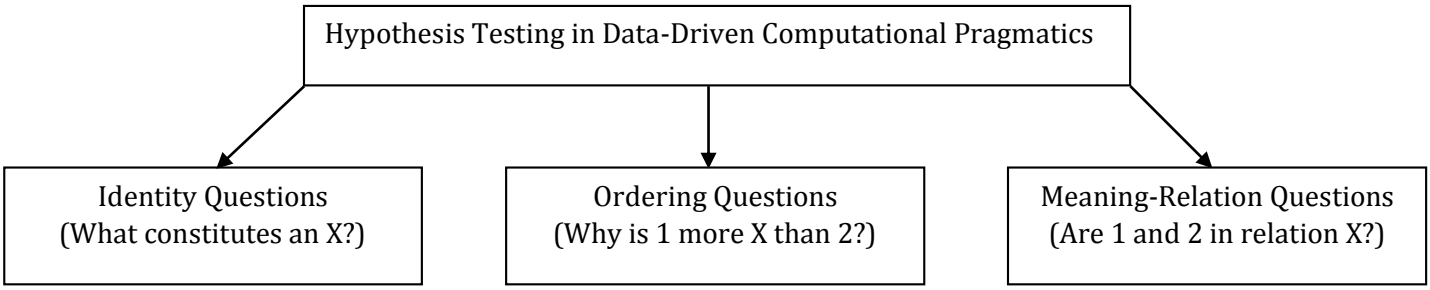
Data-Driven Computational Pragmatics

Class 2: Using Models for Hypothesis Testing



Identity Questions:

Identity questions involve modelling the essential properties of a particular phenomenon. In other words, they involve categorizing units of language as either belonging or not belonging to one or more categories. Thus, identity questions are tested using either classification or clustering algorithms. Examples of identity questions are whether a given word is used metaphorically or literally or whether a given utterance is used as a promising or questioning speech act. The hypothesis involves what features are required for making such identifications (e.g., what are the predictive features) and the relation between such features in the model.

Ordering Questions:

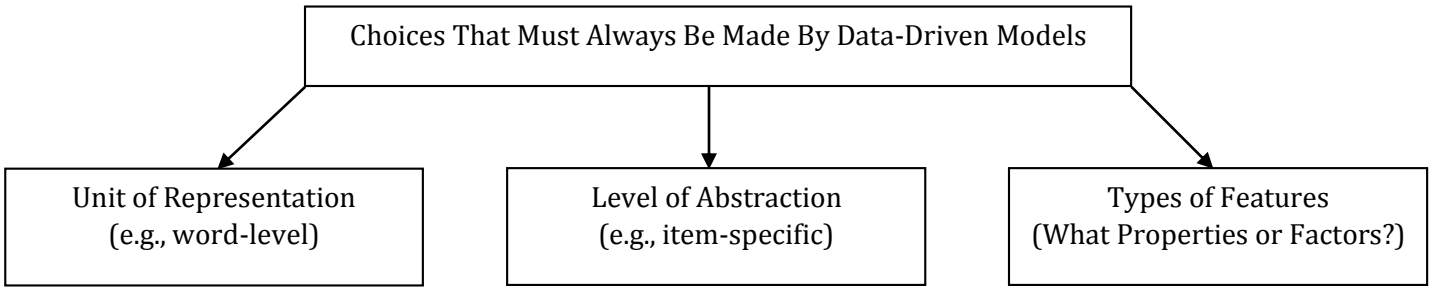
Ordering questions involve modelling the properties or factors which contribute to a particular scalar property in such a way that unit of language X is more or less than unit of language Y on that scale or ordering. In other words, ordering questions involve modelling causing or co-varying factors which can be used to predict the ordering of elements on the scale. Thus, ordering questions can be tested using regression algorithms or classifiers with ordered class variables. Examples of ordering questions are the level of abstractness or metaphoricity of a particular use of a word (e.g., is this or that word usage more metaphoric?) or the politeness of a given utterance (e.g., is this or that utterance more polite?). The hypothesis involves what features are predictive of ordering, or what features either cause or co-vary with the ordering of the dependent variable.

Meaning-Relation Questions:

Meaning-relation questions involve modelling the properties which predict whether two units stand in a particular meaning relationship. Because computational models are entirely syntactic, meaning itself is not available. However, meaning relationships such as entailment or synonymy are relationships which do or do not obtain between particular units and whose presence can be objectively (epistemically) observed. Generally, meaning-relationship questions can be tested using either classification or clustering algorithms as it involves a binary categorization of whether a particular relationship obtains between two units of language. Examples of meaning-relation questions are whether a metaphoric utterance and a literal utterance are synonyms and whether the use of one noun entails another noun. The hypothesis involves modelling the essential predictive properties of such a meaning-relationship in one or both units.

Implicit Hypotheses (Assumptions):

Data-driven computational models must make a number of choices in how they represent language, whether these choices are implicit or explicit. For hypothesis testing, it is important to make these choices explicit. There are three major types of assumptions present in these models: the unit of representation, the level of abstraction, and the type of features used.



Unit of Representation:

The first choice is the unit of representation of the model. For example, some phenomenon (e.g., abstractness) may be properties of individual words or senses of words and not depend upon larger context; other phenomenon (e.g., politeness or speech acts), however, may be properties of utterances and thus depend on much larger context. The model must decide how much of the context to include. This involves both linguistic context (in terms of whether the model makes predictions about words or sentences or whole texts) but also about non-linguistic context (in terms of whether the model has access to world-knowledge, for example).

Level of Abstraction:

The second choice is the level of abstraction of the model. For example, some phenomenon are item-specific (e.g., fixed idioms) and can be modelled in at least a partially item-specific manner (e.g., word n-grams are very item-specific features). Other phenomenon are wholly general (e.g., speech acts) and a model is less insightful when it includes more item-specific information. Level of abstraction is largely a property of features: are they specific (e.g., word-form frequencies) or general/abstract (e.g., concept domain frequencies)? Does the model include co-varying features (for example, a model of politeness could include as a co-varying feature information about the particular speech act in question). Level of abstraction is also influenced by the choice of algorithms. For example, linear SVMs are generalized in that individual feature coefficients are used to make predictions (e.g., individual cases are not remembered); however, non-linear SVMs can be item-specific in that the model remembers particular relations between variables (e.g., individual cases can be stored).

Types of Features:

The third choice is the type of features to be used in the model. Features are the only representation of language available to the model, with the result that the choice of features forms a hypothesis about what properties of language are relevant for a given phenomenon. For example, models of

metaphor often include word-level abstractness features because they hypothesize that metaphors map information from non-abstract to abstract concepts.

Difficulties and Impediments to Computational Hypothesis Testing

Some properties of data-driven models can create difficulties for hypothesis testing, including: redundant features, ensemble methods, skewed datasets, unrelated classes/clusters, non-robust clusters, and models that cannot be inspected.

Redundant/Correlated Features:

Depending on the learning algorithm, redundant features can form a problem for hypothesis testing. Some algorithms, like Naïve Bayes, cannot handle redundant/correlated features so that the presence of such features will throw off the model. The difficulty here is that some correlated features will be given too much weight in predicting the phenomenon, making the model too dependent on those correlated features. Other algorithms, like non-linear SVMS, can handle redundant/correlated features so that larger numbers of features will rarely reduce performance: only good features will be used. The difficulty here is that many features unrelated to the phenomenon will be included in the model.

Ensemble Methods:

Many models use ensemble methods by bringing together multiple algorithms or multiple iterations of algorithms. For example, a model might use three different classifiers with the same set of features and then use a second classifier to determine which classifier's predictions should be kept in particular cases. In another example, a model of co-reference might start with three different sets of entity-recognition results (e.g., entity identifications from three different classifiers) and use a clustering algorithm to determine co-reference relations across the entire set. These sorts of models become difficult to use for hypothesis testing because it is difficult to determine what features at what level from what system were involved in a particular prediction. It can also be difficult to determine whether a different ensemble classifier works best in given situation for legitimate theoretical reasons or simply because of a fluke choice made by the classifier.

Skewed Data:

Because data-driven approaches depend upon the datasets used, good data management techniques are essential for producing generalizable models. First, testing data is always required in addition to training data (for supervised approaches) in order to determine if the model formed on the training data is generalizable. Second, data from different sources or data that is differently sampled is important for validating the results of a model as both accurate and generalizable. The basic problem is that learning algorithms will usually form a decently performing model. The real question, then, is whether that model holds true across many situations and datasets. No real significance testing is available for data-driven models and rigorous data management techniques must be used to fill that void.

Unrelated Classes/Clusters:

A class-based hypothesis usually has two parts: first, that the data forms a certain set of classes; second, that those classes are formed on a particular basis or for a particular reason. Learning algorithms can be used to model the properties which separate classes, but do not specify the basis of that classification. For example, we may have a thousand sentences classified as polite or impolite and use a classifier to determine the properties that predict politeness. It may turn out that the polite sentences are also all questions and the impolite sentences statements. The model may thus be formed on the basis of speech acts and not politeness, so that the alignment of classes is accidental and not related to the hypothesis.

Clusters Can Be Fragile:

Unsupervised clustering algorithms can be used to test the learnability of a given class-based hypothesis: can the algorithm form the correct classes without any gold-standard training data and in this way mimic the human language learning process? The problem is that clustering algorithms can be quite fragile, forming different clusters with different parameter settings or even across multiple iterations with the same settings. Thus, testing hypotheses in this way should involve comparing many sets of clusters using, for example, the Measure of Concordance to evaluate the stability of the clusters.

Plausibility of the Model:

Some algorithms form models that can be inspected manually (e.g., logistic regression) and others form models that cannot be (e.g., non-linear SVMs). For the purposes of hypothesis testing it is often useful to employ an algorithm which allows the reasons for a given prediction to be determined by inspecting the model (in the case of logistic regression, using feature weights). Otherwise, the predictive power of the model may be contained in a small number of features that do not generalize in the desired manner (e.g., impoliteness might depend on the use of two or three slurs or insults). A model can be further validated, in addition to large and varied test sets, by examining the sources of its predictive power.

Power Algorithms vs. Good Features:

There is a general tension in data-driven models between the use of powerful algorithms (e.g., non-linear SVMs) and the use of good or linguistically valid features (e.g., not simple word-form n-grams). For practical purposes, a powerful algorithm with bad features may produce very good results. However, for the purposes of hypothesis testing, unless the question is specifically related to learning algorithms, it is the features or representations of language which are essential. Thus, it is important for hypothesis testing that the features can be interpreted linguistically.