Title, author list not final. Page limit: 5 + 2 bib

Learning Definiteness across Languages using Classifiers [$^{NS}_{S}$ maybe: A Classification and Classifier for Cross-linguistic Functions of Definiteness]

Abstract

Definiteness seems to be a nonhomogeneous category across languages denoting various semantic/ pragmatic aspects. The semantic - pragmatic information may be grouped differently in different languages to result in grammaticalizations corresponding to definiteness in these languages. In this paper, we attempt to find out these clusterings of semantic- pragmatic features relevant to two languages English and Hindi in an attempt to find out how definiteness is grammaticalizd in these languages. This work has twofold benefits: for the linguistic studies, it provides us a concrete set of groupings for each language which tells us how definiteness is conceptualized and repesented in these languages; in addition, these findings serve as a guide in improving machine translations across these languages.

1 Introduction

[$^{A}_{B}$ Here we discuss about why we should study definiteness- linguistically a hard problem, also it has applications in machine translation. Discuss about grammaticalization as a general problem (differences across languages), then grammaticalization of definiteness across languages, some examples to show differences across languages.] [$^{NS}_{S}$ we review the annotation scheme in §2; etc.]

2 Annotation scheme

[$^{A}_{B}$ A brief discussion of the Annotation scheme] [$^{NS}_{S}$ we should probably have a name for the scheme—something like "Functions of Definiteness Across Languages (FDAL)"? and a name for the categories (I have been calling them semantic functions).] [$^{NS}_{S}$ suggestion: lead with a discourse excerpt, ideally illustrating a few of the categories, a nested NP, and an anaphoric link across sentence boundaries. I can help format it.]

[$^{NS}_{S}$ cite papers like this: Bhatia et al. (2014) (but probably anonymize this for review) or (Reiter and Frank, 2010)]

3 Data

[$^{A}_{B}$ description of data, IAA, other statistics- n(annotations), n(annotations/ category) classifiers for English, and Hindi- baseline, accuracy etc, confusion matrix- classifier predictions and gold standard may be a brief discussion of feature ablation study- which features are most helpful at least in these languages corresponding to individual categories (a clue to grammaticalization) Principle component analysis, clustering of features]

4 Classification framework

To model the relationship between the grammar of definiteness and its semantic functions in a data-driven fashion, we work within the supervised framework of feature-rich discriminative classification, treating the functional categories from §2 as output labels y and various lexical, morphological, and syntactic characteristics of the language as features of the input x. Specifically, we learn a probabilistic log-linear model similar to multiclass logistic regression, but deviating in the following ways:

- Logistic regression treats each output label (response) as atomic; we decompose each into *attributes* based on their linguistic definitions, enabling commonalities between related labels to be recognized. Each weight in the model corresponds to a feature that mediates between *percepts* (characteristics of the input noun phrase) and attributes (characteristics of the label).
- Logistic regression assumes a prediction is either correct or incorrect. We incorporate a *cost function* that gives partial credit during learning

- **NONANAPHORA** $[-A,-B]$ **(00)**
  - **UNIQUE** $[+U]$ **(00)**
    - ∗ **UNIQ_HEARER_OLD** $[-G,+O,+S]$ **(00)**
      - · UNIQ_PHYSICAL_COPRESENCE $[+R]$ (00)
      - · UNIQ_LARGER_SITUATION $[+R]$ (00)
      - · UNIQ_PREDICATIVE_IDENTITY $[+P]$ (00)
    - ∗ UNIQ_HEARER_NEW $[-O]$
  - **NONUNIQUE** $[-U]$
    - ∗ **NONUNIQ_HEARER_OLD** $[+O]$
      - · NONUNIQ_PHYSICAL_COPRESENCE $[-G,+R,+S]$
      - · NONUNIQ_LARGER_SITUATION $[-G,+R,+S]$
      - · NONUNIQ_PREDICATIVE_IDENTITY $[+P]$
    - ∗ NONUNIQ_HEARER_NEW_SPEC $[-G,-O,+R,+S]$
    - ∗ NONUNIQ_NONSPEC $[-G,-S]$
  - **GENERIC** $[+G,-R]$
    - ∗ GENERIC_KINDLEVEL
    - ∗ GENERIC_INDIVIDUALLEVEL
- **ANAPHORA** $[+A]$
  - **BASIC** $[+O,-B]$
    - ∗ SAME_HEAD
    - ∗ DIFFERENT_HEAD
  - **EXTENDED** $[+B]$
    - ∗ BRIDGING_NOMINAL $[-G,+R,+S]$
    - ∗ BRIDGING_EVENT $[+R,+S]$
    - ∗ BRIDGING_RESTRICTIVEMODIFIER $[-G,+S]$
    - ∗ BRIDGING_SUBTYPE_INSTANCE $[-G]$
    - ∗ BRIDGING_OTHERCONTEXT $[+O]$
- **MISCELLANEOUS** $[-R]$
  - PLEONASTIC $[-B,-P]$
  - QUANTIFIED
  - PREDICATIVE_EQUATIVE_ROLE $[-B,+P]$
  - PART_OF_NONCOMPOSITIONAL_MWE
  - MEASURE_NONREFERENTIAL
  - OTHER_NONREFERENTIAL

Figure 1: Taxonomy of definiteness functions, with number of occurrences in the training data [NS TODO]. Internal (non-leaf) labels are in bold.[NS added two: Generic and Miscellaneous, and also reversed Nonreferential to Referential. is that OK?][NS TODO: normalize capitalization] +/− values are shown for ternary attributes Anaphoric, Bridging, Generic, Hearer-Old, Predicative, Referential, Specific, and Unique; these are inherited from supercategories, but otherwise default to 0.[NS nonuniq_nonspec is −generic, right?] Thus, for example, the full attribute specification for UNIQ_PHYSICAL_COPRESENCE is $[-A,-B,-G,+O,0P,+R,+S,+U]$.

when a related label is predicted, so the learned model will better match our evaluation measure.

- Logistic regression assumes the space of possible predictions matches the space of labels observed in the training data; we allow more abstract labels to be predicted, which can receive partial credit. The scoring scheme encourages the predictor to "back off" to a coarser label if it is not sufficiently confident about a fine-grained label.

These decisions are aimed at attaining better predictive accuracy as well as feature weights that better describe the form–function interactions we are interested in recovering.

Our setup is formalized below, where we discuss the mathematical model and linguistically motivated features.

4.1 Model

At test time, we model the probability of semantic label y conditional on a [NS gold?] noun phrase x as follows:

$$p_{\boldsymbol{\theta}}(y|x) = \log \frac{\exp \boldsymbol{\theta}^\top \mathbf{f}(x,y)}{\sum_{y' \in \mathcal{Y}} \exp \boldsymbol{\theta}^\top \mathbf{f}(x,y')} \qquad (1)$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ is a vector of parameters (feature weights), and $\mathbf{f} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ is the feature function over input–label pairs. The feature function is defined as follows:

$$\mathbf{f}(x,y) = \boldsymbol{\phi}(x) \times \tilde{\boldsymbol{\omega}}(y) \qquad (2)$$

where the percept function $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^c$ produces a vector of real-valued characteristics of the input, and the attribute function $\tilde{\boldsymbol{\omega}} : \mathcal{Y} \to \{0,1\}^a$ encodes characteristics of each label. There is a feature for every percept–attribute pairing: so $d = c \cdot a$ and $f_{(i-1)a+j}(x,y) = \phi_i(x)\tilde{\omega}_j(y), 1 \le i \le c, 1 \le j \le a$. The contents of the percept and attribute functions are detailed in §§4.3 and 4.2.

For prediction, having learned weights $\hat{\boldsymbol{\theta}}$ we choose the y that maximizes this probability:

$$\hat{y} \leftarrow \arg\max_{y' \in \mathcal{Y}} p_{\hat{\boldsymbol{\theta}}}(y|x) \qquad (3)$$

Training optimizes $\hat{\boldsymbol{\theta}}$ so as to maximize the L_1-regularized *softmax-margin* learning objective (**?**) over the training data $\mathcal{D}$:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D})$$

$$L(\boldsymbol{\theta}, \mathcal{D}) = -\lambda ||\boldsymbol{\theta}||_1 + \sum_{\langle x,y \rangle \in \mathcal{D}} \log \frac{\exp \boldsymbol{\theta}^\top \mathbf{f}(x,y)}{\sum_{y' \in \mathcal{Y}} \exp \left(\boldsymbol{\theta}^\top \mathbf{f}(x,y') + cost(y,y') \right)}$$

The *cost function* allows us to penalize some errors more than others during training, taking into account the linguistic functions of the labels. It

is zero for the gold label and nonnegative for the others.[$^{NS}_{S}$ intuition]

This framework gives us several ways to design a classifier appropriate to the task: the attributes, the space of labels $\mathcal{Y}$ to consider, and the cost function, and of course, the features themselves. With $cost(y,y') = 0$, $\mathcal{Y} = \{gold\ labels\ in\ training\}$, and $\tilde{\boldsymbol{\omega}}(y) = the\ identity\ of\ the\ label$, this reduces to standard logistic regression.

4.2 Features

[$^{NS}_{S}$ TODO]

4.3 Attributes

As noted above, though labels are organized into a tree hierarchy, there are actually several dimensions of commonality that suggest different groupings. These attributes are encoded as ternary characteristics; for each label (including internal labels), every one of the 8 attributes is assigned a value of +, −, or 0 (refer to fig. 1). In order to capture these similarities in the model's features and cost function, we define the attribute vector function $\boldsymbol{\omega}(y) =$

$$[y, A(y), B(y), G(y), O(y), P(y), R(y), S(y), U(y)]^\top$$

where $A : \mathcal{L} \cup \mathcal{I} \to \{+,-,0\}$ returns the value for Anaphoric, $B(y)$ for Bridging, etc. The identity of the label is also included in the vector so that different labels are always recognized as different by the attribute function. The categorical components of this vector are then binarized to form $\tilde{\boldsymbol{\omega}}(y)$; however, instead of a binary component that fires for the 0 value of each ternary attribute, there is a component that fires for *any* value of the attribute—a sort of bias term. The weights assigned to features incorporating + or − attribute values, then, are easily interpreted as deviations relative to the bias.

4.4 Cost and label space

The definiteness function hierarchy presented in fig. 1 consists of 24 *leaf labels*, which will be denoted $\mathcal{L}$, and 10 more abstract *intermediate labels*, denoted $\mathcal{I}$. All of the gold labels in the training data are from $\mathcal{L}$, but we give our model the option to predict more abstract labels to receive partial credit. We will therefore use $\mathcal{Y} = \mathcal{L} \cup \mathcal{I}$.

The relatedness of leaf label pairs is determined by the dot product of the two original attribute vectors: let $\Delta(\ell,\ell') = |\boldsymbol{\omega}(\ell) \cap \boldsymbol{\omega}(\ell')|^{-1}$.[1] For a gold leaf label ℓ and an internal label $\imath$, $\Delta(\ell,\imath)$ = the distance in the hierarchy between ℓ and $\imath$.[$^{NS}_{S}$ TODO: ensure if $\imath$ is an ancestor of ℓ, this is less than choosing another leaf dominated by $\imath$]. (There is no need to define $\Delta(\imath,\cdot)$, as the training set does not contain intermediate labels.)

5 Evaluation

The following measures will be used to evaluate our predictor against the gold standard for the held-out evaluation (dev or test) set $\mathcal{E}$:

- **Exact match:** This gives credit only where the predicted and gold labels are identical. When the model is allowed to predict internal labels, we will report overall precision and recall of leaf labels. Otherwise, we report accuracy.
- **By leaf label:** We also compute precision and recall of each leaf label to determine which categories are reliably predicted.
- **Soft match:** This accuracy measure gives partial credit where the predicted and gold labels are related. It is computed as the Δ function in §4.4 [$^{NS}_{S}$ normalized to be between 0 and 1?].
- **Perplexity:** This determines how "surprised" our model is by the gold labels in the test set; the greater the probability mass assigned to the true labels, the higher the score. It is computed as $2^{\left(\sum_{\langle x,y\rangle \in \mathcal{E}} \log_2 p_{\hat{\boldsymbol{\theta}}}(y|x)\right)/|\mathcal{E}|}$.

6 Experiments

[$^{NS}_{S}$ English: ±cost function, ±non-identity attributes, ±predicting intermediate labels]

[$^{NS}_{S}$ maybe: which attribute groupings produce the best classifier, if we want to force a hierarchy]

[$^{NS}_{S}$ feature/attribute ablations]

[$^{NS}_{S}$ Hindi?]

7 Discussion: Error analysis, comparison across languages

[$^{A}_{B}$ confusion matrix - Hindi vs Eng (for common annotations)- semantic annotations differences in annotations- why- different grammatical constructions, e.g. NP in 1 language but predicate in another etc. what do similarities tell us about grammaticalization, what do differences tell. discuss- how this analysis can help in MT]

[1] By the intersection of two attribute vectors, we mean the subset of components that have a matching (categorical) value.

8 Conclusion

References

Archna Bhatia, Mandy Simons, Lori Levin, Yulia Tsvetkov, Chris Dyer, and Jordan Bender. 2014. A unified annotation scheme for the semantic/pragmatic components of definiteness. In *Proc. of LREC*. Reykjavík, Iceland.

Nils Reiter and Anette Frank. 2010. Identifying generic noun phrases. In *Proc. of ACL*, pages 40–49. Uppsala, Sweden.