patient probably has a left-sided cerebrovascular accident; post-convulsive state is less likely.' Negation, uncertainty, and affirmation form a continuum. Uncertainty detection was the focus of a recent NLP competition.[36]

5. *Relationship extraction*: determining relationships between entities or events, such as 'treats,' 'causes,' and 'occurs with.' Lookup of problem-specific information—for example, thesauri, databases—facilitates relationship extraction.

Anaphora reference resolution[37] is a sub-task that determines relationships between 'hierarchically related' entities: such relationships include:

- *Identity*: one entity—for example, a pronoun like 's/he,' 'hers/his,' or an abbreviation—refers to a previously mentioned named entity;
- *Part/whole*: for example, city within state;
- *Superset/subset*: for example, antibiotic/penicillin.

6. *Temporal inferences/relationship extraction*[38 39]: making inferences from temporal expressions and temporal relations—for example, inferring that something has occurred in the past or may occur in the future, and ordering events within a narrative (eg, medication X was prescribed *after* symptoms began).

7. *Information extraction (IE)*: the identification of problem-specific information and its transformation into (problem-specific) structured form. Tasks 1–6 are often part of the larger IE task. For example, extracting a patient's current diagnoses involves NER, WSD, negation detection, temporal inference, and anaphoric resolution. Numerous modern clinical IE systems exist,[40–44] with some available as open-source.[25 44 45] IE and relationship extraction have been themes of several i2b2/VA NLP challenges.[46–49] Other problem areas include phenotype characterization,[50–52] biosurveillance,[53 54] and adverse-drug reaction recognition.[55]

The National Library of Medicine (NLM) provides several well-known 'knowledge infrastructure' resources that apply to multiple NLP and IR tasks. The UMLS Metathesaurus,[56] which records synonyms and categories of biomedical concepts from numerous biomedical terminologies, is useful in clinical NER. The NLM's Specialist Lexicon[57] is a database of common English and medical terms that includes part-of-speech and inflection data; it is accompanied by a set of NLP tools.[58] The NLM also provides a test collection for word disambiguation.[59]

SOME DATA DRIVEN APPROACHES: AN OVERVIEW

Statistical and machine learning involve development (or use) of algorithms that allow a program to infer patterns about example ('training') data, that in turn allows it to 'generalize'—make predictions about new data. During the *learning* phase, numerical parameters that characterize a given algorithm's underlying model are computed by optimizing a numerical measure, typically through an iterative process.

In general, learning can be *supervised*—each item in the training data is labeled with the correct answer—or *unsupervised*, where it is not, and the learning process tries to recognize patterns automatically (as in cluster and factor analysis). One pitfall in any learning approach is the potential for *over-fitting*: the model may fit the example data almost perfectly, but makes poor predictions for new, previously unseen cases. This is because it may learn the random noise in the training data rather than only its essential, desired features. Over-fitting risk is minimized by techniques such as *cross-validation*, which partition the example data randomly into training and test sets to internally validate the model's predictions. This process of data partitioning, training, and validation is repeated over several rounds, and the validation results are then averaged across rounds.

Machine-learning models can be broadly classified as either generative or discriminative. Generative methods seek to create rich models of probability distributions, and are so called because, with such models, one can 'generate' synthetic data. Discriminative methods are more utilitarian, directly estimating posterior probabilities based on observations. Srihari[60] explains the difference with an analogy: to identify an unknown speaker's language, generative approaches would apply deep knowledge of numerous languages to perform the match; discriminative methods would rely on a less knowledge-intensive approach of using differences between languages to find the closest match. Compared to generative models, which can become intractable when many features are used, discriminative models typically allow use of more features.[61] Logistic regression and conditional random fields (CRFs) are examples of discriminative methods, while Naive Bayes classifiers and hidden Markov models (HMMs) are examples of generative methods.

Some common machine-learning methods used in NLP tasks, and utilized by several articles in this issue, are summarized below.

Support vector machines (SVMs)

SVMs, a discriminative learning approach, classify inputs (eg, words) into categories (eg, parts of speech) based on a feature set. The input may be transformed mathematically using a 'kernel function' to allow *linear separation* of the data points from different categories. That is, in the simplest two-feature case, a straight line would separate them in an X–Y plot: in the general N-feature case, the separator will be an (N−1) hyperplane. The commonest kernel function used is a Gaussian (the basis of the 'normal distribution' in statistics). The separation process selects a *subset* of the training data (the 'support vectors'—data points closest to the hyperplane) that best differentiates the categories. The separating hyperplane maximizes the distance to support vectors from each category (see figure 1).

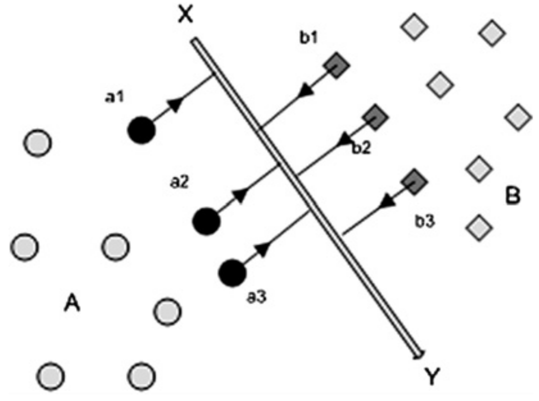

Figure 1 Support vector machines: a simple 2-D case is illustrated. The data points, shown as categories A (circles) and B (diamonds), can be separated by a straight line X–Y. The algorithm that determines X–Y identifies the data points ('support vectors') from each category that are closest to the other category (a1, a2, a3 and b1, b2, b3) and computes X–Y such that the margin that separates the categories on either side is maximized. In the general N-dimensional case, the separator will be an (N−1) hyperplane, and the raw data will sometimes need to be mathematically transformed so that linear separation is achievable.

patient probably has a left-sided cerebrovascular accident; post-convulsive state is less likely.' Negation, uncertainty, and affirmation form a continuum. Uncertainty detection was the focus of a recent NLP competition.[36]

5. *Relationship extraction*: determining relationships between entities or events, such as 'treats,' 'causes,' and 'occurs with.' Lookup of problem-specific information—for example, thesauri, databases—facilitates relationship extraction.

Anaphora reference resolution[37] is a sub-task that determines relationships between 'hierarchically related' entities: such relationships include:

- *Identity*: one entity—for example, a pronoun like 's/he,' 'hers/his,' or an abbreviation—refers to a previously mentioned named entity;
- *Part/whole*: for example, city within state;
- *Superset/subset*: for example, antibiotic/penicillin.

6. *Temporal inferences/relationship extraction*[38 39]: making inferences from temporal expressions and temporal relations—for example, inferring that something has occurred in the past or may occur in the future, and ordering events within a narrative (eg, medication X was prescribed *after* symptoms began).

7. *Information extraction (IE)*: the identification of problem-specific information and its transformation into (problem-specific) structured form. Tasks 1–6 are often part of the larger IE task. For example, extracting a patient's current diagnoses involves NER, WSD, negation detection, temporal inference, and anaphoric resolution. Numerous modern clinical IE systems exist,[40–44] with some available as open-source.[25 44 45] IE and relationship extraction have been themes of several i2b2/VA NLP challenges.[46–49] Other problem areas include phenotype characterization,[50–52] biosurveillance,[53 54] and adverse-drug reaction recognition.[55]

The National Library of Medicine (NLM) provides several well-known 'knowledge infrastructure' resources that apply to multiple NLP and IR tasks. The UMLS Metathesaurus,[56] which records synonyms and categories of biomedical concepts from numerous biomedical terminologies, is useful in clinical NER. The NLM's Specialist Lexicon[57] is a database of common English and medical terms that includes part-of-speech and inflection data; it is accompanied by a set of NLP tools.[58] The NLM also provides a test collection for word disambiguation.[59]

SOME DATA DRIVEN APPROACHES: AN OVERVIEW

Statistical and machine learning involve development (or use) of algorithms that allow a program to infer patterns about example ('training') data, that in turn allows it to 'generalize'—make predictions about new data. During the *learning* phase, numerical parameters that characterize a given algorithm's underlying model are computed by optimizing a numerical measure, typically through an iterative process.

In general, learning can be *supervised*—each item in the training data is labeled with the correct answer—or *unsupervised*, where it is not, and the learning process tries to recognize patterns automatically (as in cluster and factor analysis). One pitfall in any learning approach is the potential for *over-fitting*: the model may fit the example data almost perfectly, but makes poor predictions for new, previously unseen cases. This is because it may learn the random noise in the training data rather than only its essential, desired features. Over-fitting risk is minimized by techniques such as *cross-validation*, which partition the example data randomly into training and test sets to internally validate the model's predictions. This process of data partitioning, training, and validation is repeated over several rounds, and the validation results are then averaged across rounds.

Machine-learning models can be broadly classified as either generative or discriminative. Generative methods seek to create rich models of probability distributions, and are so called because, with such models, one can 'generate' synthetic data. Discriminative methods are more utilitarian, directly estimating posterior probabilities based on observations. Srihari[60] explains the difference with an analogy: to identify an unknown speaker's language, generative approaches would apply deep knowledge of numerous languages to perform the match; discriminative methods would rely on a less knowledge-intensive approach of using differences between languages to find the closest match. Compared to generative models, which can become intractable when many features are used, discriminative models typically allow use of more features.[61] Logistic regression and conditional random fields (CRFs) are examples of discriminative methods, while Naive Bayes classifiers and hidden Markov models (HMMs) are examples of generative methods.

Some common machine-learning methods used in NLP tasks, and utilized by several articles in this issue, are summarized below.

Support vector machines (SVMs)

SVMs, a discriminative learning approach, classify inputs (eg, words) into categories (eg, parts of speech) based on a feature set. The input may be transformed mathematically using a 'kernel function' to allow *linear separation* of the data points from different categories. That is, in the simplest two-feature case, a straight line would separate them in an X–Y plot: in the general N-feature case, the separator will be an (N−1) hyperplane. The commonest kernel function used is a Gaussian (the basis of the 'normal distribution' in statistics). The separation process selects a *subset* of the training data (the 'support vectors'—data points closest to the hyperplane) that best differentiates the categories. The separating hyperplane maximizes the distance to support vectors from each category (see figure 1).

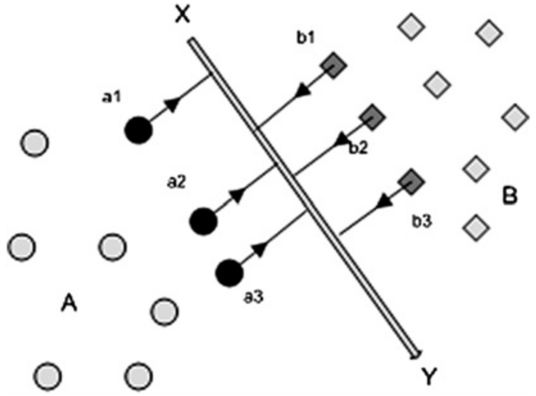

Figure 1 Support vector machines: a simple 2-D case is illustrated. The data points, shown as categories A (circles) and B (diamonds), can be separated by a straight line X—Y. The algorithm that determines X—Y identifies the data points ('support vectors') from each category that are closest to the other category (a1, a2, a3 and b1, b2, b3) and computes X—Y such that the margin that separates the categories on either side is maximized. In the general N-dimensional case, the separator will be an (N−1) hyperplane, and the raw data will sometimes need to be mathematically transformed so that linear separation is achievable.

A tutorial by Hearst *et al*[62] and the DTREG online documentation[63] provide approachable introductions to SVMs. Fradkin and Muchnik[64] provide a more technical overview.

Hidden Markov models (HMMs)

An HMM is a system where a variable can switch (with varying probabilities) between several states, generating one of several possible output symbols with each switch (also with varying probabilities). The sets of possible states and unique symbols may be large, but finite and known (see figure 2). We can observe the outputs, but the system's internals (ie, state-switch probabilities and output probabilities) are 'hidden.' The problems to be solved are:

A. *Inference*: given a particular sequence of output symbols, compute the probabilities of one or more candidate state-switch sequences.
B. *Pattern matching*: find the state-switch sequence most likely to have generated a particular output-symbol sequence.
C. *Training*: given examples of output-symbol sequence (training) data, compute the state-switch/output probabilities (ie, system internals) that fit this data best.

B and C are actually Naive Bayesian reasoning extended to sequences; therefore, HMMs use a generative model. To solve these problems, an HMM uses two simplifying assumptions (which are true of numerous real-life phenomena):

1. The probability of switching to a *new* state (or back to the same state) depends on the previous N states. In the simplest 'first-order' case (N=1), this probability is determined by the current state alone. (First-order HMMs are thus useful to model events whose likelihood depends on what happened last.)
2. The probability of generating a particular output in a particular state depends only on that state.

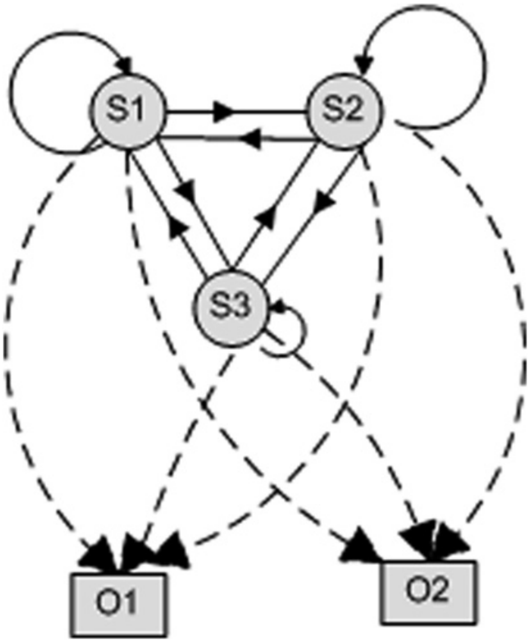

Figure 2 Hidden Markov models. The small *circles* S1, S2 and S3 represent *states*. *Boxes* O1 and O2 represent *output values*. (In practical cases, hundreds of states/output values may occur.) The *solid* lines/arcs connecting states represent *state switches*; the arrow represents the switch's direction. (A state may switch back to itself.) Each line/arc label (not shown) is the *switch probability*, a decimal number. A *dashed* line/arc connecting a state to an output value indicates 'output probability': the probability of that output value being generated from the particular state. If a particular switch/output probability is zero, the line/arc is not drawn. The sum of the switch probabilities leaving a given state (and the similar sum of output probabilities) is equal to 1. The sequential or temporal aspect of an HMM is shown in figure 3.

These assumptions allow the probability of a given state-switch sequence (and a corresponding observed-output sequence) to be computed by simple multiplication of the individual probabilities. Several algorithms exist to solve these problems.[65 66] The highly efficient Viterbi algorithm, which addresses problem B, finds applications in signal processing, for example, cell-phone technology.

Theoretically, HMMs could be extended to a multivariate scenario,[67] but the training problem can now become intractable. In practice, multiple-variable applications of HMMs (eg, NER[68]) use single, artificial variables that are uniquely determined composites of existing categorical variables: such approaches require much more training data.

HMMs are widely used for speech recognition, where a spoken word's waveform (the output sequence) is matched to the sequence of individual phonemes (the 'states') that most likely produced it. (Frederick Jelinek, a statistical-NLP advocate who pioneered HMMs at IBM's Speech Recognition Group, reportedly joked, 'every time a linguist leaves my group, the speech recognizer's performance improves.'[20]) HMMs also address several bioinformatics problems, for example, multiple sequence alignment[69] and gene prediction.[70] Eddy[71] provides a lucid bioinformatics-oriented introduction to HMMs, while Rabiner[72] (speech recognition) provides a more detailed introduction.

Commercial HMM-based speech-to-text is now robust enough to have essentially killed off academic research efforts, with dictation systems for specialized areas—eg, radiology and pathology—providing structured data entry. Phrase recognition is paradoxically more reliable for polysyllabic medical terms than for ordinary English: few word sequences sound like 'angina pectoris,' while common English has numerous homophones (eg, two/too/to).

Conditional random fields (CRFs)

CRFs are a family of discriminative models first proposed by Lafferty *et al*.[73] An accessible reference is Culotta *et al*[74]; Sutton and McCallum[75] is more mathematical. The commonest (linear-chain) CRFs resemble HMMs in that the next state depends on the current state (hence the 'linear chain' of dependency).

CRFs generalize logistic regression to sequential data in the same way that HMMs generalize Naive Bayes (see figure 3). CRFs are used to predict the state variables ('Ys') based on the observed variables ('Xs'). For example, when applied to NER, the state variables are the categories of the named entities: we want to predict a sequence of named-entity categories within a passage. The observed variables might be the word itself, prefixes/suffixes, capitalization, embedded numbers, hyphenation, and so on. The linear-chain paradigm fits NER well: for example, if the previous entity is 'Salutation' (eg, 'Mr/Ms'), the succeeding entity must be a person.

CRFs are better suited to sequential multivariate data than HMMs: the training problem, while requiring more example data than a univariate HMM, is still tractable.

N-grams

An 'N-gram'[19] is a sequence of N items—letters, words, or phonemes. We know that certain item pairs (or triplets, quadruplets, etc) are likely to occur much more frequently than