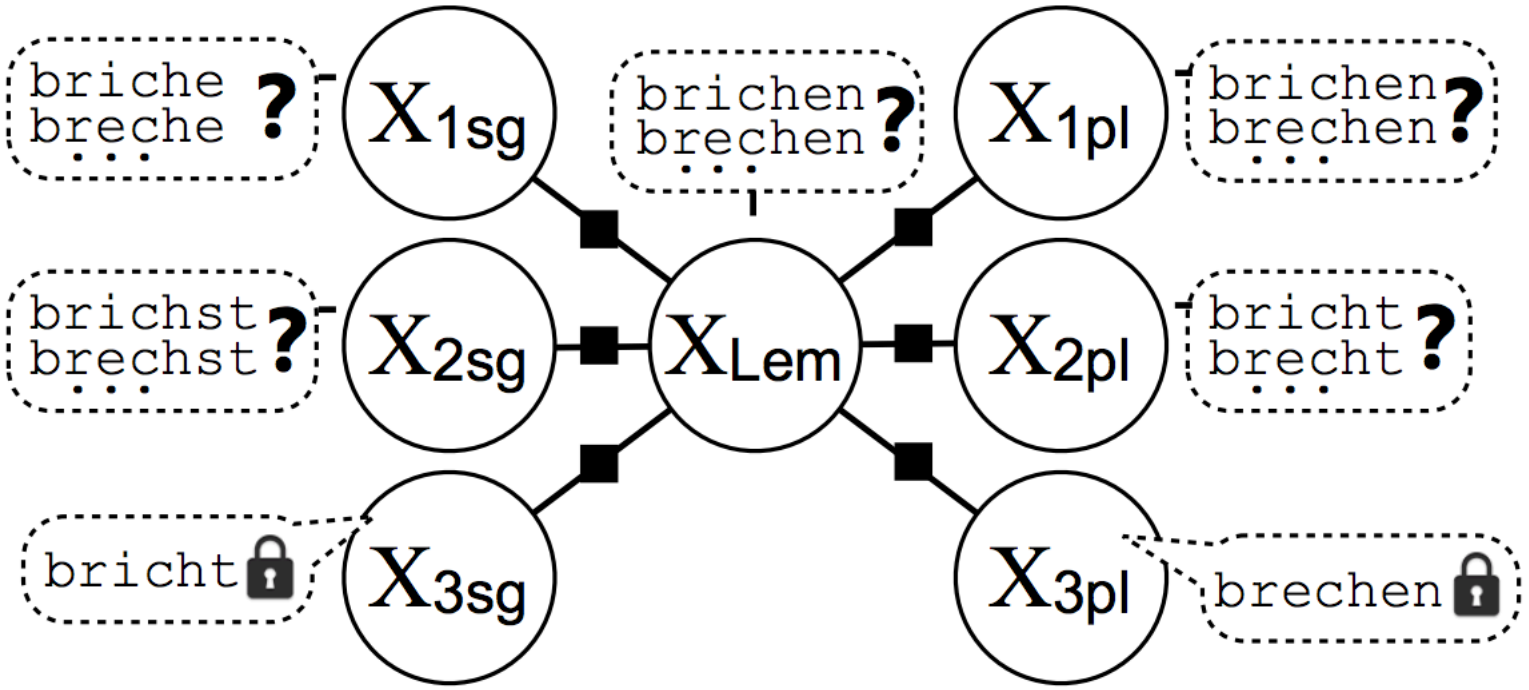
Modeling Morphological Paradigm

- The underlying presumption here is that some language specific distribution $p(\pi)$ defines whether a paradigm π is a grammatical way for a lexeme to express itself.
- Learning $p(\pi)$ helps us reconstruct paradigms.
- $p(\pi)$ is modeled as a renormalised product of many pairwise distributions *Prs(Xr,Xs)* each having log linear form.

Overview of the Model

Modeling Morphological Paradigm

This is an undirected graphical model (MRF) over *string-valued* random variables *Xs.*



Overview of the Model

Modeling the Lexicon

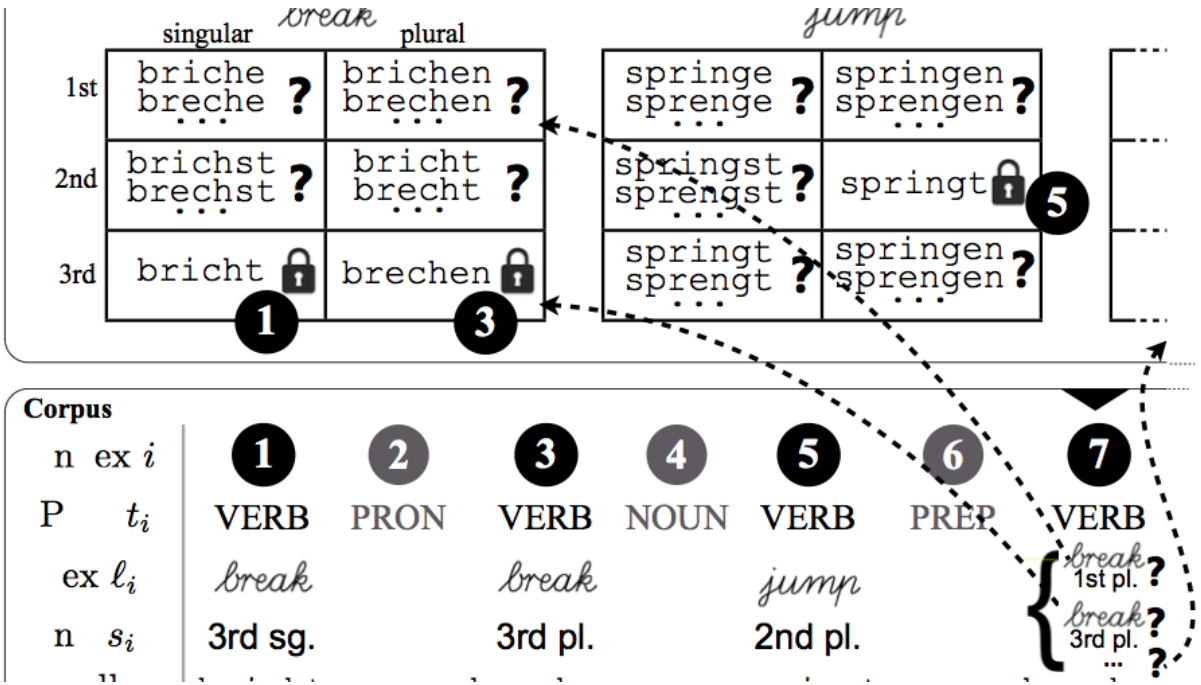
1. Choose parameter θ of the MRF which defines p(π):which paradigms are *a priori*. θ is sampled from a Gaussian prior.
2. Choose a distribution over abstract lexemes which is sampled from a Dirichlet process.
3. For each lexeme choose a distribution over its inflections. This is again sampled from a Dirichlet.
4. For each lexeme choose a paradigm that can be used to express the lexeme orthographically.

Inference and Learning

Gibbs sampling over the corpus

- The inference task is to extract the the *lexeme* and *inflection* per token.
- Using a collapsed Gibbs sampler, reanalysis of of each token is repeatedly guessed in context of all other tokens.
- Eventually similar tokens get clustered together.

Inference and Learning



A state of the Gibbs sampler. Note that each of the tokens *i* has been tagged with POS *Ti,* lexeme *Li* and inflection *Si.*

Inference and Learning

Key intuitions –

1. Current analyses of other tokens tagged with same part of speech implies a posterior distribution over that POS lexicon.
2. Belief propagation gives us which other inflection of a given lexeme maps to a token with same spelling.
3. The number of tokens associated with a lexeme suggests popularity. (e.g. Chinese Restaurant Process “Rich get richer”)

Inference and Learning
Monte Carlo EM Training of θ

- For a given θ Gibbs sampler converges to posterior distribution over analyses of the entire corpus.

- To improve the estimate, θ is periodically adjusted to maximise the probability of most recent samples.

Inference and Learning
Collapsed Representation of the Lexicon

- Lexicon is collapsed out of the sampler.
- If *(l,s)* points to at least one token *i* then we know that *(l,s)* is spelt as *Wi*.
- If the spelling of (l,s) isn't known but some other spellings in l's paradigm are known then store a truncated distribution that gives 25 most likely spellings of (l,s).
- Last case is where we know nothing about l thus all such l share the same marginal distribution over (l,s). Probabilistic finite state automata is used to approximate this marginal.

Mixture Model

- This inference model clusters words together by tagging them with the same lexeme.
- Thus the base distribution p(π) predicts word co-occurrence within a paradigm.
- Thus the model assigns words to a particular inflection slot in the paradigm.

Dirichlet Process Mixture Model

- Natural languages have an infinite lexicon although most lexemes have a very low probability.
- Thus the mixture model uses infinite number of mixture components.
- DPMM first generates a distribution over countably many lexemes and then generated a weighted paradigm per lexeme.

Formal Generative Model

1. First grammar variables need to be selected from the prior.

$$p(\vec{\theta}), p(\overrightarrow{\phi_t}), p(\alpha_t), p(\alpha'_t)$$

2. Let $D_t(\pi)$ be a distribution over paradigms of POS t. For each discovered lexeme (t, l) paradigm $\pi_{t,l}$ can be drawn from D_t.
3. For each POS t langauges has a distribution Gt(l) over lexemeswhere Gt is drawn from a Dirichlet process DP(Gt,αt) where G is the base distribution over lexemes l.
4. Inflectional distribution $p(H_{t,\ell} \mid \overrightarrow{\phi_t}, \alpha'_t)$. For each tagged lexeme (t,l) the language specifies some distribution Ht. Ht is a log linear distribution with parameters that refer to features of inflection. Ht,l is an independent draw from a finite dimensional Dirichlet distribution with mean Ht and concentration parameter α.

Formal Generative Model

5. The POS tag sequence for the experimental model is given but in a general use case to discover tags, we can model the tag sequence by a Markov model.

6. A lexeme token depends on its tag. Draw Li from Gti. Thus, $p(\ell_i \mid G_{t_i}) = \bar{G}_{t_i}(\ell_i)$

7. Inflection slot depends on the tagged lexeme. We draw Si from H_{t_i,ℓ_i}, so $p(s_i \mid H_{t_i,\ell_i}) = H_{t_i,\ell_i}(s_i)$.

8. Given a tag, lexeme and inflection at position i, word wi is generated by simply looking up its spelling in an appropriate paradigm.

9. Gt is unspecified given the sampler state but it only appears in 3 & 6 and can be integrated out of their product to get a collapsed sub-model which generates p(l | t, α) directly. This is akin to Chinese restaurant whole tables are labeled with lexemes each customer i enters restaurant ti, in turn and li denotes the table he joins.

Formal Generative Model

10. Similarly infinitely many lexeme-specific distributions Ht,l can be integrated out of product of 4 & 7 and replaced with a collapsed distribution

$$p(\vec{s} \mid \vec{\ell}, \vec{t}, \overrightarrow{\phi_t}, \overrightarrow{\alpha'})$$

Experiments

- As corpus they used first 1 million and 10 million words from WaCky.

- Verbal inflectional paradigms from CELEX morphological database were used to seed the paradigms.

Type based Evaluation

Bin	Frequency	# Verb Forms
1	0–9	116,776
2	10–99	4,623
3	100–999	1,048
4	1,000–9,999	95
5	10,000–	10
all	*any*	122,552

Table 3: The inflected verb forms from 5,615 inflectional paradigms, split into 5 token frequency bins. The frequencies are based on the 10-million word corpus.

Token based Evaluation

Bin	50 seed paradigms			100 seed paradigms		
	0	10^6	10^7	0	10^6	10^7
1	90.5	91.0	**91.3**	92.1	92.4	**92.6**
2	78.1	**84.5**	84.4	80.2	**85.5**	85.1
3	71.6	**79.3**	78.1	73.3	**80.2**	79.1
4	57.4	61.4	**61.8**	57.4	**62.0**	59.9
5	20.7	**25.0**	**25.0**	20.7	**25.0**	**25.0**
all	52.6	57.5	**57.8**	53.4	**58.5**	57.8
all (e.d.)	1.18	1.07	**1.03**	1.16	1.02	**1.01**

Table 4: Token-based analysis: Whole-word accuracy results split into different frequency bins. In the last two rows, all predictions are included, weighted by the frequency of the form to predict. Last row is edit distance.

Conclusion

- The authors formulated a a framework for obtaining both morphological annotations and the unbounded lexicon that completed the morphological paradigms.
- They were able to run the sampler over a corpus of 10 million words and by inferring everything jointly, they were able to reduce the prediction error for inflections by upto 10%.